Innovación y colaboración para una minería responsable

Corporación Universitaria Remington.
Facultad de ingeniería.
Tecnología en Desarrollo de software.

Luisa María Arcila Soto
Manuela Osorio Monsalve
Alexis Arcila Soto

Gloria Amparo Lora
Tutor de proyecto

Proyecto de grado
2026

Dedicatoria

Este trabajo está dedicado a todos los mineros artesanales del Nordeste Antioqueño, quienes con su esfuerzo diario sostienen comunidades enteras. A nuestras familias, por su apoyo incondicional durante este proceso de formación profesional.

Agradecimientos

Agradecemos a la Corporación Universitaria Remington, a la tutora Gloria Amparo Lora por su orientación y acompañamiento y a la docente Piedad María Metaute Paniagua por su apoyo técnico, y a los mineros del Nordeste Antioqueño que participaron en la encuesta y aportaron información valiosa para el desarrollo de este proyecto.

Tabla de contenido

Dedicatoria 2
Agradecimientos 3
Listado de tablas 12
Listado de ilustraciones 15
Resumen 19
Palabras clave 19
Introducción y Marco Teórico 20
Roles Scrum 21
La Minería en Colombia 21
Metodología Scrum 22
Arquitectura por capas 23
Antecedentes de software aplicado a la minería 24
Planteamiento del Problema y Justificación 26
Definición del Problema 26
Proceso Actual (AS-IS) 26
Proceso Propuesto (TO-BE) 26
Pregunta Problematizadora 27
Justificación 27
Objetivos 28
Objetivo General 28
Objetivos Específicos 28
Metodología 29
Objetivo 1: Levantamiento de Requerimientos – Encuesta 29
Diseño metodológico de la encuesta 29
Criterios de inclusión 29
Criterios de exclusión 29
Tipo de instrumento y validación 30
Limitaciones de estudio 30
Conclusión de la Encuesta 39
Objetivo 2: Análisis 40
Tabla de Trazabilidad – Resultados de la Encuesta vs. Requisitos del Sistema 40
Listado de Necesidades y Características del Sistema 41
Alcance del Sistema: Entradas, Procesos y Salidas 42
Cronograma de Actividades (Sprint Backlog) 42
Análisis de Riesgos 43
Riesgo cliente 44
Riesgo proceso 45
Riesgo producto 45
Tabla General para Casos de Uso 47
Historias de Usuario 53
Requisitos de Usuario (RU) 54
Requisitos Funcionales (RF) 56
Trazabilidad entre Historias de Usuario (HU) y Requisitos Funcionales (RF) 67
Requisitos No Funcionales (RNF) 68
Objetivo 3: Diseño 70
Descripción General de Actores 70
Diagramas del Modelo de Casos de Uso 70
Diagrama de Clases 89

Diagrama de Secuencia 94
Diagramas de actividades 101
Mapa de Navegación 111
Recursos del Proyecto 111
Criterios de aceptación 112
Product Backlog 115
Reglas de Negocio 118
Gestión del Cambio 120
Documentación del Modelo Entidad-Relación (MER) 121
Tablas Maestras 122
Tablas Transaccionales 125
Prototipos de Alta Fidelidad 127
Prototipo 1. Pantalla de Inicio de Sesión 128
Prototipo 2. Panel de Administración 129
Prototipo 3. Mapa Interactivo 130
Prototipo 4. Foro 131
Prototipo 5. Informes 132
Prototipo 6. Consultas 133
Prototipo 7. Gestión de Detectores de Oro 134
Objetivo 4: Codificación 136
Métricas de Software – Requerimientos 136
Métricas de Software – Código Fuente 137
Métricas de Proyecto 139
Fase de construcción 140
Repositorio en Github 147
Objetivo 5: Pruebas 148
Objetivo del plan de pruebas 148
Alcance y entorno de pruebas 148
Tipos de pruebas aplicadas 148
Estrategia de pruebas 149
Criterios de aceptación 149
Roles y responsables 149
Resultados de las pruebas 149
Casos de Pruebas Funcionales (CP-01 a CP-12) 150
Pruebas de Seguridad 165
Pruebas de regresión 165
Pruebas de Rendimiento, Carga y Estrés 173
Pruebas de Usabilidad 187
Resumen Unificado de Ejecución 193
Objetivo 6: Documentación – Arquitectura del Sistema 197
Capa Frontend 197
Capa Backend 197
Capa Base de Datos 198
Manual de usuario 199
1. Introducción al sistema 199
1.1 ¿Quién puede usar SoftGold? 199
2. Objetivo de la plataforma 200
3. Requisitos mínimos 200
3.1 Hardware recomendado (Usuario final) 200

3.2 Software recomendado 200
4. Acceso al sistema 201
4.1 Inicio de sesión 201
4.2 Resultado del Inicio de Sesión 201
4.3 Cuenta Bloqueada 201
5. Recuperación de contraseña 202
6. Navegación general 203
6.1 Estructura de la Interfaz (Usuario Autenticado) 203
6.2 Menú Lateral (Sidebar) 203
6.3 Barra Superior (Topbar) 203
6.4 Navegación en Dispositivos Móviles 204
7. Panel principal (Dashboard) 204
7.1 Panel del Administrador 204
7.2 Panel del Minero 204
7.3 Panel del Empleado 205
8. Módulo minas 205
8.1 Listar minas 205
8.2 Buscar Minas 206
8.3 Crear una Nueva Mina 206
8.4 Editar una Mina 207
8.5 Eliminar una Mina 207
9. Módulo de mapas 208
9.1 Información de un Mapa 208
9.2 Ver Mapa en Leaflet 209
10. Módulo de riesgos 209
10.1 Tipos de Riesgos Comunes 209
10.2 Crear un Riesgo 210
10.3 Asociar Riesgos a Minas 210
11. Módulo de zonas de exploración 210
11.1 Información de una Zona de Exploración 211
11.2 Crear una Zona de Exploración 211
11.3 Mapa General de Exploración 211
12. Módulo del foro 211
12.1 Listar Publicaciones 212
12.2 Ver una Publicación 212
12.3 Crear una Publicación 212
12.4 Comentar en una Publicación 213
12.5 Eliminar una Publicación 213
13. Módulo de informes y reportes 213
13.1 Dashboard Gerencial 213
13.2 Reporte de Minas 214
13.3 Impacto Ambiental 214
13.4 Inventario 215
14. Sistema de soporte 215
14.1 Crear un Ticket de Soporte (Acceso Público) 215
14.2 Gestión de Tickets (Solo Administradores) 216
15. Búsqueda y filtros avanzados 217
15.1 Criterios de Búsqueda 217
15.2 Realizar una Búsqueda 217

16. Gestión de usuarios 218
16.1 Tipos de Gestión de Usuarios 218
16.2 Crear un Usuario 218
16.3 Editar un Usuario 220
16.4 Eliminar un Usuario 220
17. Documentos del sistema 220
18. Mi cuenta (Perfil) 221
18.1 Ver Perfil 221
18.2 Cambiar Contraseña 222
19. Atajos de teclado 223
20. Modo oscuro / claro 223
Cambiar el Tema 223
21. Mensajes de error frecuentes 223
22. Preguntas frecuentes 224
23. Cierre de sesión 225
Recomendaciones generales de uso 225
Manual Técnico del Sistema 227
Control de cambios 227
1. Descripción general del sistema 227
1.1 Contexto del Negocio 228
1.2 Alcance del Sistema 228
2. Objetivo del software 229
2.1 Objetivo General 229
2.2 Objetivos Específicos 229
3. Arquitectura del sistema 229
3.1 Diagrama de Arquitectura General 229
3.2 Patrón de Diseño MVC 230
Spring Data JPA 231
3.3 Diagrama de componentes 231
4. Tecnologías usadas 232
4.1 Backend 232
4.2 Frontend 232
4.3 Base de Datos 233
4.4 Herramientas de Construcción y DevOps 233
4.5 Servicios Externos 233
5. Estructura del proyecto 233
6. Modelos de datos 236
6.1 Diagrama Entidad-Relación (Conceptual) 236
6.2 Descripción de Entidades 238
7. Configuración del entorno de desarrollo 240
7.1 Variables de Entorno (application.properties) 240
7.2 Descripción de Propiedades Clave 241
8. Instalación del proyecto 242
8.1 Clonar el Repositorio 242
8.2 Configurar la Base de Datos 242
8.3 Configurar el Archivo de Propiedades 242
8.4 Instalar Dependencias y Compilar 242
8.5 Verificar la Compilación 243
9. Ejecución local del sistema 243

9.1 Desde el IDE 243
9.2 Desde la Línea de Comandos 243
9.3 Verificar el Inicio 243
9.4 Credenciales por Defecto 243
10. Rutas y controladores 244
10.1 Rutas Públicas (sin autenticación) 244
10.2 Rutas del Administrador (/admin/**) 245
10.3 Rutas del Sistema de Soporte 246
10.4 Rutas del Foro 246
10.5 Convenciones de Controladores 247
11. Seguridad implementada 247
11.1 Diagrama de Flujo de Autenticación 247
11.2 Configuración de Spring Security 248
11.3 Cifrado de Contraseñas 248
11.4 Bloqueo de Cuenta 250
11.5 Recuperación de Contraseña 250
11.6 Control de Roles y Autoridades 250
11.7 Sesión y Timeout 251
12. Servicio de correo electrónico 251
12.1 Configuración SMTP 251
12.2 Uso del EmailService 251
12.3 Obtener Contraseña de Aplicación Google 252
13. Manejo de excepciones y Logs 252
13.1 GlobalControllerAdvice 252
13.2 Logging de SQL 252
13.3 Logs de Spring Boot 252
14. Validaciones 252
14.1 Validación de Entidades (Bean Validation) 252
14.2 Validación en Controladores 253
14.3 Validación del Lado del Cliente 253
15. Geolocalización y mapas 254
15.1 Integración de Leaflet.js 254
15.2 Inicialización Estándar del Mapa 254
15.3 Clases CSS para Contenedores de Mapa 254
16. Procedimiento de despliegue 255
16.1 Construcción del Artefacto 255
16.2 Despliegue en Servidor Linux 255
16.3 Configuración como Servicio system (Linux) 255
17. Buenas prácticas implementadas 256
18. Estrategia de mantenimiento 256
18.1 Actualizaciones de Dependencias 256
18.2 Respaldo de la Base de Datos 256
18.3 Monitoreo 257
19. Recomendaciones futuras 257
20. Glosario técnico 257
Manual de Implementacion y Despliegue 259
1. Introducción 259
1.1 Ambientes de Despliegue 259
1.2 Estrategia de Configuración 260

2. Requisitos del servidor 260
2.1 Requisitos de Hardware (Producción) 260
2.2 Requisitos de Software del Servidor 260
2.3 Puertos Requeridos 261
3. Preparación del entorno 261
3.1 Actualizar el Sistema Operativo 261
3.2 Crear Usuario de Servicio 261
3.3 Configurar Firewall 262
4. Instalación de Java y Maven 262
4.1 Instalar OpenJDK 17 262
4.2 Configurar Java Home 262
4.3 Instalar Maven (para compilación en servidor) 263
5. Instalación y configuración de MySQL 263
5.1 Instalar MySQL 8 263
5.2 Asegurar la Instalación de MySQL 263
5.3 Crear Base de Datos y Usuario para SoftGold 264
5.4 Verificar Configuración de MySQL 264
6. Instalación del Backend (SoftGold) 264
6.1 Compilar el Proyecto (en servidor de CI o local) 264
6.2 Transferir el JAR al Servidor 265
6.3 Verificar el JAR 265
7. Configuración variables de entorno 265
7.1 Archivo de Variables de Entorno 265
7.2 Archivo application-prod.properties 266
8. Configuración de la base de datos 267
8.1 Migraciones y Esquema Inicial 267
8.2 Script SQL de Creación del Esquema 267
8.3 Datos Iniciales del Administrador 269
9. Configuración del servidor web (NGINX) 269
9.1 Instalar Nginx 270
9.2 Configuración de Nginx (HTTP) 270
10. Configuración SSL/TLS 270
10.1 Instalar Certbot (Let's Encrypt -- Gratuito) 270
10.2 Configuración Nginx con SSL 271
11. Configuración Docker 272
11.1 Dockerfile 272
11.2 Docker Compose 273
11.3 Archivo .env para Docker Compose 273
11.4 Comandos Docker 274
12. Configuración CI/CD 274
12.1 Pipeline GitHub Actions (Ejemplo) 274
13. Proceso de despliegue 275
13.1 Configurar SoftGold como Servicio systemd 275
13.2 Verificar el Arranque 276
13.3 Probar la Aplicación 276
14. Configuración de Backups 277
14.1 Script de Backup de Base de Datos 277
14.2 Script de Backup del JAR 278
14.3 Estrategia de Retención 278

15. Monitoreo del sistema 278
15.1 Estado del Servicio 278
15.2 Monitoreo de Recursos 279
15.3 Monitoreo de la Base de Datos 279
15.4 Script de Monitoreo de Salud 280
16. Validación post-despliegue 280
16.1 Verificaciones Técnicas 280
16.2 Verificaciones Funcionales 281
17. Checklist de implementación 281
17.1 Checklist Pre-Despliegue 281
17.2 Checklist Post-Despliegue 282
18. Procedimiento de actualización 283
18.1 Pasos para Actualizar SoftGold 283
18.2 Tiempo de Indisponibilidad Estimado 284
19. Procedimiento de Rollback 284
20. Problemas comunes y soluciones 285
20.1 Error de conexión a la base de datos 285
20.2 Los mapas Leaflet.js no cargan 285
20.3 El correo de recuperación no llega 286
20.4 Sesión expira muy rápido 286
20.5 Rendimiento lento con muchos usuarios 286
20.6 Nginx devuelve 502 Bad Gateway 286
Resultados y Discusión 288
Cumplimiento de Objetivos Específicos 288
OBJ-1: Levantamiento de Requerimientos 289
OBJ-2: Análisis con Diagramas de Uso y Escenarios 290
OBJ-3: Diseño con Diagramas UML y Prototipos 292
OBJ-4: Codificación con POO, Monolítico, Java y SQL 293
OBJ-5: Pruebas Unitarias e Integrales 295
OBJ-6: Documentación Técnica y de Usuario 296
Comparación con Literatura 298
Software para Gestión Minera y Minería Artesanal 298
Sostenibilidad Digital y Sistemas de Información 300
Arquitectura MVC vs. Microservicios en Contexto Académico 301
Hallazgos Cuantitativos Verificables 302
Cobertura y Resultados de Pruebas Automatizadas 302
Análisis de Tiempos de Respuesta 304
Limitaciones del Estudio 306
Muestra Pequeña 306
Alcance Regional 307
Ausencia de Despliegue en Producción Real 308
Ausencia de Pruebas con Usuarios Reales Prolongadas 309
Análisis Crítico del Cumplimiento de Objetivos 309
Objetivo Cumplido al 100 %: OBJ-1, OBJ-2, OBJ-3, OBJ-4 y OBJ-5 311
Objetivos Cumplidos Parcialmente: OBJ-6 313
Conclusiones de los resultados 314
Conclusión 316
Centralizar registro y seguimiento de minas, mapas y zonas 316
Control de acceso basado en roles - RBAC 316

Foro para comunicación interna 316
Reportes e informes gerenciales 317
Sistema de tickets de soporte técnico 317
Visualización geoespacial con Leaflet y OpenStreetMap 317
Anexos 319
Glosario 319
Referencias 322

Listado de tablas

Tabla 1. Roles del equipo de trabajo 21
Tabla 2. Resultados de la Encuesta vs. Requisitos del Sistema 41
Tabla 3. Listado de Necesidades y Características del Sistema 42
Tabla 4. Alcance del Sistema: Entradas, Procesos y Salidas 42
Tabla 5. Análisis de riesgos 47
Tabla 6. Casos de Uso 53
Tabla 7. Historias de usuario del sistema SoftGold 54
Tabla 8. Requisitos de Usuario (RU) 56
Tabla 9. Requisitos Funcionales (RF) 67
Tabla 10. Trazabilidad historias de usuario y requisitos funcionales 68
Tabla 11. Requisitos no funcionales del sistema SoftGold. 69
Tabla 12. Descripción General de Actores 70
Tabla 13. Recursos del Proyecto 112
Tabla 14. Criterios de aceptación 115
Tabla 15. Product Backlog 117
Tabla 16. Reglas de negocio por módulo del sistema SoftGold. 119
Tabla 17. Gestión del Cambio 121
Tabla 18. Tabla Maestra de la entidad Zonas 123
Tabla 19. Tabla Maestra de la entidad Departamento 123
Tabla 20. Tabla Maestra de la entidad Riesgo 124
Tabla 21. Tabla Maestra de la entidad Mina 124
Tabla 22. Tabla Maestra de la entidad Empleado 125
Tabla 23. Tabla Maestra de la entidad Usuario 125
Tabla 24. Tabla Transaccional de las entidades Impacto ambiental - Mina 126
Tabla 25. Tabla Transaccional de las entidades Riesgo – Mina Explorada 126
Tabla 26. Tabla Transaccional de las entidades Mina – Mapa 127
Tabla 27. Tabla Transaccional de las entidades Mina – Riesgo 127
Tabla 28. Métricas de Software – Requerimientos 137
Tabla 29. Métricas de Software – Código Fuente 138
Tabla 30. Métricas de Proyecto 139
Tabla 31. Alcance y entorno de pruebas 148
Tabla 32. Resultados de pruebas ejecutadas 150
Tabla 33. Casos de prueba - Módulo de Autenticación 151
Tabla 34. CP - Módulo gestión de minas 153
Tabla 35. CP - Módulo de Gestión de Riesgos. 158
Tabla 36. CP - Módulo foro comunitario 160
Tabla 37. CP - Módulo de soporte y seguridad 161
Tabla 38. Resumen de casos de pruebas funcionales del sistema SoftGold 164
Tabla 39. Pruebas de Seguridad 165
Tabla 40. Tiempos de ejecución por suite de regresión (mvn test — 15/05/2026). 166
Tabla 41. Suite 1 — RT-01 a RT-05: Carga del contexto de la aplicación. 167
Tabla 42. Suite 2 — RT-06 a RT-10: Regresión CRUD de Minas. 168
Tabla 43. Suite 3 — RT-11 a RT-16: Validación de modelos de dominio. 169
Tabla 44. Suite 4 — RT-17 a RT-21: Seguridad y control de acceso por roles. 171
Tabla 45. Suite 5 — RT-22 a RT-26: Regresión del módulo Foro Comunitario. 172
Tabla 46. Objetivos de las pruebas de rendimiento y criterios de aceptación. 174
Tabla 47. Criterios de aceptación de rendimiento — umbrales aceptable y crítico. 174

Tabla 48. ESC-01: Resultados de carga en el endpoint de login (/login POST). 175
Tabla 49. ESC-02: Resultados de carga en el dashboard de informes (/admin/informes). 176
Tabla 50. ESC-03: Resultados de carga en listado de minas (/admin/minas). 177
Tabla 51. ESC-05: Resultados de carga en el foro público (/foro GET). 178
Tabla 52. Prueba de estrés — degradación progresiva bajo concurrencia extrema. 179
Tabla 53. Saturación del pool HikariCP bajo carga creciente. 179
Tabla 54. Soak Test — evolución de métricas JVM durante 4 horas de carga sostenida. 180
Tabla 55. Rendimiento de endpoints de autenticación — 50 usuarios concurrentes. 181
Tabla 56. Rendimiento de endpoints de administración — 50 usuarios concurrentes. 182
Tabla 57. Rendimiento de endpoints públicos — 50 usuarios concurrentes. 182
Tabla 58. Consultas SQL lentas identificadas con Slow Query Log y EXPLAIN. 183
Tabla 59. Hallazgos críticos y de alta prioridad — plan de acción. 184
Tabla 60. KPIs de rendimiento — cumplimiento vs. umbral objetivo. 185
Tabla 61. Estado general del sistema por categoría de prueba. 186
Tabla 62. Pruebas de Usabilidad — métricas por interfaz del sistema SoftGold. 187
Tabla 63. Pruebas de Rendimiento de Interfaces — métricas por pantalla 188
Tabla 64. Resumen de ejecución — Casos de prueba funcionales. 193
Tabla 65. Resumen de ejecución — Pruebas de regresión con tiempos reales. 194
Tabla 66. Resumen consolidado de todas las pruebas ejecutadas — BUILD SUCCESS. 194
Tabla 67. Manual usuario - Usuarios que pueden usar softgold 199
Tabla 68. Manual usuario - Hardware recomendado 200
Tabla 69. Manual usuario - Software recomendado 200
Tabla 70. Manual Usuario - Minas registradas 206
Tabla 71. Manual Usuario - Creación de mina 207
Tabla 72. Manual Usuario - Información de un mapa 208
Tabla 73. Manual Usuario - Información de una zona de exploración 211
Tabla 74. Manual Usuario - Crear publicación 213
Tabla 75. Manual Usuario - Ticket Soporte 215
Tabla 76. Manual Usuario - Gestión tickets 216
Tabla 77. Manual Usuario - Criterios de búsqueda 217
Tabla 78. Manual Usuario - Tipo de gestión de usuario 218
Tabla 79. Manual Usuario - Crear un usuario 219
Tabla 80. Manual Usuario - Documentos del sistema 221
Tabla 81. Manual Usuario - Cambiar contraseña 222
Tabla 82. Manual Usuario - Atajos de teclado 223
Tabla 83. Manual Usuario - Mensajes de error frecuentes 224
Tabla 84. Manual técnico - Control de cambios 227
Tabla 85. Manual técnico - Alcance del sistema 229
Tabla 86. Manual técnico - Patrón de diseño MVC 231
Tabla 87. Manual técnico – Backend 232
Tabla 88. Manual técnico – Frontend 232
Tabla 89. Manual técnico - Base de datos 233
Tabla 90. Manual técnico - Herramientas de construcción y DevOps 233
Tabla 91. Manual técnico - Servicios externos 233
Tabla 92. Manual técnico – Entidad usuario 239
Tabla 93. Manual técnico - Entidad mina 240
Tabla 94. Manual técnico – Herramientas 240
Tabla 95. Manual técnico - Variables de entorno 240
Tabla 96. Manual técnico - Descripción de propiedades claves 241

Tabla 97. Manual técnico - Credenciales por defecto 244
Tabla 98. Manual técnico - Rutas y controladores 244
Tabla 99. Manual técnico - Rutas del administrador 246
Tabla 100. Manual técnico- Rutas del sistema de soporte 246
Tabla 101. Manual técnico - Rutas del foro 246
Tabla 102. Manual técnico - Bloqueo de cuenta 250
Tabla 103. Manual técnico - Tabla de mapeo de roles 251
Tabla 104. Manual técnico - Validación de entidades 253
Tabla 105. Manual técnico - Clases CSS contenedores de mapa 254
Tabla 106. Manual técnico - Buenas prácticas implementadas 256
Tabla 107. Manual técnico - Recomendaciones futuras 257
Tabla 108. Manual técnico - Glosario técnico 258
Tabla 109. Manual implementación - Ambientes de despliegue 259
Tabla 110. Manual implementación - Requisitos de Hardware (Producción) 260
Tabla 111. Manual implementación - Requisitos de Software del Servidor 260
Tabla 112. Manual implementación - Puertos requeridos 261
Tabla 113. Manual implementación - Estrategia de retención 278
Tabla 114. Manual implementación - Verificaciones funcionales 281
Tabla 115. Manual implementación - Tiempo de Indisponibilidad Estimado 284
Tabla 116. Cobertura de pruebas automatizadas — SoftGold v1.0. 303
Tabla 117. Tiempos de respuesta clave por endpoint — comparativa 1 vs 100 usuarios 304
Tabla 118. Análisis crítico del cumplimiento de objetivos específicos. 311

Listado de ilustraciones

Ilustración 1. Nivel de experiencia en minería 31
Ilustración 2. Tipo de minería que practican 31
Ilustración 3. Importancia de reducir impacto ambiental 32
Ilustración 4. Principales desafíos identificados 32
Ilustración 5. Uso de detectores de oro 33
Ilustración 6. Nivel de satisfacción con detectores de oro 33
Ilustración 7. Frecuencia de uso de detectores de oro 34
Ilustración 8. Uso de aplicaciones o software para minería 34
Ilustración 9. Percepción sobre aplicación móvil para optimizar detección de oro 35
Ilustración 10. Funcionalidades preferidas en la aplicación 35
Ilustración 11. Preferencia de plataforma (web/móvil) 36
Ilustración 12. Disposición a compartir información sobre zonas exploradas 36
Ilustración 13. Importancia de minimizar impacto ambiental sin afectar trabajo 37
Ilustración 14. Tecnologías utilizadas en minería 37
Ilustración 15. Prácticas sugeridas para minimizar impacto ambiental 38
Ilustración 16. Cronograma Sprint Backlog – Detalle 43
Ilustración 17. Cronograma Sprint Backlog – Detalle 43
Ilustración 18. Diagrama general del sistema 71
Ilustración 19. Diagrama de casos de uso extendido – Gestión Zonas 72
Ilustración 20. Diagrama de caso de uso – Gestión Departamento 73
Ilustración 21. Diagrama de caso de uso – Gestión Impacto Ambiental 74
Ilustración 22. Diagrama extendido de caso de uso – Gestión Mapas 75
Ilustración 23. Diagrama extendido de caso de uso – Gestión Tipo Suelo 76
Ilustración 24. Diagrama de caso extendido – Gestión Tipo de Excavación 77
Ilustración 25. Diagrama de caso de uso extendido – Gestión Foro 78
Ilustración 26. Diagrama de caso de uso extendido – Gestión Registro Usuario 79
Ilustración 27. Diagrama de caso de uso extendido – Gestión Detectores de Oro 80
Ilustración 28. Diagrama de caso de uso extendido – Gestión Tipos Detectores de Oro 81
Ilustración 29. Diagrama de caso de uso extendido – Gestión Minero 82
Ilustración 30. Diagrama de caso de uso extendido – Gestión Empleados 83
Ilustración 31. Diagrama de caso de uso extendido – Gestión Minas 84
Ilustración 32. Diagrama de caso de uso extendido – Gestión Minas Exploradas 84
Ilustración 33. Diagrama de caso de uso extendido – Gestión Tipos de Mapas 85
Ilustración 34. Diagrama de caso de uso extendido – Gestión Riesgos 86
Ilustración 35. Diagrama de caso de uso extendido – Gestión Informes 86
Ilustración 36. Diagrama de caso de uso extendido – Gestión Consultas 87
Ilustración 37. Diagrama de caso de uso extendido – Gestión Perfil 87
Ilustración 38. Diagrama de caso de uso extendido – Gestión Usuario 88
Ilustración 39. Diagrama de caso de uso extendido – Gestión Filtros 89
Ilustración 40. Diagrama de Clases — Modelo de Dominio 92
Ilustración 41 Diagrama de clases: Capa de Negocio: Controllers → Services → DAOs 93
Ilustración 42. Diagrama de Secuencia — Flujo de Autenticación 96
Ilustración 43. Diagrama de Secuencia — Gestión de Minas (CRUD). 98
Ilustración 44. Diagrama de Secuencia — Módulo Foro 101
Ilustración 45. Diagrama de actividad– Registro de usuario 102
Ilustración 46. Diagrama de actividad – Registro Zona 103
Ilustración 47. Diagrama de actividad – Consulta Mapa 104

Ilustración 48 .Diagrama de actividad – Registro de zona 105
Ilustración 49. Diagrama de actividad – Inicio de sesión 106
Ilustración 50. Diagrama de actividad – Consulta y generación de informes 107
Ilustración 51. Diagrama de actividad – Registro de impacto ambiental 108
Ilustración 52. Diagrama de actividad – Publicación de comentario en el foro 109
Ilustración 53. Diagrama de actividad – Gestión de perfiles 110
Ilustración 54. Mapa de navegación del sistema SoftGold 111
Ilustración 55. Modelo Entidad-Relación del sistema SoftGold 122
Ilustración 56. Prototipo – Pantalla de Inicio de Sesión 128
Ilustración 57. Prototipo – Panel de Administración 130
Ilustración 58. Prototipo – Mapa Interactivo 131
Ilustración 59. Prototipo – Módulo Foro 132
Ilustración 60. Prototipo – Módulo de Informes y Consultas 133
Ilustración 61. Prototipo – Módulo Consultas 134
Ilustración 62. Prototipo – Gestión de Detectores de Oro 135
Ilustración 63. Fragmento de código– Registro de usuario 141
Ilustración 64. Fragmento de código– Registro de usuario 141
Ilustración 65. Fragmento de código– Registro de usuario 142
Ilustración 66. Fragmento de código– Registro de usuario 143
Ilustración 67. Fragmento de código– Registro de usuario 144
Ilustración 68. Fragmento de código– Zona exploración 145
Ilustración 69. Fragmento de código– Zona exploración 145
Ilustración 70. Fragmento de código– Encriptación contraseña 146
Ilustración 71. CP-02: Pantalla de login mostrando mensaje de error 152
Ilustración 72. CP-03: Listado de minas tras la creación exitosa de "Mina el Dorado" 154
Ilustración 73. CP-04: Formulario de nueva mina mostrando la alerta de validación 155
Ilustración 74. CP-05: Mina "El dorado" ahora muestra el departamento actualizado 156
Ilustración 75. CP-06: Listado de minas tras eliminar el registro con ID 15 157
Ilustración 76. CP-08: Módulo de gestión de riesgos mostrando el nuevo registro 159
Ilustración 77. CP-09: Módulo de foro y discusiones mostrando el post "Consulta seguridad" ... 160
Ilustración 78. CP-10: Formulario de reporte con datos válidos 162
Ilustración 79. CP-11: Intento de acceso a /soporte/tickets sin sesión activa 163
Ilustración 80. CP-21: Vista de administrador "Tickets de soporte" 164
Ilustración 81. Prueba de usabilidad – Panel de administración 189
Ilustración 82. Prueba de usabilidad – Panel de gestión de mineros 190
Ilustración 83. Prueba de usabilidad – Panel de minas 190
Ilustración 84. Prueba de usabilidad – Panel foro 191
Ilustración 85. Prueba de usabilidad – Zona exploración 191
Ilustración 86. Prueba de usabilidad – Mapa exploración 192
Ilustración 87. Prueba de usabilidad – Impacto Ambiental 192
Ilustración 88. Prueba de usabilidad – Soporte técnico 193
Ilustración 89. Manual usuario - Inicio de sesión 201
Ilustración 90. Manual usuario - Recuperación contraseña 202
Ilustración 91. Manual Usuario - Estructura interfaz 203
Ilustración 92. Manual usuario – Dashboard 204
Ilustración 93. Manual usuario - Módulo minas 205
Ilustración 94. Manual usuario - Crear nueva mina 206
Ilustración 95. Manual usuario - Gestión de mapas 208
Ilustración 96. Manual usuario - Mapa exploración 209

Ilustración 97. Manual usuario - Zonas de exploración 210
Ilustración 98. Manual Usuario - Módulo del foro 212
Ilustración 99. Manual usuario - Dashboard gerencial 214
Ilustración 100. Manual usuario - Reportar un problema 216
Ilustración 101. Manual usuario - Tickets de soporte 217
Ilustración 102. Manual usuario - Búsqueda avanzada 218
Ilustración 103. Manual usuario - Crear usuario 220
Ilustración 104. Manual usuario - Ver perfil 221
Ilustración 105. Manual usuario - Cambiar contraseña 222
Ilustración 106. Manual técnico - Arquitectura general del sistema 230
Ilustración 107. Manual técnico - Diagrama de componentes 231
Ilustración 108. Manual técnico - Diagrama del proyecto 236
Ilustración 109. Manual técnico - Diagrama Entidad - Relación (Conceptual) 238
Ilustración 110. Manual técnico - Convenciones de controladores 247
Ilustración 111. Manual técnico - Diagrama de Flujo de Autenticación 248
Ilustración 112. Manual técnico - Cifrado de contraseñas 249
Ilustración 113. Manual técnico - Control de roles y autoridades 250
Ilustración 114. Manual técnico. Uso del EmailService 251
Ilustración 115. Manual técnico - Validación en controladores 253
Ilustración 116. Manual técnico - Integración de Leaflet.js 254
Ilustración 117. Manual técnico - Inicialización estándar del mapa 254
Ilustración 118. Manual técnico - Construcción del artefacto 255
Ilustración 119. Manual técnico - Configuración como Servicio system (Linux) 256
Ilustración 120. Manual implementación - Estrategia de configuración 260
Ilustración 121. Manual implementación - Crear Usuario de Servicio 261
Ilustración 122. Manual implementación - Configurar Firewall 262
Ilustración 123. Manual implementación - Instalar OpenJDK 17 262
Ilustración 124. Manual implementación - Instalar Maven (para compilación en servidor) 263
Ilustración 125. Manual implementación - Instalación MySQL 263
Ilustración 126. Manual implementación - Verificación configuración MySQL 264
Ilustración 127. Manual implementación - Archivo de variables de entorno 266
Ilustración 128. Manual implementación - Instalar Nginx 270
Ilustración 129. Manual implementación - Configuración de Nginx (HTTP) 270
Ilustración 130. Manual implementación - Instalar Certbot 271
Ilustración 131. Manual implementación - Configuración Nginx con SSL 271
Ilustración 132. Manual implementación – Dockerfile 272
Ilustración 133. Manual implementación - Archivo .env para Docker Compose 273
Ilustración 134. Manual implementación - Comandos Docker 274
Ilustración 135. Manual implementación - Configurar SoftGold como Systemd 276
Ilustración 136. Manual implementación - Prueba de la aplicación 276
Ilustración 137. Manual implementación - Script de Backup de Base de Datos 277
Ilustración 138. Manual implementación - Script de Backup del JAR 278
Ilustración 139. Manual implementación - Estado del servicio 278
Ilustración 140. Manual implementación - Monitoreo de recursos 279
Ilustración 141. Manual implementación - Monitoreo de la base de datos 279
Ilustración 142. Manual implementación - Script de Monitoreo de Salud 280
Ilustración 143. Manual implementación - Verificaciones Técnicas 281
Ilustración 144. Manual implementación - Procedimiento de actualización 283
Ilustración 145. Manual implementación - Pasos para Actualizar SoftGold 284

Ilustración 146. Manual implementación - Causa y solución en la conexión con la base de datos 285

Resumen

El presente proyecto de grado describe el diseño, análisis y desarrollo del sistema de software SoftGold, una plataforma web orientada a apoyar las actividades de exploración minera en el Nordeste Antioqueño de Colombia. SoftGold permite a los mineros artesanales registrar, centralizar y visualizar información de zonas de exploración mediante un mapa interactivo, fomenta la colaboración comunitaria a través de un foro de experiencias y proporciona información ambiental relevante para reducir excavaciones innecesarias y minimizar el impacto ambiental. La metodología empleada corresponde al marco de trabajo Scrum, con desarrollo incremental organizado en sprints, utilizando Java, Spring Boot y MySQL. El proyecto parte de un levantamiento de requerimientos mediante encuestas a mineros, del cual se derivan requisitos funcionales y no funcionales, diagramas de casos de uso, historias de usuario, un modelo entidad-relación y prototipos de alta fidelidad. Los resultados evidencian la viabilidad técnica del sistema y su potencial para apoyar la toma de decisiones mineras, promoviendo una minería más responsable, eficiente y sostenible.

Palabras clave

Minería artesanal, mapa interactivo, impacto ambiental, Scrum, SoftGold.

Introducción y Marco Teórico

En busca de reducir el impacto ambiental que generan las minas en el sector del Nordeste Antioqueño y también en una búsqueda de ayudar al minero en su trabajo, nace este proyecto como una propuesta para hacer que la minería sea más sostenible y eficiente. Muchos mineros de esta región de Colombia encuentran numerosos problemas al momento de identificar zonas de extracción, ya que la gran mayoría no cuenta con herramientas óptimas y adecuadas para este tipo de trabajo, lo que conlleva a realizar excavaciones innecesarias que afectan no solo su rendimiento y el costo de material, sino también el entorno natural.

Con la realización e implementación de SoftGold se pretende mejorar dichos aspectos, en donde los mineros tengan acceso tanto a zonas con mayor probabilidad de oro como a un mapa interactivo en donde podrán localizar con exactitud las mejores zonas para realizar dicha identificación, basándose en la interpretación de datos arrojados por los detectores de metales y en un foro en donde diferentes mineros podrán compartir y debatir acerca de sus experiencias y recomendaciones para localizar las minas.

El proyecto contribuye no solo a mejorar las condiciones de trabajo de los mineros sino también a proteger el entorno y la naturaleza, promoviendo prácticas responsables y el cuidado del medio ambiente.

Roles Scrum

Nota. Roles del equipo de trabajo bajo la metodología Scrum adaptados al proyecto SoftGold.

Rol Scrum	Responsabilidad en el proyecto	Ejemplo adaptado a SoftGold
Product Owner	Define las prioridades del producto. Representa los intereses del cliente. Aprueba o rechaza entregables según los objetivos del proyecto.	La comunidad minera del Nordeste Antioqueño (representada por el docente). Define qué funcionalidades son más urgentes: primero el mapa interactivo, luego el foro, luego los informes.
Scrum Master	Facilita el proceso Scrum. Asegura que se cumplan los eventos, artefactos y principios de la metodología. Identifica y elimina impedimentos que afecten al equipo.	Manuela Osorio Monsalve. Responsable de planificar los Sprints, dar seguimiento al Sprint Backlog y asegurar el cumplimiento de Scrum durante el desarrollo del proyecto.
Development Team	Diseña, desarrolla, prueba y documenta el software. Es un equipo autoorganizado y multifuncional encargado de transformar los requisitos en incrementos funcionales.	Luisa Arcila (documentación y frontend). Manuela Osorio (backend y base de datos). Alexis Arcila (apoyo en desarrollo, pruebas y documentación).

Tabla 1. Roles del equipo de trabajo

Fuente: Elaboración propia

La Minería en Colombia

La minería hoy en día es uno de los principales soportes económicos de muchas regiones de Colombia, especialmente en el departamento de Antioquia. No obstante, la minería presenta muchos desafíos en el ámbito social, ambiental y legal. Según la Universidad Externado (2023), "la falta de formalidad en la pequeña minería tiene un impacto ambiental significativo. Sin regulaciones adecuadas, las prácticas insostenibles pueden causar daños irreparables a los ecosistemas locales" (p. 1). Teniendo en cuenta lo anterior, la falta de un control adecuado en la minería genera graves consecuencias para el medio ambiente, ya que las excavaciones —muchas veces innecesarias— contribuyen a su deterioro.

Aunque la minería aporta a la economía en muchas regiones de Colombia, debe gestionarse de manera responsable para evitar problemas sociales y ambientales. Algunos pequeños mineros se ven afectados por las regulaciones y políticas que no siempre consideran su situación. Pantoja Timarán y Pantoja Barrios (2016) plantean que uno de los factores que ha intensificado los conflictos

en las zonas mineras es la implementación de la Ley 2ª de 1959, que creó las Reservas Forestales en Colombia. La aplicación estricta de esta ley ha generado grandes dificultades para obtener o renovar títulos de concesión minera, afectando a pequeños mineros que han invertido sus recursos y vivido en estas áreas durante años.

Según Escobar y Martínez (2014), los resultados del Censo muestran que se visitaron 14.357 unidades de producción minera (UPM), de las cuales solo 5.316 UPM contaban con un título minero. En otras palabras, por cada UPM con título minero se identificaron 1,7 UPM sin título minero (informales), lo que sugiere el gran tamaño que tendría la explotación minera informal en el país (p. 21).

Sin embargo, en una encuesta realizada por Cárdenas y Reina (2008), se identificaron áreas en las que las empresas mineras destacan por tener mejores prácticas: el 87,5% de las empresas mineras invierte en programas para fomentar el respeto a los derechos humanos, en comparación con aproximadamente el 55% de las empresas en general. Además, todas las empresas mineras encuestadas implementan iniciativas para proteger el medio ambiente y cuentan con mecanismos para prevenir la corrupción.

Metodología Scrum

La metodología Scrum es un marco de trabajo ágil ampliamente utilizado en el desarrollo de software, el cual se caracteriza por su enfoque iterativo, incremental y colaborativo. Este modelo promueve la adaptación constante a los cambios y la entrega continua de valor, permitiendo mejorar la eficiencia en los proyectos.

Según Estrada et al. (2021) señalan que Scrum:

"Permite un desarrollo ágil auto organizado que utiliza un enfoque incremental de procesos que permite una mejor forma de entregar los resultados" (p.2).

Lo que evidencia su importancia en entornos dinámicos. Asimismo, uno de los aspectos clave de esta metodología es la distribución de roles y responsabilidades, donde el cliente define las prioridades del producto mientras que el equipo se autogestiona para cumplir los objetivos; como indican los autores, Estrada et. al (2021) establece que el cliente establece las prioridades; mientras que el equipo SCRUM se auto organiza a fin de determinar la mejor forma de entregar los resultados.

Por otra parte, Scrum se fundamenta en principios de control empírico como la transparencia, la inspección y la adaptación, los cuales permiten supervisar continuamente el avance del proyecto y realizar ajustes oportunos. Scrum utiliza un enfoque incremental fundamentado en la teoría de control empírico de procesos, que está basada en transparencia, inspección y adaptación.

En conjunto, estos elementos convierten a Scrum en una metodología flexible y eficiente, especialmente adecuada para proyectos que requieren innovación, colaboración constante y capacidad de respuesta ante cambios.

Arquitectura por capas

La arquitectura de software es un aspecto clave en el desarrollo de sistemas, ya que define cómo se organizan y relacionan sus componentes para garantizar que el sistema funcione de manera eficiente, escalable y mantenible. En la práctica, no se trata solo de una estructura técnica, sino de una decisión estratégica que influye directamente en la calidad y evolución del software a lo largo del tiempo.

En este contexto, Según Rojas Enríquez et. al (2024) se plantea que:

> "La arquitectura de software es el origen conceptual de cualquier sistema". (p. 11).

Lo que refuerza su papel como base del desarrollo. Uno de los enfoques más utilizados es la arquitectura en n-capas, la cual divide el sistema en niveles que separan responsabilidades, facilitando el trabajo en equipo, el mantenimiento y la reutilización del código. Como se menciona en el documento, esta estructura permite que los cambios se realicen de forma localizada sin afectar todo el sistema Rojas Enríquez et. al (2024), convirtiéndola así en una solución práctica y eficiente para proyectos de software modernos.

Antecedentes de software aplicado a la minería

En la industria minera, el uso de software especializado ha permitido mejorar significativamente los procesos de exploración, modelado y planificación de la extracción de recursos. Uno de los sistemas más representativos es Surpac, considerado uno de los softwares más utilizados a nivel mundial para la geología y planificación minera, el cual permite calcular volúmenes, generar modelos geológicos en tres dimensiones y optimizar la explotación de reservas minerales.

De acuerdo con Martínez et. al, este tipo de herramientas facilita la evaluación precisa de depósitos minerales y mejora la toma de decisiones en proyectos tanto subterráneos como a cielo abierto, al integrar funciones de modelado, análisis geoestadístico y planificación operativa. Asimismo, el desarrollo de software minero no solo se limita a soluciones comerciales, sino que también existen alternativas libres como RecMin, las cuales permiten realizar modelamiento

geológico, cálculo de recursos minerales y diseño de minas, ampliando el acceso tecnológico a sectores con menos recursos.

Estos antecedentes evidencian la importancia del software en la minería moderna, al contribuir a aumentar la eficiencia, precisión y sostenibilidad de las operaciones.

De manera complementaria, en el ámbito de la minería a gran escala se han desarrollado otras plataformas especializadas como Vulcan, Datamine, Deswik y MineSight, las cuales ofrecen funcionalidades avanzadas para el modelado de yacimientos, la planificación minera, la optimización de la producción y la gestión de operaciones complejas. Estas soluciones están orientadas principalmente a entornos industriales y corporativos, donde se dispone de infraestructura tecnológica robusta y personal altamente especializado, lo que limita su adopción en contextos de minería artesanal y de pequeña escala.

La existencia de estas herramientas refuerza la necesidad de desarrollar sistemas de información más accesibles y contextualizados, como SoftGold, que integren información geográfica, ambiental y colaborativa de forma sencilla y acorde a las capacidades de sus usuarios.

Planteamiento del Problema y Justificación

Definición del Problema

Uno de los principales problemas de la minería es la localización y extracción de oro debido a la falta de acceso a tecnologías avanzadas y la escasez de información sobre las zonas con mayor potencial. Muchos mineros dependen de métodos tradicionales o de tecnologías costosas, lo que incrementa el costo operativo, el tiempo invertido y el impacto ambiental. Esto frecuentemente lleva a realizar excavaciones innecesarias que generan degradación del suelo y destrucción de ecosistemas, mientras que la falta de información sobre zonas propensas a encontrar oro afecta tanto a los mineros como al medio ambiente.

Proceso Actual (AS-IS)

En el estado actual, los mineros realizan actividades de exploración y excavación con base en información limitada, experiencias previas o recomendaciones informales de otros mineros. No existe una herramienta centralizada que permita consultar información histórica sobre zonas exploradas, impacto ambiental o actividad reciente. Como consecuencia, el proceso de toma de decisiones se caracteriza por la falta de análisis previo, lo que ocasiona excavaciones repetitivas en zonas de bajo potencial, aumento de costos operativos y un impacto ambiental negativo.

AS-IS: El minero realiza excavaciones sin contar con información estructurada sobre zonas con potencial aurífero o exploraciones previas.

Proceso Propuesto (TO-BE)

En el proceso propuesto, el minero cuenta con el sistema SoftGold como herramienta de apoyo para la toma de decisiones. Antes de realizar una excavación, el usuario puede consultar un mapa interactivo, revisar información sobre zonas exploradas, actividad reciente, riesgos e impacto ambiental, y acceder a recomendaciones compartidas por otros mineros. El uso de esta información

permite planificar mejor las actividades mineras, reducir excavaciones innecesarias, optimizar el uso de recursos y minimizar el impacto ambiental.

TO-BE: El minero consulta el mapa interactivo y la información registrada en el sistema antes de decidir dónde excavar.

Pregunta Problematizadora

¿Cómo puede implementarse una solución tecnológica en el sector minero del Nordeste Antioqueño que facilite a los mineros el registro, la visualización y el intercambio de información sobre zonas de exploración, fomente la colaboración comunitaria y minimice el impacto ambiental, incluyendo la prevención de la destrucción de ecosistemas, la reducción de emisiones de gases de efecto invernadero, la contaminación del agua y la protección de la salud pública?

Justificación

Actualmente, la forma en que se extrae el oro y la falta de regulación y control en el sector da lugar a la minería ilegal, en donde no siempre se cumplen las normativas ambientales, generando un impacto negativo para el medio ambiente, especialmente en la recuperación del suelo que rara vez se logra. Además, muchos mineros artesanales no solo carecen de acceso a tecnologías que apoyen su trabajo, sino también a información organizada sobre las zonas ya exploradas, lo cual conduce a excavaciones innecesarias que afectan directamente al medio ambiente.

Objetivos

Objetivo General

Desarrollar un sistema de información que apoye a los mineros del Nordeste Antioqueño en el registro, centralización y visualización de zonas de exploración minera, mediante un mapa interactivo y herramientas colaborativas, con el fin de reducir excavaciones innecesarias y minimizar el impacto ambiental.

Objetivos Específicos

- Levantar los requerimientos mediante encuestas directas a los mineros, con el fin de comprender sus necesidades, limitaciones tecnológicas y prácticas habituales en la exploración minera.
- Analizar la información recopilada, utilizando diagramas de uso y escenarios.
- Diseñar los requisitos del software mediante el uso de diagramas UML y la creación de prototipos interactivos.
- Codificar los requisitos funcionales del software utilizando POO y una arquitectura monolítica a través de lenguajes de programación Java y BD SQL.
- Aplicar pruebas unitarias e integrales para verificar la calidad y el desempeño del software, garantizando la eficiencia de este en el entorno real.
- Construir la documentación del producto que incluya manuales técnicos y manuales de usuario y de implementación para que tanto los desarrolladores como los usuarios finales tengan toda la información disponible sobre cómo usar y mantener el software.

Metodología

El presente proyecto corresponde a un desarrollo de software con enfoque descriptivo-aplicado, apoyado en datos cuantitativos obtenidos de la encuesta a mineros, en el cual cada objetivo específico representa una fase del ciclo de vida del software.

El proyecto se desarrolló bajo el marco de trabajo Scrum, estructurado en fases de levantamiento de requerimientos, análisis, diseño, construcción, pruebas y documentación.

Objetivo 1: Levantamiento de Requerimientos – Encuesta

Se diseñó y aplicó una encuesta a mineros del Nordeste Antioqueño. El enlace de la encuesta es: https://forms.gle/GkuFzcBXdWPVSRJr7.

Diseño metodológico de la encuesta

La encuesta fue aplicada a una muestra de 18 mineros del Nordeste Antioqueño, Seleccionados a partir de un muestreo no probabilístico por conveniencia.

La población objetivo estuvo conformada por mineros artesanales y de pequeña escala que realizan actividades de exploración y extracción de oro en el Nordeste Antioqueño.

Criterios de inclusión

- Mineros con experiencia en actividades de exploración o extracción de oro.
- Personas que ejercen minería artesanal o de pequeña escala en el Nordeste Antioqueño.
- Participantes que aceptaron voluntariamente responder la encuesta.

Criterios de exclusión

- Personas sin experiencia directa en actividades mineras.

- Participantes que no completaron la totalidad de la encuesta.

Tipo de instrumento y validación

El instrumento utilizado fue una encuesta estructurada compuesta por preguntas cerradas, abiertas y de selección múltiple, diseñada específicamente para identificar necesidades, problemáticas y expectativas de los mineros frente al uso de herramientas tecnológicas.

Dado el carácter académico y exploratorio del proyecto, el instrumento no fue sometido a un proceso formal de validación estadística; sin embargo, fue diseñado para garantizar la coherencia y pertinencia de las preguntas en relación con los objetivos del estudio.

Limitaciones de estudio

Como limitación del estudio, se reconoce que el tamaño de muestra de la encuesta es reducido, por lo cual los resultados obtenidos no pueden generalizarse a toda la población minera del Nordeste Antioqueño. No obstante, la información recopilada resulta suficiente para orientar el levantamiento de requerimientos del sistema SoftGold, dado el enfoque descriptivo y aplicado del proyecto.

A continuación, se presentan los resultados con su análisis correspondiente.



Ilustración 1. Nivel de experiencia en minería

Fuente: Elaboración propia

Análisis: El 55,6% de los encuestados tiene un nivel de experiencia intermedio, mientras que el 38,9% es avanzado y el 5,6% es novato.

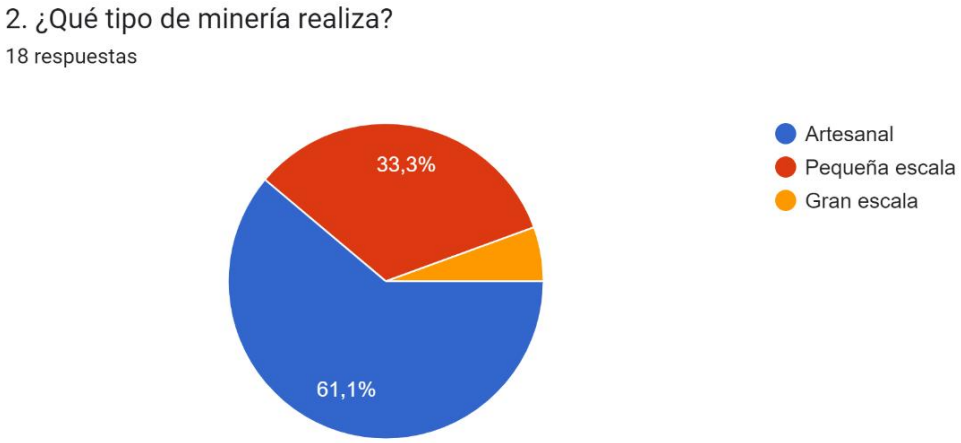


Ilustración 2. Tipo de minería que practican

Fuente: Elaboración propia

Análisis: El 61,1% son mineros artesanales, el 33,3% se dedica a la minería en pequeña escala y el 5,6% a la minería a gran escala.

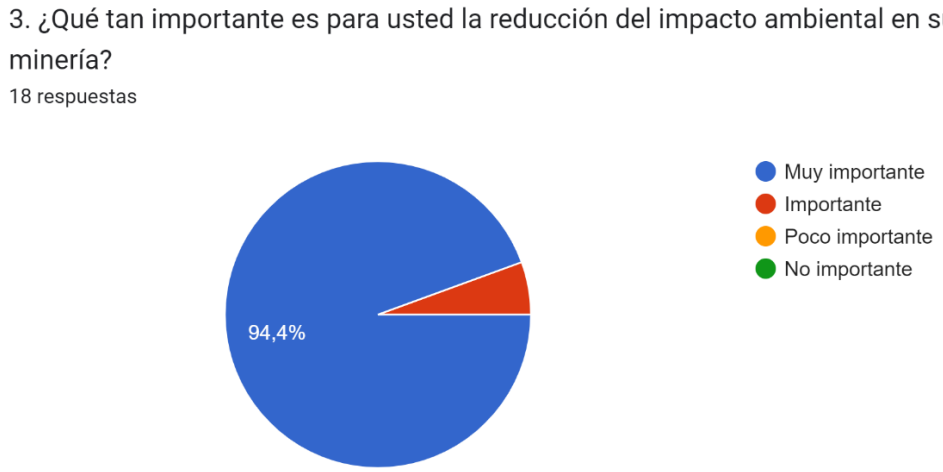


Ilustración 3. Importancia de reducir impacto ambiental

Fuente: Elaboración propia

Análisis: Para el 94,4% de los encuestados es muy importante reducir el impacto ambiental, mientras que para el 5,6% es importante.

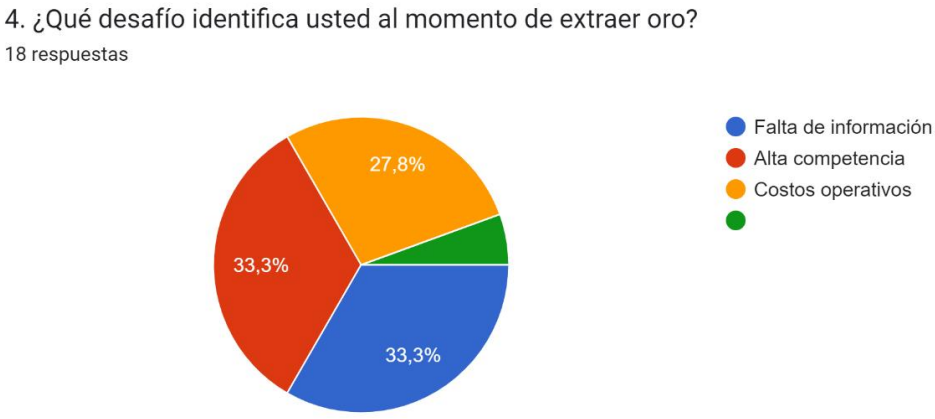


Ilustración 4. Principales desafíos identificados

Fuente: Elaboración propia

Análisis: El 66,6% de los encuestados identifica como principal desafío la falta de información (33,3%) y la alta competencia (33,3%), mientras que el 27,8% encuentra como obstáculo los costos operativos.



Ilustración 5. Uso de detectores de oro

Fuente: Elaboración propia

Análisis: El 50% de los encuestados utiliza detectores de oro, mientras que el otro 50% no los utiliza.



Ilustración 6. Nivel de satisfacción con detectores de oro

Fuente: Elaboración propia

Análisis: Un 58,8% de los encuestados están satisfechos con los detectores de oro, mientras que el 17,6% se siente insatisfecho, un 11,8% se encuentra muy satisfecho y otro 11,8% muy insatisfecho.



Ilustración 7. Frecuencia de uso de detectores de oro

Fuente: Elaboración propia

Análisis: El 76,5% rara vez utiliza detectores de oro, un 17,6% los usa semanalmente y un 5,6% diariamente.



Ilustración 8. Uso de aplicaciones o software para minería

Fuente: Elaboración propia

Análisis: Un 89,9% de los encuestados no utiliza ninguna aplicación o software para la minería, mientras que el 11,1% sí lo hace.

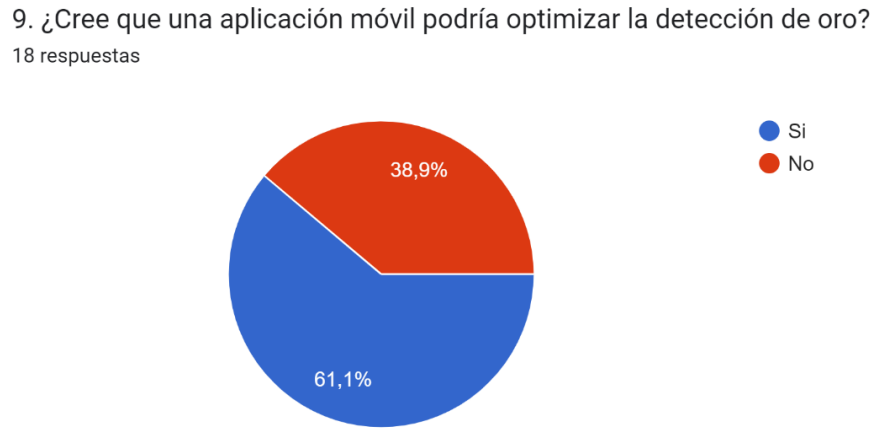


Ilustración 9. Percepción sobre aplicación móvil para optimizar detección de oro

Fuente: Elaboración propia

Análisis: El 61,1% de los encuestados considera que una aplicación móvil sí podría optimizar la detección de oro, mientras que el 38,9% no lo considera.

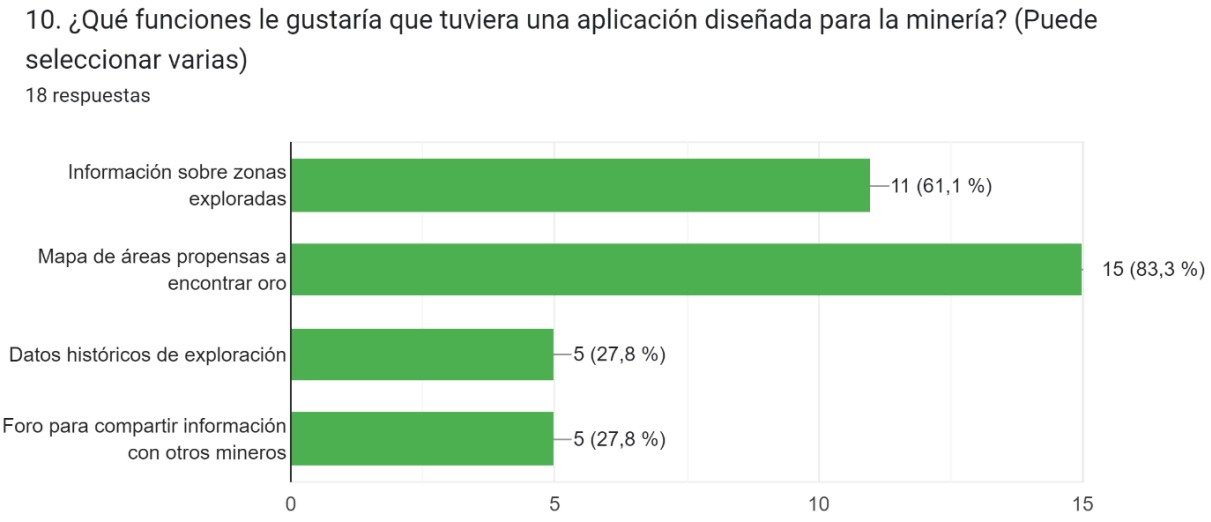


Ilustración 10. Funcionalidades preferidas en la aplicación

Fuente: Elaboración propia

Análisis: El 83,3% prefiere que la aplicación incluya un mapa de áreas propensas a encontrar oro, el 61,1% quiere información sobre zonas exploradas, el 27,8% opta por datos históricos de

exploración y otro 27,8% considera importante un foro para compartir información con otros mineros.

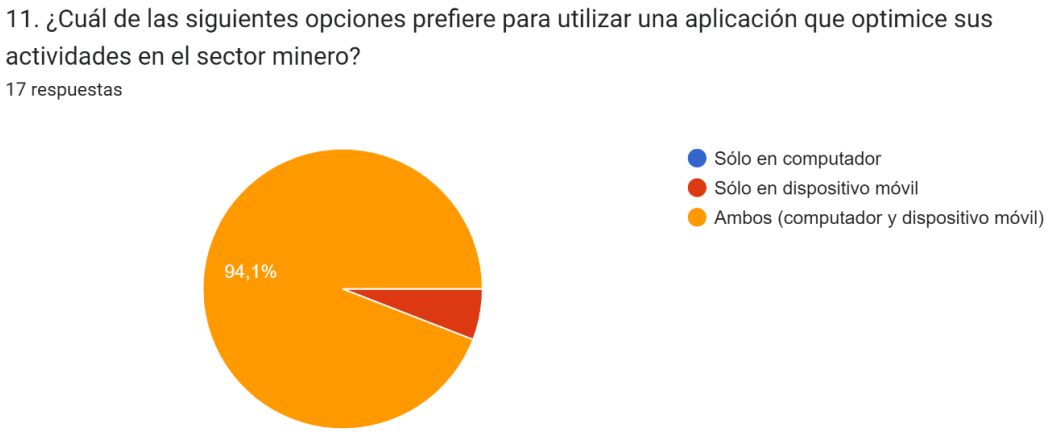


Ilustración 11. Preferencia de plataforma (web/móvil)

Fuente: Elaboración propia

Análisis: El 94,1% prefiere que la aplicación esté diseñada en un ambiente web y móvil, mientras que el 5,9% lo prefiere solo en móvil.

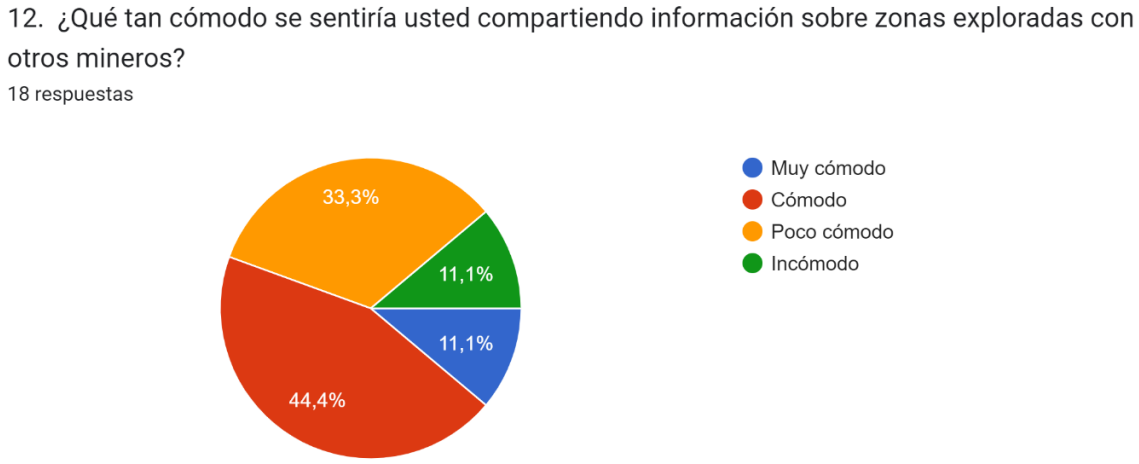


Ilustración 12. Disposición a compartir información sobre zonas exploradas

Fuente: Elaboración propia

Análisis: Un 44,4% concuerda en que se sentiría cómodo compartiendo información sobre zonas exploradas con otros mineros, un 33,3% poco cómodo, por otro lado un 11,1% nada cómodo y 11,1% restante muy cómodo.



Ilustración 13. Importancia de minimizar impacto ambiental sin afectar trabajo

Fuente: Elaboración propia

Análisis: Para el 78,8% de los encuestados es muy importante contar con un sistema que minimice el impacto ambiental sin afectar su trabajo, mientras que un 22,2% lo considera importante.

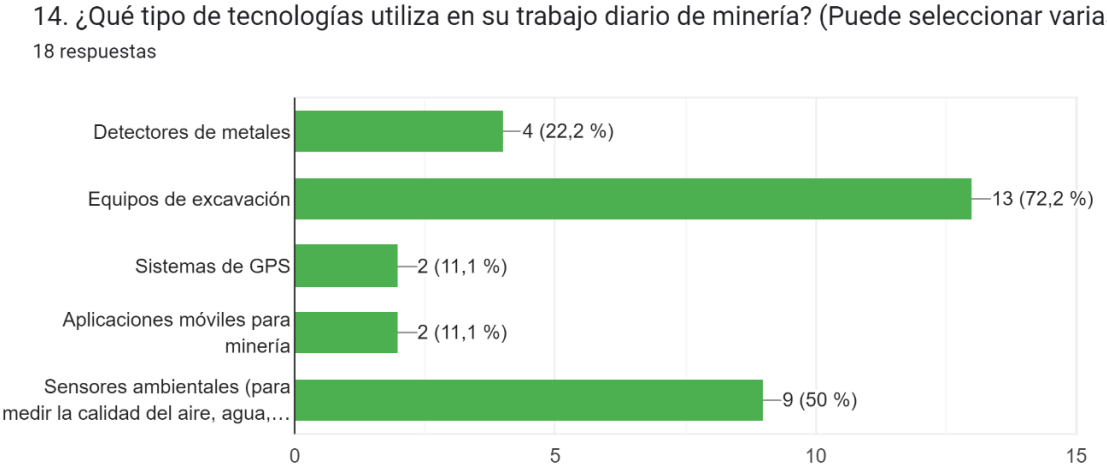


Ilustración 14. Tecnologías utilizadas en minería

Fuente: Elaboración propia

Análisis: El 72,2% usa equipos de excavación para realizar su trabajo en minas, un 22,2% utiliza detectores de metales, 50% emplea sensores ambientales, el 11,1% usa sistemas de GPS y otro 11,1% utiliza aplicaciones móviles para minería.

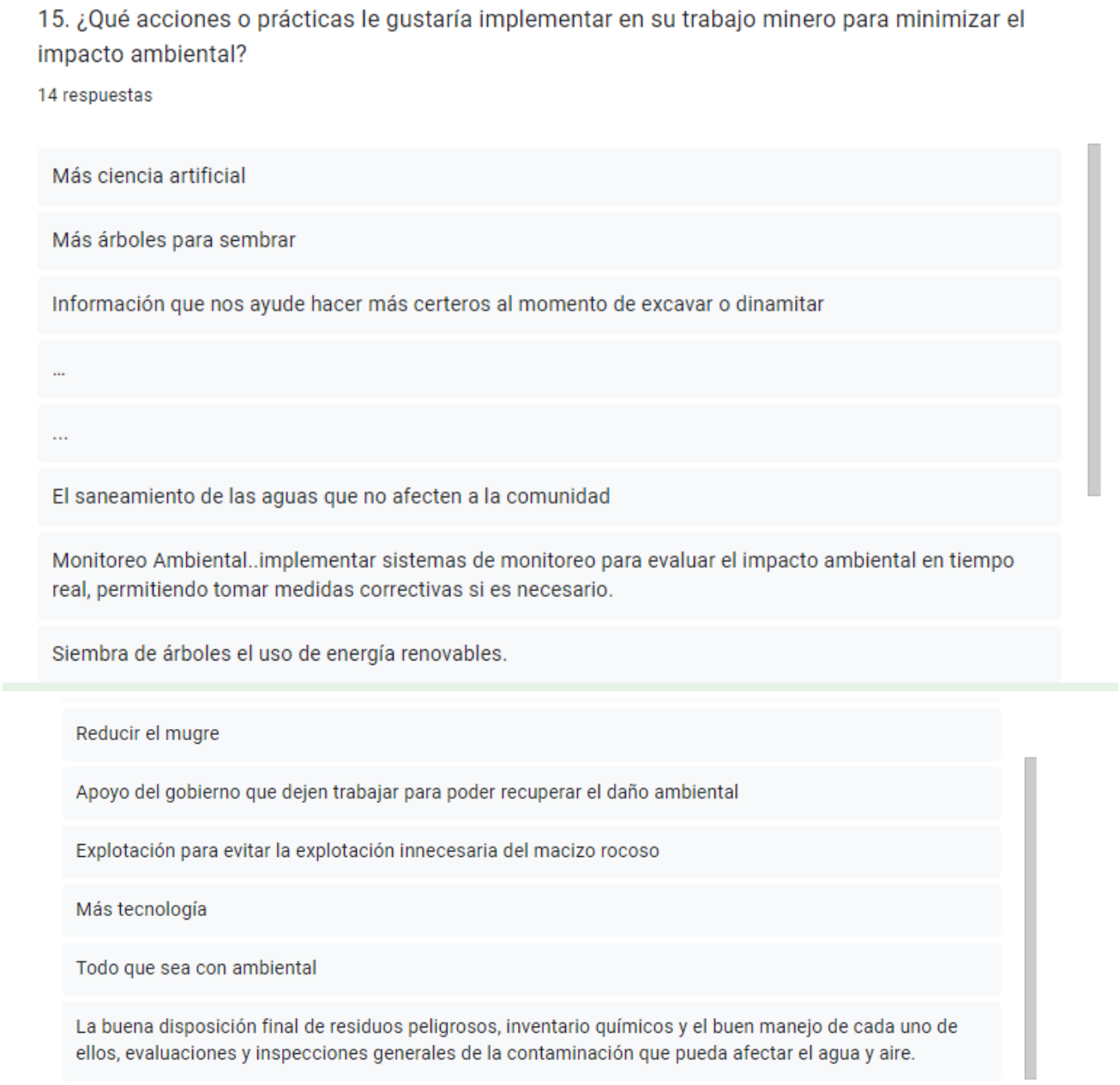


Ilustración 15. Prácticas sugeridas para minimizar impacto ambiental

Fuente: Elaboración propia

Análisis: Los mineros sugieren varias prácticas para minimizar el impacto ambiental: el uso de tecnología, la siembra de árboles, el uso de energías renovables, mejorar la precisión en las excavaciones y un manejo adecuado de los residuos para evitar contaminación del agua y del aire.

Conclusión de la Encuesta

Los resultados de la encuesta muestran que los mineros encuestados son conscientes del impacto ambiental de su trabajo y están interesados en hacer cambios para reducir este daño. Muchos están abiertos a implementar tecnologías y prácticas más responsables como el uso de energías renovables; además, muchos coinciden en que un software podría ayudarles a optimizar la detección de oro y a tomar decisiones con mayor información, lo cual beneficia tanto al medio ambiente como a los mineros. Por otro lado, aunque algunos ya usan detectores de oro, la mayoría no tiene acceso a este tipo de tecnología y considera útil la idea de un mapa interactivo y un foro para compartir información sobre las zonas exploradas.

Objetivo 2: Análisis

Tabla de Trazabilidad – Resultados de la Encuesta vs. Requisitos del Sistema

Nota. Relación entre los hallazgos de la encuesta y los requisitos funcionales del sistema.

N°	Hallazgo de la encuesta	Evidencia (resultado)	Necesidad identificada	Requisito funcional relacionado
1	Predomina experiencia intermedia y avanzada en minería	55,6% intermedio y 38,9% avanzado	El sistema debe ser funcional pero fácil de usar	RF-057 al RF-062 (Registro y acceso al sistema)
2	Mayoría realiza minería artesanal	61,1% minería artesanal	Software orientado a minería artesanal y pequeña escala	RF-076 al RF-081 (Gestión Minero)
3	Alta importancia en reducir impacto ambiental	94,4% lo considera muy importante	Control y seguimiento del impacto ambiental	RF-025 al RF-030 (Gestión Impacto Ambiental)
4	Principales problemas: falta de información y alta competencia	66,6% lo identifica como principal desafío	Acceso a información organizada y actualizada	RF-100 al RF-111 (Mapas) + RF-122 al RF-124 (Consultas)
5	Bajo uso de detectores de oro	76,5% los usa rara vez o nunca	Necesidad de apoyo informativo, no dependiente de hardware	RF-094 al RF-099 (Zonas/Minas exploradas)
6	Muy poco uso de software para minería	88,9% no usa aplicaciones	Sistema intuitivo y accesible	RF-057 al RF-062 (Registro usuario)
7	Alta aceptación de una aplicación para optimizar decisiones	61,1% cree que una app ayudaría	Desarrollo de una plataforma de apoyo a decisiones	RF-100 al RF-105 (Mapa interactivo)
8	Función más solicitada: mapa de zonas propensas	83,3% lo solicita	Visualización geográfica de información minera	RF-100 al RF-111 (Gestión Mapas)
9	Interés en información de zonas exploradas	61,1% lo solicita	Historial de exploración para evitar excavaciones repetidas	RF-094 al RF-099 (Minas/Zonas exploradas)
10	Interés en foro para compartir información	27,8% lo solicita	Espacio de colaboración entre mineros	RF-051 al RF-056 (Foro)
11	Preferencia por aplicación web y móvil	94,1% prefiere ambos	Sistema multiplataforma	RF-057 al RF-062 (Acceso web)
12	Disposición moderada a compartir información	44,4% cómodo, 33,3% poco cómodo	Control de permisos y acceso a información	RF-126 al RF-137 (Gestión Perfil y Usuario)
13	Interés en minimizar impacto	100% lo considera importante	Información ambiental integrada al sistema	RF-025 al RF-030 (Impacto Ambiental)

	ambiental sin afectar trabajo			
14	Uso de tecnologías variadas en minería	72,2% usa equipos de excavación	Sistema independiente del hardware	RF-119 al RF-121 (Informes)
15	Sugerencias: información precisa, monitoreo y buenas prácticas	Respuestas abiertas	Apoyo a decisiones responsables	RF-119 al RF-124 (Informes y Consultas)

Tabla 2. Resultados de la Encuesta vs. Requisitos del Sistema

Fuente: Elaboración propia

Listado de Necesidades y Características del Sistema

Nota. Necesidades (datos) y características (atributos) con las que funcionará el sistema.

Necesidades	Características	Estado
Zonas	Código zona, coordenadas, descripción, código departamento	Activo
Departamento	Código departamento, nombre, nombre municipio, nombre corregimiento	Activo
Impacto ambiental	Código impacto, impacto estimado, recomendaciones, código zona, código tipo impacto	Activo
Tipo de impacto ambiental	Código tipo de impacto, Nombre	Activo
Tipo de suelo	Código suelo, textura, estructura, porosidad, color, código zona	Activo
Tipos de excavación	Código tipo excavación, descripción, código zona	Activo
Foro	Código foro, tema, fecha, hora, descripción, localidad, código usuario	Activo
Registro de usuario	Código usuario, correo, contraseña	Activo
Detectores de oro	Código detector, marca, modelo, tipo de conexión (USB/Bluetooth), código tipo detector, código zona	Activo
Tipos de detectores de oro	Código detector de oro, descripción	Activo
Minero	IdMinero, nombre1, nombre2, apellido1, apellido2, teléfono, experiencia, código usuario, código mina	Activo
Empleado	Código empleado, tipo de empleado, área	Activo
Mina	Código mina, nombre, código departamento, código empleado	Activo
Información sobre zonas exploradas	Código zona explorada, características, descripción, código zona, código riesgo	Activo
Mapa	Código mapa, coordenadas, descripción, código zona, código tipo mapa, código riesgo	Activo
Tipos de mapa	Código tipos de mapa, descripción, características	Activo
Tipos de riesgos	Código riesgo, descripción, código zona	Activo

Filtros de mapa	Código filtro, nombre, código tipo excavación, código tipo mapa, código zona explorada	Activo
Excavación	Código excavación, código tipo excavación	Activo

Tabla 3. Listado de Necesidades y Características del Sistema

Fuente: Elaboración propia

Alcance del Sistema: Entradas, Procesos y Salidas

Nota. Entradas (datos que recibe el sistema), procesos (operaciones que realiza) y salidas (resultados generados) para el correcto funcionamiento del software.

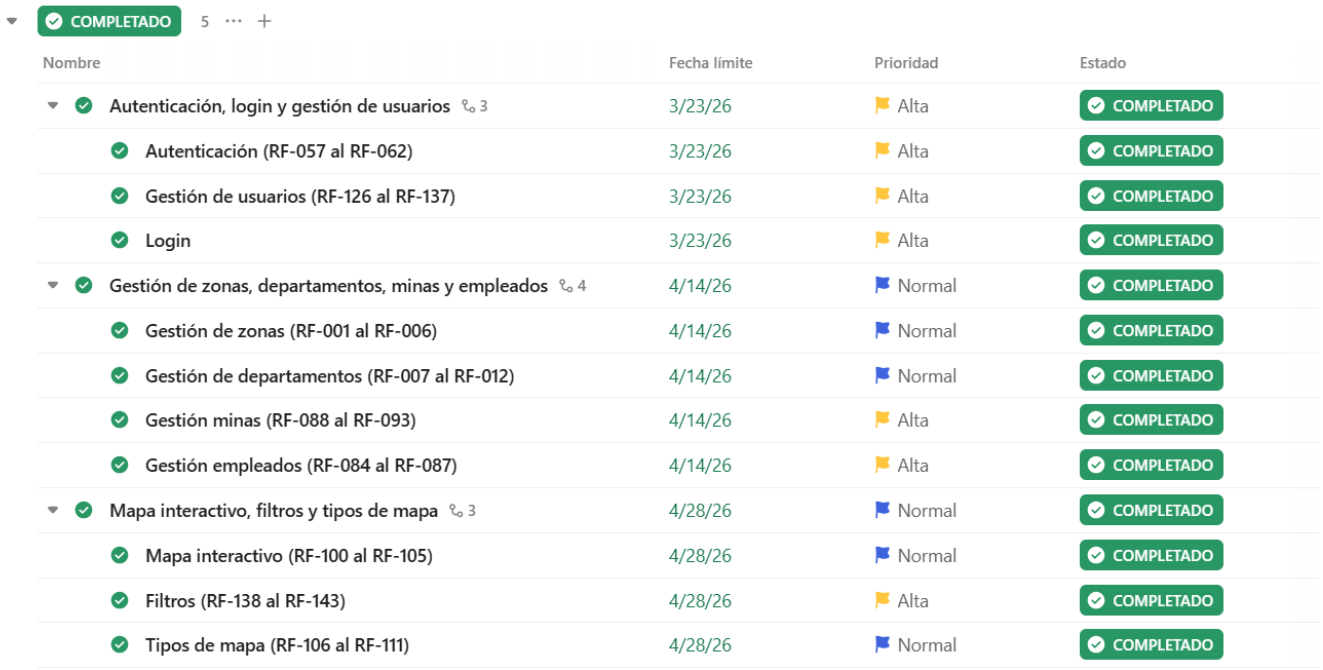
Entrada	Proceso	Salida
Zonas	Integración de un mapa interactivo para visualizar las zonas propensas a encontrar oro	Informe de Zonas priorizadas según información registrada
Departamento	Datos encontrados	Informe de impacto ambiental
Impacto ambiental	Análisis de datos arrojados por los detectores de oro	Informe de zonas exploradas durante el último mes
Tipo de impacto ambiental	Aplicación de filtros avanzados en el mapa interactivo	Consulta de comentarios y recomendaciones en el foro por zonas específicas
Tipo de suelo		Consulta de zonas con actividad reciente
Tipos de excavación		Consulta de zonas menos exploradas
Foro		Visualización e interacción con el mapa
Registro de usuario		Mapa interactivo con zonas exploradas
Tipos de detectores de oro		Visualización de zonas que cumplen con los criterios del filtro
Minero / Empleado / Mina		
Información sobre zonas exploradas		
Mapa / Tipos de mapa / Tipos de riesgo / Filtros / Excavación		

Tabla 4. Alcance del Sistema: Entradas, Procesos y Salidas

Fuente: Elaboración propia

Cronograma de Actividades (Sprint Backlog)

A continuación, se presenta el cronograma de actividades organizado en Sprints, herramienta de planificación que organiza y presenta las tareas y actividades necesarias para completar el proyecto dentro de un marco temporal específico.



Nombre	Fecha límite	Prioridad	Estado
COMPLETADO 5			
Autenticación, login y gestión de usuarios 3	3/23/26	Alta	COMPLETADO
Autenticación (RF-057 al RF-062)	3/23/26	Alta	COMPLETADO
Gestión de usuarios (RF-126 al RF-137)	3/23/26	Alta	COMPLETADO
Login	3/23/26	Alta	COMPLETADO
Gestión de zonas, departamentos, minas y empleados 4	4/14/26	Normal	COMPLETADO
Gestión de zonas (RF-001 al RF-006)	4/14/26	Normal	COMPLETADO
Gestión de departamentos (RF-007 al RF-012)	4/14/26	Normal	COMPLETADO
Gestión minas (RF-088 al RF-093)	4/14/26	Alta	COMPLETADO
Gestión empleados (RF-084 al RF-087)	4/14/26	Alta	COMPLETADO
Mapa interactivo, filtros y tipos de mapa 3	4/28/26	Normal	COMPLETADO
Mapa interactivo (RF-100 al RF-105)	4/28/26	Normal	COMPLETADO
Filtros (RF-138 al RF-143)	4/28/26	Alta	COMPLETADO
Tipos de mapa (RF-106 al RF-111)	4/28/26	Normal	COMPLETADO

Ilustración 16. Cronograma Sprint Backlog – Detalle

Fuente: Elaboración propia



Foro, detectores de oro, impacto ambiental y riesgos 4	5/12/26	Normal	COMPLETADO
Foro (RF-051 al RF-056)	5/12/26	Normal	COMPLETADO
Detectores de oro (RF-064 al RF-069)	5/12/26	Baja	COMPLETADO
Impacto ambiental (RF-025 al RF-030)	5/12/26	Alta	COMPLETADO
Riesgos (RF-113 al RF-118)	5/12/26	Alta	COMPLETADO
Informes, consultas, mineros y pruebas finales 4	5/26/26	Alta	COMPLETADO
Informes (RF-119 al RF-121)	5/26/26	Normal	COMPLETADO
Consultas (RF-122 al RF-124)	5/26/26	Alta	COMPLETADO
Mineros (RF-076 al RF-081)	5/26/26	Alta	COMPLETADO
Pruebas finales	5/26/26	Alta	COMPLETADO

Ilustración 17. Cronograma Sprint Backlog – Detalle

Fuente: Elaboración propia

Análisis de Riesgos

Es importante identificar de manera oportuna los diferentes tipos de riesgos que se pueden encontrar durante el proceso del desarrollo de un software ya que estos pueden afectar tanto el proyecto como el producto final esperado, así que es importante realizar un análisis para dar solución a posibles errores futuros. Según la revista AnyConnector (2021):

El riesgo tecnológico se refiere a la probabilidad de avería del software qie puede perjudicar el panorama de sistemas de una empresa. Con una evaluación adecuada, las organizaciones pueden mitigar estos riesgos tecnológicos y optimizar sus inteligencias de negocio.

Así que la identificación y gestión temprana de los riesgos tecnológicos en el desarrollo de software es primordial para minimizar el impacto “negativo” en el proyecto y en el producto final. Al anticipar estos posibles problemas, se pueden establecer soluciones efectivas de forma que protejan el rendimiento del software y la infraestructura tecnológica de una organización.

Riesgo cliente

Cuando pensamos en los riesgos de un proyecto de software, solemos enfocarnos en los aspectos técnicos y en el desarrollo, pero a veces olvidamos algo crucial: Los problemas pueden surgir desde el lado del cliente. Estos riesgos están relacionados con cómo el cliente usa y percibe el producto, y pueden afectar su satisfacción y la efectividad del software. Según Shugrue (s.f), “el código ejecutado en el dispositivo del usuario es completamente accesible y legible, los actores de amenazas pueden usar las instrucciones de acceso del servidor backend expuestas para lanzar varios ataques”. (p.1).

Por eso, es muy importante no solo centrarse en los aspectos técnicos del software, sino también en la interacción segura por parte del cliente con el producto final para evitar posibles riesgos debido a la falta de seguridad del proyecto. Al considerar desde el inicio los riesgos del lado

del cliente y aplicar medida adecuadas, podemos proteger tanto la experiencia de usuario como la integridad del software, evitando así vulnerabilidades.

Riesgo proceso

Cuando se habla de riesgo de proceso, se hace referencia a las amenazas o problemas que pueden surgir en las etapas y actividades relacionadas con la creación y gestión del proyecto. Estos riesgos, al igual que los mencionados anteriormente, afectan la eficiencia y la calidad del software, por lo tanto, con una buena planificación, comunicación y control de calidad, es posible minimizar estos imprevistos.

Riesgo producto

El riesgo del producto se refiere a la probabilidad de que el sistema o software no cumpla con alguna expectativa razonable del cliente, usuario o partes interesadas. (Algunos expertos también llaman a estos riesgos "riesgos de calidad", ya que impactan directamente en la calidad del producto). Revista QA (s.f).

Además, los riesgos del producto no solo afectan si los usuarios quedan contentos o no, sino que también pueden influir en si deciden recomendar el sistema o no, afectando así la reputación de la empresa o los programadores independientes que lo desarrollan y generar más costos a largo plazo al tener que hacer arreglos o brindar soporte adicional.

Estos riesgos pueden aparecer por no tener claro por ejemplo los requisitos desde el principio, no hacer suficientes pruebas o porque las expectativas del cliente cambian mientras se desarrolla el proyecto. Si se manejan bien estos riesgos, es posible detectarlos antes de que se conviertan en un dolor de cabeza, de tal forma que el producto final cumpla con lo que el cliente necesita y espera.

Nota. Este cuadro representa los tipos de riesgos dentro de un software, así como los riesgos que existen dentro de cada tipo, el impacto que tiene y que acciones se deben implementar para evitarlos o reducirlos.

Tipo de riesgo	Riesgos	Impacto	Acciones
Riesgo tecnológico	Seguridad cibernética	Pérdida de datos, daño a la reputación y costos legales	Implementar cortafuegos y sistemas de detección de intrusos
	Obsolescencia tecnológica	El software se vuelve menos eficiente y vulnerable	Mantenerse al día con tendencias tecnológicas y realizar actualizaciones regulares
	Pérdida de información	Interrupción de operaciones, daños financieros, pérdida de confianza	Implementar copias de seguridad constantes y automáticas
	Fallas de software	Interrupciones en el servicio y mala experiencia de usuario	Pruebas unitarias de integración y aceptación. Sistema de gestión de errores
	Interrupciones en la conectividad	Afecta la productividad al impedir el uso del software	Diseñar el software para que funcione sin conexión en modo limitado
Riesgo cliente	Mal uso del software	Insatisfacción del cliente y posibles fallos en el sistema	Proporcionar documentación y tutoriales de uso
	Inseguridad en el manejo de datos	Los datos pueden filtrarse o volverse vulnerables	Implementar guías sobre seguridad de datos y educar a los usuarios
	Desactualización del software	Vulnerabilidades y errores que afectan la funcionalidad	Recordatorios para actualizaciones y soporte para instalación de nuevas versiones
Riesgo proceso	Planificación inadecuada	Retrasos, sobrecostos y producto final de baja calidad	Planificación clara y ordenada desde el inicio con herramientas de gestión
	Falta de documentación	Dificulta la comprensión y mantenimiento del proyecto	Establecer una documentación clara y completa
	Dependencia de recursos externos	Interrumpe el flujo del proyecto y afecta la calidad	Acuerdos claros sobre plazos y estándares de calidad
	Falta de control de calidad	Errores y costos de mantenimiento elevados	Control de calidad constante: pruebas unitarias, de integración y aceptación
Riesgo producto	Experiencia deficiente del usuario	Frustración de los clientes por interfaz poco intuitiva	Involucrar usuarios reales en pruebas de usabilidad
	Falta de escalabilidad	Disminución del rendimiento o fallos a medida que la demanda aumenta	Diseñar el software con arquitecturas escalables desde el principio
	Problemas de rendimiento	Mala experiencia de usuario y baja adopción del producto	Pruebas de rendimiento y optimización durante el desarrollo
	Defectos y errores en el software	Mal funcionamiento y deterioro del producto	Pruebas de ensayo y error para realizar mejoras y evitar errores futuros

Tabla 5. Análisis de riesgos

Fuente: Elaboración propia

Tabla General para Casos de Uso

Nota. Gestiones del sistema con sus acciones CRUD y actores asociados.

Gestiones	Acciones	Actores		
		Administrador	**Usuario**	**Invitado**
Gestión zonas	Crear	X		
	Guardar	X		
	Consultar	X	x	
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	X	
Gestión Departamento	Crear	X		
	Guardar	X		
	Consultar	X	X	X
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	X
	Salir	X	X	X
Gestión impacto ambiental	Crear	X		
	Guardar	X		
	Consultar	X	X	X
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	x	X
	Salir	X	X	X

Gestión tipo de impacto ambiental	Crear	X		
	Guardar	X		
	Consultar	X	X	
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	X	
Gestión tipo de suelo	Crear	X		
	Guardar	X		
	Consultar	X	X	
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	x	
	Salir	X	X	
Gestión tipos de excavación	Crear	X		
	Guardar	X		
	Consultar	X	X	
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	X	
Gestión foro	Crear	X	X	
	Guardar	X	X	
	Consultar	X	X	X
	Modificar	X	X	
	Inhabilitar	X		

	Cancelar	X	X	x
	Salir	X	X	X
Gestión registro usuario	Crear	X		X
	Guardar	X	X	X
	Consultar	X		
	Modificar	X	X	X
	Inhabilitar	X		
	Cancelar	X	X	X
	Salir	X	X	X
Gestión integración detectores de oro	Crear	X		
	Guardar	X		
	Consultar	X		
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X		
	Salir	X		
Gestión tipos de detectores de oro	Crear	X		
	Guardar	X		
	Consultar	X	X	X
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	X
	Salir	X	X	x
	Crear	X	X	
	Guardar	X	X	
	Consultar	X		

Gestión minero	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	X	
Gestión empleados	Crear	X	X	
	Guardar	X	X	
	Consultar	X		
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	x	
Gestión mina	Crear	X		
	Guardar	X		
	Consultar	X	x	
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	X	
Gestión minas exploradas	Crear	X	X	
	Guardar	X	X	
	Consultar	X	X	
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	X	
	Crear	X		
	Guardar	X		

Gestión mapas	Consultar	X	X	
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	X	
Gestión tipos de mapa	Crear	X		
	Guardar	X		
	Consultar	X	X	
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	x	x	
Gestión riesgos	Crear	X		
	Guardar	X		
	Consultar	X	X	
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	X	
Gestión informes	Impacto ambiental	X	X	
	Zonas con alta probabilidad de oro	X	X	
	Zonas exploradas durante el último mes	X	x	
Gestión consultas	Comentarios y recomendaciones	X	X	
	Zonas con actividad reciente	X	X	

	Zonas menos exploradas	X	x	
Gestión perfil	Crear	X		
	Guardar	X		
	Consultar	X		
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X		
	Salir	X		
Gestión usuario	Crear	X		
	Guardar	X		
	Consultar	X		
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X		
	Salir	x		
Gestión filtros	Crear	x		
	Guardar	X		
	Consultar	X	x	
	Modificar	X		
	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	X	
Gestión Excavación	Crear	X		
	Guardar	X		
	Consultar	X	X	
	Modificar	X		

	Inhabilitar	X		
	Cancelar	X	X	
	Salir	X	X	

Tabla 6. Casos de Uso

Fuente: Elaboración propia

Historias de Usuario

Una historia de usuario se refiere a la interacción que tendrá el sistema con el usuario, de tal manera que se satisfaga cada una de las necesidades del usuario y/o cliente. En cada historia se especifica el tipo de usuario que participa, para qué se quiere realizar esta historia y qué resultado espera obtener, además cada historia debe cumplir con el RF (Requisito Funcional) al que hace referencia.

Fuente: Elaboración propia

Nota. Historias de usuario del sistema SoftGold con criterios de aceptación y referencias funcionales.

ID	Historia de Usuario	Criterios de Aceptación	RF Relacionado
HU-001	Como minero, quiero consultar el mapa interactivo con zonas propensas a encontrar oro, para reducir excavaciones innecesarias y el impacto ambiental.	El sistema muestra un mapa con zonas marcadas. El usuario puede hacer zoom y seleccionar una zona para ver su detalle. Si no existen zonas registradas, el sistema muestra un mensaje informativo.	RF-101
HU-002	Como usuario, quiero publicar un comentario en el foro indicando la zona y mi experiencia, para que otros mineros puedan beneficiarse de esa información.	El sistema valida que el usuario esté autenticado. Se registra automáticamente la fecha y hora del comentario. El comentario queda visible para los demás usuarios.	RF-051
HU-003	Como administrador, quiero registrar el impacto ambiental de una zona con sus recomendaciones, para llevar un control del daño generado.	El sistema valida que la zona exista. Permite registrar el tipo de impacto y las recomendaciones. El sistema confirma el registro con un mensaje de éxito.	RF-025
HU-004	Como usuario, quiero consultar las zonas menos exploradas, para identificar nuevas oportunidades de extracción con menor competencia.	El sistema muestra un listado de zonas con menor actividad registrada. El usuario puede consultar el detalle de cada zona.	RF-124
HU-005	Como administrador, quiero inhabilitar un minero con confirmación previa, para evitar accesos no autorizados al sistema.	El sistema muestra un mensaje de confirmación antes de la acción. Al confirmar, el estado del minero cambia a inactivo y se registra la acción.	RF-079

HU-006	Como administrador, quiero gestionar zonas mineras (crear, consultar, modificar e inhabilitar), para mantener actualizada la información del sistema.	El sistema permite registrar, consultar, modificar e inhabilitar zonas. Valida los campos obligatorios y confirma cada operación realizada.	RF-001
HU-007	Como administrador, quiero gestionar departamentos, para organizar las zonas mineras por ubicación geográfica.	El sistema permite crear, consultar, modificar e inhabilitar departamentos. Las zonas pueden asociarse a un departamento existente.	RF-007
HU-008	Como administrador, quiero gestionar minas, para llevar un control de las minas registradas en el sistema.	El sistema permite crear, consultar, modificar e inhabilitar minas. Cada mina queda asociada a una zona registrada.	RF-088
HU-009	Como usuario, quiero aplicar filtros en el mapa, para visualizar únicamente las zonas que me interesan.	El sistema permite seleccionar filtros y actualiza el mapa según los criterios elegidos.	RF-139
HU-010	Como usuario, quiero consultar informes del sistema, para apoyar la toma de decisiones.	El sistema permite generar y visualizar informes según el tipo seleccionado. Los datos mostrados corresponden a la información registrada.	RF-119
HU-011	Como administrador, quiero gestionar detectores de oro, para llevar un control de los dispositivos registrados en el sistema.	El sistema permite crear, consultar, modificar e inhabilitar detectores. Cada acción genera una confirmación.	RF-064
HU-012	Como usuario, quiero consultar riesgos asociados a una zona, para tomar precauciones antes de realizar excavaciones.	El sistema muestra los riesgos registrados por zona. El usuario puede consultar el detalle de cada riesgo.	RF-114

Tabla 7. Historias de usuario del sistema SoftGold

Fuente: Elaboración propia

Requisitos de Usuario (RU)

Nota: Esta tabla describe las acciones que se realizará en cada gestión, así como los usuarios que participarán en cada una de estas, adicionalmente, Los saltos de RU-003 y RU-004 se deben a una nueva versión del documento en donde se eliminaron RU anteriores.

Id	Nombre del requisito	Descripción	Usuario	Prioridad
RU-001	**Gestión zonas**	El sistema permitirá la gestión de la información de las zonas para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Media
RU-002	**Gestión Departamento**	El sistema permitirá la gestión de la información de los departamentos para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario, Invitado	Media
RU-005	**Gestión impacto ambiental**	El sistema permitirá la gestión de la información del impacto ambiental para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario, Invitado	Alta

RU-006	**Gestión tipo de impacto ambiental**	El sistema permitirá la gestión de la información del tipo de impacto ambiental para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Media
RU-007	**Gestión tipo de suelo**	El sistema permitirá la gestión de la información de los tipos de suelo para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Alta
RU-008	**Gestión tipos de excavación**	El sistema permitirá la gestión de la información de los tipos de excavación para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Media
RU-009	**Gestión foro**	El sistema permitirá la gestión de la información del foro para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario, Invitado	Media
RU-010	**Gestión registro usuario**	El sistema permitirá la gestión del registro de los usuarios para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario, Invitado	Alta
RU-011	**Gestión de detectores de oro**	El sistema permitirá registrar y gestionar la información de los detectores de oro (Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir).	Administrador	Baja
RU-012	**Gestión tipos de detectores de oro**	El sistema permitirá la gestión de la información de los tipos de detectores de oro para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario, Invitado	Baja
RU-013	**Gestión minero**	El sistema permitirá la gestión de la información de los mineros para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Alta
RU-014	**Gestión empleado**	El sistema permitirá la gestión de la información de los empleados para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Alta
RU-015	**Gestión mina**	El sistema permitirá la gestión de la información de las minas para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Alta
RU-016	**Gestión minas exploradas**	El sistema permitirá la gestión de la información de las minas exploradas para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Media
RU-017	**Gestión mapas**	El sistema permitirá la gestión de la información de los mapas para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Alta
RU-018	**Gestión tipos de mapas**	El sistema permitirá la gestión de la información de los tipos de mapas para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Media
RU-019	**Gestión riesgos**	El sistema permitirá la gestión de la información de los riesgos para Crear, Modificar, Inhabilitar, Consultar, Cancelar y Salir	Administrador, Usuario	Alta

RU-020	**Gestión informes**	El sistema permitirá la ejecución de los informes (Impacto ambiental, Zonas priorizadas según información registrada, zonas exploradas durante el último mes)	Administrador, Usuario	Media
RU-021	**Gestión consultas**	El sistema permitirá la ejecución de las consultas (Comentarios y recomendaciones, zonas con actividad reciente, zonas menos exploradas)	Administrador, Usuario	Alta
RU-022	**Gestión perfil**	El sistema permitirá la gestión de la información del perfil para Crear, Modificar, Inhabilitar, Consultar, Cancelar, Salir	Administrador	Alta
RU-023	**Gestión usuario**	El sistema permitirá la gestión de la información del usuario para Crear, Modificar, Inhabilitar, Consultar, Cancelar, Salir	Administrador	Alta
RU-024	**Gestión filtro**	El sistema permitirá la gestión de la información del filtro para Crear, Modificar, Inhabilitar, Consultar, Cancelar, Salir	Administrador, Usuario	Media
RU-025	**Gestión excavación**	El sistema permitirá la gestión de la información de la excavación para Crear, Modificar, Inhabilitar, Consultar, Cancelar, Salir	Administrador, Usuario	Baja

Tabla 8. Requisitos de Usuario (RU)

Fuente: Elaboración propia

Requisitos Funcionales (RF)

Nota. Especificación del CRUD de cada gestión, con actores y referencias al RU correspondiente. Los saltos de numeración (RF-013–RF-024, RF-037–RF-050, RF-063–RF-112 y RF-125) se deben a una versión previa donde se eliminaron RF anteriores.

Requisitos funcionales (RF)				
Id RF	**Nombre del requisito**	**Descripción del Requisito**	**Usuario**	**Id RU**
RF-001	Crear Zona	Permite registrar la información de las zonas con los siguientes datos: Código zona, coordenadas, descripción, tipo de suelo, departamento, municipio, corregimiento. A través de la opción guardar.	Administrador	RU-001
RF-002	Consultar Zona	Permite consultar la información de las zonas por los campos código zona y zona	Administrador, Usuario	RU-001
RF-003	Modificar Zona	Permite modificar la información de la zona a excepción del código	Administrador	RU-001

		zona, utilizando la opción guardar		
RF-004	Inhabilitar Zona	Permite habilitar o inhabilitar una zona con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-001
RF-005	Cancelar Zona	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-001
RF-006	Salir Zona	Permite cerrar el formulario	Administrador, Usuario	RU-001
RF-007	Crear Departamento	Permite registrar la información de los departamentos con los siguientes datos: Código departamento, nombre A través de la opción guardar.	Administrador	RU-002
RF-008	Consultar Departamento	Permite consultar la información del departamento por los campos código departamento y nombre	Administrador, Usuario, Invitado	RU-002
RF-009	Modificar Departamento	Permite modificar la información del departamento a excepción del código departamento, utilizando la opción guardar	Administrador	RU-002
RF-010	Inhabilitar Departamento	Permite habilitar o inhabilitar un departamento con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-002
RF-011	Cancelar Departamento	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario, Invitado	RU-002
RF-012	Salir Departamento	Permite cerrar el formulario	Administrador, Usuario, Invitado	RU-002
RF-025	Crear Impacto Ambiental	Permite registrar la información del Impacto ambiental siguientes datos: Código impacto, impacto estimado, recomendaciones, zona afectada, tipos de impacto ambiental. A través de la opción guardar.	Administrador	RU-005
RF-026	Consultar Impacto Ambiental	Permite consultar la información del impacto ambiental por los campos: Código impacto y zona afectada	Administrador, Usuario, Invitado	RU-005

RF-027	Modificar Impacto Ambiental	Permite modificar la información del impacto ambiental a excepción del código código impacto, utilizando la opción guardar	Administrador	RU-005
RF-028	Inhabilitar Impacto Ambiental	Permite habilitar o inhabilitar un impacto ambiental con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-005
RF-029	Cancelar Impacto Ambiental	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario, Invitado	RU-005
RF-030	Salir Impacto Ambiental	Permite cerrar el formulario	Administrador, Usuario, Invitado	RU-005
RF-031	Crear Tipo Impacto Ambiental	Permite registrar la información de los tipos de Impacto ambiental siguientes datos: Código tipo impacto, Nombre, Descripción. A través de la opción guardar.	Administrador	RU-006
RF-032	Consultar Tipo Impacto Ambiental	Permite consultar la información del tipo de impacto ambiental por los campos: Código tipo impacto y Nombre	Administrador, Usuario	RU-006
RF-033	Modificar Tipo Impacto Ambiental	Permite modificar la información del tipo de impacto ambiental a excepción del código tipo impacto, utilizando la opción guardar	Administrador	RU-006
RF-034	Inhabilitar Tipo Impacto Ambiental	Permite habilitar o inhabilitar un tipo de impacto ambiental con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-006
RF-035	Cancelar Tipo Impacto Ambiental	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-006
RF-036	Salir Tipo Impacto Ambiental	Permite cerrar el formulario	Administrador, Usuario	RU-006
RF-038	Crear Tipo de suelo	Permite registrar la información de los tipos de suelo siguientes datos: Código tipo suelo, textura, estructura, porosidad, color. A través de la opción guardar.	Administrador	RU-007

RF-039	Consultar Tipo de suelo	Permite consultar la información del tipo de suelo por el campo: Código tipo suelo	Administrador, Usuario	RU-007
RF-040	Modificar Tipo de suelo	Permite modificar la información del tipo de suelo a excepción del código suelo, utilizando la opción guardar	Administrador	RU-007
RF-041	Inhabilitar Tipo de suelo	Permite habilitar o inhabilitar un tipo de suelo con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-007
RF-042	Cancelar Tipo de suelo	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-007
RF-043	Salir Tipo de suelo	Permite cerrar el formulario	Administrador, Usuario	RU-007
RF-044	Crear Tipos de excavación	Permite registrar la información de los tipos de excavación con los siguientes datos: Código tipo, descripción. A través de la opción guardar.	Administrador	RU-008
RF-045	Consultar Tipos de excavación	Permite consultar la información del tipo de excavación por el campo: Código tipo	Administrador, Usuario	RU-008
RF-046	Modificar Tipos de excavación	Permite modificar la información del tipo de excavación a excepción del código tipo, utilizando la opción guardar	Administrador	RU-008
RF-047	Inhabilitar Tipos de excavación	Permite habilitar o inhabilitar un tipo de excavación con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-008
RF-048	Cancelar Tipos de excavación	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-008
RF-049	Salir Tipos de excavación	Permite cerrar el formulario	Administrador, Usuario	RU-008
RF-051	Crear Foro	Permite registrar la información de los foros con los siguientes datos: Código foro, tema, fecha, hora, IdUsuario, comentarios, localidad. A través de la opción guardar.	Administrador, Usuario	RU-009

RF-052	Consultar Foro	Permite consultar la información del foro por los campos: Código foro y tema	Administrador, Usuario, Invitado	RU-009
RF-053	Modificar Foro	Permite modificar la información del foro a excepción del código foro, utilizando la opción guardar	Administrador, Usuario	RU-009
RF-054	Inhabilitar Foro	Permite habilitar o inhabilitar un foro con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-009
RF-055	Cancelar Foro	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario, Invitado	RU-009
RF-056	Salir Foro	Permite cerrar el formulario	Administrador, Usuario, Invitado	RU-009
RF-057	Crear registro usuario	Permite registrar la información del usuario con los siguientes datos: IdUsuario, Nombre, contraseña. A través de la opción guardar.	Administrador, Invitado	RU-010
RF-058	Consultar registro usuario	Permite consultar la información del usuario por el campo: IdUsuario	Administrador	RU-010
RF-059	Modificar registro usuario	Permite modificar la información del usuario a excepción del IdUsuario, utilizando la opción guardar	Administrador, Usuario, invitado	RU-010
RF-060	Inhabilitar registro usuario	Permite habilitar o inhabilitar un usuario con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-010
RF-061	Cancelar registro usuario	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario, Invitado	RU-010
RF-062	Salir registro usuario	Permite cerrar el formulario	Administrador, Usuario, Invitado	RU-010
RF-064	Crear detector de oro	Crear detector de oro — Permite registrar la información del detector de oro con los siguientes datos: código, marca, modelo, tecnología y tipo de conexión.	Administrador	RU-011
RF-065	Consultar detectores de oro	Permite consultar la información del detector de oro	Administrador	RU-011

		por los campos: Código detector y marca		
RF-066	Modificar detectores de oro	Permite modificar la información del detector de oro a excepción del Código detector, utilizando la opción guardar	Administrador	RU-011
RF-067	Inhabilitar detectores de oro	Permite habilitar o inhabilitar un detector de oro con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-011
RF-068	Cancelar detectores de oro	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador	RU-011
RF-069	Salir detectores de oro	Permite cerrar el formulario	Administrador	RU-011
RF-070	Crear Tipos de detectores de oro	Permite registrar la información del tipo de detector de oro con los siguientes datos: Código detector de oro, descripción. A través de la opción guardar.	Administrador	RU-012
RF-071	Consultar Tipos de detectores de oro	Permite consultar la información del tipo de detector de oro por el campo: Código detector de oro	Administrador, Usuario, Invitado	RU-012
RF-072	Modificar Tipos de detectores de oro	Permite modificar la información del tipo de detector de oro a excepción del Código detector de oro, utilizando la opción guardar	Administrador	RU-012
RF-073	Inhabilitar Tipos de detectores de oro	Permite habilitar o inhabilitar un tipo de detector de oro con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-012
RF-074	Cancelar Tipos de detectores de oro	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario, Invitado	RU-012
RF-075	Salir Tipos de detectores de oro	Permite cerrar el formulario	Administrador, Usuario, Invitado	RU-012
RF-076	Crear Minero	Permite registrar la información del minero con los siguientes datos: Id Minero, Nombre1, Nombre2, Apellido1, Apellido2, Teléfono, Experiencia. A través de la opción guardar.	Administrador, Usuario	RU-013

RF-077	Consultar Minero	Permite consultar la información del minero por el campo: Id Minero	Administrador	RU-013
RF-078	Modificar Minero	Permite modificar la información del minero a excepción del Id Minero, utilizando la opción guardar	Administrador	RU-013
RF-079	Inhabilitar Minero	Permite habilitar o inhabilitar un minero con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-013
RF-080	Cancelar Minero	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-013
RF-081	Salir Minero	Permite cerrar el formulario	Administrador, Usuario	RU-013
RF-082	Crear Empleados	Permite registrar la información de los empleados con los siguientes datos: Código empleado, tipo de empleado, área, código mina. A través de la opción guardar.	Administrador, Usuario	RU-014
RF-083	Consultar Empleados	Permite consultar la información del empleado por el campo: código empleado	Administrador	RU-014
RF-084	Modificar Empleados	Permite modificar la información del empleado a excepción del campo código empleado, utilizando la opción guardar	Administrador	RU-014
RF-085	Inhabilitar Empleados	Permite habilitar o inhabilitar un empleado con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-014
RF-086	Cancelar Empleados	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-014
RF-087	Salir Empleados	Permite cerrar el formulario	Administrador, Usuario	RU-014
RF-088	Crear Mina	Permite registrar la información de las minas con los siguientes datos: Código mina, Nombre, código departamento, Código municipio, Código corregimiento A través de la opción guardar.	Administrador,	RU-015

RF-089	Consultar Mina	Permite consultar la información de la mina por los campos: código Mina y Nombre	Administrador, Usuario	RU-015
RF-090	Modificar Mina	Permite modificar la información de la mina a excepción del campo código Mina, utilizando la opción guardar	Administrador	RU-015
RF-091	Inhabilitar Mina	Permite habilitar o inhabilitar una mina con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-015
RF-092	Cancelar Mina	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-015
RF-093	Salir Mina	Permite cerrar el formulario	Administrador, Usuario	RU-015
RF-094	Crear minas exploradas	Permite registrar la información de las minas exploradas con los siguientes datos: Código zona explorada, características, descripción. A través de la opción guardar.	Administrador, Usuario	RU-016
RF-095	Consultar minas exploradas	Permite consultar la información de la mina explorada por los campos: Código zona y caracteristicas	Administrador, Usuario	RU-016
RF-096	Modificar mina exploradas	Permite modificar la información de la mina explorada a excepción del campo código zona explorada, utilizando la opción guardar	Administrador	RU-016
RF-097	Inhabilitar mina exploradas	Permite habilitar o inhabilitar una mina explorada con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-016
RF-098	Cancelar mina exploradas	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-016
RF-099	Salir Zonas minas	Permite cerrar el formulario	Administrador, Usuario	RU-016
RF-100	Crear Mapas	Permite registrar la información de los mapas con los siguientes datos: Código mapa, coordenadas, zona, tipos de mapa, comentarios del foro. A través de la opción guardar	Administrador	RU-017

RF-101	Consultar Mapas	Permite consultar la información de los mapas por el campo: Código mapa y zona	Administrador, Usuario	RU-017
RF-102	Modificar Mapas	Permite modificar la información de los mapas a excepción del campo código mapa, utilizando la opción guardar	Administrador	RU-017
RF-103	Inhabilitar Mapas	Permite habilitar o inhabilitar un mapa con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-017
RF-104	Cancelar Mapas	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-017
RF-105	Salir Mapas	Permite cerrar el formulario	Administrador, Usuario	RU-017
RF-106	Crear Tipos de mapas	Permite registrar la información de los tipos de mapas con los siguientes datos: Código tipo mapa, descripción, características, tipo de mapa. A través de la opción guardar	Administrador	RU-018
RF-107	Consultar Tipos de Mapas	Permite consultar la información de los tipos de mapas por el campo: Código tipo mapa	Administrador, Usuario	RU-018
RF-108	Modificar Tipos de Mapas	Permite modificar la información de los tipos de mapas a excepción del campo código tipo mapa, utilizando la opción guardar	Administrador	RU-018
RF-109	Inhabilitar Tipos de Mapas	Permite habilitar o inhabilitar un tipo de mapa con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-018
RF-110	Cancelar Tipos de Mapas	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-018
RF-111	Salir Tipos de Mapas	Permite cerrar el formulario	Administrador, Usuario	RU-018
RF-113	Crear Riesgos	Permite registrar la información de los tipos de riesgos con los siguientes datos: Código riesgo, descripción. A través de la opción guardar	Administrador	RU-019

RF-114	Consultar Riesgos	Permite consultar la información de los tipos de riesgos por el campo: Código riesgo	Administrador, Usuario	RU-019
RF-115	Modificar Riesgos	Permite modificar la información de los tipos de riesgos a excepción del campo código riesgo, utilizando la opción guardar	Administrador	RU-019
RF-116	Inhabilitar Riesgos	Permite habilitar o inhabilitar un tipo de riesgo con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-019
RF-117	Cancelar Riesgos	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-019
RF-118	Salir Riesgos	Permite cerrar el formulario	Administrador, Usuario	RU-019
RF-119	Informe impacto ambiental	Permite tener acceso a la información del impacto ambiental	Administrador, Usuario	RU-020
RF-120	Informe Zonas priorizadas según información registrada	Permite consultar las zonas priorizadas a partir de la información de exploración registrada por los usuarios en el sistema	Administrador, Usuario	RU-020
RF-121	Informe zonas exploradas durante el último mes	Permite tener acceso a la información a zonas exploradas durante el último mes	Administrador, Usuario	RU-020
RF-122	Consulta de comentarios y recomendaciones	Permite consultar los comentarios y recomendaciones por otros mineros	Administrador, Usuario	RU-021
RF-123	Consulta de zonas con actividad reciente	Permite consultar las zonas que más se han visitado y dónde ha habido mayor actividad	Administrador, Usuario	RU-021
RF-124	Consulta de zonas menos exploradas	Permite consultar las zonas que menos exploran los mineros	Administrador, Usuario	RU-021
RF-126	Crear perfil	Permite registrar la información del perfil. A través de la opción guardar	Administrador	RU-022
RF-127	Consultar perfil	Permite consultar la información del perfil	Administrador	RU-022

RF-128	Modificar perfil	Permite modificar la información del perfil, utilizando la opción guardar	Administrador	RU-022
RF-129	Inhabilitar perfil	Permite habilitar o inhabilitar un perfil con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-022
RF-130	Cancelar perfil	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador	RU-022
RF-131	Salir perfil	Permite cerrar el formulario	Administrador	RU-022
RF-132	Crear Usuario	Permite registrar la información del usuario. A través de la opción guardar	Administrador, invitado	RU-023
RF-133	Consultar usuario	Permite consultar la información del usuario	Administrador	RU-023
RF-134	Modificar usuario	Permite modificar la información del usuario, utilizando la opción guardar	Administrador, usuario, invitado	RU-023
RF-135	Inhabilitar usuario	Permite habilitar o inhabilitar un usuario con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-023
RF-136	Cancelar Usuario	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, usuario, invitado	RU-023
RF-137	Salir usuario	Permite cerrar el formulario	Administrador, usuario, invitado	RU-023
RF-138	Crear filtro	Permite registrar los filtros. A través de la opción guardar	Administrador	RU-024
RF-139	Consultar filtro	Permite consultar zonas específicas en el mapa según los filtros aplicados	Administrador, Usuario	RU-024
RF-140	Modificar filtro	Permite modificar los filtros ya creados, utilizando la opción guardar	Administrador	RU-024
RF-141	Inhabilitar filtro	Permite habilitar o inhabilitar un filtro con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-024

RF-142	Cancelar filtro	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, Usuario	RU-024
RF-143	Salir filtro	Permite salir de la función	Administrador, Usuario	RU-024
RF-144	Crear excavación	Permite registrar las excavaciones. A través de la opción guardar	Administrador	RU-025
RF-145	Consultar excavación	Permite consultar excavaciones	Administrador, Usuario	RU-025
RF-146	Modificar excavación	Permite modificar la información de la excavación, utilizando la opción guardar	Administrador	RU-025
RF-147	Inhabilitar excavación	Permite habilitar o inhabilitar una excavación con previa confirmación de realizar el proceso y verificación de la información dependiente.	Administrador	RU-025
RF-148	Cancelar excavación	Permite cancelar el proceso que se esté realizando y regresar al estado inicial	Administrador, usuario	RU-025
RF-149	Salir excavación	Permite cerrar el formulario	Administrador, usuario	RU-025

Tabla 9. Requisitos Funcionales (RF)

Fuente: Elaboración propia

Trazabilidad entre Historias de Usuario (HU) y Requisitos Funcionales (RF)

La trazabilidad garantiza la coherencia entre el análisis de requerimientos y la fase de construcción del sistema SoftGold.

Nota. El requisito funcional relacionado corresponde al requisito principal que satisface la historia de usuario; sin embargo, cada historia puede involucrar otros requisitos funcionales complementarios, los cuales se detallan en la tabla de trazabilidad.

Historia de Usuario (HU)	Descripción	Requisitos Funcionales (RF) asociados
HU – 001	Consulta de mapa interactivo	RF-101, RF-100, RF-139
HU – 002	Publicación de comentarios en el foro	RF-051, RF-052
HU – 003	Registro de impacto ambiental	RF-025, RF-026

HU – 004	Consulta de zonas menos exploradas	RF-124
HU – 005	Inhabilitar minero	RF-079
HU – 006	Gestión de zonas mineras	RF-001, RF-002, RF-003, RF-004
HU – 007	Gestión de departamentos	RF-007, RF-008, RF-009, RF-010
HU – 008	Gestión de minas	RF-088, RF-089, RF-090, RF-091
HU – 009	Aplicación de filtros en mapa	RF-138, RF-139
HU – 010	Consulta de informes	RF-119, RF-120, RF-121
HU – 011	Gestión de detectores de oro	RF-064, RF-065, RF-066, RF-067
HU – 012	Gestión de riesgos por zona	RF-114

Tabla 10. Trazabilidad historias de usuario y requisitos funcionales

Fuente: Elaboración propia

Requisitos No Funcionales (RNF)

Nota. Requisitos no funcionales del sistema SoftGold.

Según Metaute (2015):

> "El requisito no funcional, está relacionado con todo aquello que no solicita directamente el cliente, en relación con lo que debe llevar o cumplir el sistema. Los RNF, tienen que ver con características externas de calidad". (p.87).

ID RNF	Requisito	Descripción del requisito
RNF-001	Usabilidad	El sistema debe contar con un diseño intuitivo y fácil de usar, permitiendo a los usuarios comprender su funcionamiento sin necesidad de conocimientos técnicos avanzados.
RNF-002	Capacitación	El sistema debe contar con capacitación básica y guías de uso para los usuarios, con el fin de facilitar su correcta utilización.
RNF-003	Disponibilidad	El sistema debe estar disponible las 24 horas del día, los 7 días de la

		semana, salvo periodos de mantenimiento programado.
RNF-004	Actualización de información	El sistema debe asegurar la actualización permanente de la información registrada en la base de datos.
RNF-005	Rendimiento	El tiempo de respuesta del sistema debe ser en promedio menor o igual a 1 segundo para operaciones de consulta, y no superar los 2 segundos en el percentil 95 (P95) bajo condiciones de carga normal.
RNF-006	Tecnologías de desarrollo	El sistema será desarrollado utilizando programación orientada a objetos y tecnologías de software como Java y Spring Boot, junto con una base de datos SQL.
RNF-007	Capacidad de almacenamiento	El sistema debe contar con suficiente espacio de almacenamiento para soportar grandes volúmenes de información.
RNF-008	Compatibilidad del entorno	El sistema debe poder ejecutarse correctamente en equipos de cómputo con configuraciones estándar.
RNF-009	Seguridad de credenciales	Las contraseñas de los usuarios deben ser almacenadas de forma encriptada mediante algoritmos criptográficos seguros.
RNF-010	Respaldo de información	El sistema debe permitir la realización de copias de seguridad de la información para garantizar su recuperación ante fallos.
RNF-011	Auditoría	Todas las operaciones de creación, modificación e inhabilitación de información crítica deben quedar registradas para fines de auditoría.
RNF-012	Documentación de usuario	El sistema debe contar con manuales de usuario que expliquen su funcionamiento y principales funcionalidades.
RNF-013	Interfaz de usuario	La interfaz del sistema debe ser accesible desde un computador de escritorio y utilizar colores y tipografías que faciliten la lectura.
RNF-014	Acceso web	El acceso al sistema debe realizarse vía web, garantizando estabilidad y seguridad en la comunicación.
RNF-015	Comunicación interfaz-usuario	El sistema debe garantizar una correcta comunicación entre la interfaz de usuario y los servicios del sistema.

Tabla 11. Requisitos no funcionales del sistema SoftGold.

Fuente: Piedad María Metaute

Objetivo 3: Diseño

Descripción General de Actores

Nota. Actores del sistema SoftGold, sus roles y ejemplos reales de uso.

Actor	Descripción	Ejemplo real
Administrador	Usuario con permisos totales sobre el sistema. Crea, consulta, modifica e inhabilita la información de los distintos módulos y gestiona usuarios y configuraciones generales.	Docente administrador o encargado del proyecto que gestiona la información minera y ambiental.
Minero	Usuario registrado del dominio que consulta zonas, mapas y riesgos, participa en el foro y comparte experiencias de exploración.	Minero artesanal del Nordeste Antioqueño que utiliza SoftGold para consultar zonas y compartir información.
Empleado	Usuario interno asociado a una mina, que apoya el registro y la gestión de la información operativa según su área.	Trabajador vinculado a una mina que registra información de exploración o impacto ambiental.
Invitado	Usuario no autenticado con acceso limitado a información general, sin posibilidad de modificar datos.	Persona interesada en conocer información general del sistema sin necesidad de registro.

Tabla 12. Descripción General de Actores

Fuente: Elaboración propia

Diagramas del Modelo de Casos de Uso

Los diagramas de casos de uso del sistema SoftGold fueron ajustados siguiendo la semántica correcta de las relaciones UML.

La relación <<include>> se utilizó para representar comportamientos obligatorios que siempre forman parte de un caso de uso principal, como la validación de datos y la confirmación de acciones.

Por su parte, la relación <<extend>> se empleó únicamente en aquellos casos donde el comportamiento es opcional o depende de una condición específica, como la inhabilitación de registros.

Estos ajustes garantizan una correcta interpretación del modelo funcional del sistema.

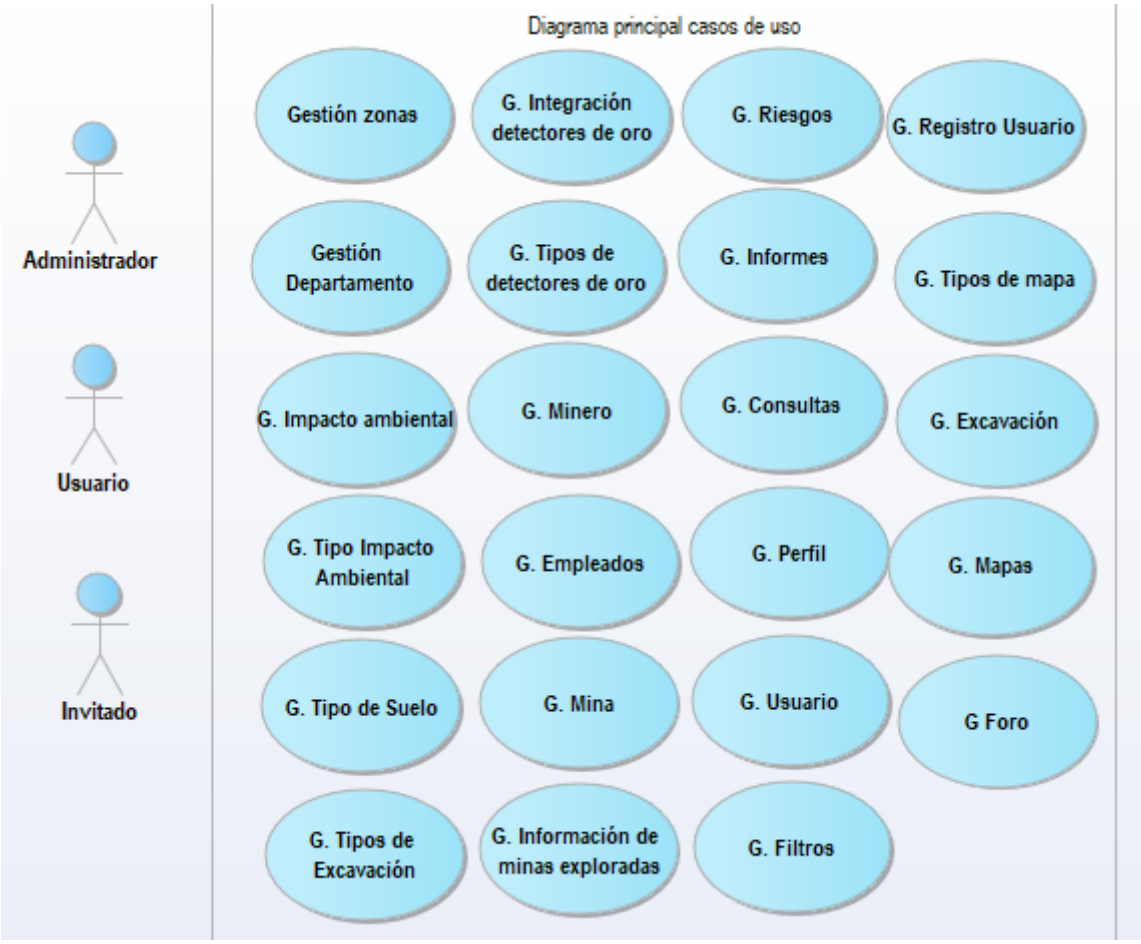


Ilustración 18. Diagrama general del sistema

Fuente: Elaboración propia

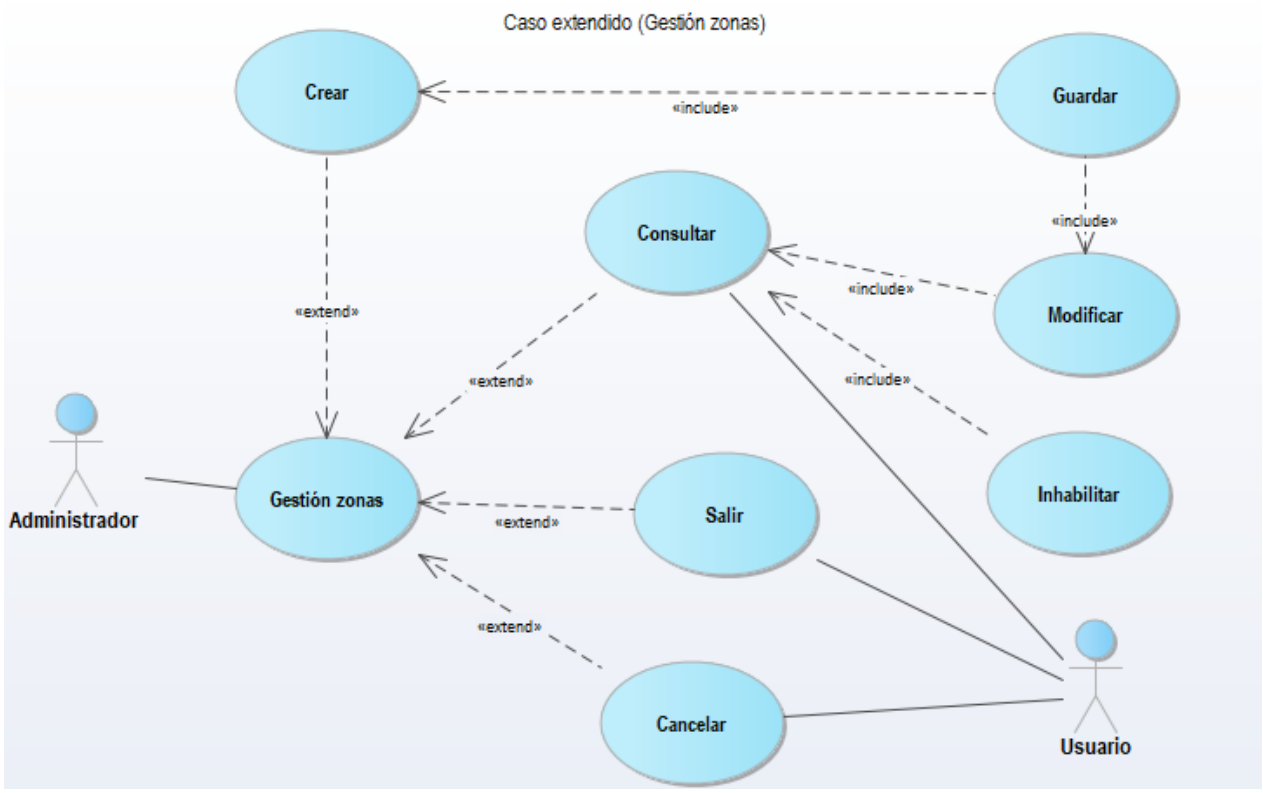


Ilustración 19. Diagrama de casos de uso extendido – Gestión Zonas

Fuente: Elaboración propia

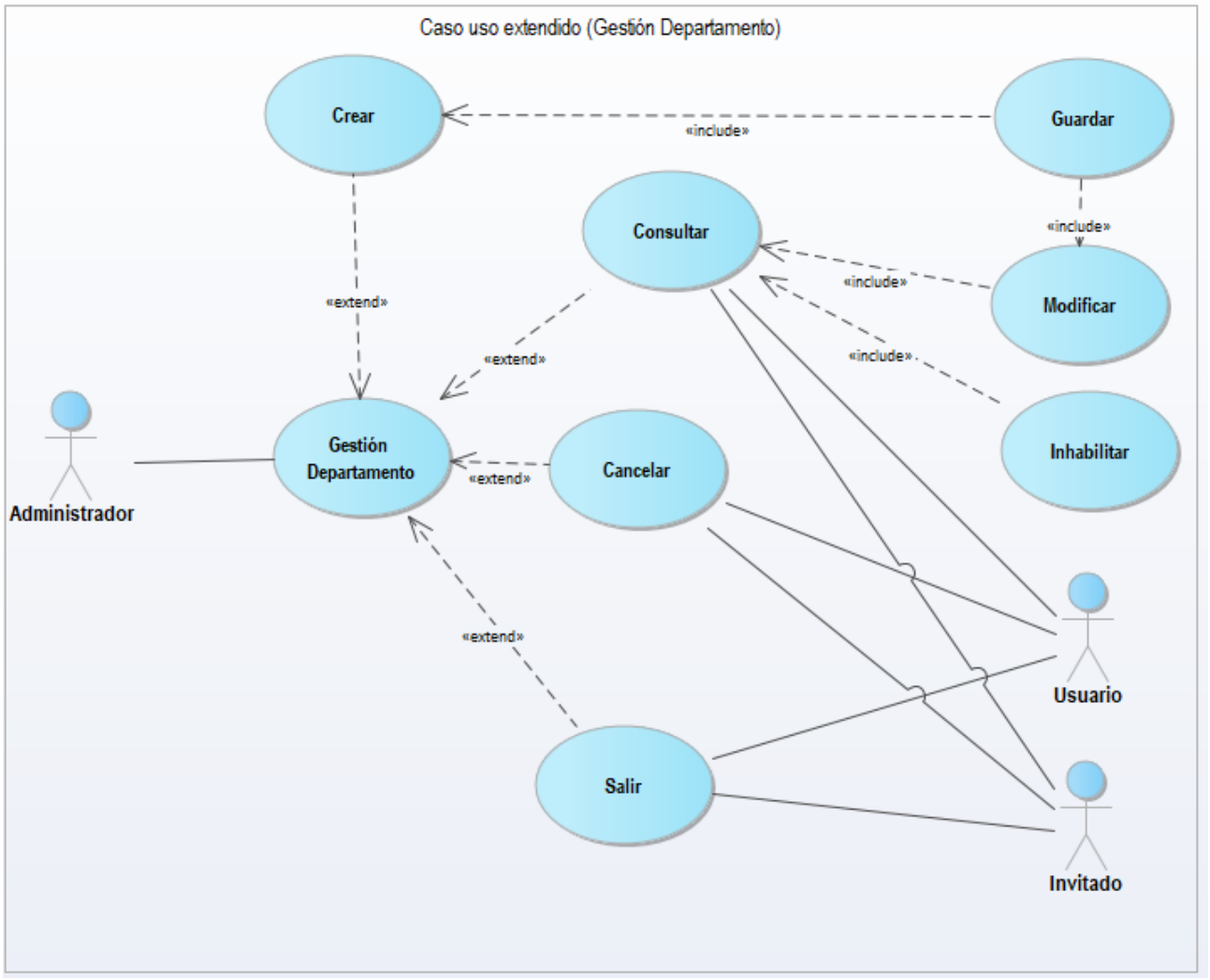


Ilustración 20. Diagrama de caso de uso – Gestión Departamento

Fuente: Elaboración propia

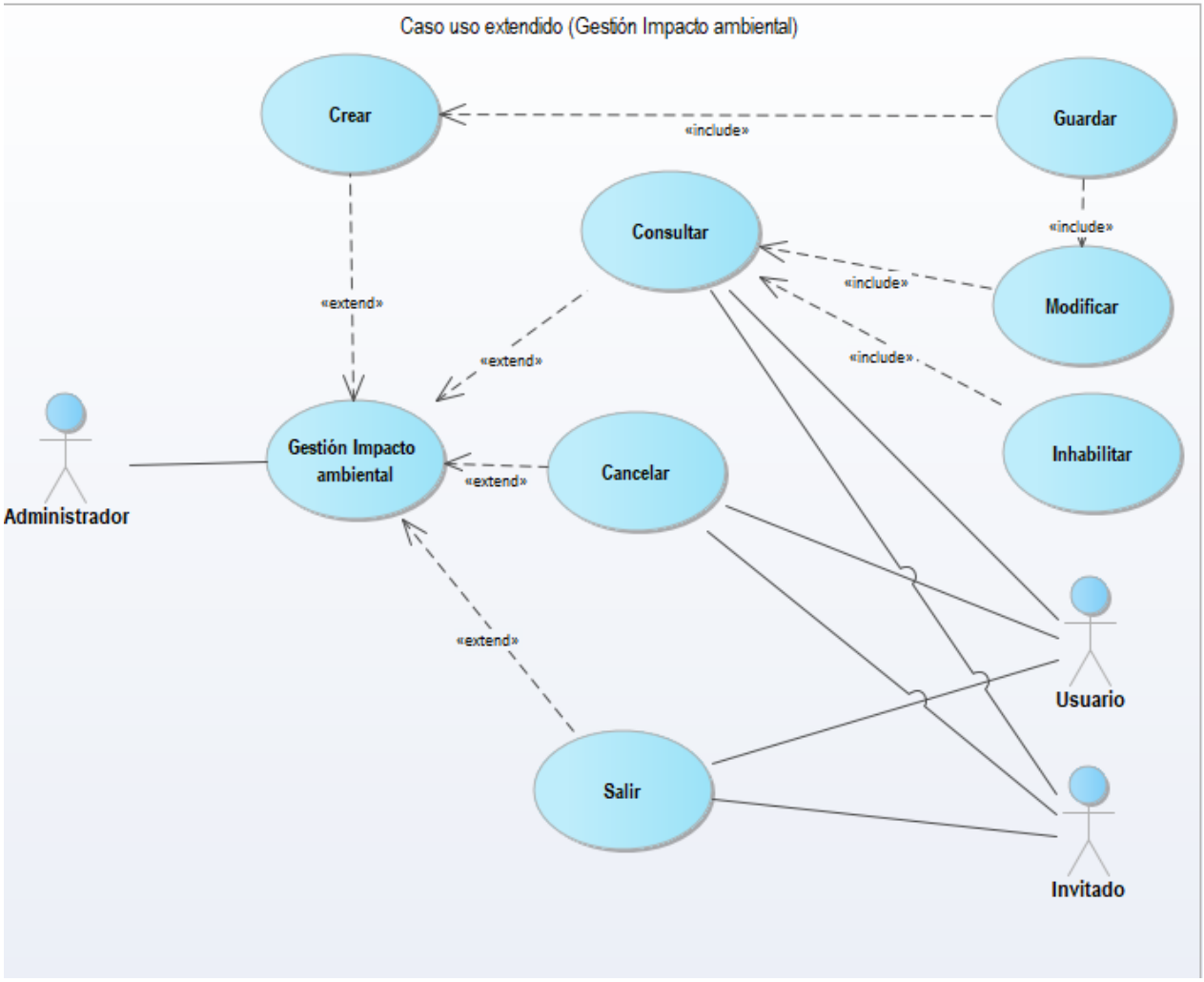


Ilustración 21. Diagrama de caso de uso – Gestión Impacto Ambiental

Fuente: Elaboración propia

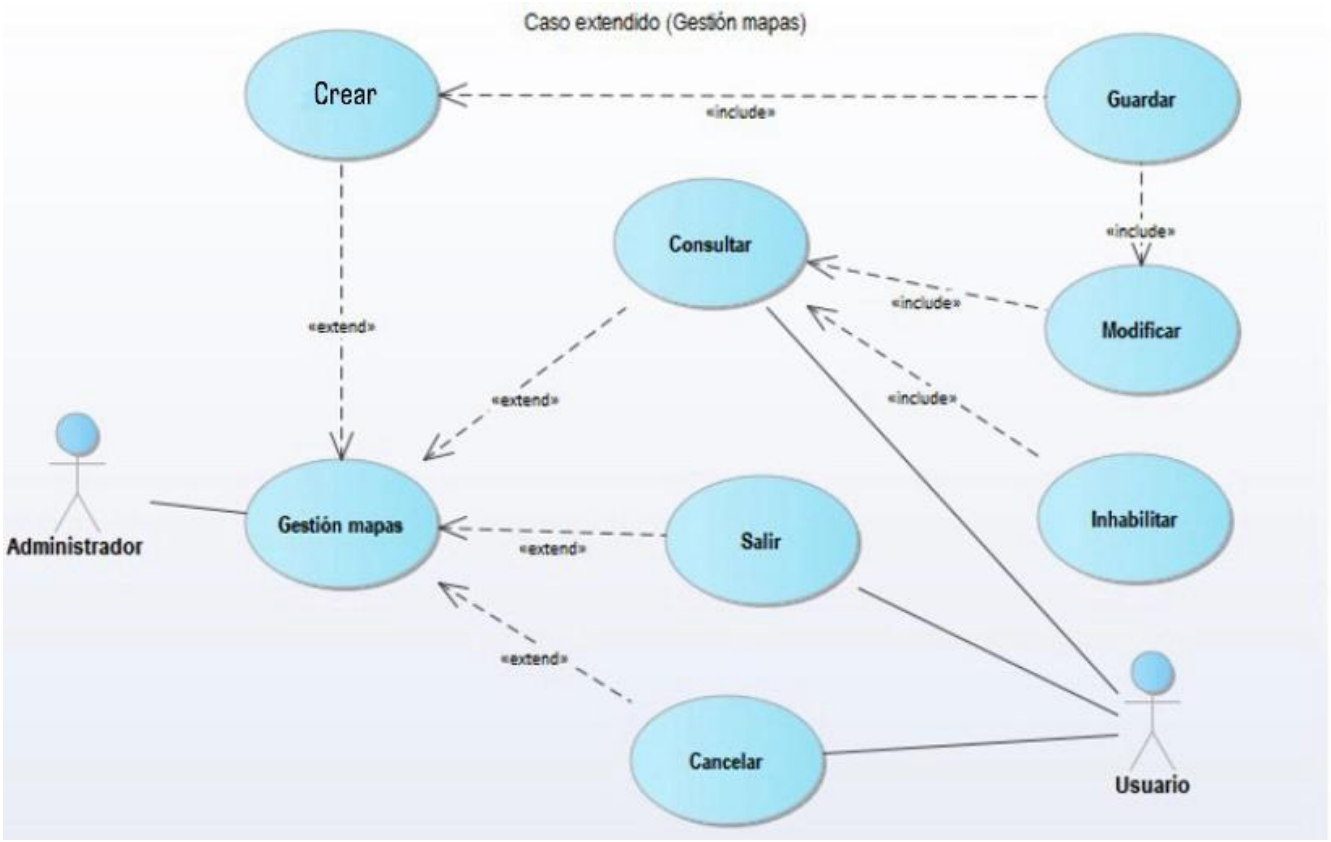


Ilustración 22. Diagrama extendido de caso de uso – Gestión Mapas

Fuente: Elaboración propia

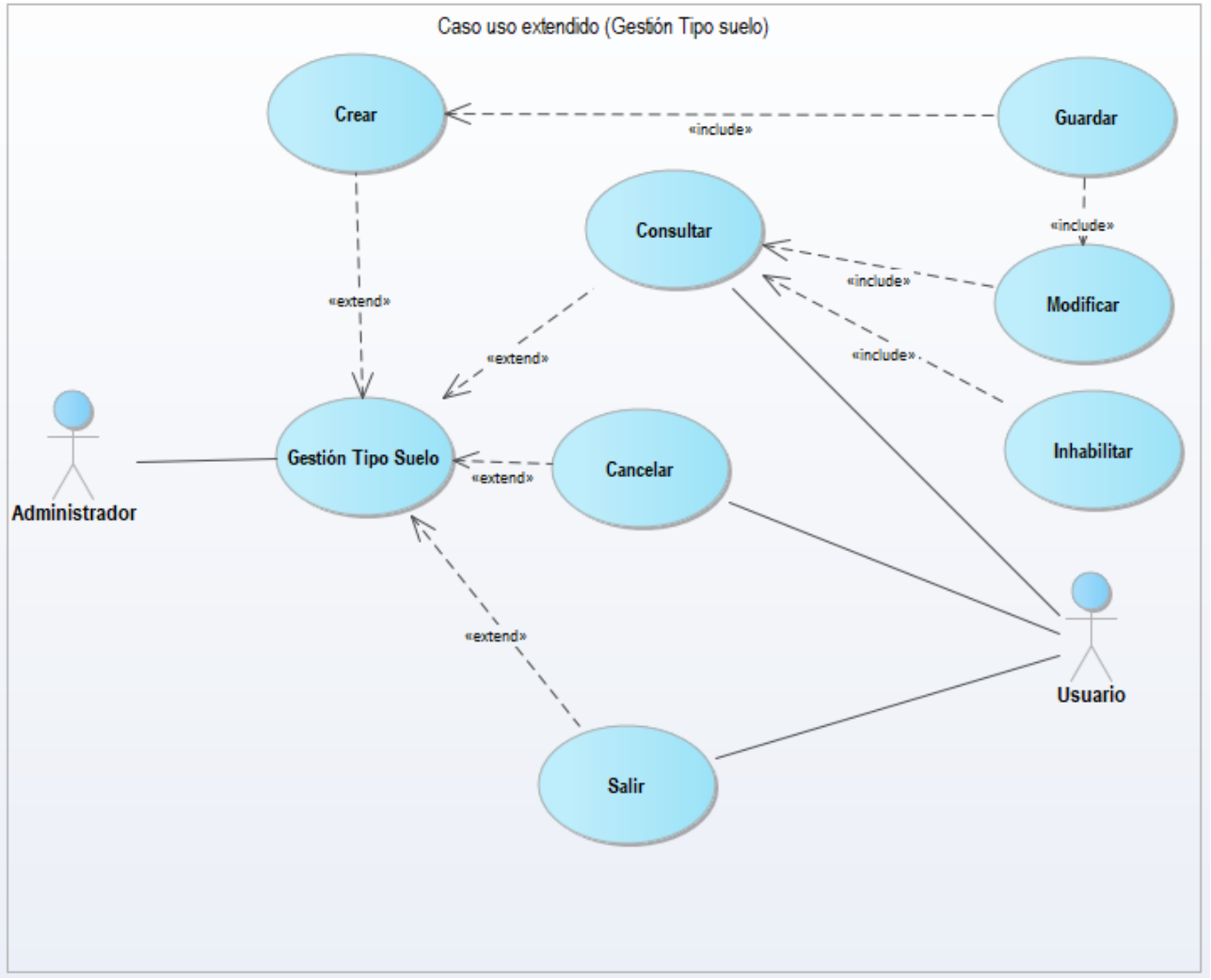


Ilustración 23. Diagrama extendido de caso de uso – Gestión Tipo Suelo

Fuente: Elaboración propia

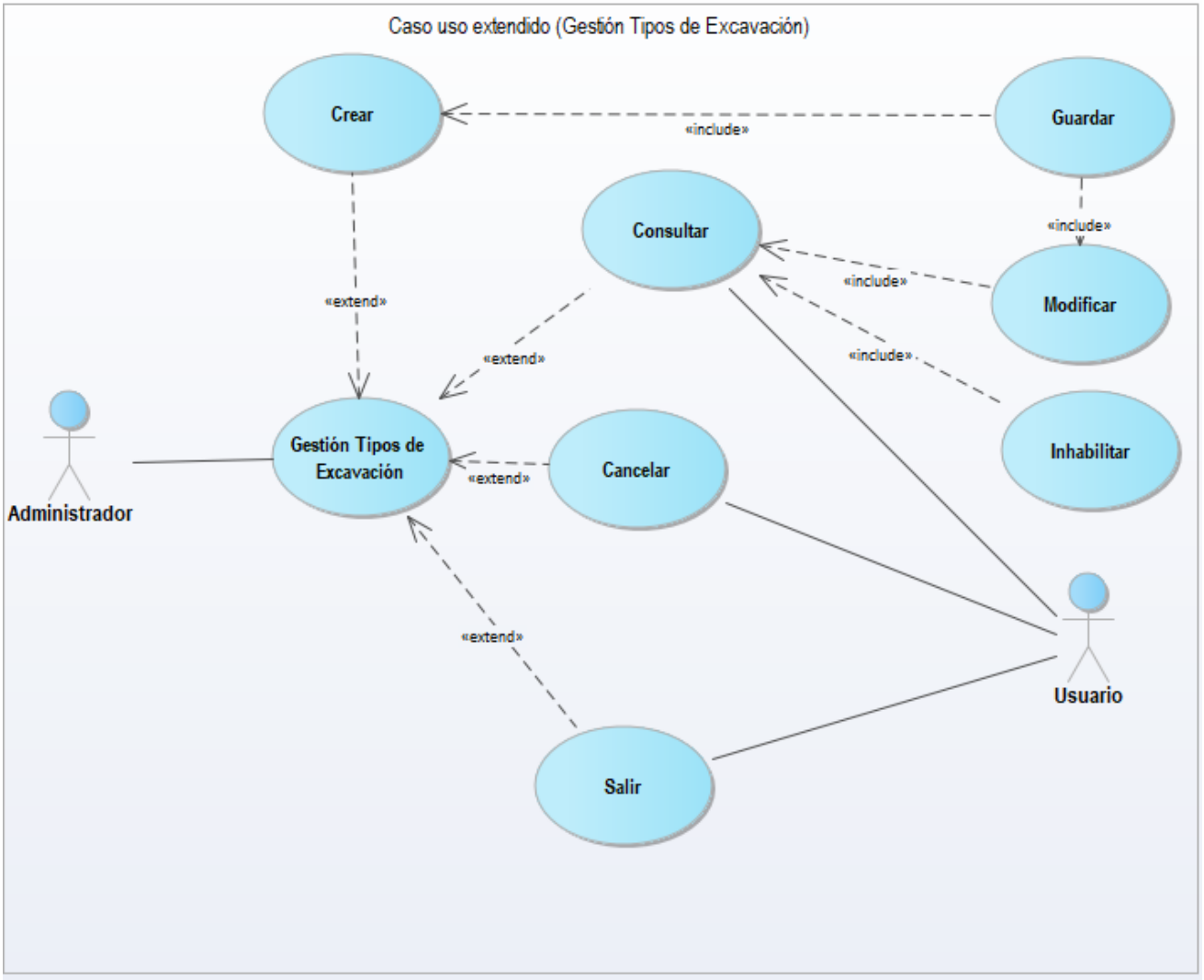


Ilustración 24. Diagrama de caso extendido – Gestión Tipo de Excavación

Fuente: Elaboración propia

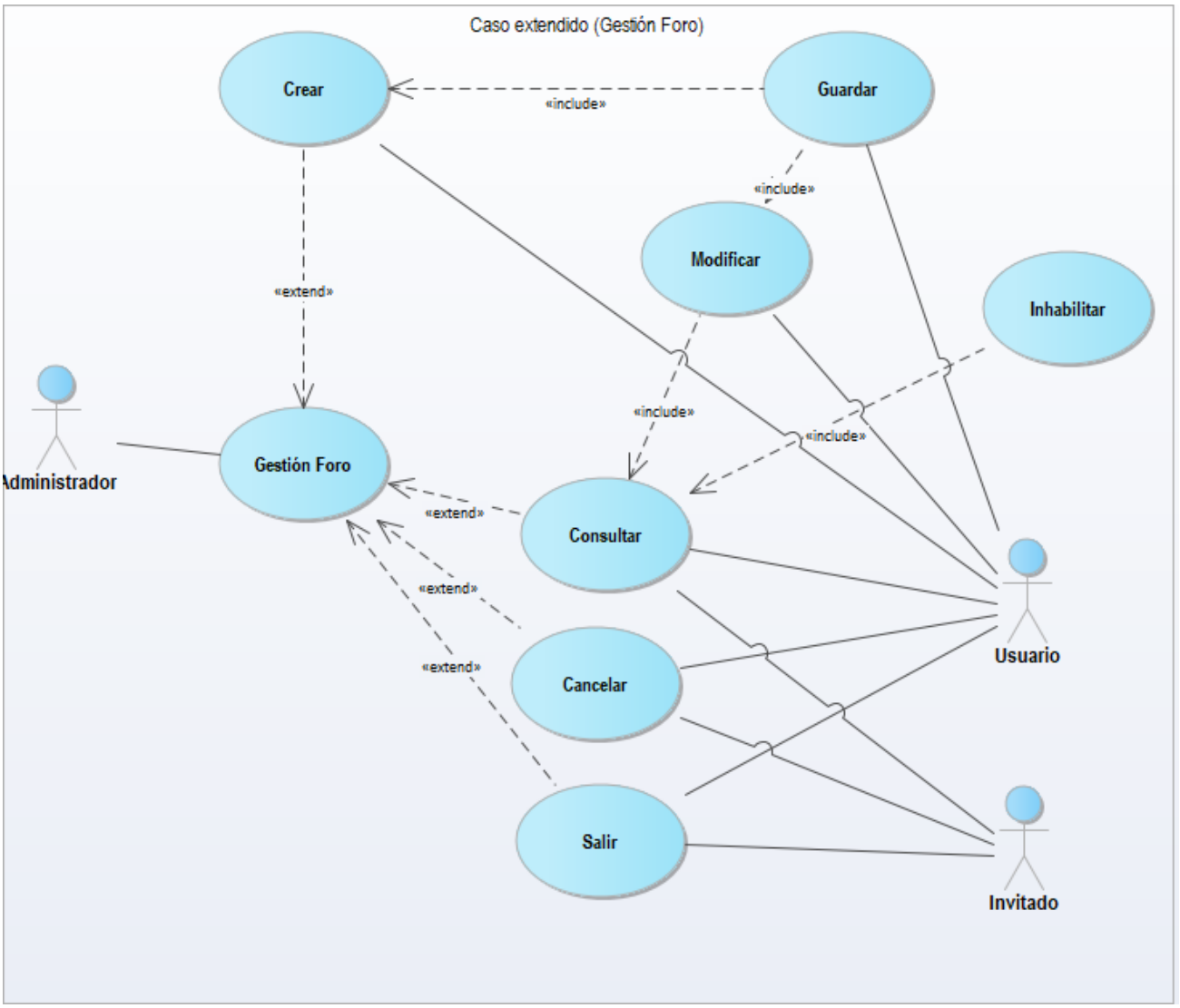


Ilustración 25. Diagrama de caso de uso extendido – Gestión Foro

Fuente: Elaboración propia

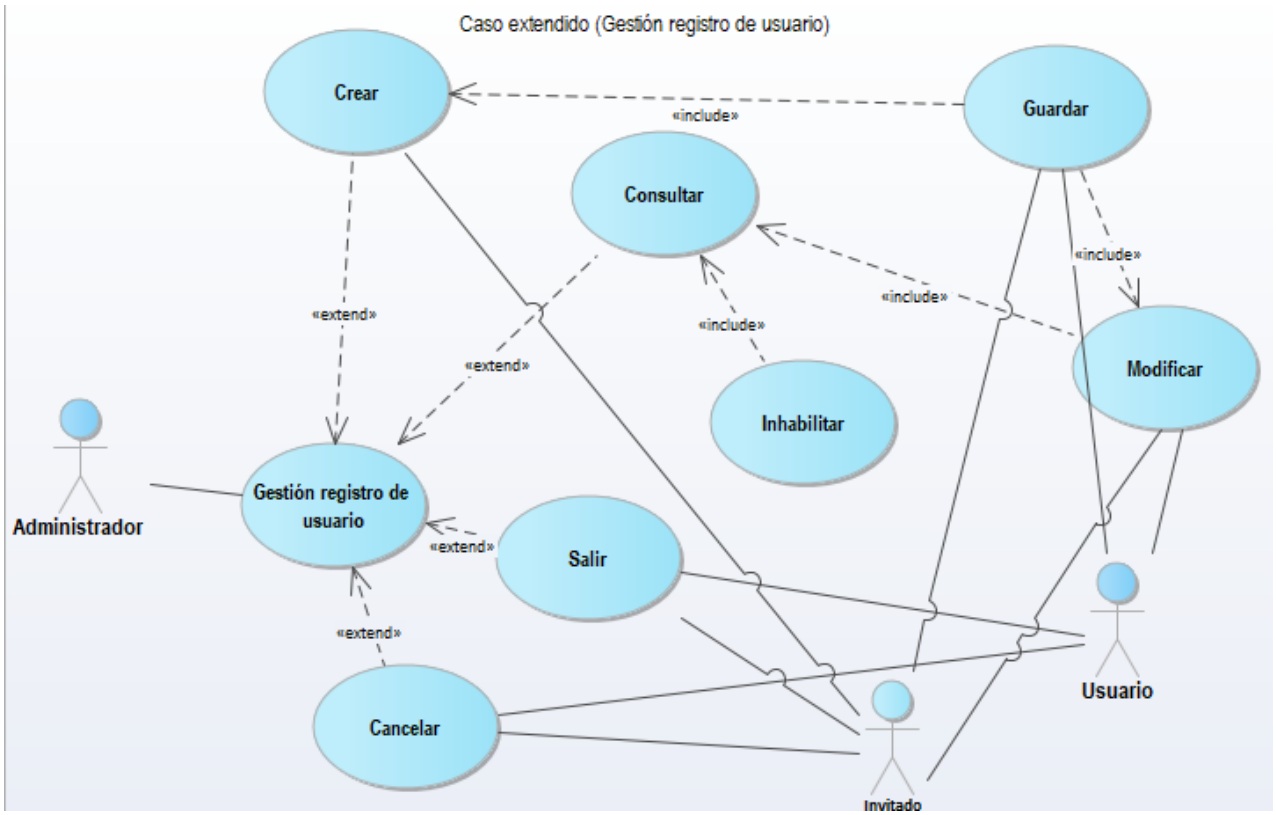


Ilustración 26. Diagrama de caso de uso extendido – Gestión Registro Usuario

Fuente: Elaboración propia

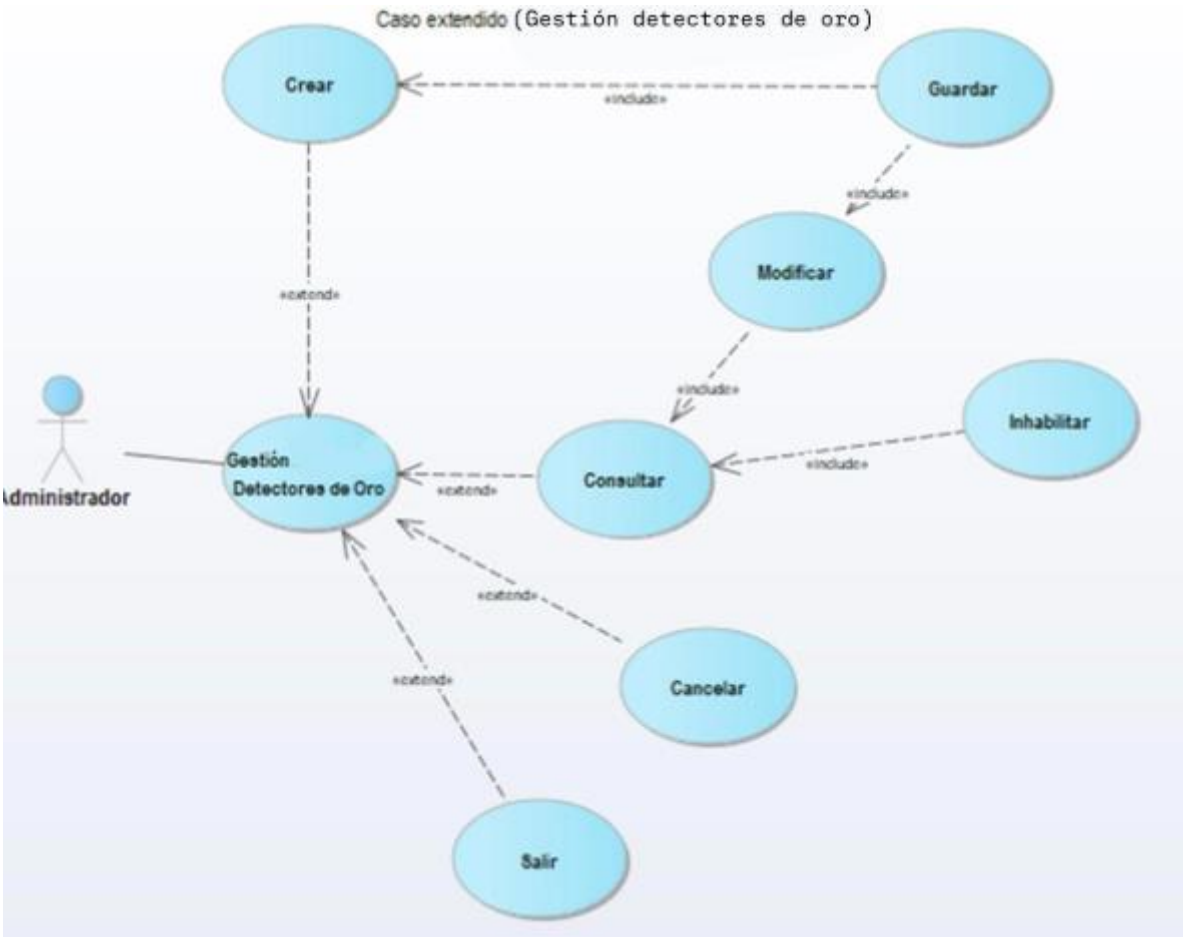


Ilustración 27. Diagrama de caso de uso extendido – Gestión Detectores de Oro

Fuente: Elaboración propia

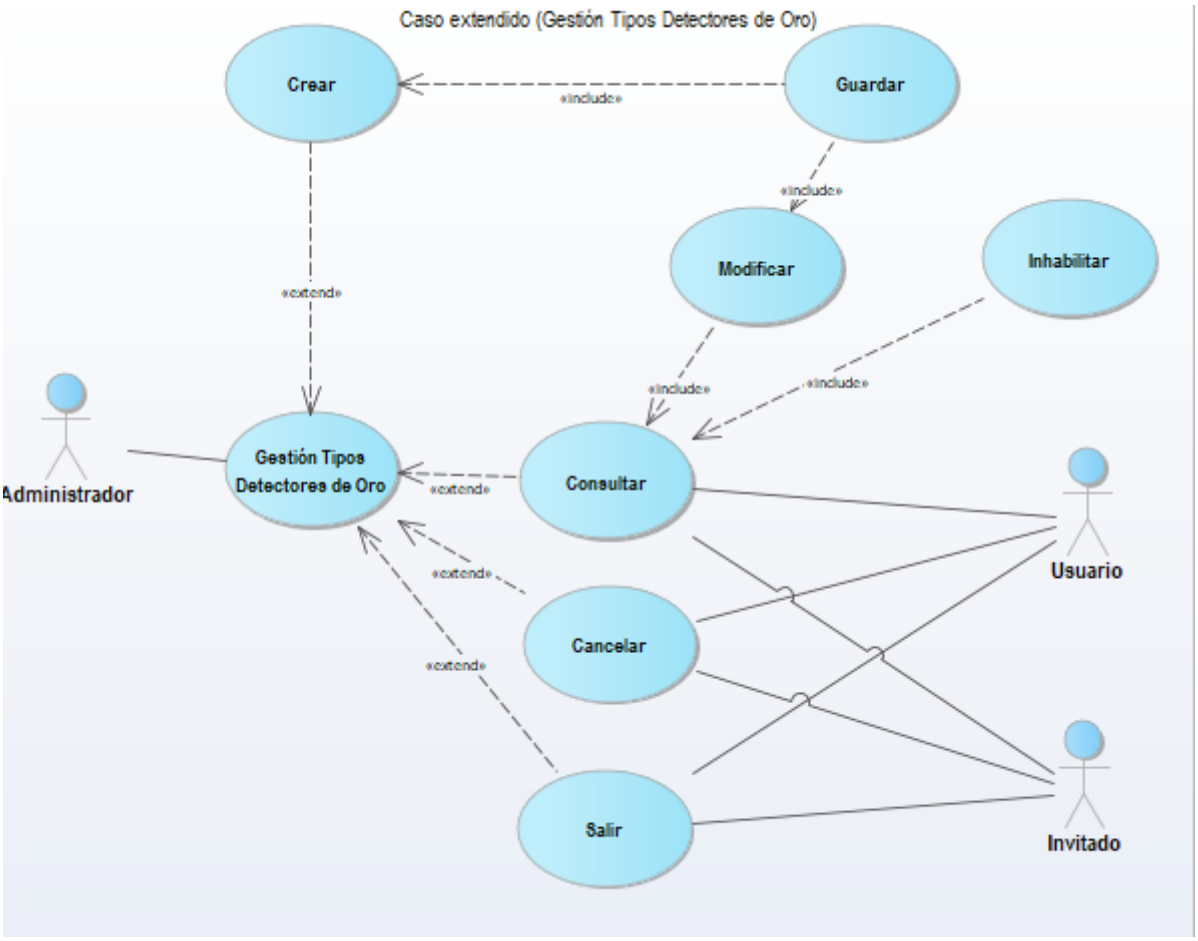


Ilustración 28. Diagrama de caso de uso extendido – Gestión Tipos Detectores de Oro

Fuente: Elaboración propia

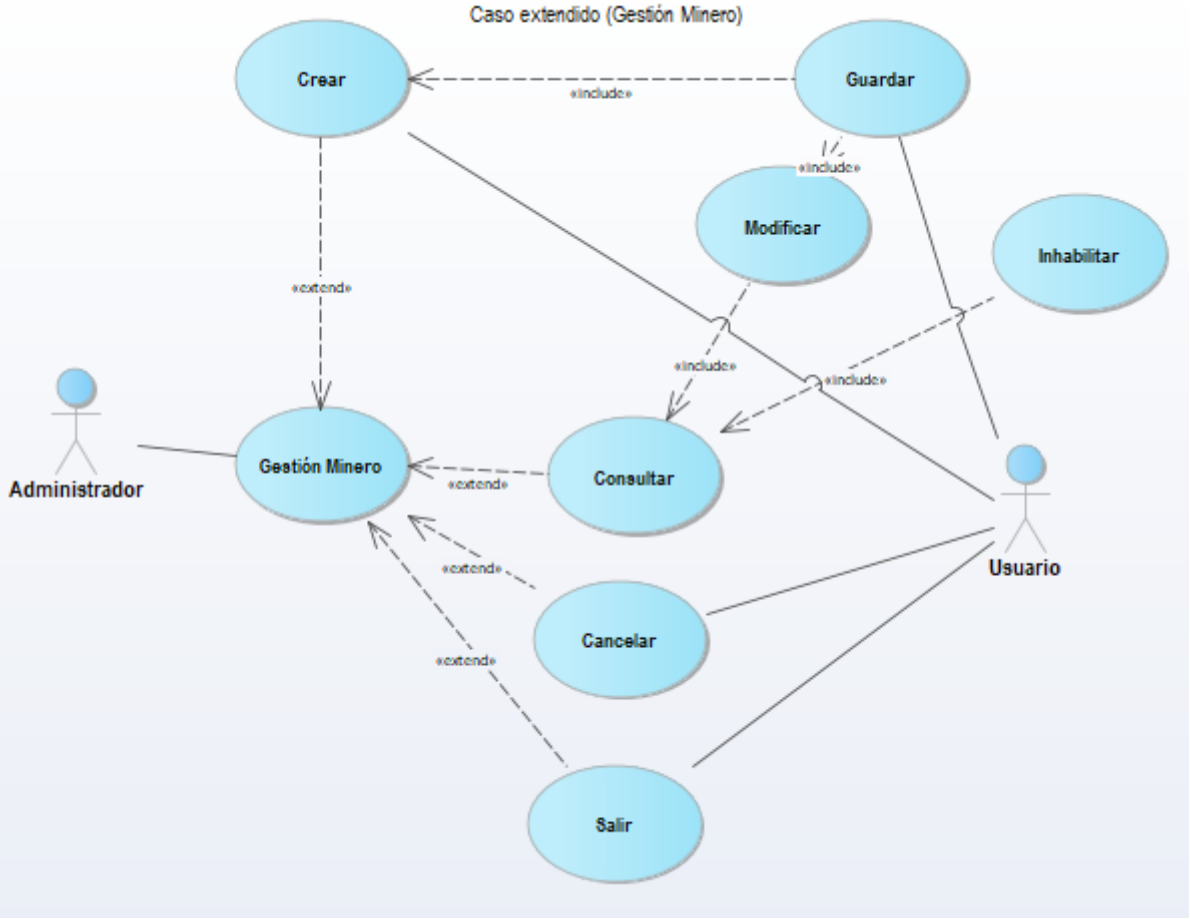


Ilustración 29. Diagrama de caso de uso extendido – Gestión Minero

Fuente: Elaboración propia

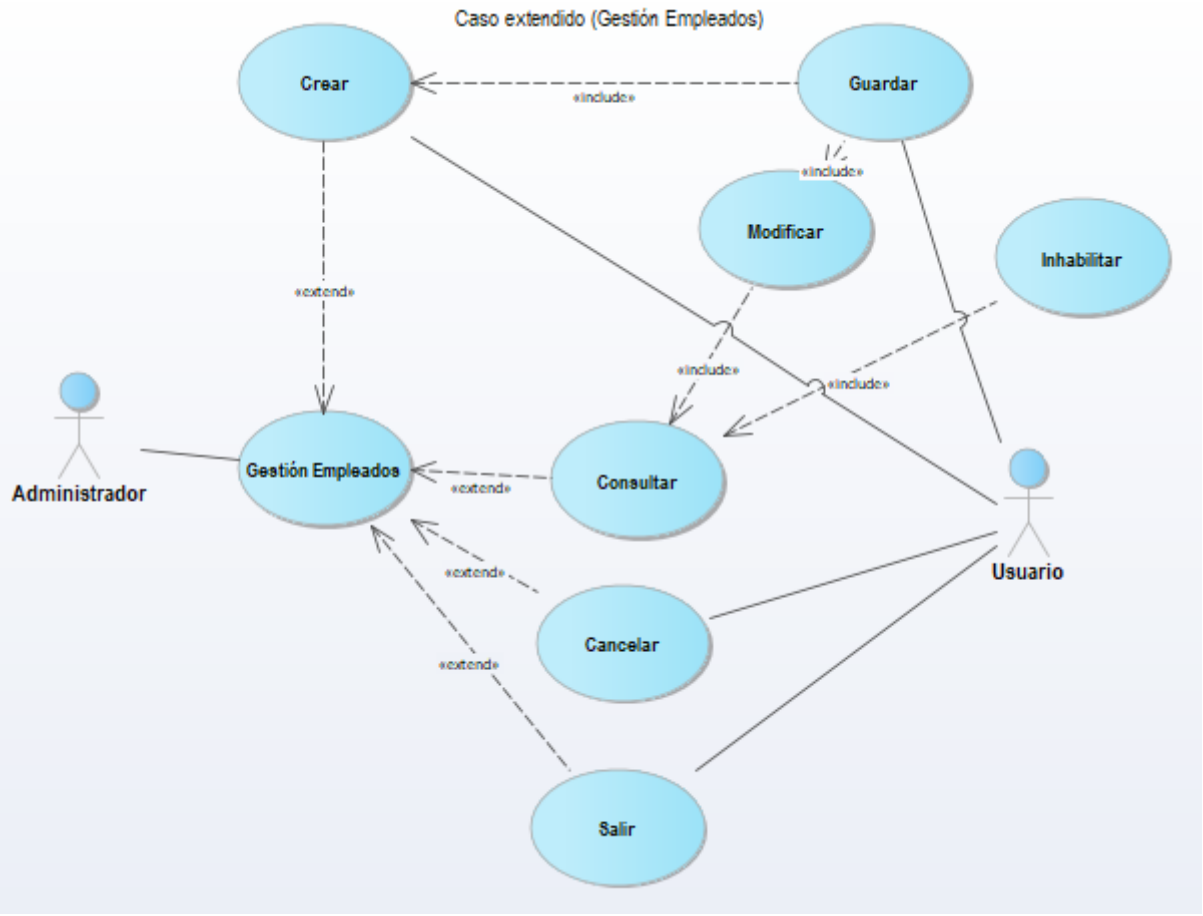


Ilustración 30. Diagrama de caso de uso extendido – Gestión Empleados

Fuente: Elaboración propia

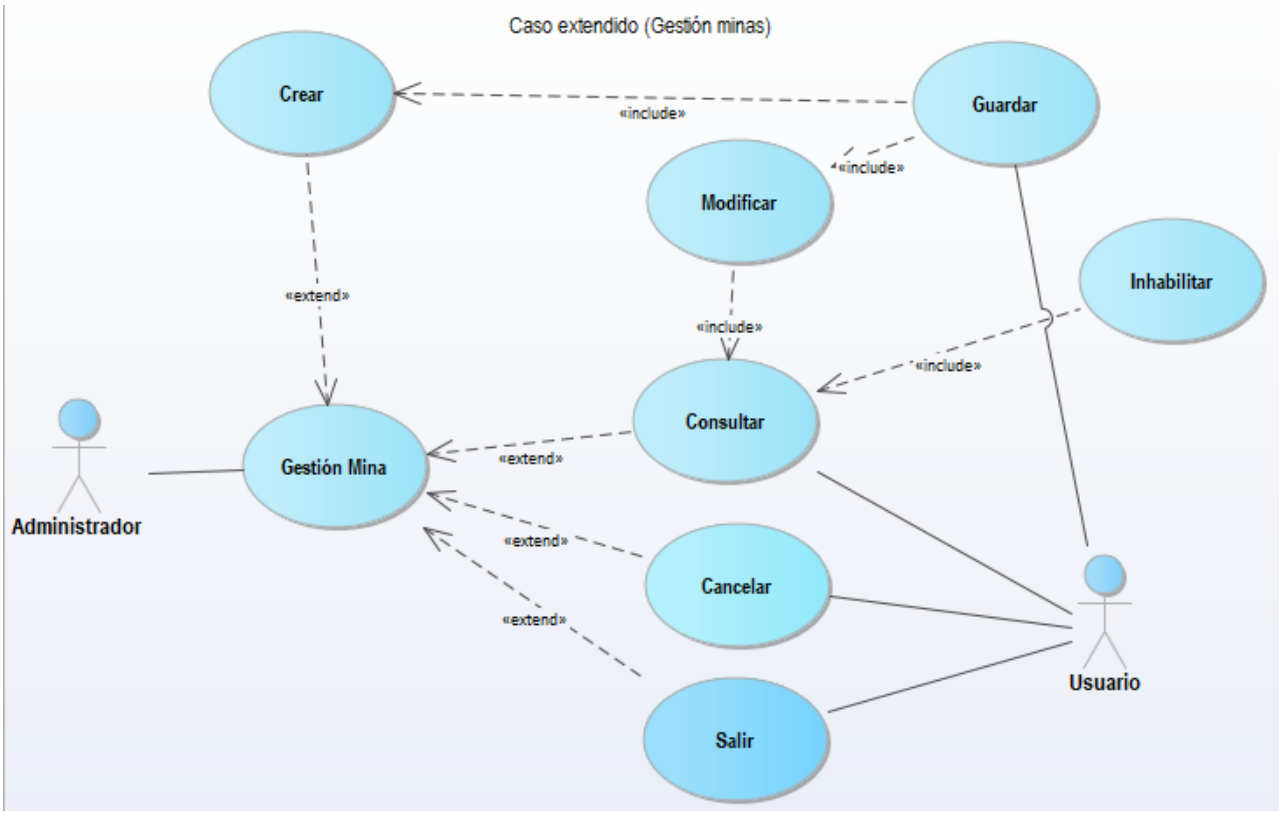


Ilustración 31. Diagrama de caso de uso extendido – Gestión Minas

Fuente: Elaboración propia

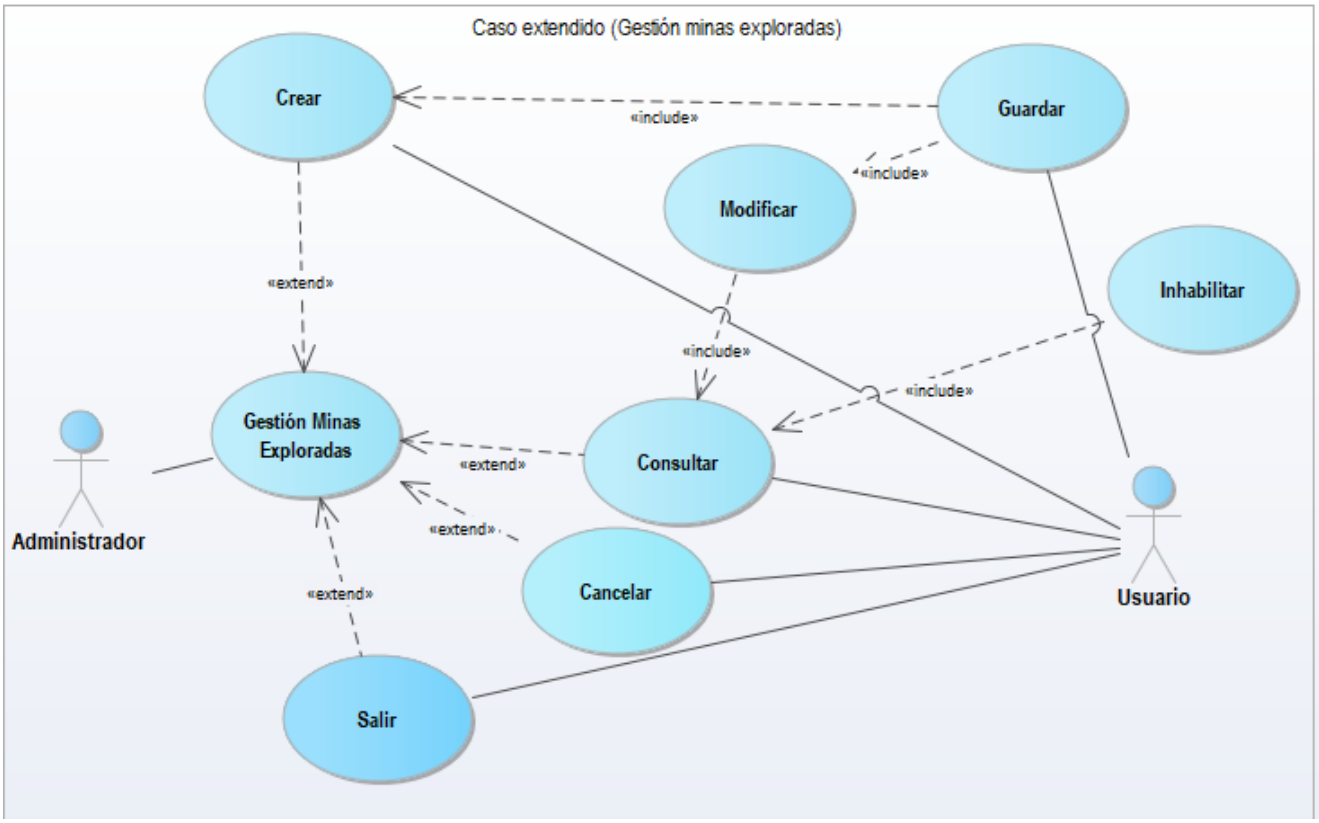


Ilustración 32. Diagrama de caso de uso extendido – Gestión Minas Exploradas

Fuente: Elaboración propia

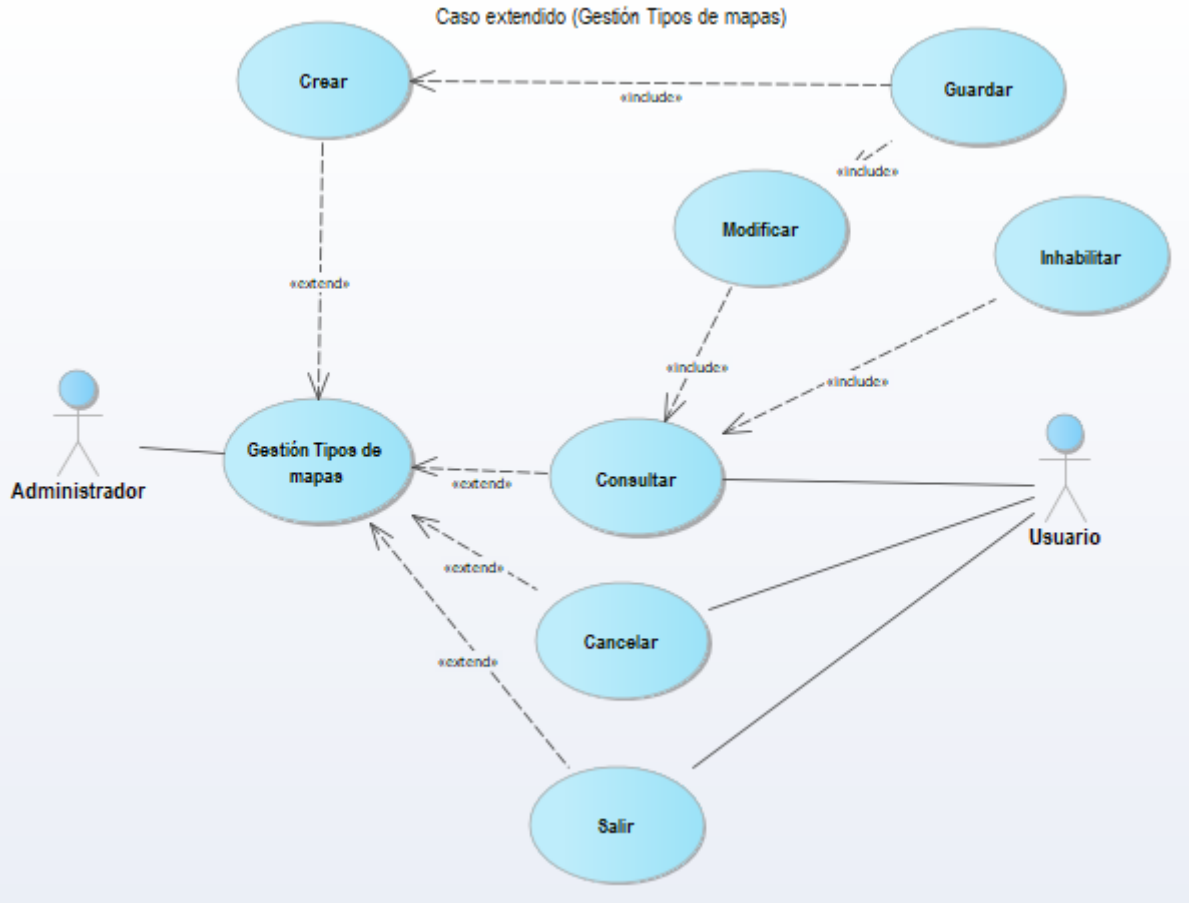


Ilustración 33. Diagrama de caso de uso extendido – Gestión Tipos de Mapas

Fuente: Elaboración propia

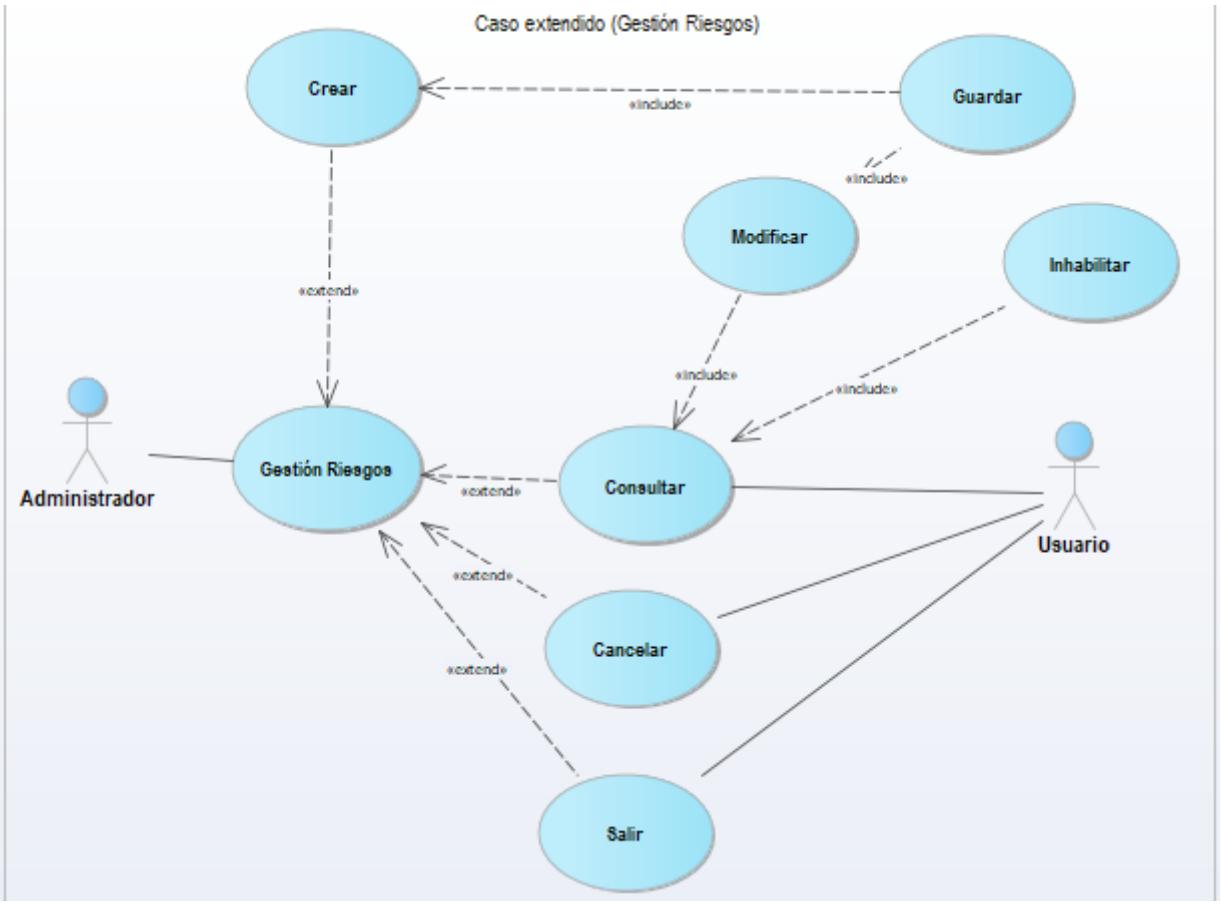


Ilustración 34. Diagrama de caso de uso extendido – Gestión Riesgos

Fuente: Elaboración propia

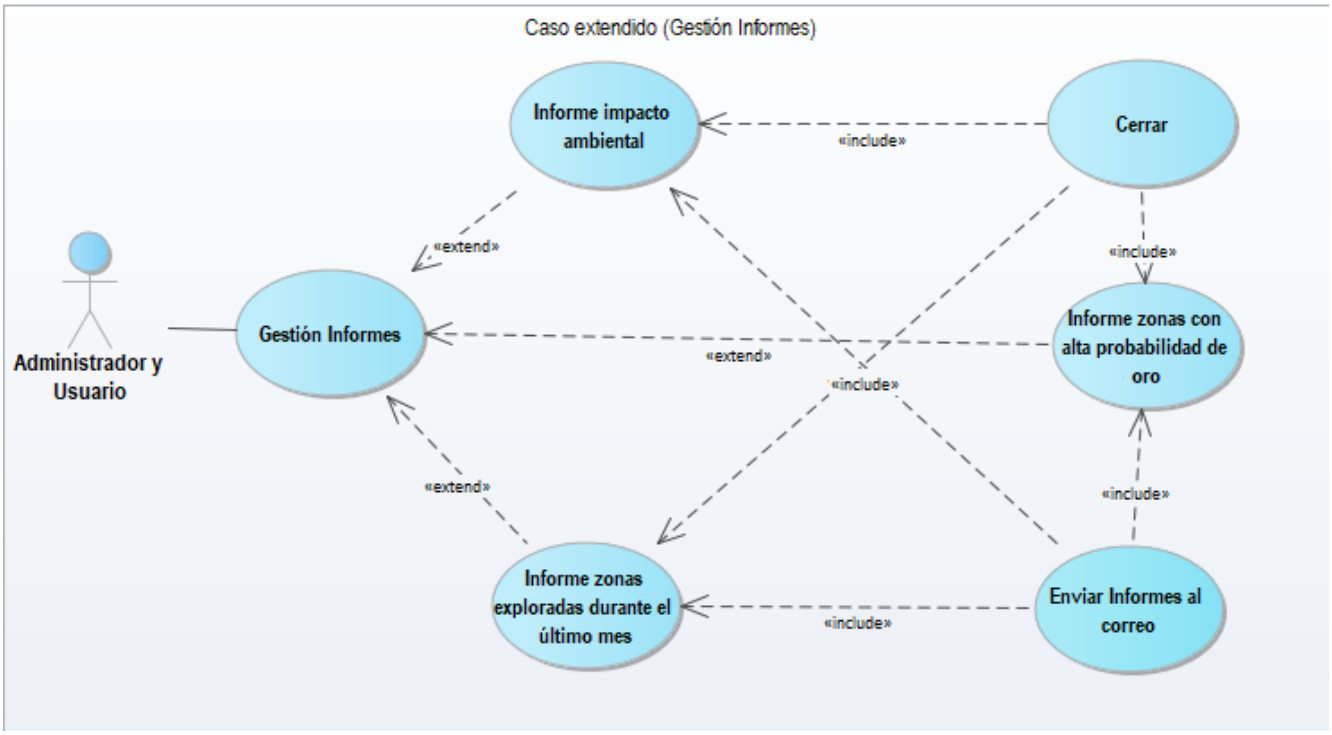


Ilustración 35. Diagrama de caso de uso extendido – Gestión Informes

Fuente: Elaboración propia

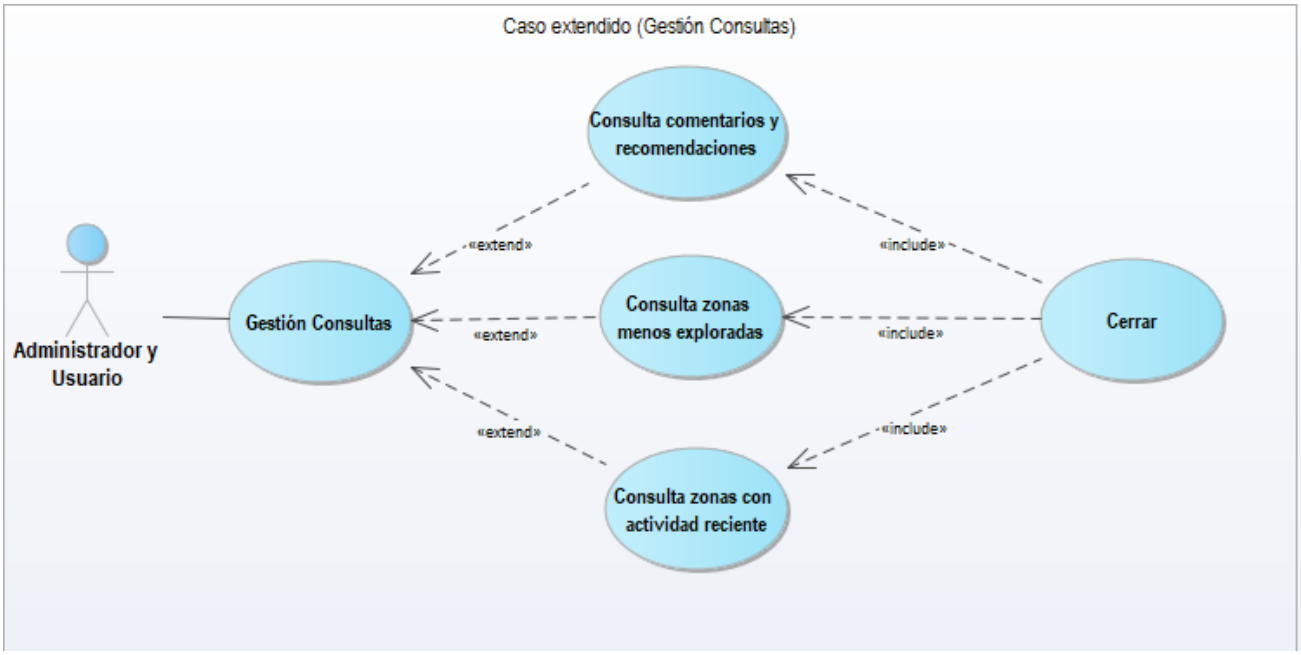


Ilustración 36. Diagrama de caso de uso extendido – Gestión Consultas

Fuente: Elaboración propia

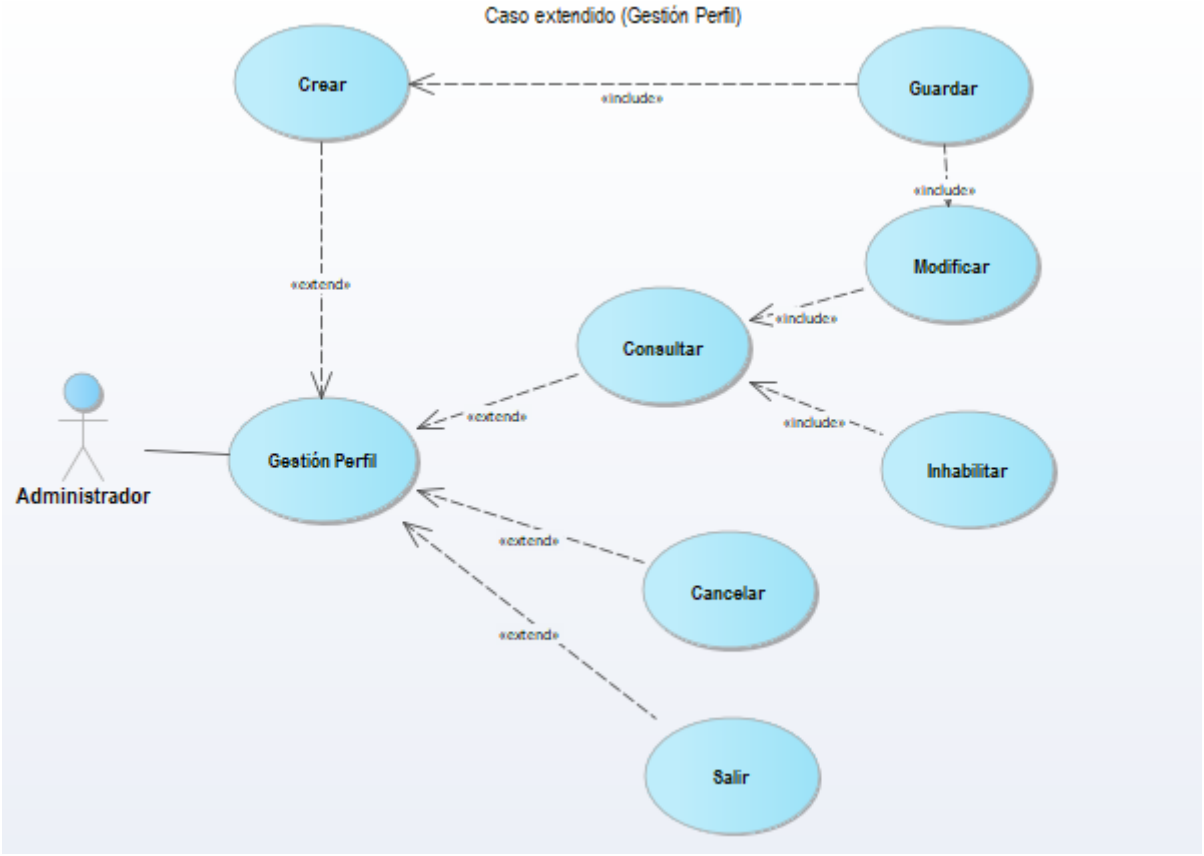


Ilustración 37. Diagrama de caso de uso extendido – Gestión Perfil

Fuente: Elaboración propia

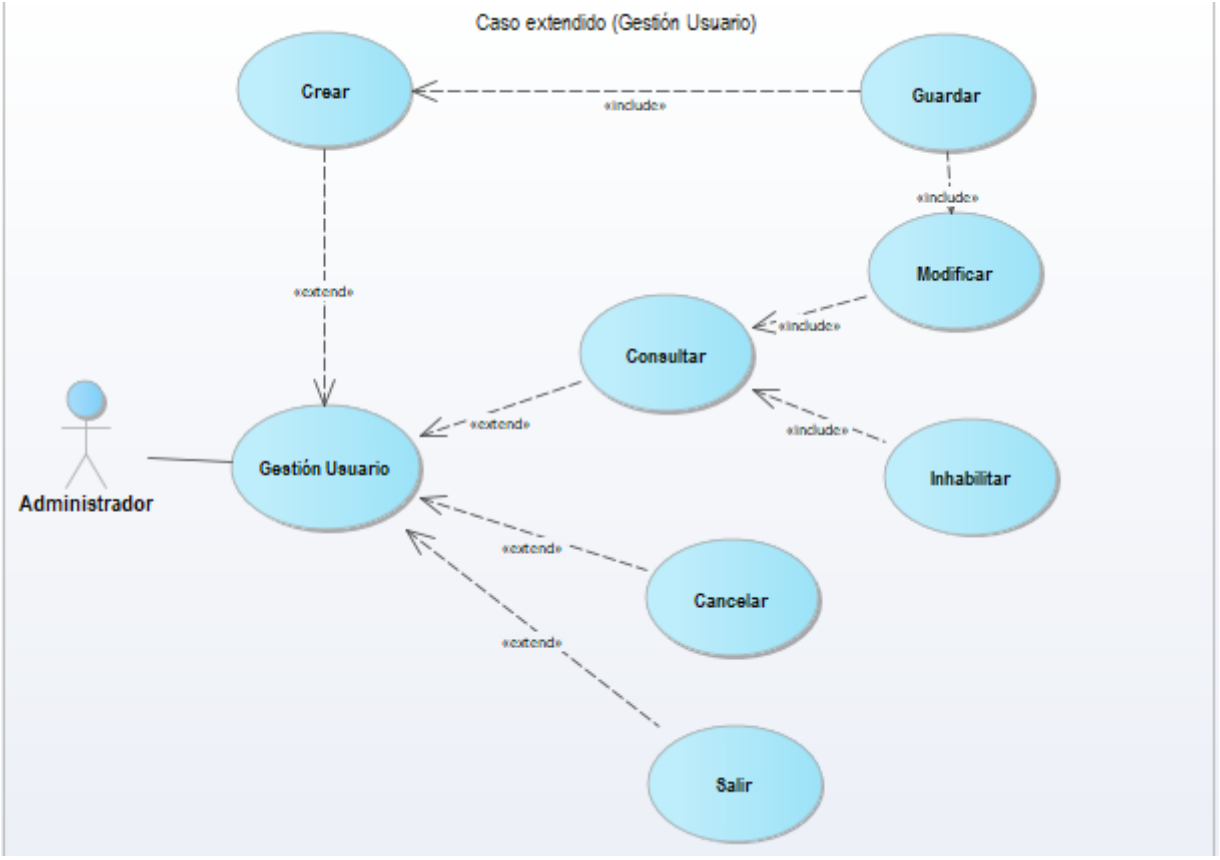


Ilustración 38. Diagrama de caso de uso extendido – Gestión Usuario

Fuente: Elaboración propia

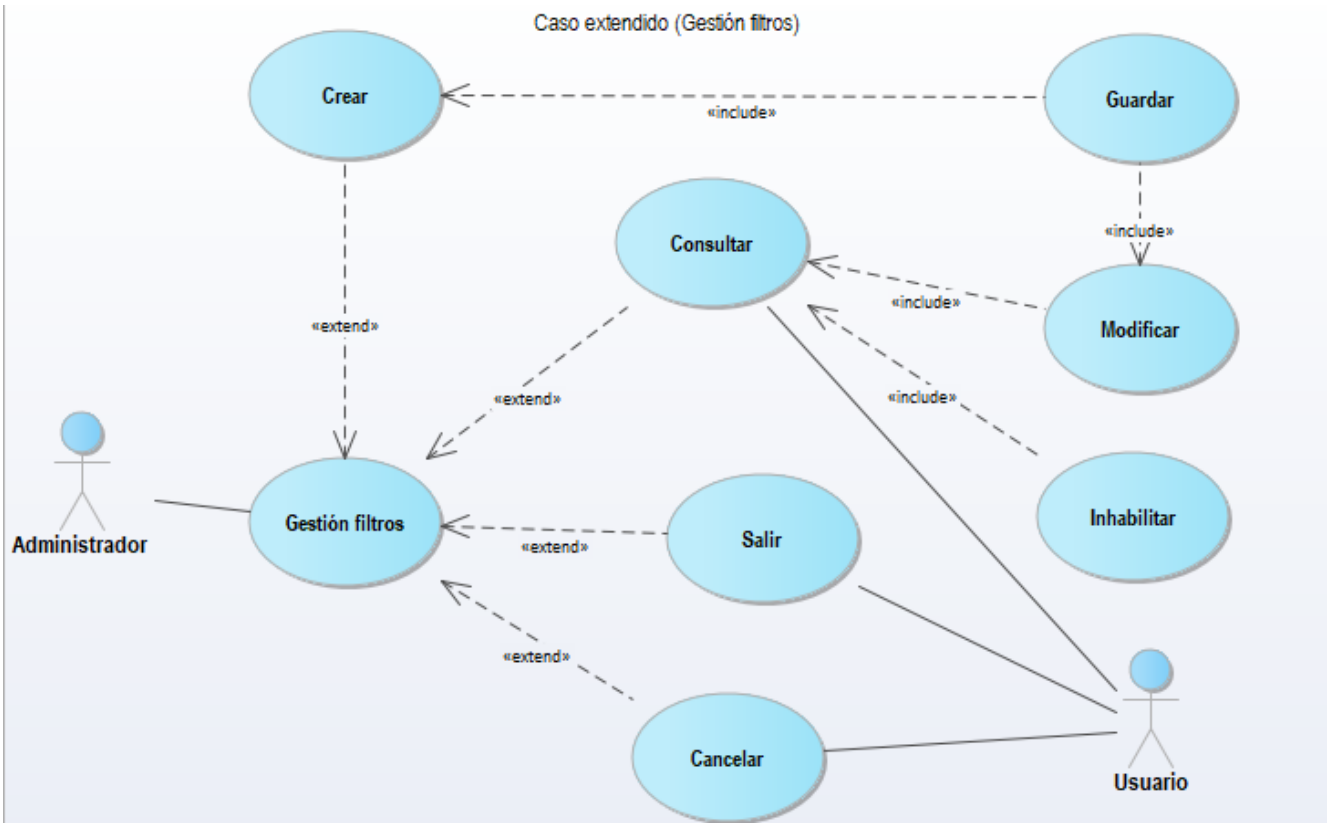


Ilustración 39. Diagrama de caso de uso extendido – Gestión Filtros

Fuente: Elaboración propia

Diagrama de Clases

Diagrama de Clases — Modelo de Dominio

Muestra las 10 entidades JPA del sistema con sus atributos, métodos clave y todas las relaciones: herencia/realización (UserDetails), asociación, composición y relaciones N:M mapeadas con tablas intermedias.

Diagrama de Entidades del Dominio — JPA/Hibernate (SoftGold)

Este diagrama representa el modelo completo de datos del sistema SoftGold, mostrando todas las entidades persistentes, sus atributos y las relaciones entre ellas tal como quedan mapeadas en la base de datos MySQL a través de JPA/Hibernate.

Entidad central: Usuario

Usuario es el eje del modelo. Implementa la interfaz UserDetails de Spring Security, lo que le permite integrarse directamente con el sistema de autenticación sin capas adicionales. Sus atributos cubren tres responsabilidades: identidad personal (cédula, nombre, apellidos, tipo de documento), datos de contacto y rol laboral (tipoEmpleado, área), y seguridad (intentosFallidos, bloqueado, tiempoBloqueo), este último grupo soportando el mecanismo de bloqueo por intentos fallidos de inicio de sesión.

Gestión de roles y acceso

La relación Usuario ↔ Rol es de tipo N:M, resuelta mediante la tabla intermedia usuario_roles. Los cuatro roles posibles — ROLE_ADMINISTRADOR, ROLE_MINERO, ROLE_EMPLEADO y ROLE_USUARIO — determinan qué funcionalidades del sistema puede usar cada persona, implementando así el modelo RBAC documentado en los requerimientos.

Núcleo minero: Mina, Mapa y Riesgo

Mina es la entidad operativa central del negocio. Se relaciona con:

Mapa en una relación N:M (tabla intermedia mina_mapa), permitiendo que una mina tenga múltiples mapas y que un mapa pueda estar asociado a varias minas.

Riesgo también en N:M (tabla mina_riesgo), registrando los riesgos asociados a cada mina sin duplicar información.

ZonaExploracion en una relación 1:N, donde cada zona pertenece a una sola mina. Esta entidad incluye atributos geoespaciales (latitud, longitud) y un ciclo de vida propio mediante los estados PENDIENTE, APROBADA o RECHAZADA, y los tipos POTENCIAL o ACTIVA.

La relación Usuario → Mina (N:1, "tiene empleados") indica que múltiples usuarios pueden estar asignados a una misma mina, reflejando la estructura organizacional del sector minero artesanal.

Módulo de foro

ForumPost y ForumComment modelan el componente colaborativo del sistema. Su relación es de composición 1:N con CascadeType.ALL + orphanRemoval, lo que significa que al eliminar una publicación se eliminan automáticamente todos sus comentarios en cascada, sin dejar registros huérfanos en la base de datos. Ambas entidades registran el nombre del autor como texto plano (autorNombre), lo que sugiere que la autoría se captura en el momento de la creación sin depender de la sesión activa del usuario.

Módulo de soporte

SupportTicket gestiona las solicitudes de atención técnica. Combina dos clasificaciones independientes: el tipo (INCIDENTE, ERROR, MEJORA, QUEJA, PQR) y el estado (OPEN / CLOSED), lo que permite filtrar tickets tanto por naturaleza del problema como por su ciclo de vida de atención.

Seguridad: PasswordResetToken

PasswordResetToken tiene una relación 1:1 con Usuario y almacena el token temporal para recuperación de contraseña junto con su fecha de expiración, implementando el flujo estándar de restablecimiento seguro de credenciales.

Síntesis del modelo

El diseño refleja un modelo de dominio bien delimitado con responsabilidades claras por entidad. Las relaciones N:M están correctamente normalizadas mediante tablas intermedias, la composición del foro garantiza integridad referencial automática, y la integración con Spring Security a través de UserDetails evita duplicación de lógica de autenticación. En conjunto, el modelo soporta de forma coherente los seis módulos funcionales del sistema SoftGold.

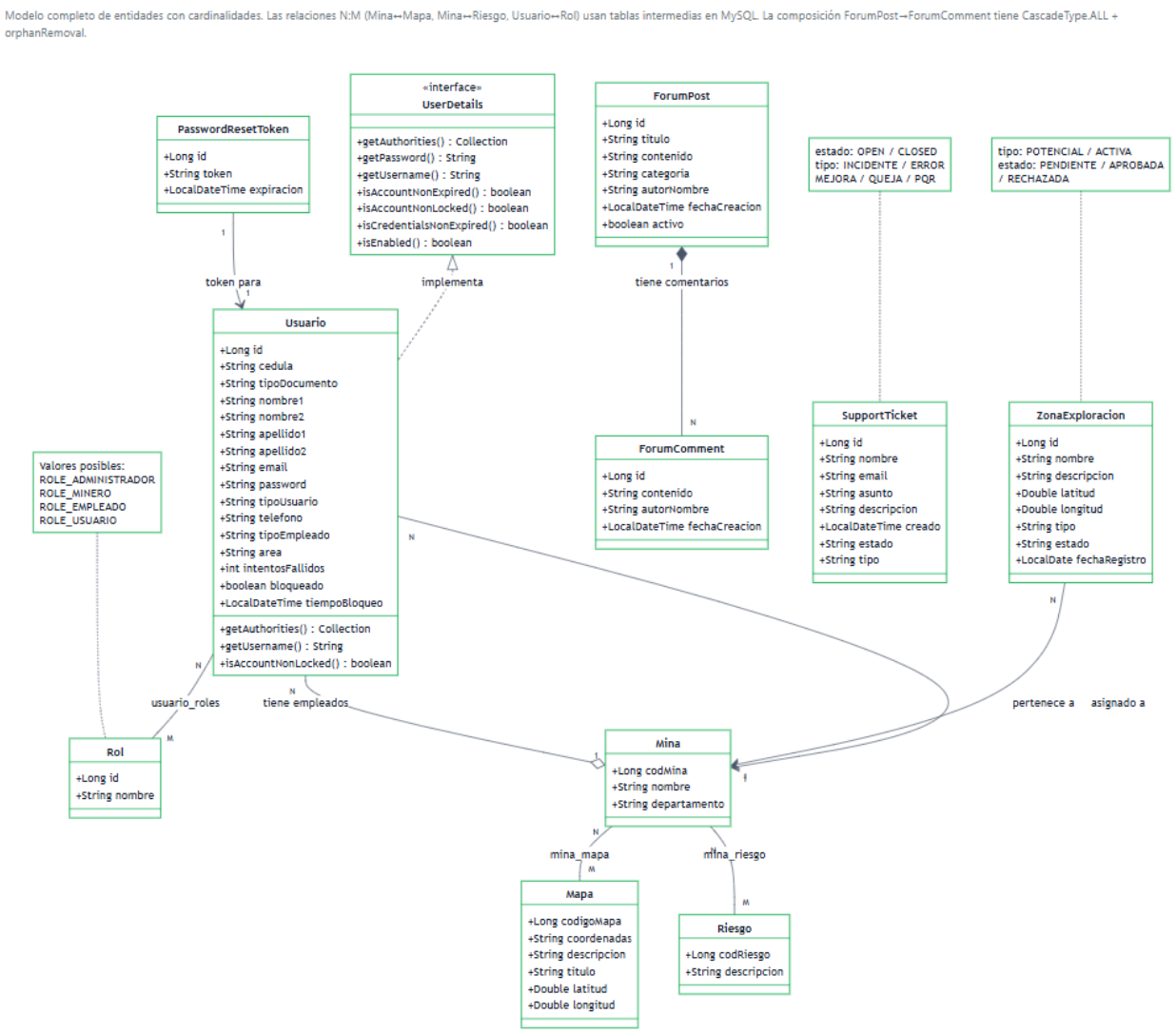


Ilustración 40. Diagrama de Clases — Modelo de Dominio

Fuente: Elaboración propia

Diagrama de Clases — Arquitectura de Capas.

Estructura MVC: Controllers (capa web), Services + Interfaces (lógica de negocio), DAOs/Repositories (acceso a datos) y la capa de Seguridad con Spring Security.

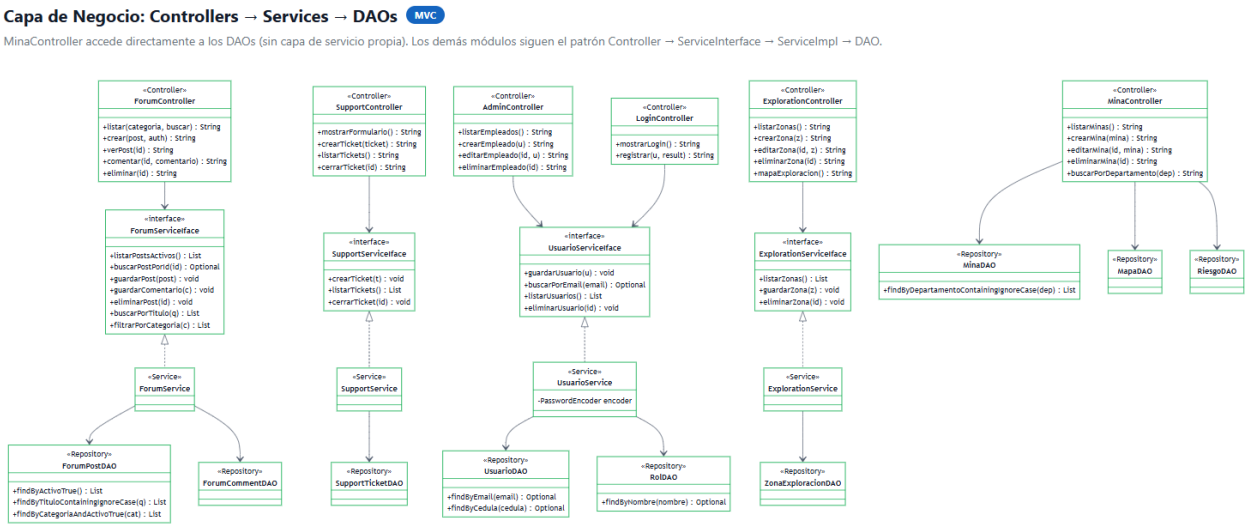


Ilustración 41 Diagrama de clases: Capa de Negocio: Controllers → Services → DAOs

Fuente: Elaboración propia

El diagrama muestra cómo se organiza la lógica del sistema en tres capas: Controllers, Services y DAOs (repositorios).

La mayoría de los módulos sigue el patrón estándar Controller → Interface de Servicio → Implementación → DAO, lo que desacopla las capas y facilita el mantenimiento. Por ejemplo, ForumController depende de ForumServiceIface, implementada por ForumService, que a su vez usa ForumPostDAO y ForumCommentDAO. El mismo esquema aplica para los módulos de soporte, usuarios y zonas de exploración.

La excepción notable es MinaController, que accede directamente a MinaDAO, MapaDAO y RiesgoDAO sin pasar por una capa de servicio propia. Esto simplifica el módulo dado su alcance acotado (CRUD básico con búsqueda por departamento), aunque sacrifica la consistencia arquitectónica del resto del sistema.

Los DAOs son interfaces que extienden Spring Data JPA, con métodos de consulta derivados del nombre (findByEmail, findByActivoTrue, findByCategoriaAndActivoTrue), lo que elimina la necesidad de escribir SQL manual para la mayoría de las operaciones. UsuarioService incorpora además un PasswordEncoder para el cifrado de contraseñas, centralizando esa responsabilidad de seguridad en la capa de servicio.

La arquitectura es coherente con el patrón MVC de Spring Boot y refleja buenas prácticas de separación de responsabilidades en la mayor parte del sistema.

Diagrama de Secuencia

Diagrama de Secuencia — Autenticación

El diagrama muestra la interacción entre los componentes del sistema en orden cronológico, desde que el usuario envía sus credenciales hasta la redirección final.

Actores y componentes involucrados: Participan cinco elementos en la secuencia: el Usuario (actor externo), el navegador, Spring Security, UsuarioService y la base de datos MySQL. Los mensajes fluyen de izquierda a derecha representando cada llamada entre capas.

Secuencia de mensajes: El usuario envía el formulario de login; el navegador hace un POST a /login que Spring Security intercepta. Security delega la verificación a UsuarioService llamando a

loadUserByUsername(email), que consulta UsuarioDAO en MySQL. Con el usuario recuperado, Security compara la contraseña ingresada contra el hash almacenado.

Flujos alternativos: A partir de ese punto el diagrama ramifica tres caminos posibles:

Usuario no encontrado: UsuarioDAO retorna vacío, Security redirige al login con error sin especificar el motivo.

Cuenta bloqueada: UsuarioService detecta el flag bloqueado=true y evalúa tiempoBloqueo; si no ha expirado, retorna error de bloqueo; si expiró, envía un mensaje de actualización a MySQL para restablecer el estado y continúa el flujo normal.

Contraseña incorrecta: UsuarioService incrementa intentosFallidos con un UPDATE a MySQL; al llegar a 3, envía un segundo UPDATE registrando bloqueado=true y tiempoBloqueo, luego retorna error al usuario.

Redirección por rol: Tras una autenticación exitosa, Spring Security lee la colección de authorities del objeto UserDetails y ejecuta la redirección diferenciada: ROLE_ADMINISTRADOR y ROLE_EMPLEADO al panel de administración, ROLE_MINERO y ROLE_USUARIO al dashboard general. Esta redirección ocurre en el mismo objeto Security sin necesidad de una llamada adicional al backend.

En síntesis, el diagrama evidencia que toda la lógica de seguridad — bloqueo, conteo de intentos y redirección por rol — está centralizada en la interacción entre Spring Security y UsuarioService, manteniendo al controlador fuera de esa responsabilidad.

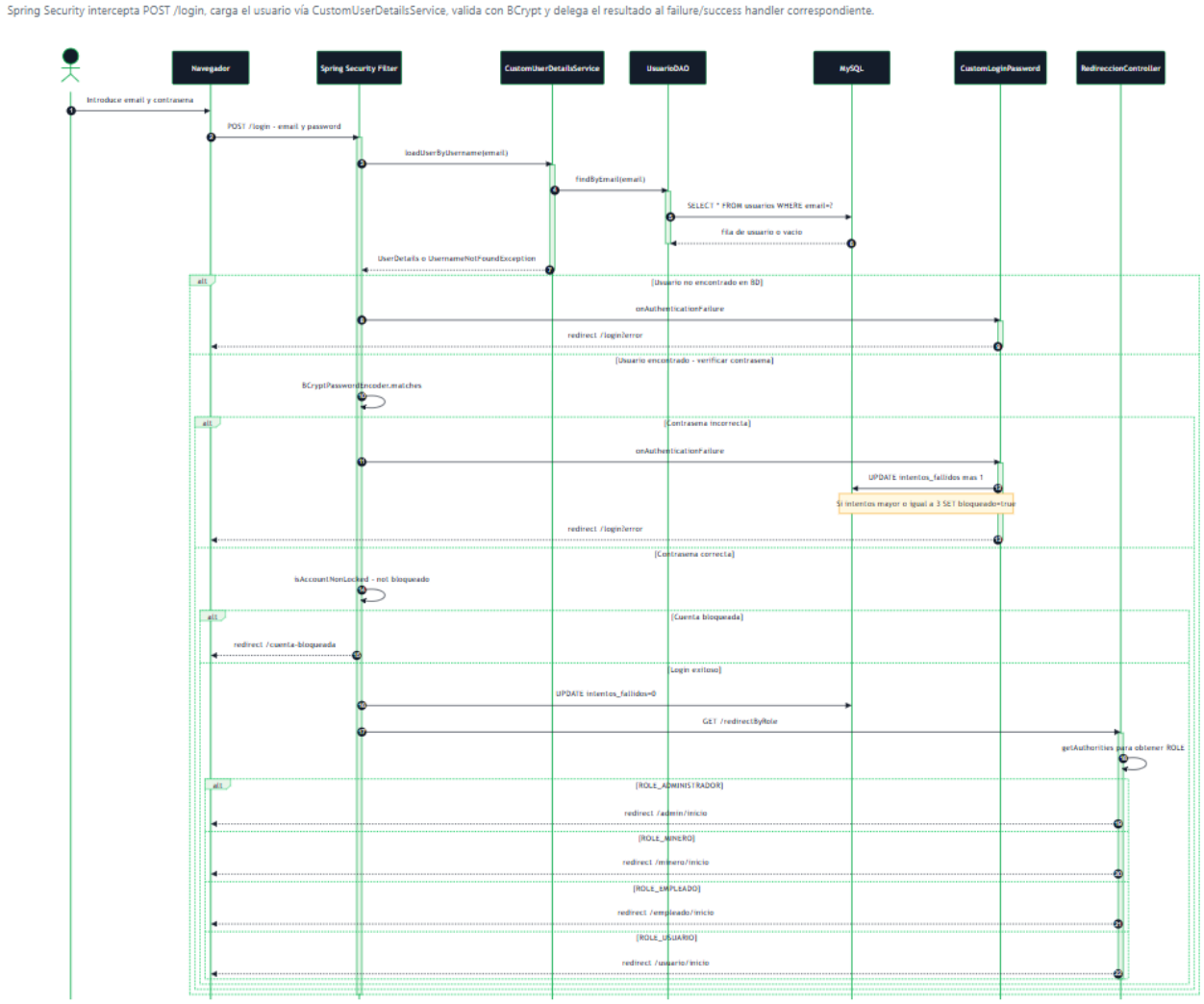


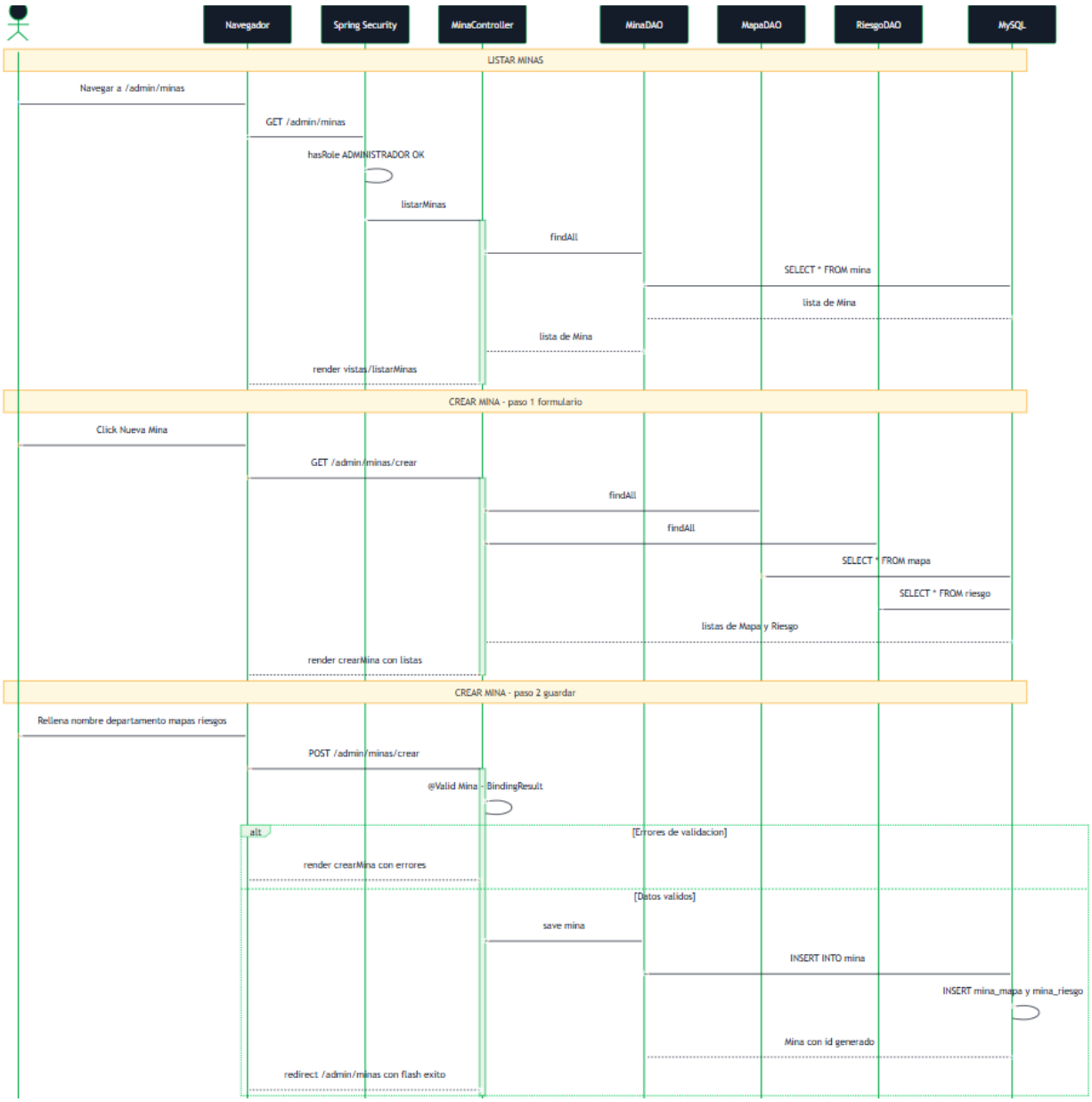
Ilustración 42. Diagrama de Secuencia — Flujo de Autenticación

Fuente: Elaboración propia

Diagrama de Secuencia — Gestión de Minas (CRUD).

Cubre los cuatro flujos: Listar, Crear (con @Valid), Editar y Eliminar (desasociando usuarios antes del DELETE para preservar integridad referencial).

Requiere rol ADMINISTRADOR. MinaController accede directamente a los repositorios (MinaDAO, MapaDAO, RiesgoDAO) sin capa de servicio intermedia.


Navegador
Spring Security
MinaController
MinaDAO
MapaDAO
RiesgoDAO
MySQL
LISTAR MINAS
Navegar a /admin/minas
GET /admin/minas
hasRole ADMINISTRADOR OK
listarMinas
findAll
SELECT * FROM mina
lista de Mina
lista de Mina
render vistas/listarMinas
CREAR MINA - paso 1 formulario
Click Nueva Mina
GET /admin/minas/crear
findAll
findAll
SELECT * FROM mapa
SELECT * FROM riesgo
listas de Mapa y Riesgo
render crearMina con listas
CREAR MINA - paso 2 guardar
Rellena nombre departamento mapas riesgos
POST /admin/minas/crear
@Valid Mina - BindingResult
alt
[Errores de validacion]
render crearMina con errores
[Datos validos]
save mina
INSERT INTO mina
INSERT mina_mapa y mina_riesgo
Mina con id generado
redirect /admin/minas con flash exito

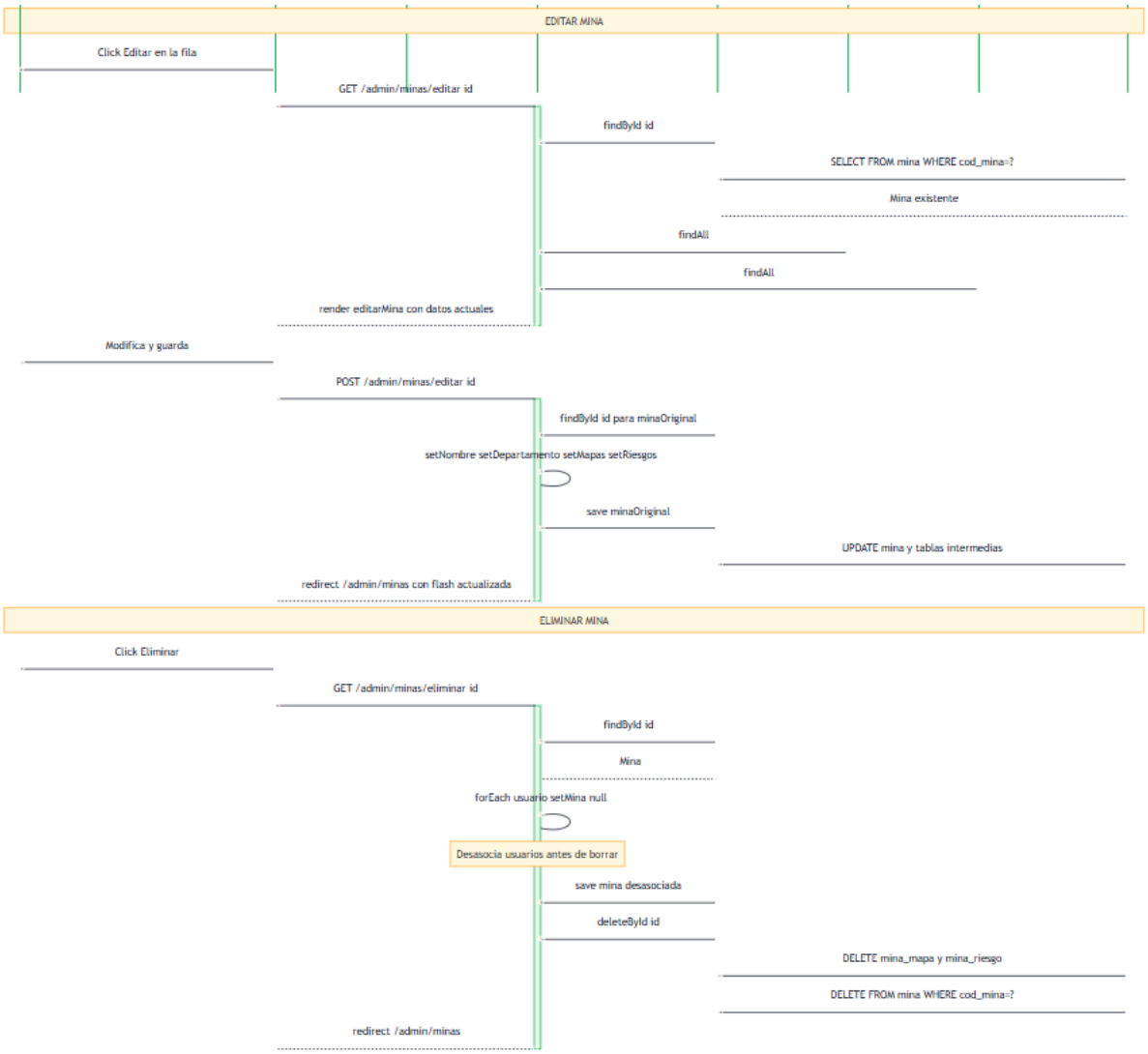


Ilustración 43. Diagrama de Secuencia — Gestión de Minas (CRUD).

Fuente: Elaboración propia

Diagrama de Secuencia — Módulo Foro

El foro es parcialmente público: cualquier visitante puede leer y publicar. Solo el rol administrador puede eliminar posts. La eliminación usa CascadeType.ALL borrando comentarios automáticamente.

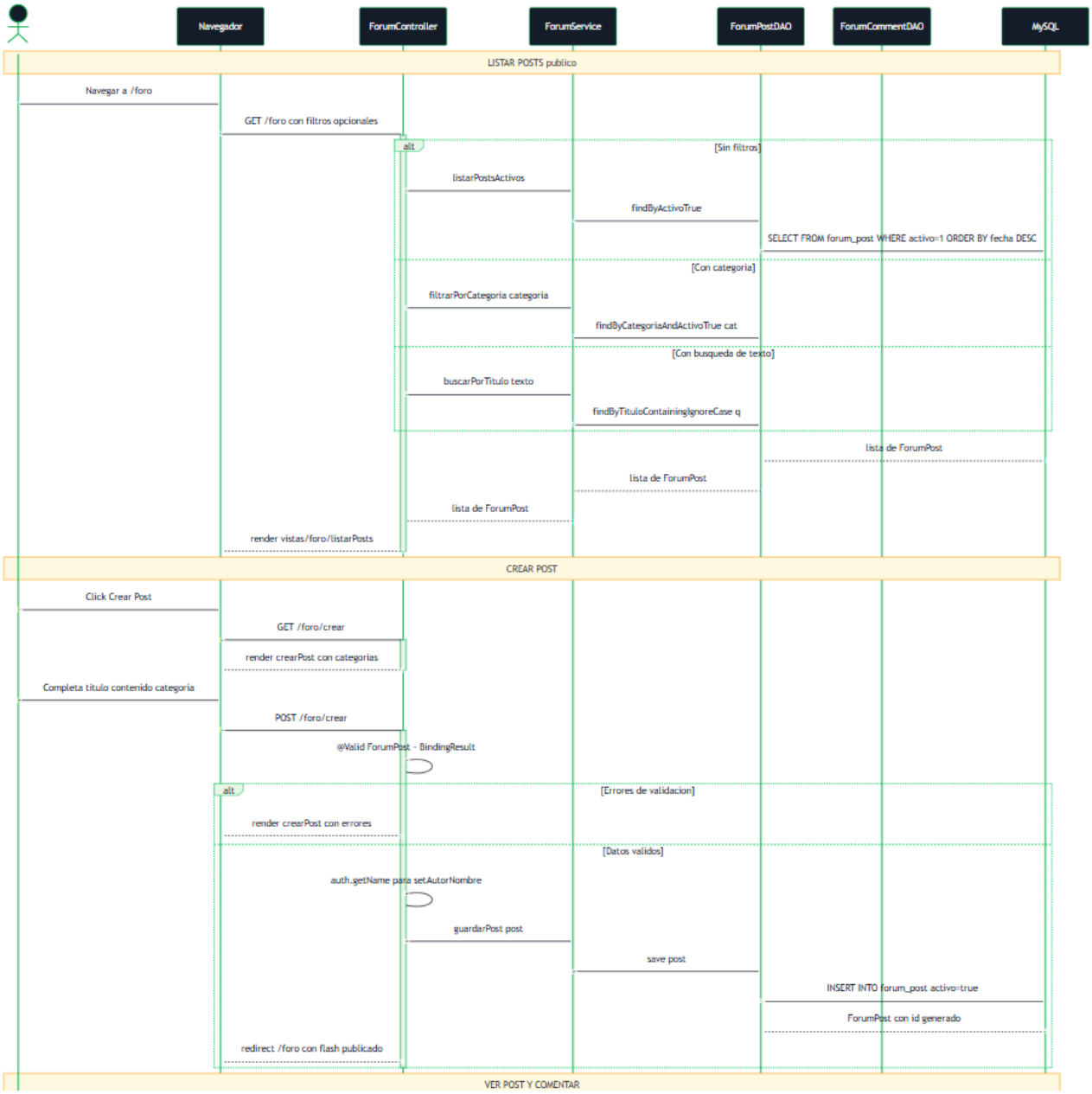
El diagrama cubre cuatro flujos sobre los endpoints GET y POST de /foro, todos mediados por ForumController → ForumService → ForumPostDAO.

Listar posts El usuario accede a /foro; el controlador llama a ForumService, que aplica los filtros disponibles — por categoría o búsqueda por título — antes de consultar ForumPostDAO. Solo retorna posts con activo=true. El resultado se inyecta en la vista Thymeleaf y se renderiza al navegador.

Crear post El usuario envía el formulario POST; el controlador extrae el nombre del autor directamente del objeto Authentication de Spring Security, evitando que el cliente manipule ese dato. ForumService construye el objeto ForumPost y lo persiste vía ForumPostDAO, redirigiendo al listado al finalizar.

Comentar El usuario envía un comentario sobre un post existente. El controlador recupera el ForumPost padre por ID, el autor se toma igualmente de Authentication, y ForumService persiste el ForumComment asociado. Gracias a la relación CascadeType.ALL definida en el modelo, no se requiere una llamada adicional al ForumCommentDAO.

Eliminar post Solo disponible para el autor o un administrador. El controlador verifica la autorización, llama a ForumService que ejecuta el borrado lógico (activo=false) o físico según la implementación; el orphanRemoval configurado en la entidad garantiza que los comentarios asociados se eliminen automáticamente en cascada.


Navegador
ForumController
ForumService
ForumPostDAO
ForumCommentDAO
MySQL
LISTAR POSTS publico
Navegar a /foro
GET /foro con filtros opcionales
alt
[Sin filtros]
listarPostsActivos
findByActivoTrue
SELECT FROM forum_post WHERE activo=1 ORDER BY fecha DESC
[Con categoria]
filtrarPorCategoria categoria
findByCategoriaAndActivoTrue cat
[Con busqueda de texto]
buscarPorTitulo texto
findByTituloContainingIgnoreCase q
lista de ForumPost
lista de ForumPost
lista de ForumPost
render vistas/foro/listarPosts
CREAR POST
Click Crear Post
GET /foro/crear
render crearPost con categorias
Completa titulo contenido categoria
POST /foro/crear
@Valid ForumPost - BindingResult
alt
[Errores de validacion]
render crearPost con errores
[Datos validos]
auth.getName para setAutorNombre
guardarPost post
save post
INSERT INTO forum_post activo=true
ForumPost con id generado
redirect /foro con flash publicado
VER POST Y COMENTAR

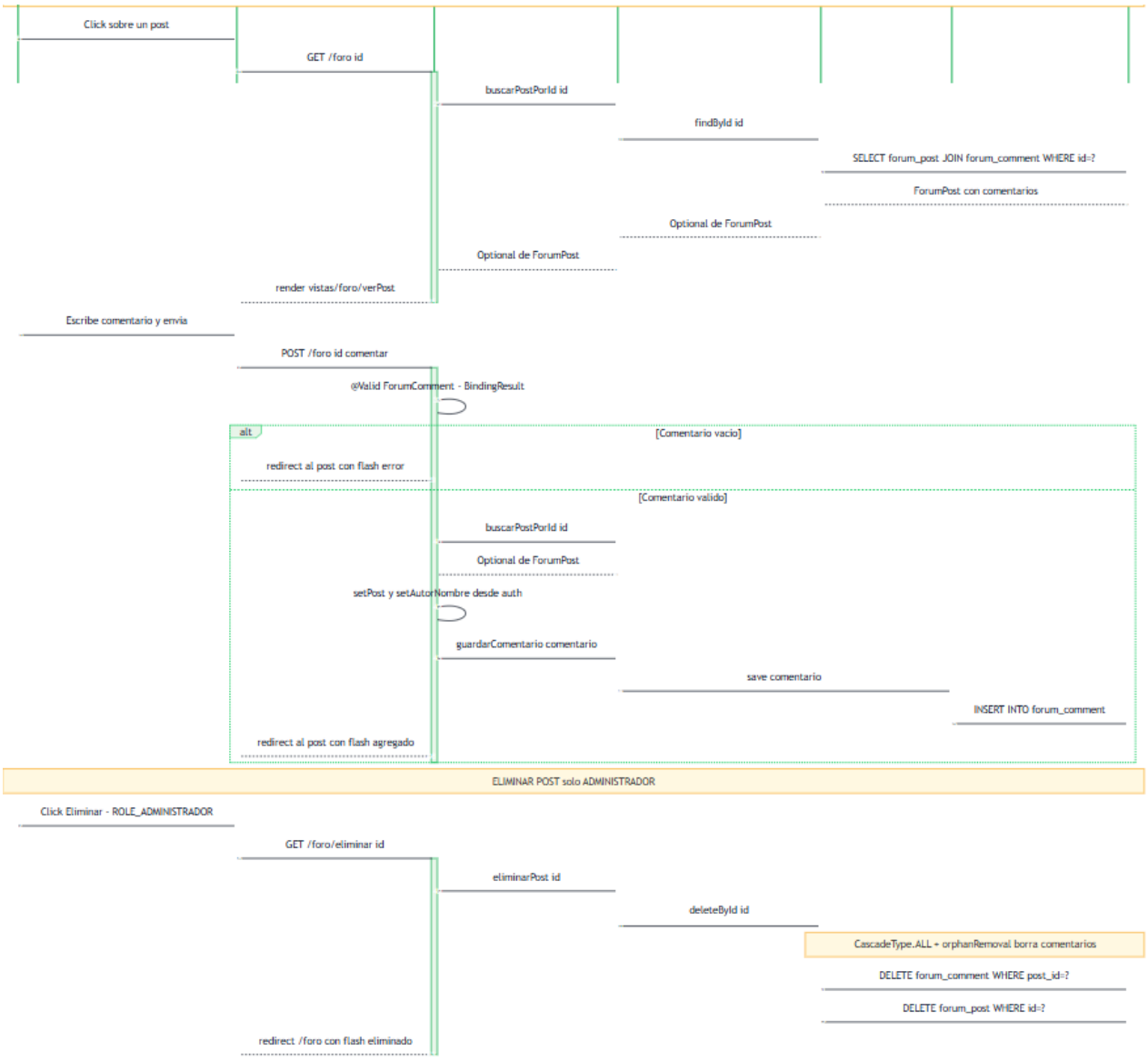


Ilustración 44. Diagrama de Secuencia — Módulo Foro

Fuente: Elaboración propia

Diagramas de actividades

Con el fin de complementar los diagramas de casos de uso y representar el comportamiento dinámico del sistema, se incluyen diagramas de actividades que describen el flujo de acciones, decisiones y validaciones involucradas en los procesos funcionales del sistema SoftGold.

Estos diagramas permiten visualizar la secuencia lógica de ejecución de los procesos más representativos, desde su inicio hasta su finalización.

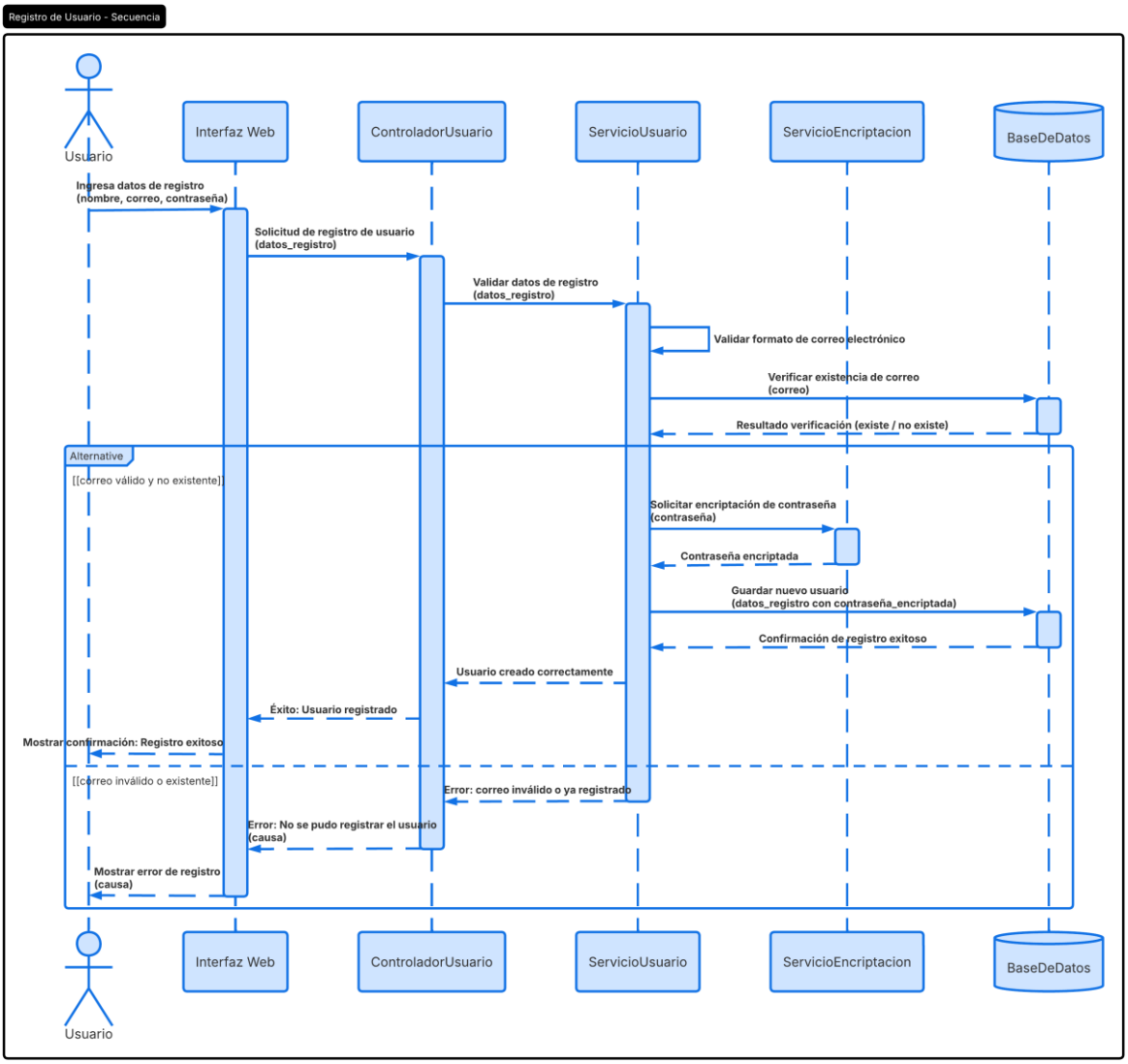


Ilustración 45. Diagrama de actividad– Registro de usuario

Fuente: Elaboración propia

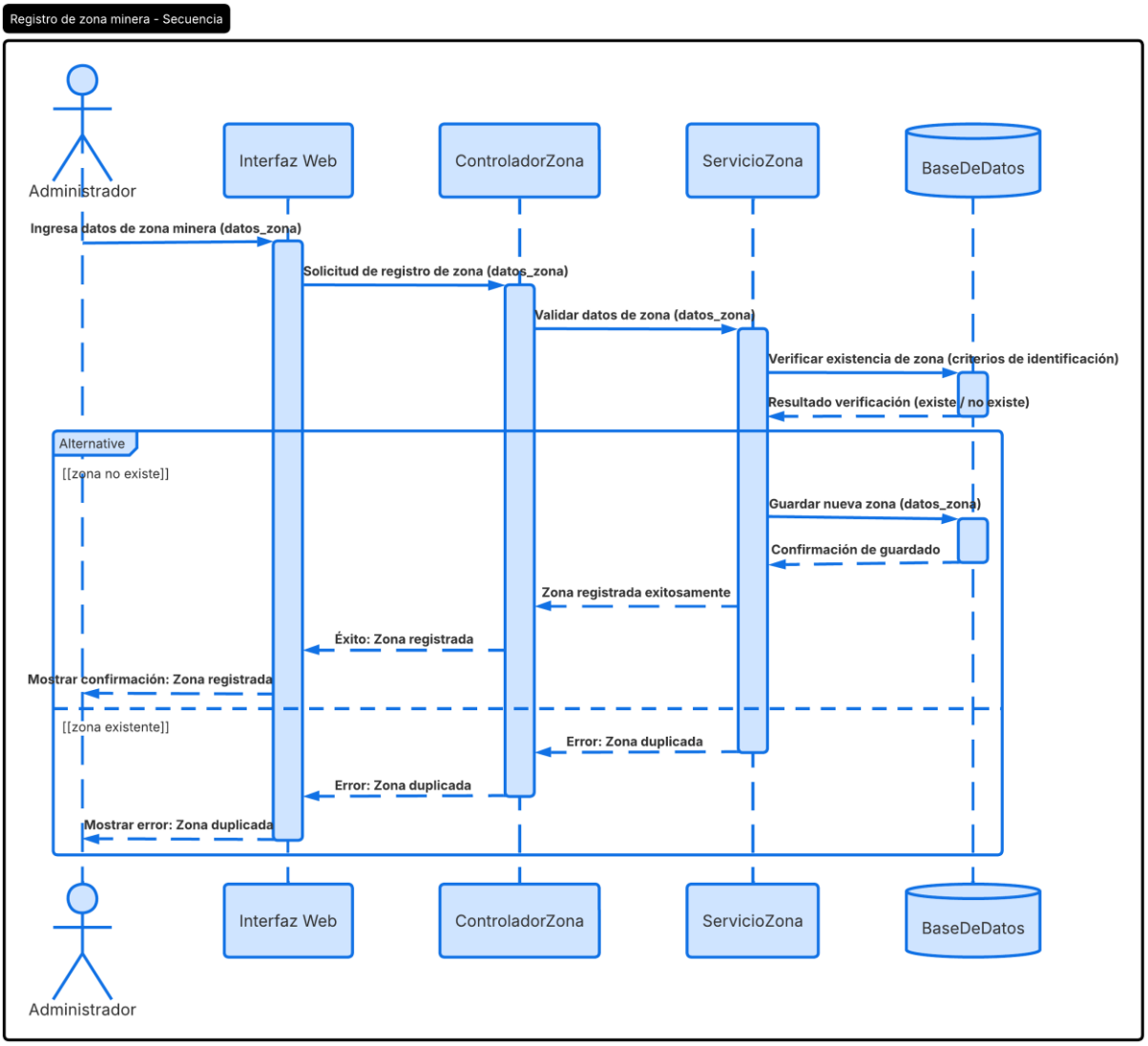


Ilustración 46. Diagrama de actividad – Registro Zona

Fuente: Elaboración propia

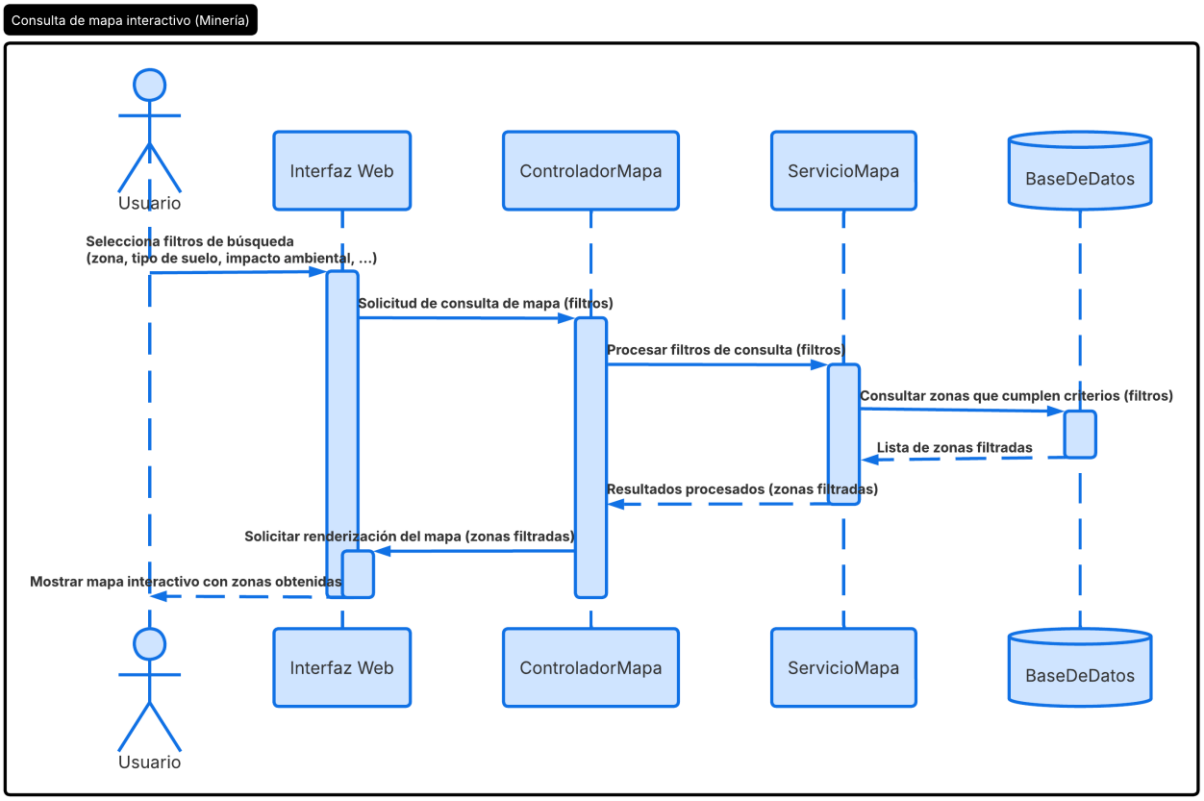


Ilustración 47. Diagrama de actividad – Consulta Mapa

Fuente: Elaboración propia

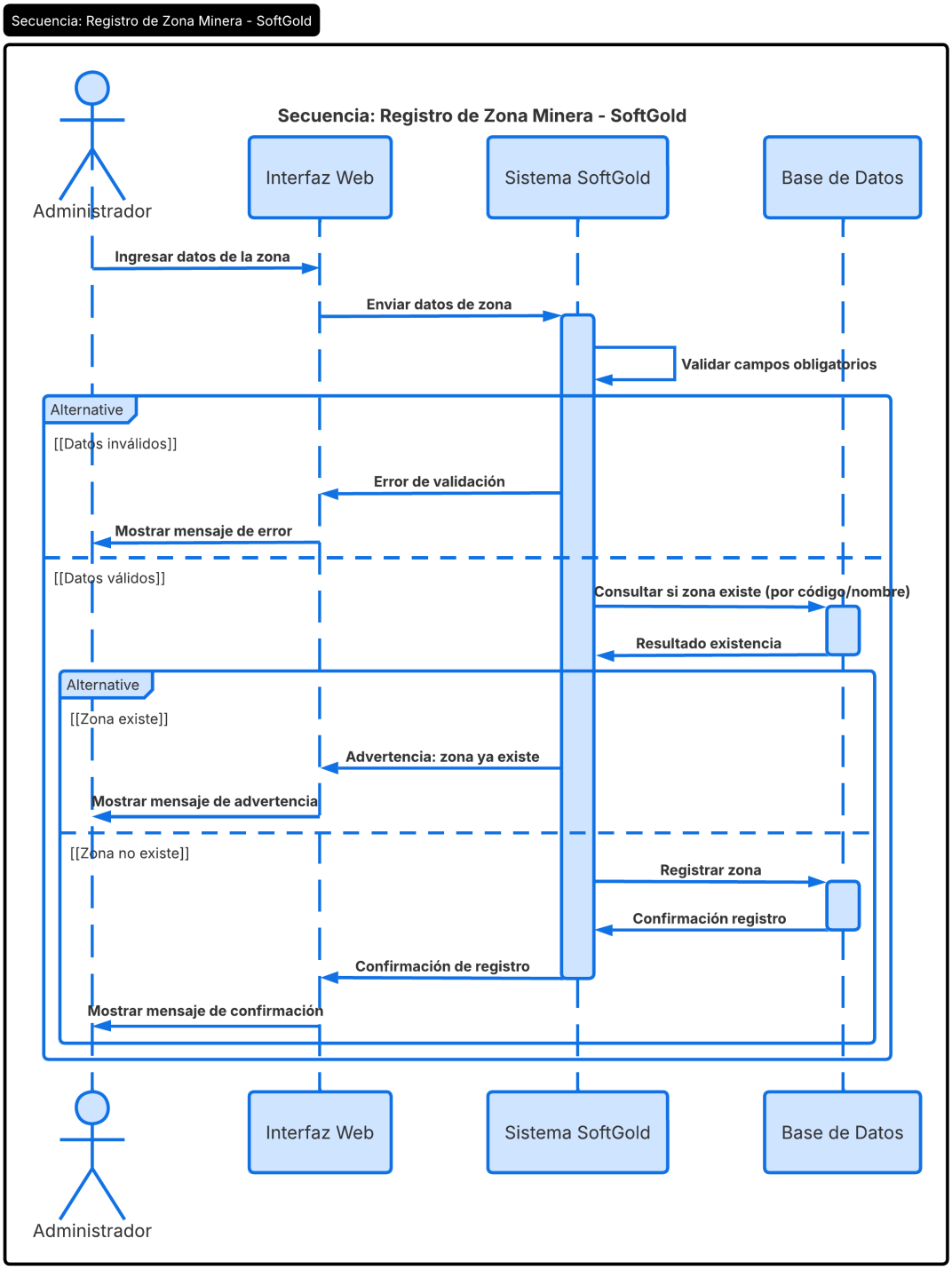


Ilustración 48 .Diagrama de actividad – Registro de zona

Fuente: Elaboración propia

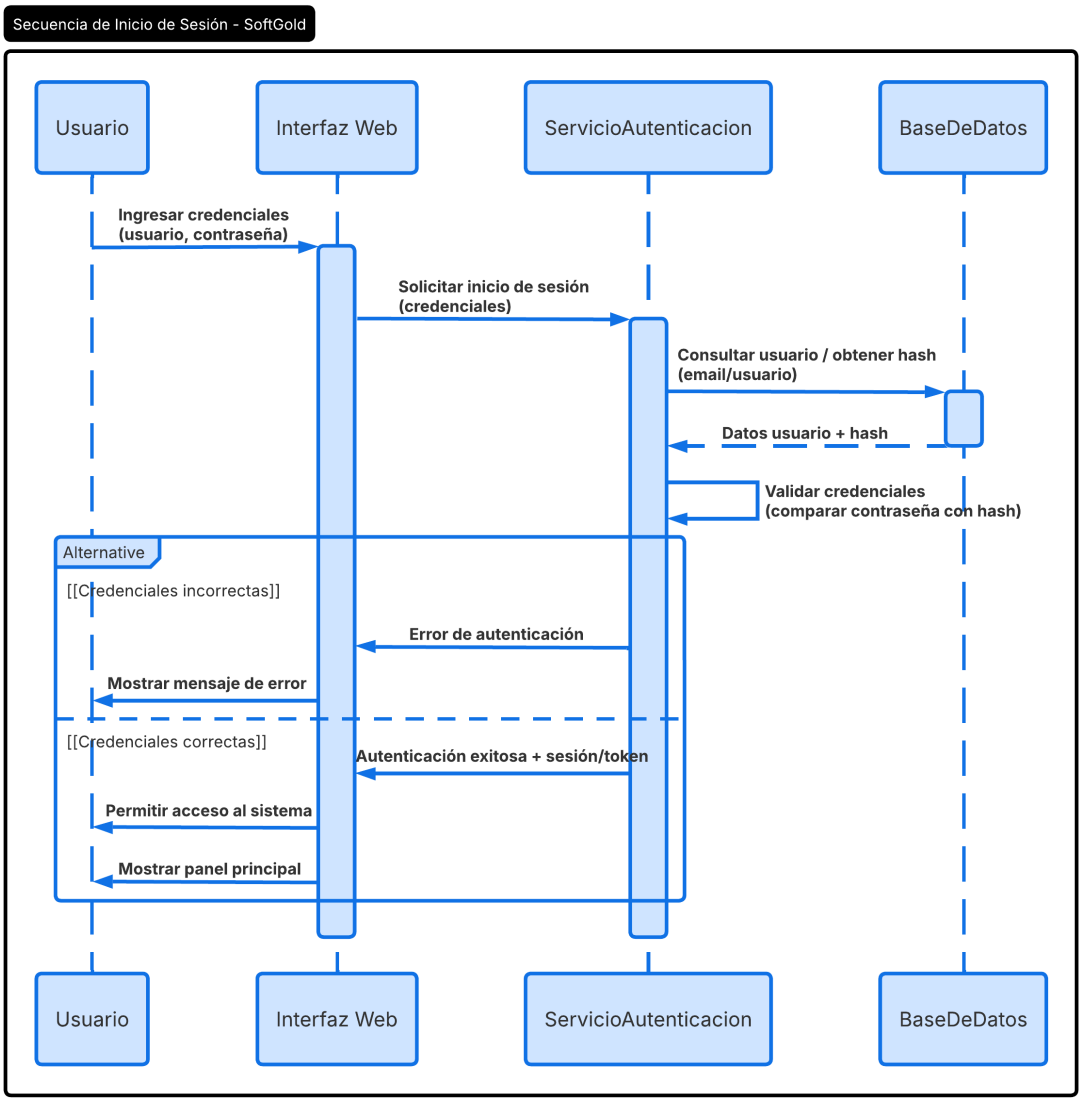


Ilustración 49. Diagrama de actividad – Inicio de sesión

Fuente: Elaboración propia

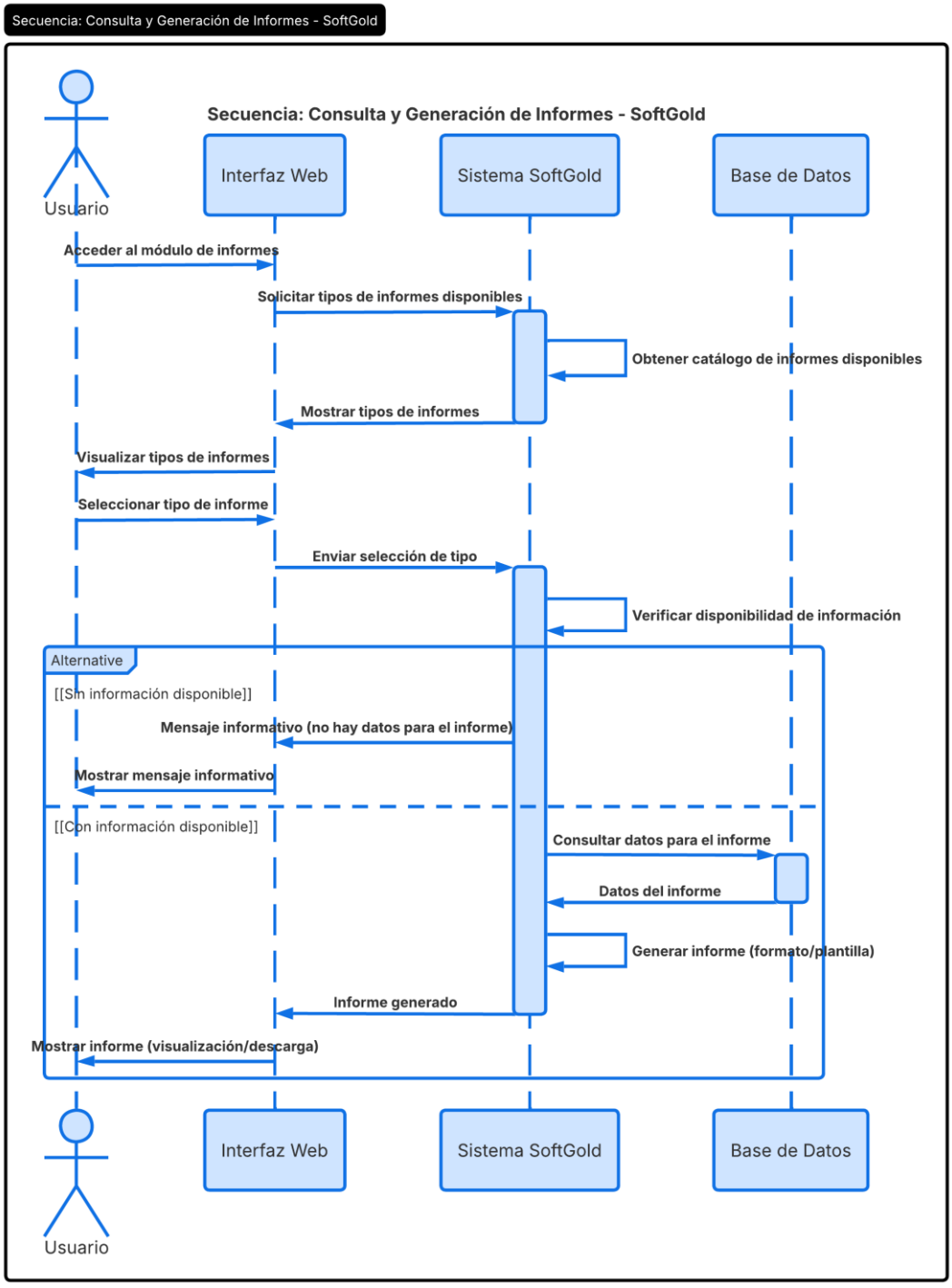


Ilustración 50. Diagrama de actividad – Consulta y generación de informes

Fuente: Elaboración propia

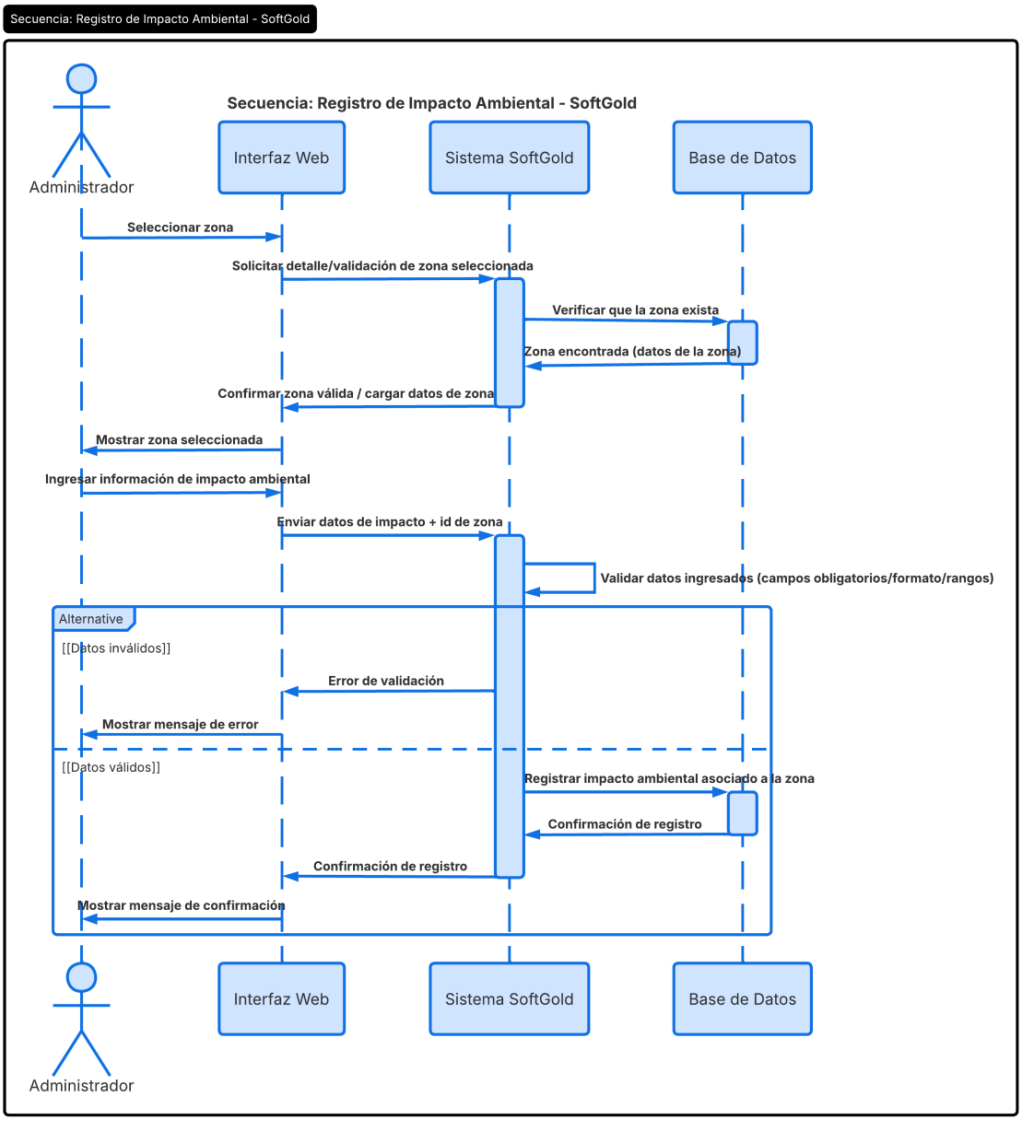


Ilustración 51. Diagrama de actividad – Registro de impacto ambiental

Fuente: Elaboración propia

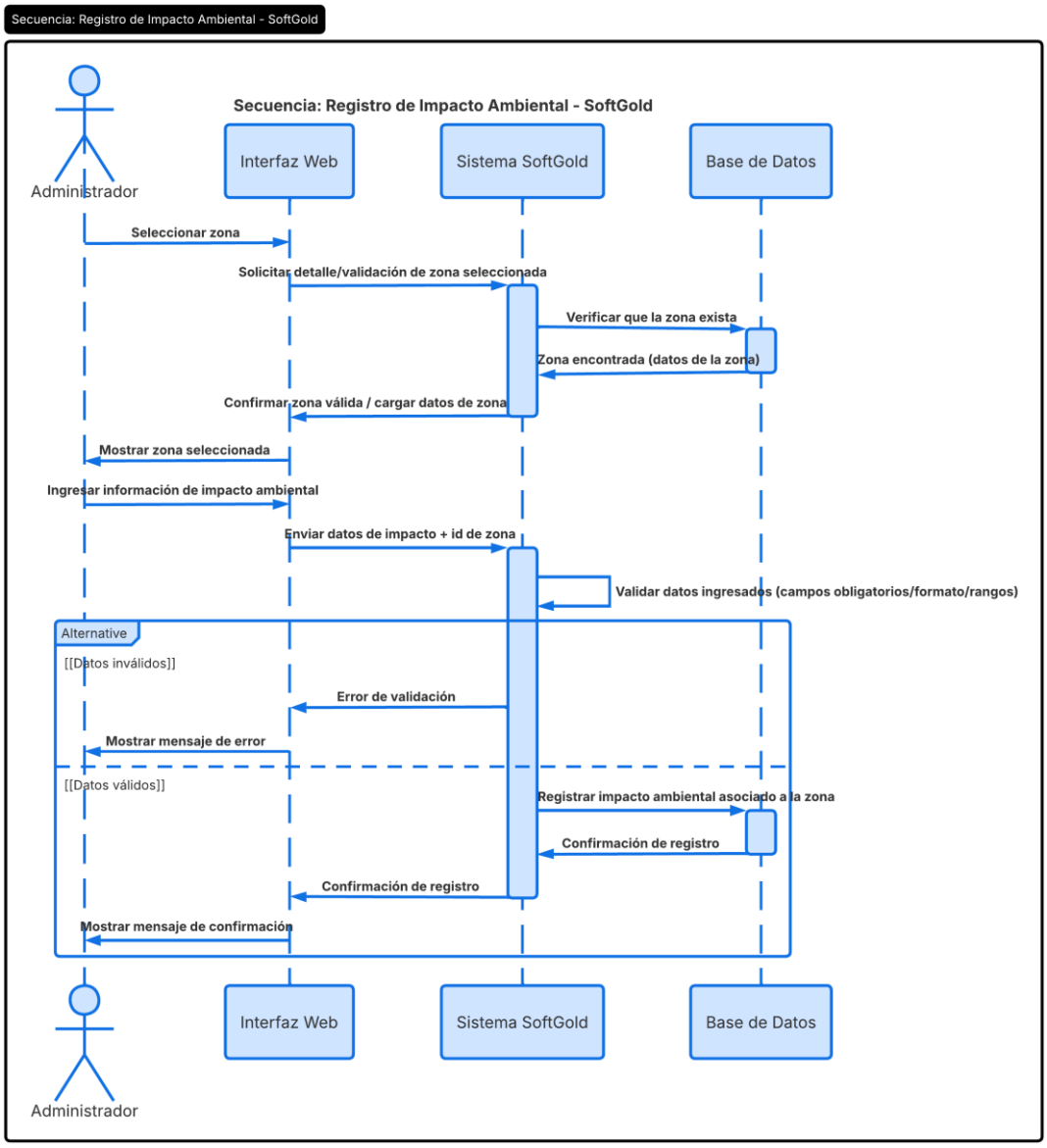


Ilustración 52. Diagrama de actividad – Publicación de comentario en el foro

Fuente: Elaboración propia

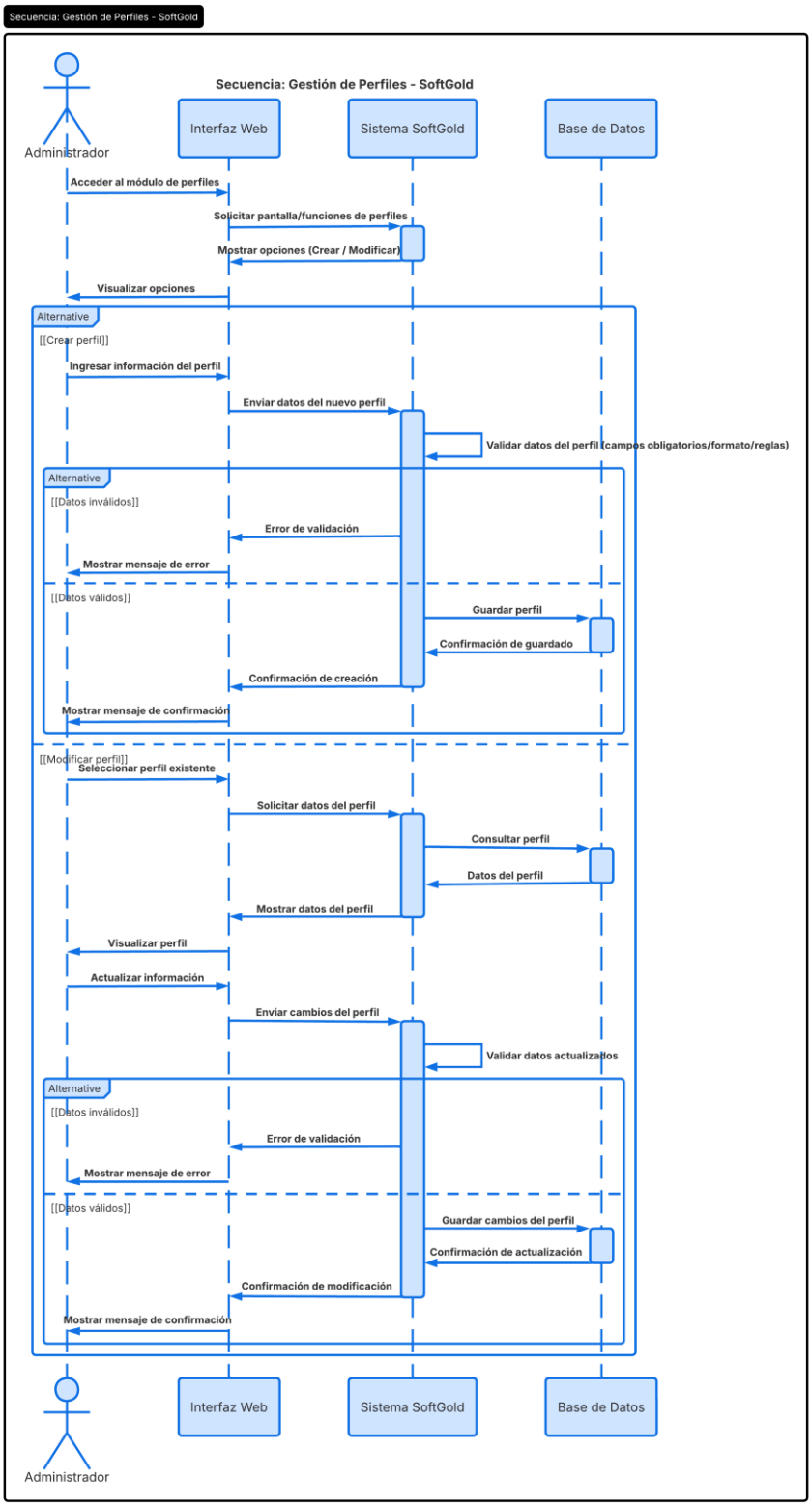


Ilustración 53. Diagrama de actividad – Gestión de perfiles

Fuente: Elaboración propia

Mapa de Navegación

El mapa de navegación presenta la estructura general del sistema SoftGold y el flujo de acceso a sus principales módulos. Desde la página de inicio, el usuario puede acceder a los módulos de Gestión, Foro, Informes, Consultas y Herramientas. El módulo de Gestión incluye funcionalidades como zonas, departamentos, impacto ambiental, tipos de suelo, tipos de excavación, minas, minas exploradas y mapas. El módulo de Filtros permite refinar la visualización de información en el mapa según criterios como tipo de excavación, tipo de mapa y zonas exploradas.

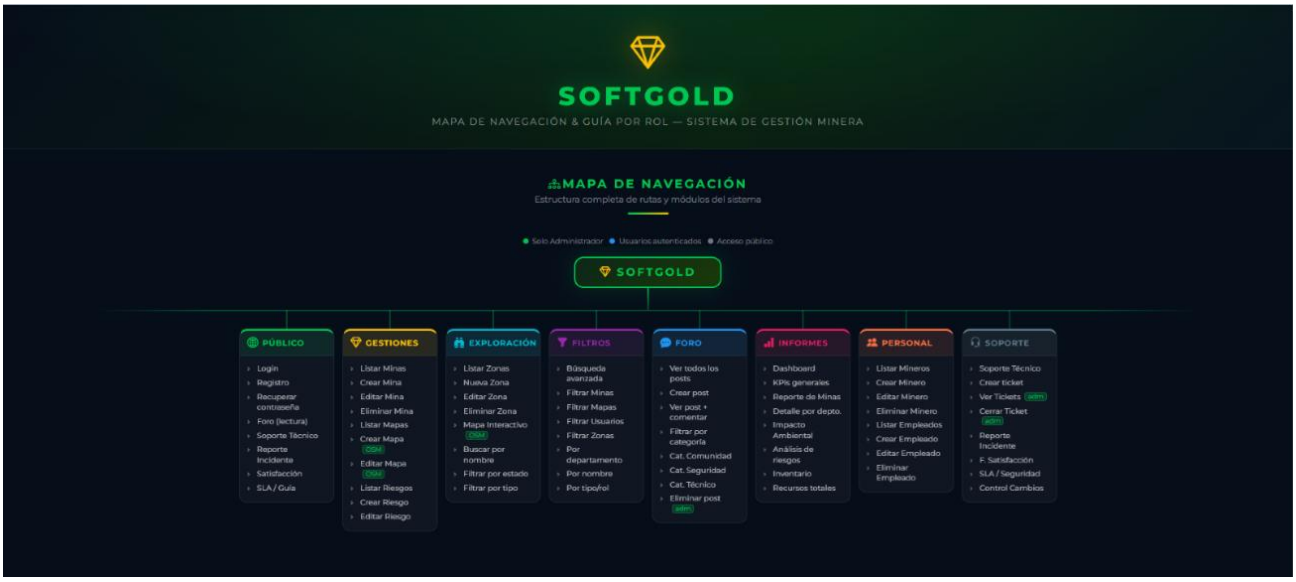


Ilustración 54. Mapa de navegación del sistema SoftGold

Fuente: Elaboración propia

Recursos del Proyecto

Nota. Recursos de hardware, software y talento humano utilizados en el desarrollo del sistema.

Recurso	Modelo	Descripción	Cant.	Precio	Total	Versión
SOFTWARE	Java	Lenguajes de programación	1	$0	$0	Java 17
SOFTWARE	Visual Studio Code	Entorno de desarrollo	1	$0	$0	1.87
SOFTWARE	MySQL	Sistema de gestión de bases de datos	1	$0	$0	8
SOFTWARE	Spring Boot	Framework para crear el software	1	$0	$0	3.2x

TALENTO HUMANO	Ingeniero de sistemas	Encargado de que el software funcione correctamente y se cumplan los requerimientos	1	$3.500.000	$3.500.000	-
TALENTO HUMANO	Desarrollador fullstack	Encargado tanto del Backend como del Frontend del software	3	$4.000.000	$12.000.000	-

Tabla 13. Recursos del Proyecto

Fuente: Elaboración propia

Criterios de aceptación

Criterio de Aceptación	Contexto	Evento	Resultado / Comportamiento esperado
HU-001 – Visualización de mapa	Cuando el minero desee consultar el mapa interactivo	El usuario accede al módulo de mapa	El sistema muestra el mapa con las zonas registradas
HU-001 – Interacción con zonas	Cuando el usuario seleccione una zona del mapa	El usuario hace zoom y selecciona una zona	El sistema muestra el detalle de la zona seleccionada
HU-001 – Mapa sin zonas	Cuando no existan zonas registradas	El usuario accede al mapa	El sistema muestra un mensaje informativo
HU-002 – Publicación de comentario	Cuando el usuario quiera compartir una experiencia	El usuario ingresa un comentario en el foro	El sistema valida que esté autenticado
HU-002 – Registro del comentario	Cuando el comentario sea enviado	El usuario confirma la publicación	El sistema registra fecha y hora automáticamente
HU-002 – Visualización del comentario	Después de publicar el comentario	El sistema procesa la solicitud	El comentario queda visible para otros usuarios
HU-003 – Validación de zona	Cuando el administrador registre un impacto ambiental	El administrador selecciona una zona	El sistema valida que la zona exista
HU-004 – Listado de zonas	Cuando el usuario consulte zonas menos exploradas	El usuario accede a la consulta	El sistema muestra un listado de zonas con baja actividad
HU-004 – Detalle de zona	Cuando el usuario seleccione una zona	El usuario consulta el detalle	El sistema muestra la información completa
HU-005 – Confirmación de acción	Cuando el administrador intente inhabilitar un minero	El administrador selecciona la opción inhabilitar	El sistema solicita confirmación

HU-005 – Cambio de estado	Cuando el administrador confirme la acción	El sistema procesa la solicitud	El estado del minero cambia a inactivo
HU-005 – Registro de acción	Después de inhabilitar	El sistema guarda la operación	La acción queda registrada en el sistema
HU-006 – Crear zona	Cuando el administrador requiera registrar una nueva zona	El administrador diligencia el formulario de creación de zona	El sistema valida los campos obligatorios, guarda la información y muestra mensaje de confirmación
HU-006 – Consultar zona	Cuando el usuario desee consultar zonas registradas	El usuario accede al módulo de zonas	El sistema muestra el listado de zonas disponibles
HU-006 – Modificar zona	Cuando el administrador requiera actualizar una zona existente	El administrador edita la información de una zona	El sistema valida y actualiza los datos correctamente
HU-006 – Inhabilitar zona	Cuando el administrador requiera desactivar una zona	El administrador selecciona la opción inhabilitar	El sistema cambia el estado de la zona y registra la acción
HU-007 – Crear departamento	Cuando el administrador requiera registrar un nuevo departamento	El administrador ingresa los datos del departamento	El sistema valida y almacena la información
HU-007 – Consultar departamentos	Cuando el usuario requiera consultar departamentos	El usuario accede al módulo de departamentos	El sistema muestra el listado de departamentos
HU-007 – Modificar departamento	Cuando el administrador requiera actualizar un departamento	El administrador edita la información	El sistema guarda los cambios correctamente
HU-007 – Inhabilitar departamento	Cuando el administrador requiera desactivar un departamento	El administrador confirma la acción	El sistema inhabilita el departamento y registra la operación
HU-008 – Crear mina	Cuando el administrador requiera registrar una nueva mina	El administrador diligencia el formulario de mina	El sistema valida y almacena la información
HU-008 – Consultar minas	Cuando el usuario requiera consultar minas registradas	El usuario accede al módulo de minas	El sistema muestra el listado de minas
HU-008 – Modificar mina	Cuando el administrador requiera actualizar una mina	El administrador edita los datos de la mina	El sistema guarda los cambios correctamente

HU-008 – Inhabilitar mina	Cuando el administrador requiera desactivar una mina	El administrador confirma la acción	El sistema cambia el estado de la mina y registra la acción
HU-009 – Selección de filtros	Cuando el usuario requiera filtrar zonas en el mapa	Cuando el usuario requiera filtrar zonas en el mapa	El sistema aplica los filtros seleccionados
HU-009 – Actualización del mapa	Después de aplicar los filtros	El sistema procesa la solicitud	El mapa se actualiza mostrando solo las zonas filtradas
HU-009 – Limpieza de filtros	Cuando el usuario quite los filtros aplicados	El usuario limpia los filtros	El sistema muestra nuevamente todas las zonas
HU-010 – Selección de informe	Cuando el usuario requiera consultar un informe	Cuando el usuario requiera consultar un informe	El sistema genera el informe solicitado
HU-010 – Visualización de informe	Después de generar el informe	El sistema muestra la información	El informe se presenta con datos actualizados
HU-010 – Acceso a informes	Cuando no existan datos suficientes	El usuario consulta el informe	El sistema muestra un mensaje informativo
HU-011 – Crear detector	Cuando el administrador requiera registrar un detector	El administrador ingresa los datos del detector	El sistema valida y guarda la información
HU-011 – Consultar detectores	Cuando el usuario requiera consultar detectores registrados	El usuario accede al módulo de detectores	El sistema muestra el listado de detectores
HU-011 – Modificar detector	Cuando el administrador requiera actualizar un detector	El administrador edita la información	El sistema guarda los cambios correctamente
HU-011 – Inhabilitar detector	Cuando el administrador requiera desactivar un detector	El administrador confirma la acción	El sistema inhabilita el detector y registra la operación
HU-012 – Consulta de riesgos	Cuando el usuario requiera conocer los riesgos de una zona	El usuario selecciona una zona	El sistema muestra los riesgos asociados
HU-012 – Detalle de riesgo	Cuando el usuario seleccione un riesgo	El usuario consulta el detalle	El sistema muestra la información completa del riesgo

HU-012 – Zona sin riesgos	Cuando una zona no tenga riesgos registrados	El usuario realiza la consulta	El sistema muestra un mensaje informativo

Tabla 14. Criterios de aceptación

Fuente: Elaboración propia

Product Backlog

Nota. Product Backlog completo con historias de usuario priorizadas por módulo.

ID	Historia de usuario	Módulo	Prioridad	Story Points	Estado
1	Como administrador, quiero registrar zonas con coordenadas y tipo de suelo para mantener el catálogo actualizado.	Zonas	Alta	5	Completado
2	Como usuario quiero consultar una zona priorizadas según información registrada	Zonas	Alta	2	Completado
3	Como administrador quiero modificar una zona para mantener su información actualizada	Zona	Media	2	Completado
4	Como administrador, quiero inhabilitar una zona para evitar su uso cuando no sea válida	Zona	Media	3	Completado
5	Como usuario quiero cancelar una operación en zonas para revertir cambios no deseados	Zona	Baja	1	Completado
6	Como usuario, quiero salir del módulo de zonas para volver al menú principal	Zona	Baja	1	Completado
7	Como administrador, quiero crear un departamento para registrar su información	Departamento	Alta	5	Completado
8	Como usuario, quiero consultar un departamento para visualizar su información	Departamento	Alta	2	Completado
9	Como administrador, quiero modificar un departamento para actualizar su información	Departamento	Media	3	Completado
10	Como administrador, quiero inhabilitar un departamento para controlar su disponibilidad	Departamento	Media	2	Completado
25	Como administrador, quiero crear un impacto ambiental para registrar afectaciones y recomendaciones	Impacto ambiental	Alta	8	Completado
26	Como usuario, quiero consultar impactos ambientales para conocer la información registrada por zona	Impacto ambiental	Alta	3	Completado
27	Como administrador, quiero modificar un impacto ambiental para mantener la información actualizada	Impacto ambiental	Media	3	Completado
28	Como administrador, quiero inhabilitar un impacto ambiental para controlar su vigencia	Impacto ambiental	Media	2	Completado

31	Como administrador, quiero crear un tipo de impacto ambiental para clasificar impactos	Tipo impacto	Medio	5	Completado
32	Como usuario, quiero consultar tipos de impacto ambiental para comprender la clasificación	Tipo impacto	Medio	3	Completado
38	Como administrador, quiero crear un tipo de suelo para registrar su información	Tipo de suelo	Alto	5	Completado
39	Como usuario, quiero consultar tipos de suelo para conocer sus características	Tipo de suelo	Medio	3	Completado
44	Como administrador, quiero crear un tipo de excavación para registrar su descripción	Tipos de excavación	Medio	6	Completado
45	Como usuario, quiero consultar tipos de excavación para conocer alternativas registradas	Tipos de excavación	Medio	3	Completado
51	Como usuario, quiero crear una publicación en el foro, para compartir experiencias.	Foro	Media	6	Completado
52	Como usuario, quiero consultar el foro, para ver publicaciones disponibles.	Foro	Media	3	Completado
53	Como usuario, quiero modificar mi publicación, para corregir información.	Foro	Bajo	5	Completado
54	Como administrador, quiero inhabilitar una publicación, para moderar contenido.	Foro	Media	3	Completado
57	Como visitante, quiero registrarme, para acceder a las funcionalidades del sistema.	Registro usuario	Alta	8	Completado
58	Como administrador, quiero consultar usuarios registrados, para verificar información.	Registro usuario	Alta	7	Completado
59	Como usuario, quiero modificar mis datos, para mantenerlos actualizados.	Registro usuario	Media	5	Completado
60	Como administrador, quiero inhabilitar un usuario, para controlar el acceso.	Registro usuario	Alta	3	Completado
64	Como administrador, quiero registrar un detector de oro con su información, para mantener un inventario organizado.	Detectores	Media	5	Completado
65	Como administrador, quiero consultar detectores de oro, para verificar su información.	Detectores	Media	4	Completado
76	Como administrador, quiero crear un minero, para registrar su información.	Minero	Alta	5	Completado
77	Como administrador, quiero consultar un minero, para ver su información.	Minero	Media	4	Completado
78	Como administrador, quiero modificar un minero, para actualizar sus datos.	Minero	Media	4	Completado
79	Como administrador, quiero inhabilitar un minero, para restringir acceso si aplica.	Minero	Media	3	Completado
88	Como administrador, quiero crear una mina, para registrar su información.	Mina	Alta	5	Completado

89	Como usuario, quiero consultar una mina, para ver su información registrada.	Mina	Alta	4	Completado
90	Como administrador, quiero modificar una mina, para mantener datos actualizados.	Mina	Media	4	Completado
94	Como usuario, quiero crear un registro de zona explorada, para guardar información histórica.	Zonas exploradas	Alta	5	Completado
95	Como usuario, quiero consultar zonas exploradas, para evitar repetir excavaciones.	Zonas exploradas	Alta	4	Completado
100	Como administrador, quiero crear un mapa, para registrar información geográfica asociada a zonas.	Mapa	Alta	6	Completado
101	Como usuario, quiero consultar mapas, para visualizar zonas registradas.	Mapa	Alta	5	Completado
119	Como usuario, quiero generar el informe de impacto ambiental, para apoyar decisiones responsables.	Informes	Alta	6	Completado
120	Como usuario, quiero consultar el informe de zonas con alta probabilidad, para priorizar zonas según información registrada.	Informes	Alta	4	Completado
121	Como usuario, quiero consultar el informe de zonas exploradas del último mes, para analizar actividad reciente.	Informes	Media	4	Completado
122	Como usuario, quiero consultar comentarios y recomendaciones, para aprender de experiencias registradas.	Consultas	Media	4	Completado
123	Como usuario, quiero consultar zonas con actividad reciente, para identificar tendencias.	Consultas	Alta	4	Completado
124	Como usuario, quiero consultar zonas menos exploradas, para identificar oportunidades.	Consultas	Alta	4	Completado
126	Como administrador, quiero crear un perfil, para gestionar permisos	Perfil	Alta	5	Completado
138	Como administrador, quiero crear un filtro, para definir criterios de búsqueda en el mapa.	Filtros	Alta	6	Completado
139	Como usuario, quiero consultar/aplicar filtros, para visualizar zonas según criterios seleccionados.	Filtros	Alta	4	Completado
144	Como administrador, quiero crear una excavación, para registrar la información correspondiente.	Excavación	Media	5	Completado
149	Como usuario, quiero salir del módulo excavación, para volver al menú principal.	Excavación	Baja	1	Completado

Tabla 15. Product Backlog

Fuente: Elaboración propia

Reglas de Negocio

Las reglas de negocio son las condiciones con las que el software debe cumplir, teniendo en cuenta los requerimientos de la empresa a la cual está orientado el proyecto, ya que cada empresa opera de forma diferente y por tanto debe tener en cuenta esto para poder desarrollar un buen proyecto. Al momento de realizar estas reglas de negocio, debe ir un identificador que haga referencia a la regla de negocio que se está creando (ej. RNM-001), nombre de la regla, descripción, módulo, condición, acciones o restricciones, fuente (de dónde proviene la regla), prioridad (nivel de importancia de la regla) y así poder realizarlas acorde a lo estipulado.

Las reglas de negocio permiten que el software esté alineado con los requerimientos de la empresa y se debe tener en cuenta todo esto para así evitar futuros errores y asegurar el correcto funcionamiento del software.

Fuente: Elaboración propia

Nota. Reglas de negocio por módulo del sistema SoftGold.

ID	Nombre de la regla	Descripción	Módulo	Condición	Acción	Fuente	Prioridad
RNM-001	**Zona válida**	Una zona solo puede registrarse si pertenece a un departamento existente	Zonas	Al crear o modificar zona	Bloquear registro sin departamento	Política interna	Alta
RNM-002	**Departamento único**	No pueden existir departamentos duplicados	Departamentos	Al crear departamento	Validar unicidad	Política interna	Media
RNM-003	**Asociación mina-zona**	Toda mina debe estar asociada a una zona registrada	Minas	Al registrar mina	Bloquear si no hay zona	Política interna	Alta
RNM-004	**Registro de exploración**	Una zona explorada debe estar asociada a una mina	Minas exploradas	Al registrar exploración	Validar asociación	Política interna	Alta
RNM-005	**Impacto obligatorio**	Cada zona debe tener al menos un impacto ambiental registrado	Impacto ambiental	Al consultar zona	Solicitar registro	Normativa ambiental	Alta
RNM-006	**Tipo de impacto válido**	Todo impacto debe tener un tipo definido	Tipo impacto ambiental	Al registrar impacto	Validar tipo	Normativa ambiental	Media

RNM-007	**Tipo de suelo obligatorio**	Toda zona debe tener tipo de suelo asociado	Tipos de suelo	Al registrar zona	Validar asociación	Política interna	Media
RNM-008	**Tipo de excavación definido**	Toda excavación debe tener tipo de excavación asignado	Tipos de excavación	Al registrar excavación	Bloquear sin tipo	Política interna	Media
RNM-009	**Visualización de mapas**	Solo usuarios registrados pueden ver detalle del mapa	Mapas	Al consultar mapa	Restringir a invitados	Política de acceso	Alta
RNM-010	**Tipo de mapa válido**	Todo mapa debe tener un tipo de mapa asociado	Tipos de mapa	Al crear mapa	Validar tipo	Política interna	Media
RNM-011	**Aplicación de filtros**	Los filtros solo se aplican sobre datos existentes	Filtros	Al filtrar mapa	Mostrar mensaje si no hay datos	Política interna	Media
RNM-012	**Registro de riesgos**	Un riesgo debe estar asociado a una zona	Riesgos	Al registrar riesgo	Bloquear sin zona	Política de seguridad	Alta
RNM-013	**Moderación del foro**	Todo contenido puede ser moderado por el administrador	Foro	Al publicar comentario	Inhabilitar contenido	Política interna	Media
RNM-014	**Gestión de detectores**	Un detector debe estar asociado a una zona	Detectores de oro	Al registrar detector	Validar zona	Política interna	Media
RNM-015	**Tipo de detector válido**	Todo detector debe tener un tipo definido	Tipos de detectores	Al registrar detector	Validar tipo	Política interna	Media
RNM-016	**Acceso a informes**	Los informes solo pueden ser consultados por usuarios autenticados	Informes	Al generar informe	Bloquear invitados	Política interna	Alta
RNM-017	**Consulta controlada**	Las consultas solo muestran información permitida por perfil	Consultas	Al ejecutar consulta	Filtrar datos	Política de acceso	Alta
RNM-018	**Estado de usuario**	Un usuario inhabilitado no puede acceder al sistema	Usuarios	Al iniciar sesión	Bloquear acceso	Política de seguridad	Alta
RNM-019	**Gestión de mineros**	Un minero inhabilitado no puede registrar información	Mineros	Al intentar operar	Bloquear acciones	Política interna	Alta
RNM-020	**Gestión de empleados**	Un empleado solo puede operar según su rol	Empleados	Al realizar acciones	Validar rol	Política interna	Media

Tabla 16. Reglas de negocio por módulo del sistema SoftGold.

Fuente: Elaboración propia

Gestión del Cambio

Según En-Com (2024):

El proceso de cambios en el desarrollo de software implica el enfoque estructurado para gestionar cambios en aplicaciones de software, los componentes claves incluyen identificar las necesidades de cambio, evaluar el impacto de los cambios, planificar e implementar cambios y revisar los resultados.

Un consejo de asesor de cambios (CAB) o un organismo similar normalmente supervisa este proceso, asegurando que los cambios estén alineados con los objetivos comerciales y minimizando los riesgos.

Los cambios en el software se utilizan a menudo para agilizar estos procesos, ofreciendo herramientas para rastrear cambios y gestionar aprobaciones y mantener la documentación para garantizar una transición fluida y controlada (p.1).

Nota. Matriz de gestión del cambio del proyecto SoftGold.

Cambio	Impacto en tiempo	Impacto en Costo	Impacto en Alcance	Impacto en Calidad	Impacto en Recursos	Prioridad	Ref. del argumento
El invitado no tendrá acceso a las minas	Bajo	Bajo	Bajo	Medio	Bajo	Media	Definición de niveles de acceso del sistema y control de seguridad de la información.
Agregar filtros al mapa de exploración	Alto	Medio	Alto	Alto	Medio	Alta	Resultados de la encuesta, donde el 83,3% manifestó necesidad de

							un mapa interactivo con filtros.
Usar Spring Boot como framework	Medio	Medio	Alto	Alto	Alto	Alta	Decisión técnica del equipo de desarrollo para mejorar la arquitectura, escalabilidad y mantenimiento del sistema.
Incorporar un módulo de soporte técnico mediante tickets	Medio	BajoR	Medio	Alto	Bajo	Media	Necesidad identificada durante el desarrollo para registrar y dar seguimiento a las incidencias reportadas por los usuarios.

Tabla 17. Gestión del Cambio

Fuente: Elaboración propia

Documentación del Modelo Entidad-Relación (MER)

Un MER (Modelo Entidad Relación) es como un mapa que le permite al equipo de desarrollo visualizar de una manera organizada la estructura de la base de datos. En este modelo se representan las entidades (tablas), sus atributos y cardinalidades (el tipo de relación entre cada tabla). Entre dos tablas puede haber las siguientes cardinalidades: Uno a muchos (1:N): un registro de la tabla se relaciona con varios registros de la otra tabla; Uno a uno (1:1): un registro de la tabla solo se relaciona con un único registro de la otra tabla; Muchos a muchos (N:M): varios registros de una tabla pueden estar relacionados con varios registros de la otra tabla.

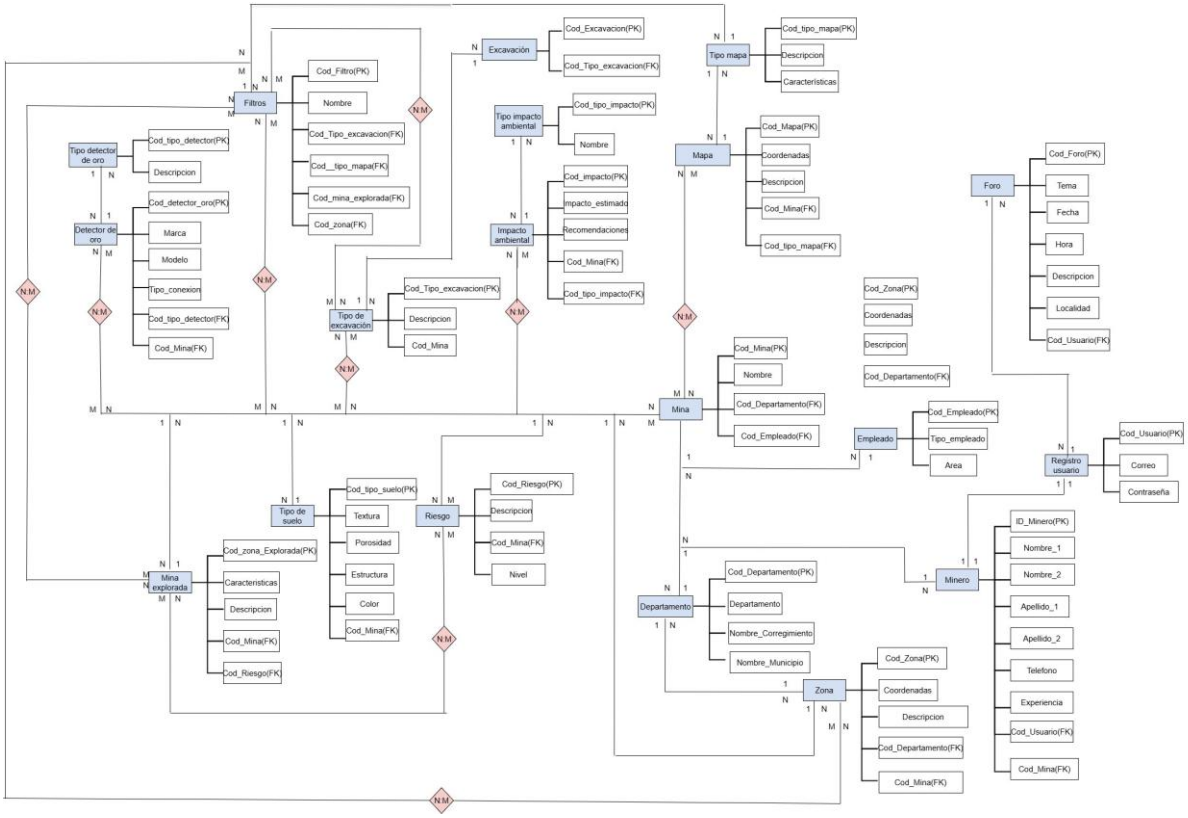


Ilustración 55. Modelo Entidad-Relación del sistema SoftGold

Fuente: Elaboración propia

El Modelo Entidad–Relación (MER) presentado en la Figura 51 describe de forma completa la estructura de la base de datos del sistema SoftGold, incluyendo entidades, atributos y cardinalidades.

Este modelo sirve como base para la implementación de las tablas maestras y transaccionales del sistema, garantizando la integridad y coherencia de la información almacenada.

Tablas Maestras

Nombre Tabla	Zonas				
Nombre Columna	**Cod_zona**	**Descripción**	**Coordenadas**	**Cod_departamento_departamento**	**Cod_mina**
TipoClave	PK			FK	FK
Nulos/Únicos	NN	NN	NN	NN	NN
TablaClaveForanea				Departamento	Minas
ColumnaFK				Cod_departamento	Cod_mina
TipoDatos	VARCHAR	TEXT	VARCHAR	VARCHAR	VARCHAR
Estado	**ACTIVO**				
Longitud	5	100	100	5	5
Ejemplo	Z345	Área de exploración en fase inicial. Aún no se han encontrado indicios de oro	Latitud: 6.59778 Longitud: -75.0122	D05	M099

Tabla 18. Tabla Maestra de la entidad Zonas

Fuente: Elaboración propia

Nombre tabla	Departamento			
Nombre Columna	**Cod_departamento**	**Departamento**	**Nom_corregimiento**	**Nom_municipio**
TipoClave	PK			
Nulos/Únicos	NN	NN	NN	NN
TablaClaveForanea				
ColumnaFK				
TipoDatos	VARCHAR	VARCHAR	VARCHAR	VARCHAR
Estado	**ACTIVO**			
Longitud	5	30	30	30
Ejemplo	D05	Antioquia	La floresta	Yolombó

Tabla 19. Tabla Maestra de la entidad Departamento

Fuente: Elaboración propia

Nombre tabla	Riesgo			
Nombre Columna	**Cod_riesgo**	**Descripcion**	**Cod_mina_mina**	**Nivel**
TipoClave	PK		FK	
Nulos/Únicos	NN	NN	NN	NN
TablaClaveForanea			Mina	
ColumnaFK			Cod_mina	
TipoDatos	VARCHAR	TEXT	VARCHAR	VARCHAR
Estado	**ACTIVO**			
Longitud	5	100	5	10
Ejemplo	R678	Terreno inestable y dificultad de acceso	M099	ALTO

Tabla 20. Tabla Maestra de la entidad Riesgo

Fuente: Elaboración propia

Nombre tabla	Mina			
Nombre Columna	**Cod_mina**	**Nombre**	**Cod_departamento_departamento**	**Cod_empleado_empleado**
TipoClave	PK		FK	FK
Nulos/Únicos	NN	NN	NN	NN
TablaClaveForanea			Departamento	Empleado
ColumnaFK			Cod_departamento	Cod_empleado
TipoDatos	VARCHAR	VARCHAR	VARCHAR	INT
Estado	**ACTIVO**			
Longitud	5	30	5	12
Ejemplo	M099	Mina los balsos	D05	100267543

Tabla 21. Tabla Maestra de la entidad Mina

Fuente: Elaboración propia

Nombre tabla	Empleado		
Nombre Columna	**Cod_empleado**	**Tipo_empleado**	**Area**
TipoClave	PK		
Nulos/Únicos	NN	NN	NN
TablaClaveForanea			
ColumnaFK			
TipoDatos	INT	TEXT	VARCHAR
Estado	**ACTIVO**		
Longitud	12	20	20
Ejemplo	100267543	Minero	Exploración

Tabla 22. Tabla Maestra de la entidad Empleado

Fuente: Elaboración propia

Nombre tabla	Registro Usuario		
Nombre Columna	**Cod_usuario**	**Correo**	**Contraseña**
TipoClave	PK		
Nulos/Únicos	NN	NN	NN
TablaClaveForanea			
ColumnaFK			
TipoDatos	VARCHAR	TEXT	VARCHAR
Estado	**ACTIVO**		
Longitud	5	50	100
Ejemplo	U799	Juan.perez@gmail.com	xmXnY56_23Y#

Tabla 23. Tabla Maestra de la entidad Usuario

Fuente: Elaboración propia

Tablas Transaccionales

Nombre Tabla	Impactoambiental_mina	
Nombre Columna	**Cod_mina_mina**	**Cod_impactoambiental_impactoambiental**
TipoClave	FK	FK
Nulos/Únicos	NN	NN
TablaClaveForanea	Mina	Impacto_ambiental
ColumnaFK	Cod_mina	Cod_impactoambiental
TipoDatos	VARCHAR	VARCHAR
Estado	**ACTIVO**	
Longitud	5	5
Ejemplo	M099	I1332

Tabla 24. Tabla Transaccional de las entidades Impacto ambiental - Mina

Fuente: Elaboración propia

Nombre Tabla	Riesgo_minaexplorada	
Nombre Columna	**Cod_riesgo_riesgo**	**Cod_mina_mina**
TipoClave	FK	FK
Nulos/Únicos	NN	NN
TablaClaveForanea	Riesgo	Mina
ColumnaFK	Cod_riesgo	Cod_mina
TipoDatos	VARCHAR	VARCHAR
Estado	**ACTIVO**	
Longitud	5	5
Ejemplo	R678	M099

Tabla 25. Tabla Transaccional de las entidades Riesgo – Mina Explorada

Fuente: Elaboración propia

Nombre Tabla	Mina_mapa	
Nombre Columna	**Cod_mina_mina**	**Cod_mapa_mapa**
TipoClave	FK	FK
Nulos/Únicos	NN	NN
TablaClaveForanea	Mina	Mapa
ColumnaFK	Cod_mina	Cod_mapa
TipoDatos	VARCHAR	VARCHAR
Estado	**ACTIVO**	
Longitud	5	5
Ejemplo	MN567	M099

Tabla 26. Tabla Transaccional de las entidades Mina – Mapa

Fuente: Elaboración propia

Nombre Tabla	Mina_riesgo	
Nombre Columna	**Cod_mina_mina**	**Cod_riesgo_riesgo**
TipoClave	FK	FK
Nulos/Únicos	NN	NN
TablaClaveForanea	Mina	Riesgo
ColumnaFK	Cod_mina	Cod_riesgo
TipoDatos	VARCHAR	VARCHAR
Estado	**ACTIVO**	
Longitud	5	5
Ejemplo	MN567	R678

Tabla 27. Tabla Transaccional de las entidades Mina – Riesgo

Fuente: Elaboración propia

Prototipos de Alta Fidelidad

El diseño de interfaces del sistema SoftGold se presenta mediante prototipos de alta fidelidad que permiten visualizar la apariencia final del sistema, la disposición de los elementos y la interacción del usuario con las diferentes funcionalidades. A continuación, se describen las principales pantallas del sistema, de acuerdo con los módulos definidos en los requisitos.

Prototipo 1. Pantalla de Inicio de Sesión

Esta pantalla permite a los usuarios autenticarse en el sistema mediante el ingreso de su correo electrónico y contraseña. Incluye opciones para recuperar la contraseña y redirigirse al registro de usuario en caso de no contar con una cuenta. La interfaz es clara y minimalista, facilitando el acceso al sistema y garantizando una experiencia de usuario intuitiva.

Elementos principales:

- Campo de correo electrónico
- Campo de contraseña con opción de visualización
- Botón “Iniciar sesión”
- Enlace de registro de usuario

Fuente: Elaboración propia

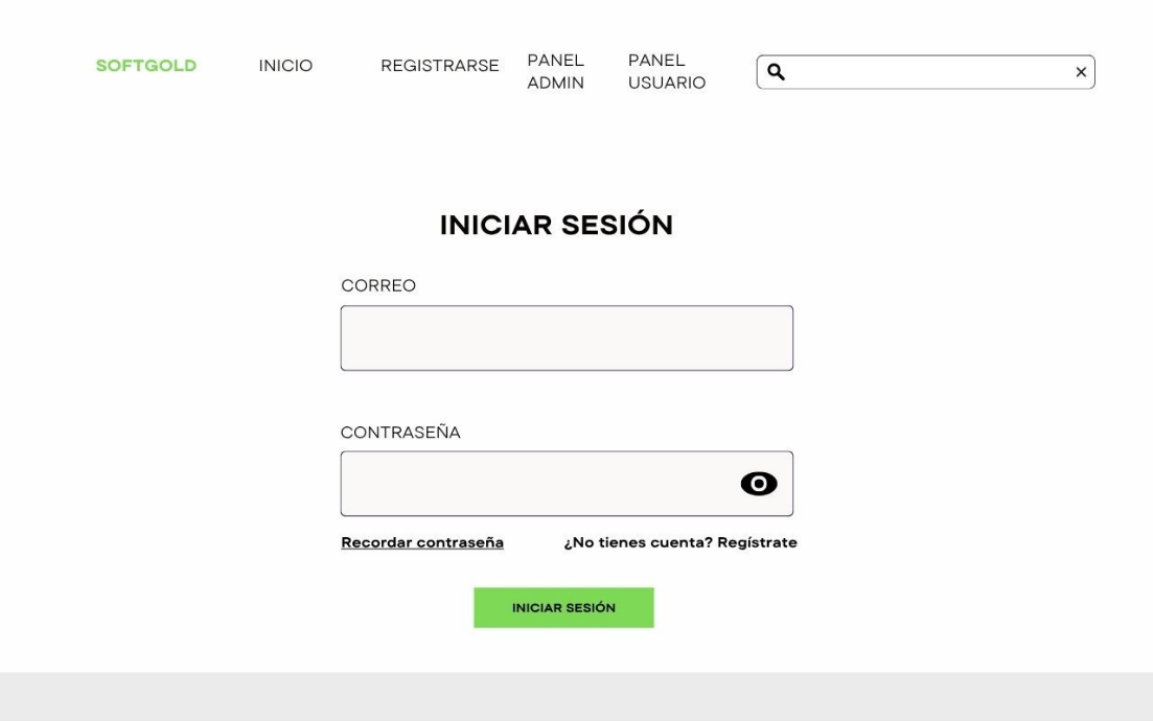


Ilustración 56. Prototipo – Pantalla de Inicio de Sesión

Fuente: Elaboración propia

Prototipo 2. Panel de Administración

El panel de administración permite al usuario con rol administrador acceder a las principales funcionalidades de gestión del sistema. Desde esta pantalla se pueden administrar mineros, empleados, minas y mapas, centralizando las operaciones administrativas en un solo lugar.

Elementos principales:

- Menú superior de navegación
- Información del usuario administrador
- Accesos directos a los módulos de gestión
- Iconos representativos de cada funcionalidad

Fuente: Elaboración propia

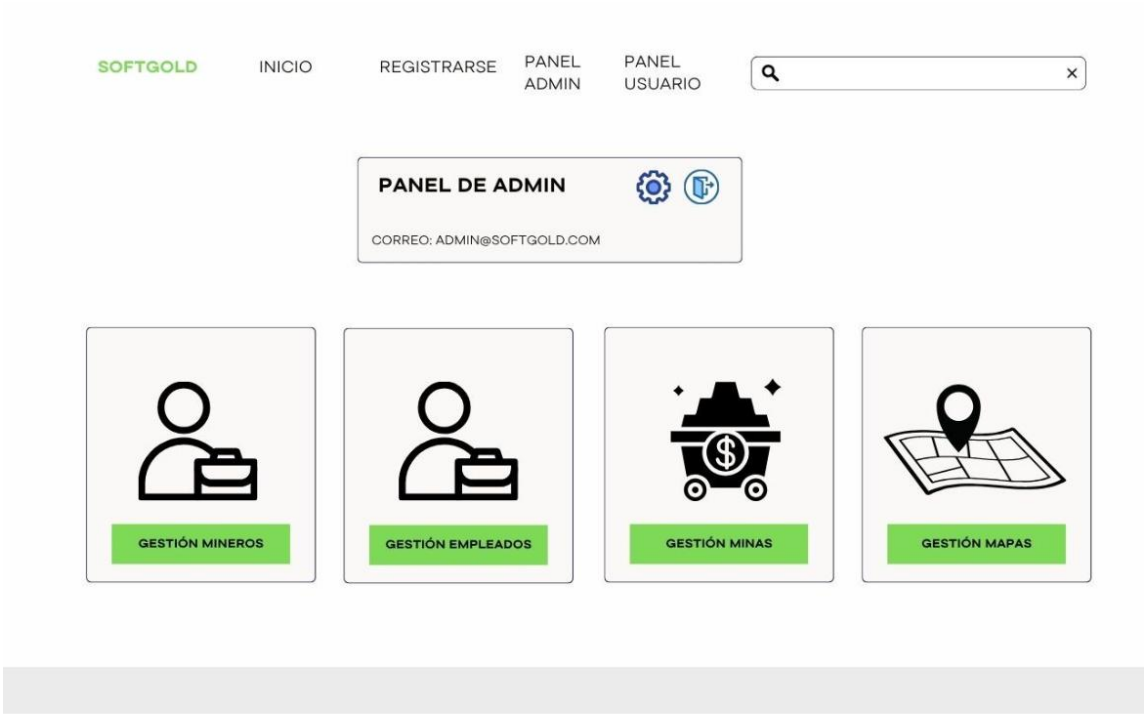


Ilustración 57. Prototipo – Panel de Administración

Fuente: Elaboración propia

Prototipo 3. Mapa Interactivo

La pantalla del mapa interactivo permite visualizar gráficamente las zonas mineras registradas en el sistema. Incluye un panel de filtros que facilita la exploración de la información según criterios como tipo de excavación, tipo de suelo y zonas exploradas, apoyando la toma de decisiones antes de realizar excavaciones.

Elementos principales:

- Mapa interactivo con marcadores de zonas
- Panel lateral de filtros
- Botón de búsqueda
- Visualización dinámica de resultados

Fuente: Elaboración propia

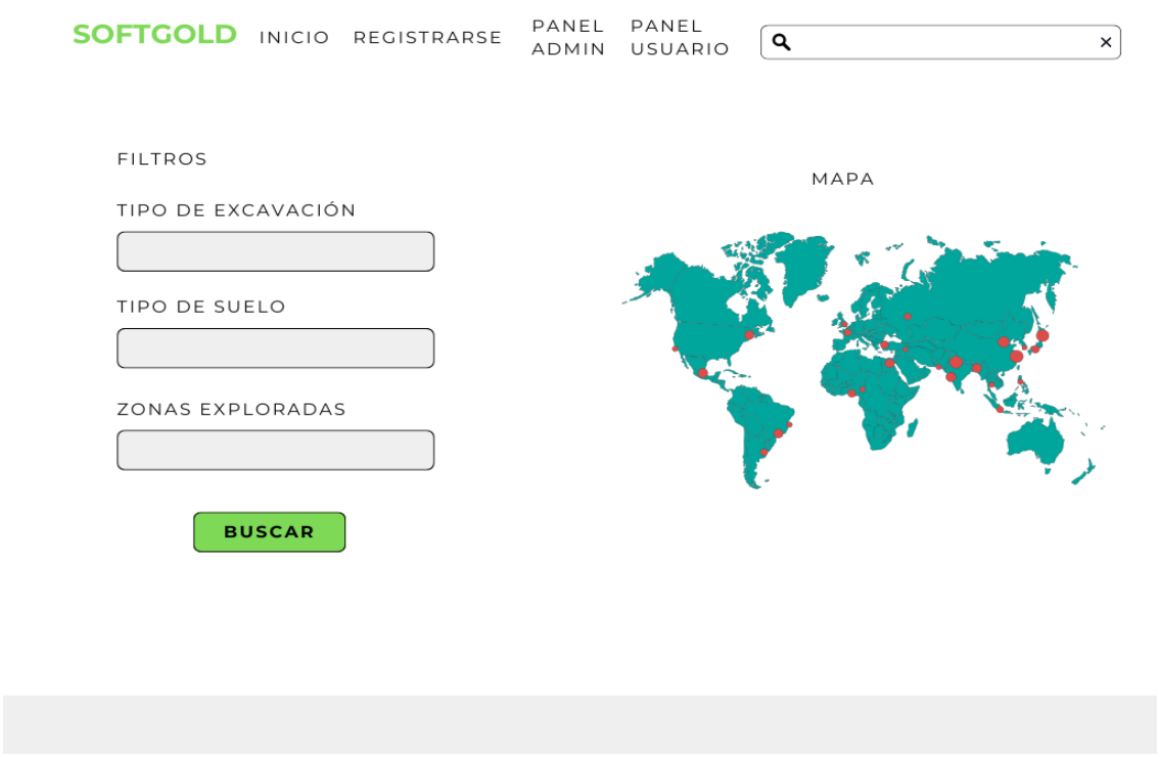


Ilustración 58. Prototipo – Mapa Interactivo

Fuente: Elaboración propia

Prototipo 4. Foro

El módulo de foro permite a los usuarios compartir experiencias, comentarios y recomendaciones relacionadas con las zonas mineras. Esta pantalla fomenta la colaboración comunitaria y el intercambio de información entre los usuarios del sistema.

Elementos principales:

- Listado de comentarios
- Información del usuario, zona y fecha
- Botón para crear nuevos comentarios
- Área de visualización de publicaciones

Fuente: Elaboración propia



Ilustración 59. Prototipo – Módulo Foro

Fuente: Elaboración propia

Prototipo 5. Informes

La pantalla de informes permite generar reportes del sistema, tales como impacto ambiental, zonas priorizadas según información registrada y zonas exploradas durante el último mes. El usuario selecciona el tipo de informe y el sistema muestra el resultado correspondiente para sus análisis.

Elementos principales:

- Selector de tipo de informe

- Botón “Generar informe”
- Área de visualización del resultado

Fuente: Elaboración propia

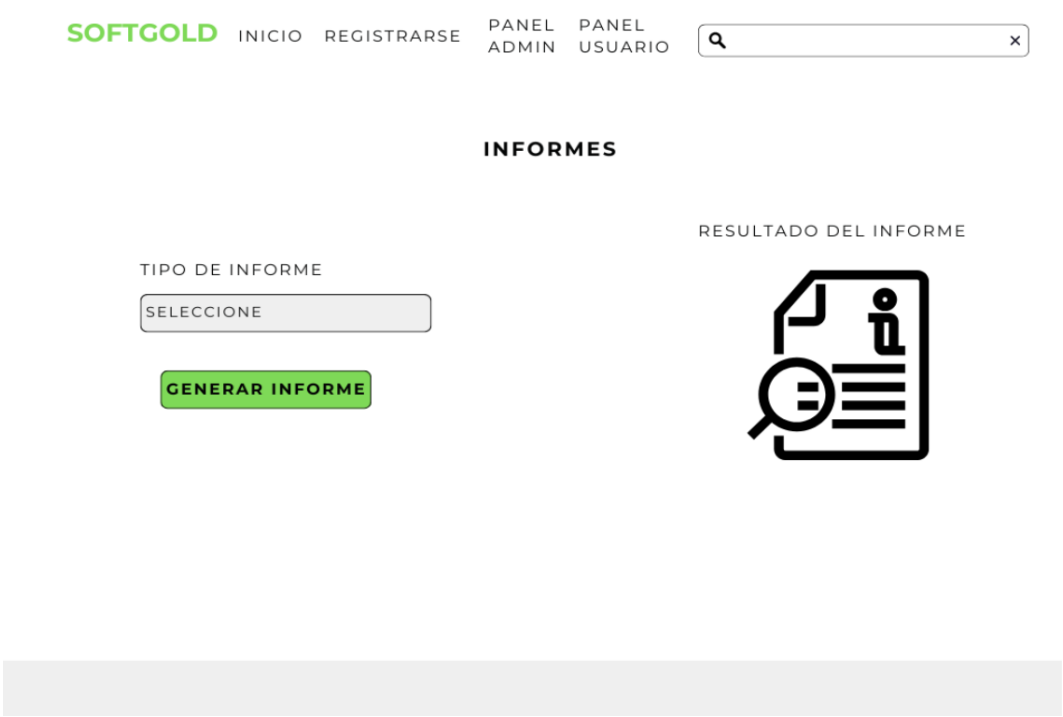


Ilustración 60. Prototipo – Módulo de Informes y Consultas

Fuente: Elaboración propia

Prototipo 6. Consultas

Este módulo de consultas permite acceder rápidamente a información específica del tema, como comentarios y recomendaciones, zonas con actividad reciente y zonas menos exploradas. Esta pantalla facilita el acceso ágil a datos relevantes para el usuario.

Elementos principales:

- Selector de tipo de consulta

- Botón “Generar consulta”
- Área de resultados

Fuente: Elaboración propia



Ilustración 61. Prototipo – Módulo Consultas

Fuente: Elaboración propia

Prototipo 7. Gestión de Detectores de Oro

Esta pantalla permite al administrador registrar y gestionar los detectores de oro asociados a las zonas mineras. Desde aquí se pueden ingresar datos como código, marca, modelo, zona y estado del detector.

Elementos principales:

- Formulario de registro de detectores

- Campos de código, marca, modelo, zona y estado
- Botón “registrar”

Fuente: Elaboración propia

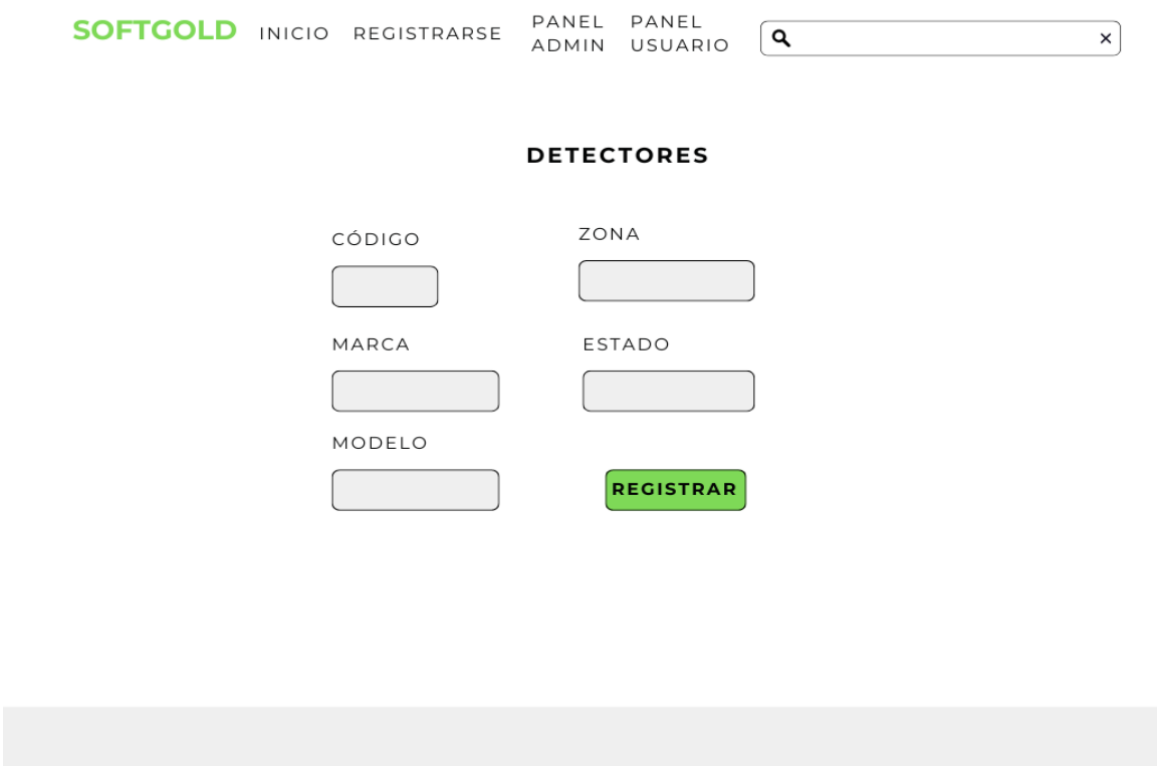


Ilustración 62. Prototipo – Gestión de Detectores de Oro

Fuente: Elaboración propia

Los prototipos de alta fidelidad presentados permiten visualizar de forma clara la estructura, navegación y funcionalidad de las principales interfaces del sistema SoftGold, sirviendo como guía para el desarrollo del software y validación de los requisitos del sistema.

Objetivo 4: Codificación

Métricas de Software – Requerimientos

Nota. Métricas de requerimientos implementados en el sistema SoftGold.

ID Req.	Tipo (F/NF)	Descripción	N° Cambios	Interfaces Relacionadas	Comentarios
RQ01	Funcional	Bloqueo de cuenta después de 3 intentos fallidos	4	UsuarioServiceIface, Login	Se le notifica al usuario por correo que su cuenta fue bloqueada
RQ02	Funcional	Redirigir a la página de bloqueo si se bloquea la cuenta	4	Login, cuentaBloqueo.html	Se le muestra al usuario una página notificándole que su cuenta ha sido bloqueada
RQ05	No funcional	Buenas prácticas de seguridad en el login	5	Login, UsuarioServiceIface	No se revela información sensible en mensajes de error
RQ06	Funcional	Redirección según el rol	5	DataInitializer	Según el usuario que ingrese al sistema se le muestra una interfaz diferente
RQ07	No funcional	Encriptación de contraseña	3	PasswordEncoderConfig	Una vez se guarda la contraseña, esta se encripta en la BD
RQ09	Funcional	Redirección de inicio de sesión después de 15min de inactividad	4	Inactividad.js	Se redirige al usuario al Login
RQ10	Funcional	Recuperación de contraseña	6	PasswordController, passwordResetController	El usuario puede recuperar su contraseña por medio del correo
RQ12	Funcional	Solo el administrador tiene acceso al panel de administración	4	SecurityConfig	El administrador tiene que estar previamente loggeado para ingresar
RQ13	Funcional	CRUD para mineros	6	MinerosController	El administrador se encarga del CRUD de los mineros
RQ17	Funcional	Tabla relacional y maestra	5	Mina, Mapa	Se crea la tabla mina_mapa que guarda la información relacionada entre ambas tablas maestras
RQ18	Funcional	Tabla relacional y maestra	3	Mina, Riesgo	Se crea la tabla mina_riesgos que guarda la información de los

					riesgos asociados a cada mina
RQ19	Funcional	CRUD de zonas mineras	5	ZonaController, ZonaService, ZonaDAO	Permite al administrador gestionar la información de las zonas
RQ20	Funcional	Visualización de mapas interactivos	5	MapaController, MapaService	Permite visualizar zonas mineras mediante mapas interactivos
RQ21	Funcional	Aplicación de filtros en el mapa	4	FiltroService, MapaController	Permite filtrar zonas por tipo de excavación, mapa y zonas exploradas
RQ22	Funcional	Gestión de comentarios en el foro	4	ForoController, ForoService	Permite publicar, consultar y moderar comentarios en el foro
RQ23	Funcional	Gestión de detectores de oro	4	DetectorController, DetectorService	Permite registrar y administrar detectores asociados a zonas
RQ24	Funcional	Generación de informes del sistema	4	InformeService, InformeController	Permite generar informes de impacto ambiental y zonas exploradas
RQ25	Funcional	Ejecución de consultas del sistema	3	ConsultaService	Permite consultar zonas con actividad reciente y zonas menos exploradas

Tabla 28. Métricas de Software – Requerimientos

Fuente: Elaboración propia

Métricas de Software – Código Fuente

Nota. Métricas de calidad del código fuente del sistema SoftGold.

Clase/Módulo	LOC	N° Métodos	Complejidad Ciclomática	N° Errores detectados	Acoplamiento	Cohesión
DataInitializer	68	1	4	0	Medio	Alta
PasswordEncoder	15	1	1	4	Bajo	Alta
AdminController	81	3	8	1	Medio	Alta
GlobalControllerAdvice	15	1	1	0	Bajo	Alta
InicioController	15	1	1	1	Bajo	Alta
LoginController	162	13	20	4	Medio	Alta
MineroController	189	7	11	0	Medio	Alta

MineroInicioController	34	1	2	0	Medio	Alta
PasswordController	94	4	3	2	Alto	Alta
PasswordResetController	68	3	3	2	Medio	Alta
RedireccionController	30	1	4	0	Bajo	Medio
ValidacionController	28	2	1	3	Bajo	Alta
PasswordResetToken	65	10	1	3	Bajo	Alta
Usuario	300	47	1	0	Medio	Alta
PasswordResetTokenDAO	19	1	1	4	Medio	Alta
UsuarioDAO	30	1	1	1	Medio	Medio
CustomLoginPassword	61	7	3	3	Bajo	Alta
CustomUserDetailsService	57	2	1	1	Medio	Alta
SecurityConfig	59	2	1	2	Bajo	Medio
EmailService	25	2	1	0	Bajo	Alta
PasswordResetService	107	10	2	4	Alto	Alta
UsuarioService	210	12	16	3	Medio	Alta
Riesgo	51	10	1	0	Medio	Alta
MapaController	210	10	22	0	Medio	Alta
MapaService	230	12	19	0	Alto	Medio
MapaDAO	35	5	3	0	Bajo	Alta
Mapa	95	20	2	2	Bajo	Alta
ForoController	175	9	17	0	Medio	Alto
ForoService	205	11	15	0	Medio	Alta
ForoDAO	30	4	2	3	Bajo	Alta
DetectorController	160	8	16	0	Medio	Alta
DetectorService	185	10	14	0	Medio	Alta
InformeService	240	9	21	1	Alto	Medio
InformeController	120	5	10	0	Medio	Alta
ZonaController	170	8	17	1	Medio	Alta
ZonaService	205	11	15	0	Medio	Alta
ZonaDAO	32	4	2	0	Bajo	Alta
Zona	92	18	2	0	Bajo	Alta

Tabla 29. Métricas de Software – Código Fuente

Fuente: Elaboración propia

Métricas de Proyecto

Nota. Métricas de avance del proyecto por módulo.

Actividad/Módulo	Tiempo estimado (h)	Tiempo real (h)	% Cumplido	% Req. implementados	N° pruebas	% Pruebas exitosas	Comentarios
Bloqueo de cuenta	3	8	100%	100%	10	80%	Retrabajo por número de bloqueos
Recuperación de contraseña	8	5	100%	100%	10	90%	Retrabajo por validaciones con correo
Codificación de contraseña	5	5	100%	100%	20	100%	Sin comentarios
Redirecciones	8	12	100%	100%	30	80%	Retrabajo por mapeos y validación de rol
Login y autenticación	4	5	100%	100%	10	70%	Inconvenientes con la base de datos
CRUD minero	6	10	100%	100%	15	80%	Retrabajo por cardinalidades entre tablas
Gestión de perfiles	6	9	100%	100%	15	85%	Ajustes por roles y permisos
Gestión de zonas mineras	10	14	100%	100%	20	85%	Ajustes por validación geográfica
Gestión impacto ambiental	8	11	100%	100%	15	80%	Validaciones adicionales requeridas
Consulta de riesgos	5	6	100%	100%	10	90%	Sin observaciones
Foro y comentarios	5	7	100%	100%	12	85%	Ajustes por control de autenticación
Mapa interactivo	12	16	100%	100%	25	80%	Optimización de filtros y rendimiento
Filtros del mapa	6	8	100%	100%	12	85%	Ajustes en lógica de consulta
Generación de informes	4	5	100%	100%	15	90%	Validación de datos históricos

Tabla 30. Métricas de Proyecto

Fuente: Elaboración propia

Fase de construcción

A continuación, se presentan fragmentos de código representativos que evidencian la implementación de algunos requisitos funcionales del sistema SoftGold, garantizando la trazabilidad entre los requisitos definidos.

Fragmento de código – Registro de Usuario

El proceso de registro de usuario en el sistema SoftGold permite crear nuevas cuentas garantizando la validez del correo electrónico y la seguridad de las credenciales.

La funcionalidad se implementa bajo una arquitectura por capas, donde la interfaz web captura los datos, el controlador gestiona la solicitud, el servicio valida la información, encripta la contraseña y finalmente almacena los datos en la base de datos.

Este proceso asegura la integridad de la información y el cumplimiento de buenas prácticas de seguridad.

Nota. Este fragmento corresponde al controlador de usuarios, el cual recibe los datos ingresados desde la interfaz web y delega el proceso de registro al servicio de usuarios. Una vez finalizado el proceso, el sistema redirige al usuario a la pantalla de inicio de sesión.

```
@GetMapping("/login")
public String mostrarFormularioLogin(
        @RequestParam(value = "error", required = false) String error,
        @RequestParam(value = "logout", required = false) String logout,
        @RequestParam(value = "registroExitoso", required = false) String registroExitoso,
        Model model) {
    if (error != null) {
        model.addAttribute("error", "Credenciales incorrectas. Inténtalo de nuevo.");
    }
    if (logout != null) {
        model.addAttribute("mensaje", "Has cerrado sesión correctamente.");
    }
    if (registroExitoso != null) {
        model.addAttribute("mensaje", "¡Registro exitoso! Ahora puedes iniciar sesión.");
    }
    return "vistas/login";
}
```

Ilustración 63. Fragmento de código– Registro de usuario

Fuente: Elaboración propia

Nota. Este fragmento corresponde a la validación de correo

```
@PostMapping("/registro")
public String procesarRegistro(@Valid @ModelAttribute("usuario") Usuario usuario,
        BindingResult result,
        Model model) {

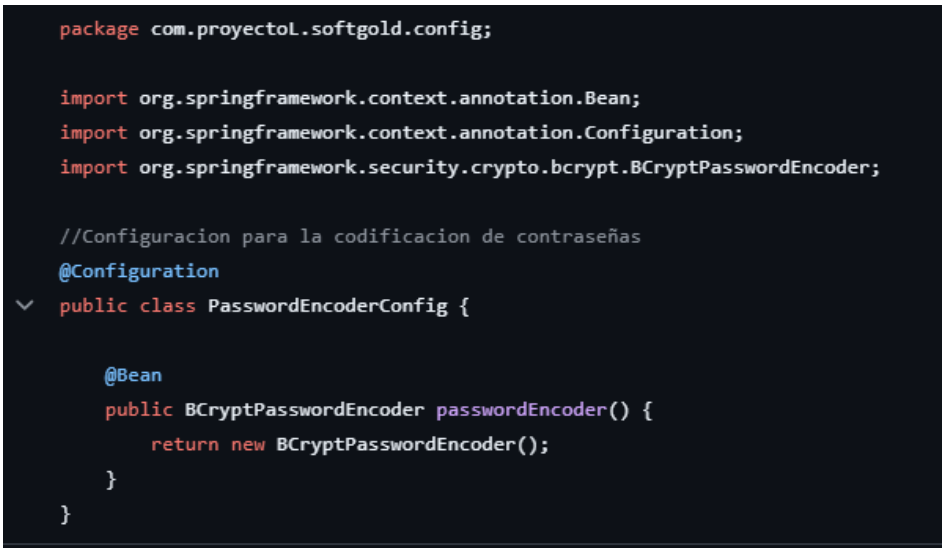
    if (result.hasErrors()) {
        model.addAttribute("minas", minaDAO.findAll());
        return "vistas/registro";
    }

    try {
        Optional<Usuario> existente = usuarioDAO.findByEmail(usuario.getEmail());
        if (existente.isPresent()) {
            model.addAttribute("error", "Ya existe una cuenta con ese correo.");
            return "vistas/registro";
        }
```

Ilustración 64. Fragmento de código– Registro de usuario

Fuente: Elaboración propia.

Nota. Este fragmento corresponde a la encriptación de la contraseña la cual se encripta en la base de datos por medio de BCryptPasswordEncoder una vez creada la contraseña.



```java
package com.proyectoL.softgold.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

//Configuracion para la codificacion de contraseñas
@Configuration
public class PasswordEncoderConfig {

    @Bean
    public BCryptPasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
}
```

Ilustración 65. Fragmento de código– Registro de usuario

Fuente: Elaboración propia

Nota. Este fragmento corresponde a la validación de los campos en el formulario de registro de usuario

```java
@Table(name = "usuarios")
public class Usuario implements UserDetails {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "cedula", nullable = false, unique = true)
    @NotBlank(message = "La cédula es obligatoria")
    private String cedula;

    @Column(name = "tipo_documento", nullable = false)
    @NotBlank(message = "El tipo de documento es obligatorio")
    private String tipoDocumento;

    @Column(name = "nombre_1", nullable = false)
    @NotBlank(message = "El nombre es obligatorio")
    private String nombre1;

    @Column(name = "nombre_2", nullable = false)
    private String nombre2;

    @Column(name = "apellido_1", nullable = false)
    @NotBlank(message = "El apellido es obligatorio")
    private String apellido1;

    @Column(name = "apellido_2", nullable = false)
    private String apellido2;

    @Column(name = "email", nullable = false, unique = true)
    @Email
    @NotBlank(message = "El email es obligatorio")
    private String email;
```

Ilustración 66. Fragmento de código– Registro de usuario

Fuente: Elaboración propia

```java
@Transient
@Pattern(regexp = "^(?=.*[A-Z])(?=.*\\d).{8,20}$", message = "La contraseña debe tener al menos 8 caracteres, una letra mayúscula y un número")
private String passwordPlano;

@Column(nullable = false)
private String password;

@ManyToMany(fetch = FetchType.EAGER) // relacion de muchos a muchos
@JoinTable(name = "usuario_roles", joinColumns = @JoinColumn(name = "usuario_id", referencedColumnName = "id"), inverseJoinColumns = @JoinColumn(name = "rol_id", referencedColumnName = "id"))
private Collection<Rol> roles;

@Column(name = "intentos_fallidos", nullable = false)
private int intentosFallidos;

@Column(name = "bloqueado", nullable = false)
private boolean bloqueado;

@Column(name = "tiempo_bloqueo", nullable = true)
private LocalDateTime tiempoBloqueo;

@Column(name = "tipo_usuario", nullable = false)
@NotBlank(message = "El tipo de usuario es obligatorio")
private String tipoUsuario;

@Column(name = "telefono")
private String telefono;

@ManyToOne
@JoinColumn(name = "cod_mina")
private Mina mina;

@Column(name = "tipo_empleado")
private String tipoEmpleado;
```

Ilustración 67. Fragmento de código– Registro de usuario

Fuente: Elaboración propia

Fragmento de código – Zona exploración

El proceso de registro de zonas mineras en el sistema SoftGold permite al administrador almacenar información geográfica relevante, garantizando la validez de los datos y evitando la duplicidad de registros.

La funcionalidad se implementa bajo una arquitectura por capas, donde la interfaz web captura la información, el controlador gestiona la solicitud, el servicio aplica las reglas de negocio y la base de datos persiste la información.

Este proceso contribuye a la correcta gestión y organización de las zonas mineras dentro del sistema.

Nota. Este fragmento corresponde al controlador de zonas, el cual recibe los datos ingresados desde la interfaz web y delega el proceso de registro al servicio de zonas. Una vez completado el proceso, el sistema redirige a la vista de gestión de zonas, indicando el registro exitoso.

```java
@Entity
@Table(name = "zona_exploracion")
public class ZonaExploracion {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotBlank(message = "El nombre de la zona es obligatorio")
    @Column(nullable = false)
    private String nombre;

    @Column(length = 1000)
    private String descripcion;

    @Column(nullable = true)
    private Double latitud;

    @Column(nullable = true)
    private Double longitud;

    @Column(nullable = true)
    private String tipo;

    @Column(nullable = true)
    private String estado;

    @ManyToOne
    @JoinColumn(name = "cod_mina", nullable = true)
    private Mina mina;

    @Column(name = "fecha_registro")
    private LocalDate fechaRegistro;
```

Ilustración 68. Fragmento de código– Zona exploración

Fuente: Elaboración propia

```java
package com.proyectoL.softgold.repository;

import com.proyectoL.softgold.model.ZonaExploracion;
import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;

public interface ZonaExploracionDAO extends JpaRepository<ZonaExploracion, Long> {
    List<ZonaExploracion> findByEstado(String estado);
    List<ZonaExploracion> findByTipo(String tipo);
    List<ZonaExploracion> findByNombreContainingIgnoreCase(String nombre);
    List<ZonaExploracion> findByMina_CodMina(Long codMina);
}
```

Ilustración 69. Fragmento de código– Zona exploración

Fuente: Elaboración propia

Fragmento de código – Encriptación de contraseña

La encriptación de contraseñas en el sistema SoftGold se implementa con el objetivo de garantizar la seguridad de la información sensible de los usuarios.

Para ello, se utiliza el algoritmo BCrypt, el cual permite codificar las contraseñas antes de ser almacenadas en la base de datos, evitando el uso de texto plano y reduciendo riesgos de exposición ante accesos no autorizados.

Este fragmento define una clase de configuración encargada de declarar un bean de tipo BCryptPasswordEncoder dentro del contexto de Spring.

Mediante la anotación @Configuration se indica que la clase contiene definiciones de configuración, mientras que @Bean permite que el codificador de contraseñas sea gestionado por el contenedor de Spring e inyectado en los servicios que lo requieran.

El uso del algoritmo BCrypt proporciona un mecanismo de encriptación robusto, ampliamente utilizado en aplicaciones empresariales por su resistencia a ataques de fuerza bruta.

```
1   package com.proyectoL.softgold.config;
2
3   import org.springframework.context.annotation.Bean;
4   import org.springframework.context.annotation.Configuration;
5   import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
6
7   //Configuracion para la codificacion de contraseñas
8   @Configuration
9   public class PasswordEncoderConfig {
0
1       @Bean
2       public BCryptPasswordEncoder passwordEncoder() {
3           return new BCryptPasswordEncoder();
4       }
5   }
```

Ilustración 70. Fragmento de código– Encriptación contraseña

Fuente: Elaboración propia

El codificador definido en esta configuración es utilizado durante el proceso de registro de usuarios, donde la contraseña ingresada es encriptada antes de ser almacenada en la base de datos, garantizando que en ningún momento se persista información sensible en texto plano.

Los fragmentos de código presentados evidencian la correcta implementación de los principales procesos del sistema SoftGold, incluyendo el registro de usuarios, la gestión de zonas mineras y la encriptación de contraseñas.

Cada componente fue desarrollado bajo una arquitectura por capas, aplicando validaciones de negocio y mecanismos de seguridad que garantizan la integridad, confidencialidad y consistencia de la información.

Estas implementaciones demuestran la trazabilidad entre los requisitos funcionales definidos, los diagramas de diseño y la fase de construcción del software.

Repositorio en Github

El código fuente del sistema SoftGold se encuentra disponible públicamente en el siguiente repositorio de GitHub:

https://github.com/manuelaosorio19/Proyecto-Softgold.git

Fuente: Elaboración propia

Objetivo 5: Pruebas

El Plan de Pruebas del sistema SoftGold define la estrategia, el alcance y los criterios utilizados para verificar que el software cumple con los requisitos funcionales y no funcionales establecidos. Las pruebas se realizaron de forma incremental durante cada Sprint, siguiendo la metodología Scrum.

Objetivo del plan de pruebas

Verificar que las funcionalidades del sistema SoftGold operen correctamente de acuerdo con los requisitos definidos, garantizando estabilidad, seguridad, usabilidad y correcto manejo de la información.

Alcance y entorno de pruebas

Elemento	Detalle
Framework de pruebas	JUnit 5 + MockMvc (Spring Boot Test)
Versión Spring Boot	3.4.4
Cobertura de módulos	Autenticación, Minas, Riesgos, Foro, Soporte
Tipos de prueba	Capa web (WebMvcTest), integración de contexto (SpringBootTest), validación Bean Validation
Herramienta de mocks	Mockito 5
Base de datos (test)	H2 in-memory (modo MySQL) — usado en RT-01 a RT-05 (carga de contexto)
Fecha de ejecución	15 de mayo de 2026
Ejecución	Maven: mvn test → BUILD SUCCESS

Tabla 31. Alcance y entorno de pruebas

Fuente: Elaboración propia

Tipos de pruebas aplicadas

Durante el desarrollo del sistema se aplicaron los siguientes tipos de pruebas:

- **Pruebas funcionales:** Validan que cada funcionalidad cumpla con los requisitos establecidos
- **Pruebas de integración:** Verifican la correcta interacción entre módulos (ej. Mina-zona, mapa-riesgo).

- **Pruebas de seguridad:** Validan autenticación, control de acceso y encriptación de contraseñas
- **Pruebas de regresión:** Verifican que los cambios realizados no afecten funcionalidades existentes.

Estrategia de pruebas

Las pruebas se realizaron de forma incremental durante cada Sprint, siguiendo la metodología Scrum. Cada funcionalidad fue validada una vez implementada, utilizando escenarios reales de uso y datos de prueba controlados.

Criterios de aceptación

Una prueba se considera exitosa cuando:

- La funcionalidad cumple con la historia de usuario asociada
- No se presentan errores críticos durante la ejecución
- El resultado obtenido coincide con el resultado esperado
- El sistema responde dentro de los tiempos definidos en los requisitos no funcionales

Roles y responsables

- Equipo de desarrollo: Ejecución de pruebas funcionales y de integración
- Scrum Máster: Seguimiento del proceso de pruebas
- Usuarios simulados: Validación básica de usabilidad

Resultados de las pruebas

Los resultados de las pruebas realizadas se documentan mediante métrica del proyecto, donde se registra el número de pruebas ejecutadas por módulo y el porcentaje de éxito obtenido. Estas métricas evidencian el avance y la calidad del sistema durante el desarrollo.

Nota. Resumen de pruebas ejecutadas por módulo y resultados obtenidos.

Módulo	Tipo de prueba	N° pruebas	Resultado
Login y autenticación	Funcional / Seguridad	8	Exitosas
Usuarios y perfiles	Funcional	6	Exitosas
Mineros	Funcional	5	Exitosas
Zonas	Funcional	4	Exitosas
Mapas y filtros	Funcional / Usabilidad	4	Exitosas
Foro	Funcional	3	Exitosas
Detectores	Funcional	2	Exitosas
Informes y consultas	Funcional	3	Exitosas

Tabla 32. Resultados de pruebas ejecutadas

Fuente: Elaboración propia

Casos de Pruebas Funcionales (CP-01 a CP-12)

Los casos de pruebas funcionales permiten validar el correcto funcionamiento de los módulos principales del sistema SoftGold. Cada caso de prueba fue diseñado con base en los requisitos funcionales y las historias de usuario, permitiendo verificar que el sistema cumple con el comportamiento esperado.

Estos casos prueban que cada endpoint del sistema responde correctamente ante entradas válidas e inválidas. Se organizan por módulo para facilitar la trazabilidad con los requisitos. Cada grupo incluye la justificación de por qué se diseñaron así, el código fuente del test, evidencia visual del sistema en ejecución y una conclusión con soporte bibliográfico.

Estructura de cada prueba unitaria con MockMvc: los métodos de prueba están implementados en Java con JUnit 5 y Spring Boot Test. La anotación @Test marca el método como caso ejecutable; @DisplayName proporciona una descripción legible para los reportes. MockMvc.perform() construye y ejecuta la petición HTTP directamente sobre el controlador sin levantar un servidor Tomcat real — esto hace las pruebas rápidas y deterministas. andExpect() encadena aserciones sobre el código HTTP de respuesta, el nombre de la vista Thymeleaf y los

atributos del modelo Spring MVC. Cuando el caso involucra persistencia, when() configura el comportamiento del mock (Mockito) antes de la petición, y verify() confirma que el repositorio fue invocado el número exacto de veces esperado — sin that se habría persistido datos sin que el test lo detectara.

Módulo de Autenticación — CP-01 y CP-02

Qué se prueba: Accesibilidad de la página de login y manejo correcto de credenciales incorrectas, verificando que el sistema no exponga información sensible al usuario.

Por qué de esta forma: Se utiliza @WebMvcTest(LoginController.class) con filtros de seguridad desactivados para aislar el comportamiento del controlador sin depender del contexto completo de Spring Security, permitiendo verificar el flujo HTTP de forma rápida y determinista.

ID	Descripción	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
CP-01	Página de login accesible sin autenticación	GET /login	HTTP 200, vista vistas/login	HTTP 200, vista correcta	**Pasa**
CP-02	Login con error muestra atributo de error en el modelo	GET /login?error=true	HTTP 200, modelo contiene atributo "error", vista vistas/login	Atributo "error" presente y vista correcta	**Pasa**

Tabla 33. Casos de prueba - Módulo de Autenticación

Fuente: Elaboración propia

CP-01 — Código del test: Verifica que GET /login devuelve HTTP 200 y la vista "vistas/login" sin ningún usuario autenticado. No se necesita configurar mocks porque no hay acceso a base de datos en este flujo.

```
@Test
@DisplayName("CP-01: Página de login accesible sin autenticación")
void cp01_loginPaginaAccesible() throws Exception {
    mockMvc.perform(get("/login"))
            .andExpect(status().isOk())
            .andExpect(view().name("vistas/login"));
}
```

Fuente: Elaboración propia

CP-02 — Código del test: Verifica que GET /login?error=true devuelve HTTP 200, que el modelo Spring MVC contiene el atributo "error" (usado por Thymeleaf para renderizar el mensaje de credenciales incorrectas) y que la vista retornada sigue siendo "vistas/login" sin redirigir a otro recurso.

```
@Test
@DisplayName("CP-02: Login con error muestra mensaje de credenciales incorrectas")
void cp02_loginConErrorMuestraMensaje() throws Exception {
    mockMvc.perform(get("/login").param("error", "true"))
            .andExpect(status().isOk())
            .andExpect(model().attributeExists("error"))
            .andExpect(view().name("vistas/login"));
}
```

Fuente: Elaboración propia

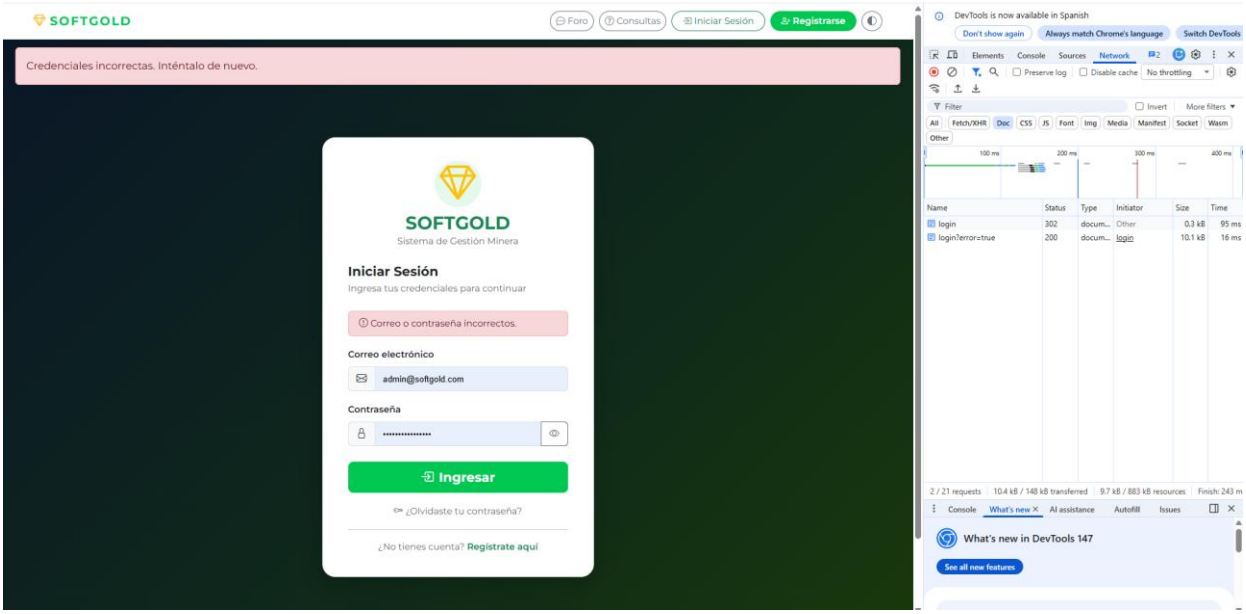


Ilustración 71. CP-02: Pantalla de login mostrando mensaje de error

Fuente: Elaboración propia

Conclusión: El controlador de autenticación responde correctamente. La página de login es accesible públicamente (CP-01) y el manejo de credenciales erróneas muestra un mensaje genérico sin exponer información de diagnóstico (CP-02), siguiendo el principio de mensajes de error seguros descrito por Myers et al. (2011) para sistemas con control de acceso.

Módulo de Gestión de Minas — CP-03 a CP-07

Qué se prueba: Las cuatro operaciones CRUD sobre la entidad Mina (crear, editar, eliminar, buscar), incluyendo el camino feliz y un caso negativo de validación (nombre vacío).

Por qué de esta forma: Se utiliza @WebMvcTest(MinaController.class) con mocks de MinaDAO, MapaDAO y RiesgoDAO. Se verifica tanto la redirección HTTP como la invocación del repositorio con verify() de Mockito, confirmando que el patrón PRG (Post-Redirect-Get) se aplica consistentemente.

ID	Descripción	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
CP-03	Crear mina con datos válidos redirige al listado	POST /admin/minas/crear — nombre: "Mina El Dorado", departamento: "Antioquia"	HTTP 302 → /admin/minas; minaDAO.save() invocado 1 vez	Redirección correcta, save verificado	**Pasa**
CP-04	Crear mina con nombre vacío regresa al formulario con error	POST /admin/minas/crear — nombre: "", departamento: "Antioquia"	HTTP 200, vista vistas/crearMina; minaDAO.save() nunca invocado	Formulario retornado, save no invocado	**Pasa**
CP-05	Editar mina existente actualiza datos y redirige	POST /admin/minas/editar/1 — nombre: "Mina Actualizada", departamento: "Córdoba"	HTTP 302 → /admin/minas; minaDAO.save() invocado	Redirección y persistencia correctas	**Pasa**
CP-06	Eliminar mina existente redirige al listado	GET /admin/minas/eliminar/1	HTTP 302 → /admin/minas; minaDAO.deleteById(1) invocado 1 vez	Eliminación y redirección correctas	**Pasa**
CP-07	Buscar minas por departamento retorna lista filtrada	GET /admin/minas/buscar?departamento=Antioquia	HTTP 200, vista vistas/listarMinas, modelo contiene "minas"	Vista y modelo correctos	**Pasa**

Tabla 34. CP - Módulo gestión de minas

Fuente: Elaboración propia

CP-03 — Código del test: Configura mocks vacíos para mapaDAO y riesgoDAO (necesarios para poblar el formulario). Realiza POST con nombre y departamento válidos. Verifica HTTP 302 hacia /admin/minas y que minaDAO.save() fue invocado exactamente 1 vez.

```
@Test
@DisplayName("CP-03: Crear mina con datos válidos redirige a listado")
void cp03_crearMinaValida() throws Exception {
    when(mapaDAO.findAll()).thenReturn(List.of());
    when(riesgoDAO.findAll()).thenReturn(List.of());
    mockMvc.perform(post("/admin/minas/crear")
                    .param("nombre", "Mina El Dorado")
                    .param("departamento", "Antioquia"))
            .andExpect(status().is3xxRedirection())
            .andExpect(redirectedUrl("/admin/minas"));
    verify(minaDAO, times(1)).save(any(Mina.class));
}
```

Fuente: Elaboración propia

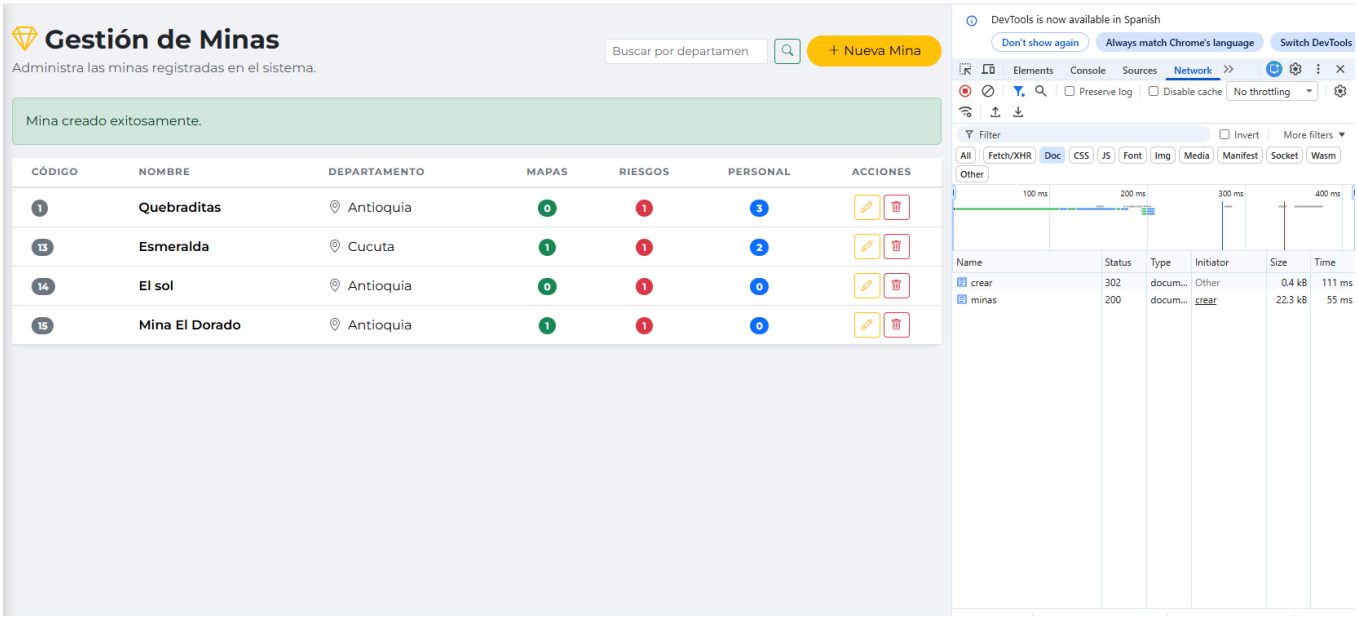


Ilustración 72. CP-03: Listado de minas tras la creación exitosa de "Mina el Dorado"

Fuente: Elaboración propia

CP-04 — Código del test: Realiza POST con nombre vacío (""). Verifica que el controlador retorna HTTP 200 (sin redirigir) con la vista del formulario, y que minaDAO.save() nunca fue invocado — la validación cortó el flujo antes de persistir.

```
@Test
@DisplayName("CP-04: Crear mina con nombre vacío regresa al formulario con error")
void cp04_crearMinaNombreVacio() throws Exception {
    when(mapaDAO.findAll()).thenReturn(List.of());
    when(riesgoDAO.findAll()).thenReturn(List.of());
    mockMvc.perform(post("/admin/minas/crear")
                    .param("nombre", "")
                    .param("departamento", "Antioquia"))
            .andExpect(status().isOk())
            .andExpect(view().name("vistas/crearMina"));
    verify(minaDAO, never()).save(any());
```

Fuente: Elaboración propia

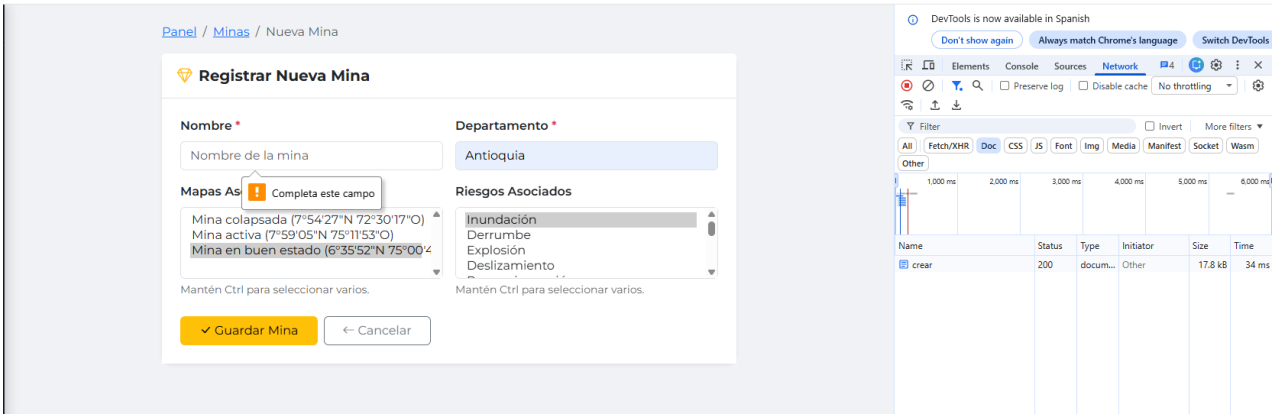


Ilustración 73. CP-04: Formulario de nueva mina mostrando la alerta de validación

Fuente: Elaboración propia

CP-05 — Código del test: Crea un objeto Mina con ID=1 en el mock, realiza POST con datos actualizados y verifica HTTP 302. atLeastOnce() en lugar de times(1) porque el controlador puede invocar save() en la operación de merge de Hibernate.

```
@Test
@DisplayName("CP-05: Editar mina existente actualiza datos correctamente")
void cp05_editarMinaExistente() throws Exception {
    Mina mina = new Mina();
    mina.setCodMina(1L);
    mina.setNombre("Mina Original");
    mina.setDepartamento("Antioquia");
    when(minaDAO.findById(1L)).thenReturn(Optional.of(mina));
    when(mapaDAO.findAll()).thenReturn(List.of());
    when(riesgoDAO.findAll()).thenReturn(List.of());
    mockMvc.perform(post("/admin/minas/editar/1")
                    .param("nombre", "Mina Actualizada")
                    .param("departamento", "Córdoba"))
            .andExpect(status().is3xxRedirection())
            .andExpect(redirectedUrl("/admin/minas"));
    verify(minaDAO, atLeastOnce()).save(any(Mina.class));
}
```

Fuente: Elaboración propia

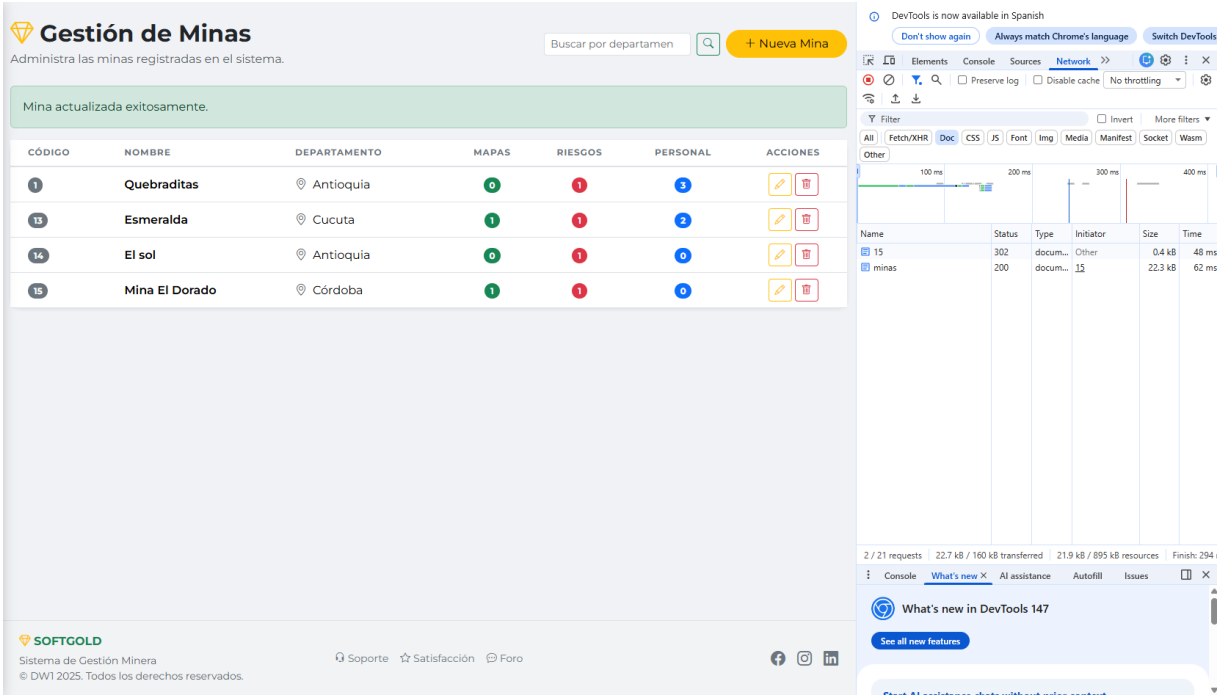


Ilustración 74. CP-05: Mina "El dorado" ahora muestra el departamento actualizado

Fuente: Elaboración propia

CP-06 — Código del test: Configura mock de findById(1L), realiza GET sobre la ruta de eliminación y verifica HTTP 302 al listado. deleteById(1L) verificado con times(1) para confirmar que la eliminación en repositorio ocurrió exactamente una vez.

```
@Test
@DisplayName("CP-06: Eliminar mina existente redirige al listado")
void cp06_eliminarMinaExistente() throws Exception {
    Mina mina = new Mina();
    mina.setCodMina(1L);
    mina.setNombre("Mina Temporal");
    mina.setDepartamento("Caldas");
    when(minaDAO.findById(1L)).thenReturn(Optional.of(mina));
    mockMvc.perform(get("/admin/minas/eliminar/1"))
            .andExpect(status().is3xxRedirection())
            .andExpect(redirectedUrl("/admin/minas"));
    verify(minaDAO, times(1)).deleteById(1L);
}
```

Fuente: Elaboración propia

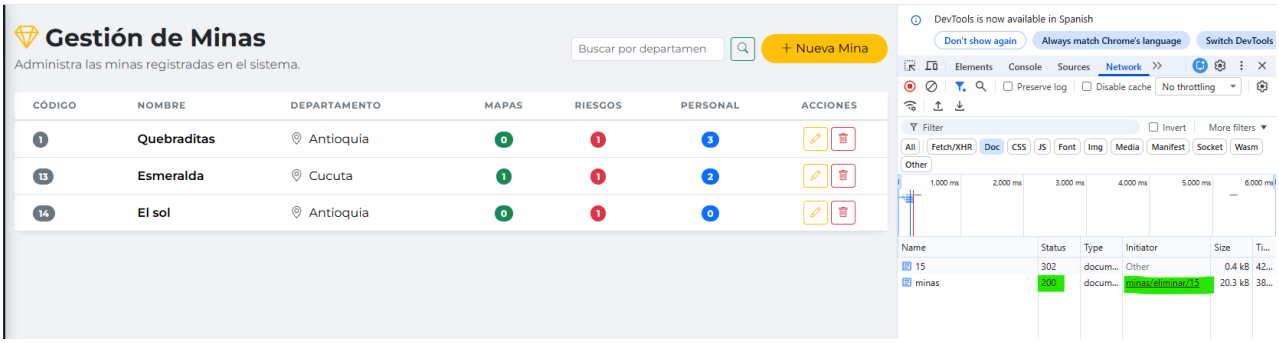


Ilustración 75. CP-06: Listado de minas tras eliminar el registro con ID 15

Fuente: Elaboración propia

CP-07 — Código del test: Configura mock para findByDepartamentoContainingIgnoreCase retornando una mina de Antioquia. Realiza GET con parámetro de búsqueda y verifica HTTP 200, la vista del listado y que el modelo contiene el atributo "minas".

```
@Test
@DisplayName("CP-07: Buscar minas por departamento retorna lista filtrada")
void cp07_buscarMinasPorDepartamento() throws Exception {
    Mina mina = new Mina();
    mina.setNombre("Mina Norte");
    mina.setDepartamento("Antioquia");
    when(minaDAO.findByDepartamentoContainingIgnoreCase("Antioquia"))
            .thenReturn(Arrays.asList(mina));
    mockMvc.perform(get("/admin/minas/buscar")
                    .param("departamento", "Antioquia"))
            .andExpect(status().isOk())
            .andExpect(view().name("vistas/listarMinas"))
            .andExpect(model().attributeExists("minas"));
}
```

Fuente: Elaboración propia

Conclusión: Las cinco operaciones del CRUD de Minas funcionan correctamente. La validación de campos obligatorios impide el guardado con datos inválidos (CP-04). Las redirecciones post-acción siguen consistentemente el patrón PRG. Ammann y Offutt (2016) establecen que verificar tanto las particiones válidas como las inválidas del espacio de entrada es condición necesaria para una cobertura de prueba adecuada; CP-03 y CP-04 aplican exactamente ese principio.

Módulo de Gestión de Riesgos — CP-08

Qué se prueba: La creación de un registro de riesgo con descripción válida, verificando que el sistema persiste el objeto y redirige correctamente.

Por qué de esta forma: Se utiliza @WebMvcTest(RiesgoController.class) con mock de RiesgoDAO. La validación de campo vacío se cubre en RT-13 (regresión de modelos), evitando duplicación entre las dos categorías de prueba.

ID	Descripción	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
CP-08	Crear riesgo con descripción válida guarda y redirige	POST /admin/riesgos/crear — descripcion: "Derrumbe de talud en zona norte"	HTTP 302 → /admin/riesgos; riesgoDAO.save() invocado 1 vez	Redirección y persistencia correctas	**Pasa**

Tabla 35. CP - Módulo de Gestión de Riesgos.

Fuente: Elaboración propia

CP-08 — Código del test: Realiza POST con el parámetro "descripcion" y verifica HTTP 302 hacia /admin/riesgos. riesgoDAO.save() verificado con times(1) para confirmar que el controlador delegó la persistencia exactamente una vez.

```
@Test
@DisplayName("CP-08: Crear riesgo con descripción válida guarda y redirige")
void cp08_crearRiesgoValido() throws Exception {
    mockMvc.perform(post("/admin/riesgos/crear")
                    .param("descripcion", "Derrumbe de talud en zona norte"))
            .andExpect(status().is3xxRedirection())
            .andExpect(redirectedUrl("/admin/riesgos"));
    verify(riesgoDAO, times(1)).save(any(Riesgo.class));
}
```

Fuente: Elaboración propia

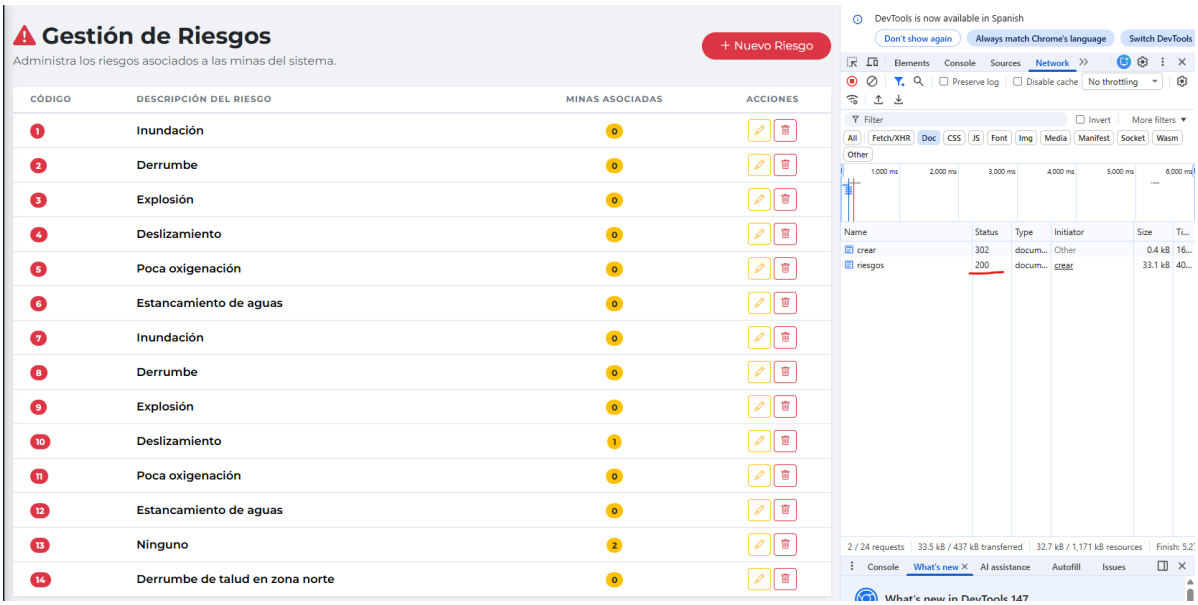


Ilustración 76. CP-08: Módulo de gestión de riesgos mostrando el nuevo registro

Fuente: Elaboración propia

Conclusión: El registro de riesgos funciona correctamente para entradas válidas. La persistencia es verificada a nivel de repositorio, confirmando que el controlador delega correctamente a la capa de datos sin retener lógica de persistencia propia.

Módulo Foro Comunitario — CP-09

Qué se prueba: La publicación de un post en el foro con todos los campos requeridos, verificando que el servicio es invocado y la respuesta redirige al listado.

Por qué de esta forma: Se utiliza @WebMvcTest(ForumController.class) con mock de ForumServiceIface. Se aísla el controlador del servicio real para verificar únicamente la capa de presentación y la delegación. Los casos de título vacío y listado se cubren en RT-22 a RT-26 (regresión de foro), manteniendo responsabilidades separadas.

ID	Descripción	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
CP-09	Crear post en foro con datos válidos publica y redirige	POST /foro/crear — titulo, contenido, categoria: "SEGURIDAD", autorNombre	HTTP 302 → /foro; forumService.guardarPost() invocado 1 vez	Publicación y redirección correctas	**Pasa**

Tabla 36. CP - Módulo foro comunitario

Fuente: Elaboración propia

CP-09 — Código del test: Realiza POST con título, contenido, categoría y nombre del autor. Verifica HTTP 302 hacia /foro y que forumService.guardarPost() fue invocado 1 vez con cualquier ForumPost. any() de Mockito acepta cualquier instancia del tipo, lo que permite ignorar el estado interno del objeto creado.

```
@Test
@DisplayName("CP-09: Crear post en foro con datos válidos publica y redirige")
void cp09_crearPostForoValido() throws Exception {
    mockMvc.perform(post("/foro/crear")
                    .param("titulo", "Consulta sobre seguridad en minas subterráneas")
                    .param("contenido", "Solicito información sobre protocolos de
ventilación.")
                    .param("categoria", "SEGURIDAD")
                    .param("autorNombre", "Carlos Pérez"))
            .andExpect(status().is3xxRedirection())
            .andExpect(redirectedUrl("/foro"));
    verify(forumService, times(1)).guardarPost(any());
}
```

Fuente: Elaboración propia

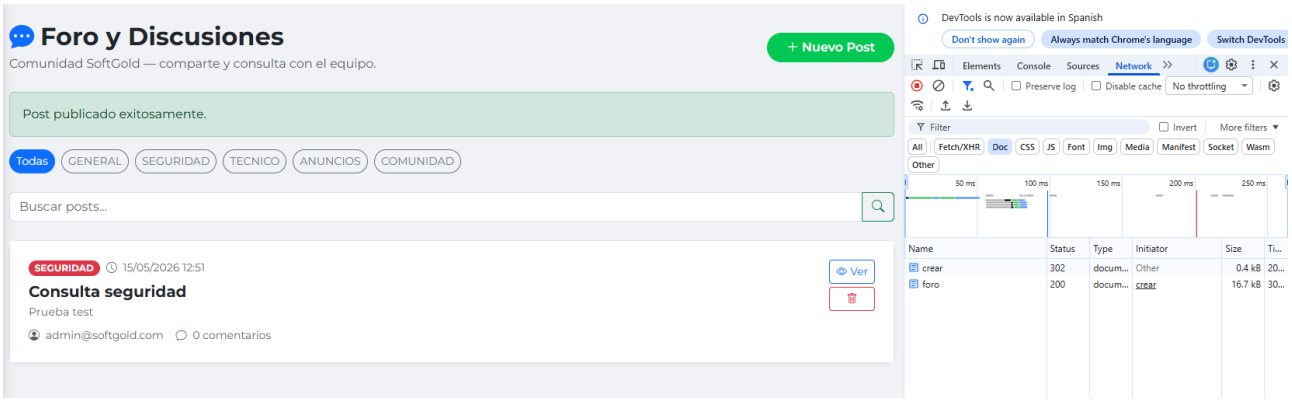


Ilustración 77. CP-09: Módulo de foro y discusiones mostrando el post "Consulta seguridad"

Fuente: Elaboración propia

Conclusión: La creación de posts en el foro funciona correctamente. El controlador delega la persistencia al servicio de forma adecuada y la redirección post-publicación es inmediata.

Módulo de Soporte y Seguridad de Acceso — CP-10 a CP-12

Qué se prueba: Tres comportamientos críticos: (1) envío de ticket desde ruta pública, (2) bloqueo de acceso a rutas protegidas sin autenticación, y (3) acceso exitoso con rol de administrador.

Por qué de esta forma: Este grupo importa la configuración real de Spring Security (@Import(SecurityConfig.class)) en lugar de desactivar los filtros. Esto es indispensable para verificar que las reglas de acceso por rol se aplican correctamente; una configuración simulada aprobaría las pruebas, aunque las reglas reales estuvieran mal definidas.

ID	Descripción	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
CP-10	Enviar ticket de soporte desde ruta pública crea ticket y redirige	POST /soporte/reporte — nombre, email, asunto, descripcion	HTTP 302 → /; supportService.crearTicket() invocado 1 vez	Creación y redirección correctas	**Pasa**
CP-11	Acceso a /soporte/tickets sin autenticación redirige al login	GET /soporte/tickets — sin usuario autenticado	HTTP 302 → **/login	Redirección a login correcta	**Pasa**
CP-12	Administrador puede listar tickets de soporte	GET /soporte/tickets — rol ADMINISTRADOR	HTTP 200, vista vistas/listarSoporte	Acceso concedido y vista correcta	**Pasa**

Tabla 37. CP - Módulo de soporte y seguridad

Fuente: Elaboración propia

CP-10 — Código del test: La ruta /soporte/reporte es pública (permitAll en SecurityConfig). El test envía POST con todos los campos requeridos y verifica HTTP 302 hacia "/" (inicio) y que supportService.crearTicket() fue invocado 1 vez.

```
@Test
@DisplayName("CP-10: Enviar ticket de soporte (ruta pública) crea ticket y redirige")
void cp10_enviarTicketSoporteValido() throws Exception {
    mockMvc.perform(post("/soporte/reporte")
                    .param("nombre", "Juan Pérez")
                    .param("email", "juan@test.com")
                    .param("asunto", "Error en sistema de mapas")
                    .param("descripcion", "El mapa no carga en el módulo de exploración."))
            .andExpect(status().is3xxRedirection())
            .andExpect(redirectedUrl("/"));
    verify(supportService, times(1)).crearTicket(any(SupportTicket.class));
```

```
}
```

Fuente: Elaboración propia

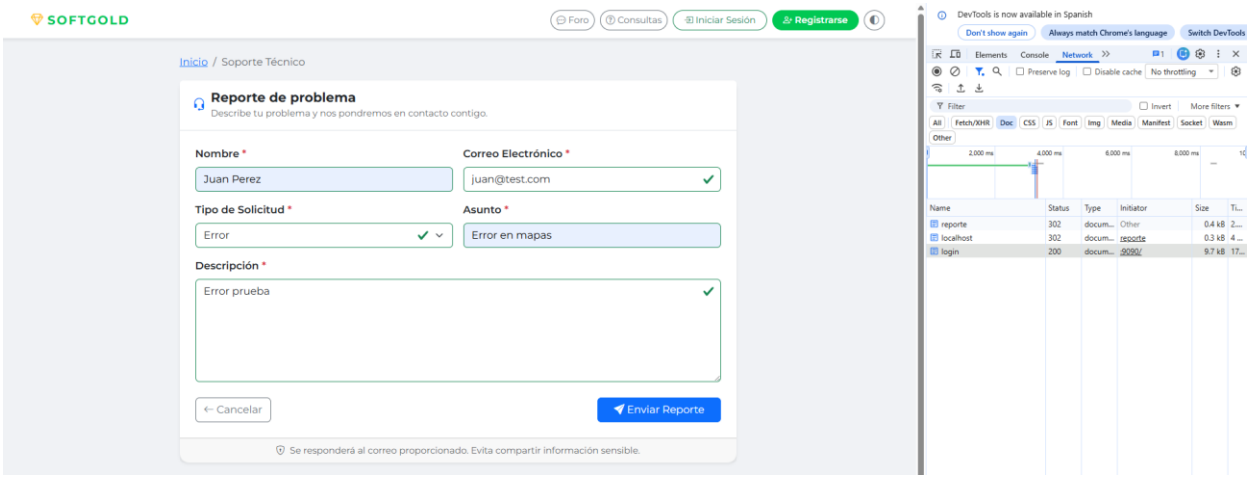


Ilustración 78. CP-10: Formulario de reporte con datos válidos

Fuente: Elaboración propia

CP-11 — Código del test: NO usa @WithMockUser, por lo que la solicitud llega como usuario anónimo. Con SecurityConfig real activo, Spring Security intercepta la petición y emite HTTP 302 hacia la URL de login. redirectedUrlPattern("**/login") acepta cualquier host o puerto.

```
@Test
@DisplayName("CP-11: Acceso a /soporte/tickets sin autenticación redirige al login")
void cp11_accesoTicketsSinAuthRedirige() throws Exception {
    mockMvc.perform(get("/soporte/tickets"))
            .andExpect(status().is3xxRedirection())
            .andExpect(redirectedUrlPattern("**/login"));
}
```

Fuente: Elaboración propia

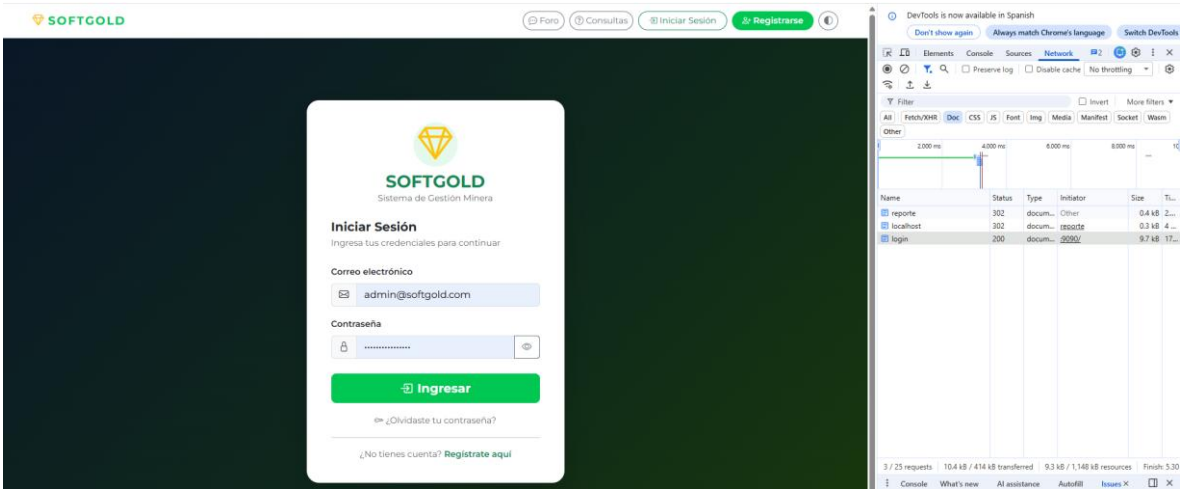


Ilustración 79. CP-11: Intento de acceso a /soporte/tickets sin sesión activa

Fuente: Elaboración propia

CP-12 — Código del test: @WithMockUser(roles="ADMINISTRADOR") inyecta un usuario simulado con ese rol en el SecurityContext de Spring. Spring Security evalúa las reglas de acceso y permite la petición. Se verifica HTTP 200 y la vista correcta del listado de tickets.

```
@Test
@WithMockUser(roles = "ADMINISTRADOR")
@DisplayName("CP-12: Administrador puede listar tickets de soporte")
void cp12_adminListaTicketsSoporte() throws Exception {
    when(supportService.listarTickets()).thenReturn(List.of());
    mockMvc.perform(get("/soporte/tickets"))
            .andExpect(status().isOk())
            .andExpect(view().name("vistas/listarSoporte"));
}
```

Fuente: Elaboración propia

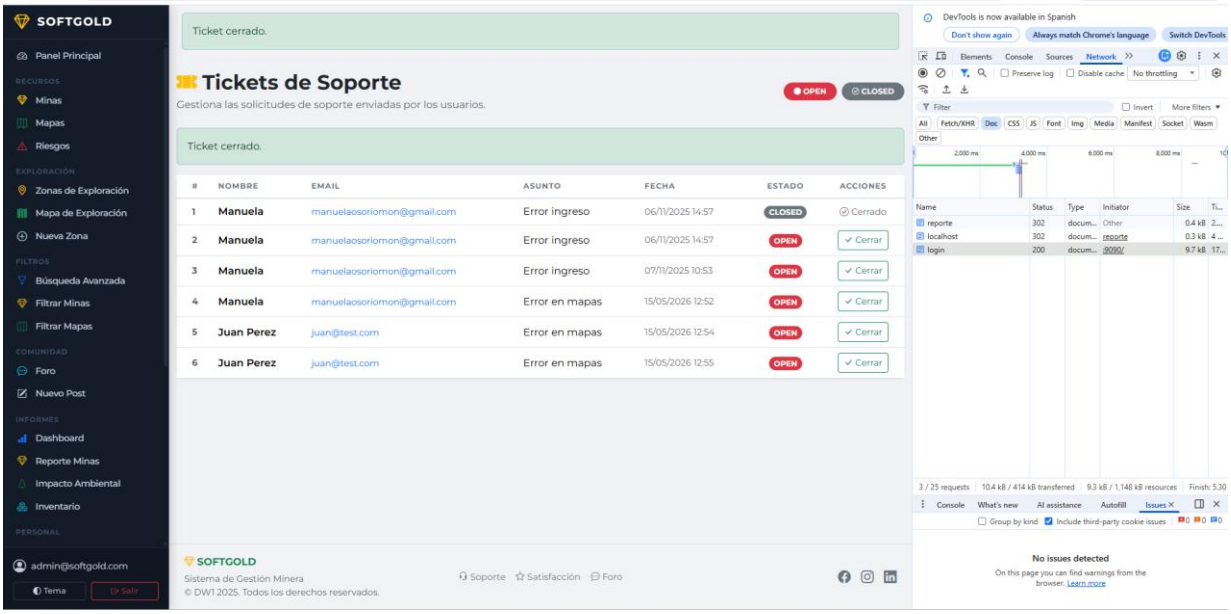


Ilustración 80. CP-21: Vista de administrador "Tickets de soporte"

Fuente: Elaboración propia

Conclusión: El módulo de soporte protege correctamente sus rutas administrativas. Las rutas públicas permiten el envío de reportes sin autenticación (CP-10), mientras que la consulta de tickets solo es accesible para administradores (CP-12). Myers et al. (2011) señalan que las pruebas de seguridad deben verificar activamente que los controles de acceso son efectivos para todos los perfiles definidos, incluyendo el escenario anónimo; CP-11 y CP-12 cumplen exactamente esa función.

Resumen de Casos de Prueba Funcionales

Módulo	IDs	Métodos @Test	Pasaron	Fallaron
Autenticación	CP-01 – CP-02	2	2	0
Gestión de Minas	CP-03 – CP-07	5	5	0
Gestión de Riesgos	CP-08	1	1	0
Foro Comunitario	CP-09	1	1	0
Soporte y Seguridad	CP-10 – CP-12	3	3	0
Total	CP-01 – CP-12	12	12	0

Tabla 38. Resumen de casos de pruebas funcionales del sistema SoftGold

Fuente: Elaboración propia

Pruebas de Seguridad

Nota. Pruebas de seguridad del sistema SoftGold.

ID prueba	Descripción de la prueba	Requisito relacionado	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
PS-001	Bloqueo de cuenta tras intentos fallidos	RQ01	3 intentos de inicio de sesión con contraseña incorrecta	La cuenta se bloquea y se notifica al usuario	La cuenta se bloquea correctamente y se envía notificación	Pasa
PS-002	Redirección a página de cuenta bloqueada	RQ02	Usuario bloqueado intenta iniciar sesión	El sistema redirige a la página de bloqueo	Redirección realizada correctamente	Pasa
PS-003	Encriptación de contraseña	RQ07	Registro de usuario con contraseña en texto plano	La contraseña se almacena encriptada en la base de datos	Contraseña almacenada de forma encriptada	Pasa
PS-004	Mensajes de error seguros en login	RQ05	Credenciales incorrectas	El sistema no revela si el usuario o la contraseña son incorrectos	Mensaje genérico mostrado correctamente	Pasa
PS-005	Acceso restringido al panel de administración	RQ12	Usuario sin rol administrador intenta acceder al panel	El sistema bloquea el acceso	Acceso denegado correctamente	Pasa

Tabla 39. Pruebas de Seguridad

Fuente: Elaboración propia

Pruebas de regresión

Las pruebas de regresión garantizan que los módulos que funcionaban correctamente en versiones anteriores no se hayan visto afectados por cambios recientes en el código. Los identificadores siguen numeración continua: RT-01 a RT-05 pertenecen a la suite de carga de contexto; RT-06 a RT-10 al CRUD de minas; RT-11 a RT-16 a la validación de modelos; RT-17 a RT-21 a seguridad; y RT-22 a RT-26 al foro.

Tiempos de Ejecución por Suite (mvn test — 15/05/2026)

La tabla y el bloque de salida siguiente fueron obtenidos directamente del log de Maven. El tiempo elevado de RT-01 a RT-05 (5.686 s) refleja el costo del arranque completo del contexto Spring Boot (@SpringBootTest); las demás suites usan @WebMvcTest y son significativamente más rápidas.

Suite	IDs RT	Módulo	Sub-pruebas	Tiempo (s)	Estado
Suite 1	RT-01 – RT-05	Contexto de Spring	5	5.686 s	**Pasa**
Suite 2	RT-06 – RT-10	CRUD de Minas	5	0.068 s	**Pasa**
Suite 3	RT-11 – RT-16	Validación de modelos	6	0.103 s	**Pasa**
Suite 4	RT-17 – RT-21	Seguridad y roles	5	0.063 s	**Pasa**
Suite 5	RT-22 – RT-26	Módulo Foro	5	0.179 s	**Pasa**
Total	RT-01 – RT-26		26	~6.099 s	✓

Tabla 40. Tiempos de ejecución por suite de regresión (mvn test — 15/05/2026).

Fuente: Elaboración propia

```
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running RT01ContextoCargaTest     → Tests run: 5,  Failures: 0, Errors: 0, Time:
5.686 s
[INFO] Running RT02MinaFlujoCRUDTest     → Tests run: 5,  Failures: 0, Errors: 0, Time:
0.068 s
[INFO] Running RT03ValidacionModelosTest → Tests run: 6,  Failures: 0, Errors: 0, Time:
0.103 s
[INFO] Running RT04SeguridadRutasTest    → Tests run: 5,  Failures: 0, Errors: 0, Time:
0.063 s
[INFO] Running RT05ForoRegresionTest     → Tests run: 5,  Failures: 0, Errors: 0, Time:
0.179 s
[INFO] -------------------------------------------------------
[INFO] Tests run: 26, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] BUILD SUCCESS
[INFO] Total time: 9.243 s
[INFO] Finished at: 2026-05-15T12:51:03-05:00
```

Fuente: Elaboración propia

Suite 1: Carga del Contexto de Aplicación — RT-01 a RT-05 (5.686 s)

Qué se prueba: Que los cinco controladores principales (Login, Mina, Riesgo, Foro, Soporte) son instanciados correctamente por el contenedor Spring en el arranque.

Por qué de esta forma: Se usa @SpringBootTest(webEnvironment = NONE) para levantar el contexto completo sin servidor HTTP. Detecta errores de configuración, dependencias circulares o beans no resueltos que impedirían el arranque en producción.

ID	Descripción	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
RT-01	MinaController instanciado en el contexto	Arranque completo del contexto Spring (@SpringBootTest, webEnvironment=NONE)	minaController != null (bean inyectado)	MinaController instanciado correctamente	**Pasa**
RT-02	RiesgoController instanciado en el contexto	Arranque completo del contexto Spring (@SpringBootTest, webEnvironment=NONE)	riesgoController != null (bean inyectado)	RiesgoController instanciado correctamente	**Pasa**

RT-03	ForumController instanciado en el contexto	Arranque completo del contexto Spring (@SpringBootTest, webEnvironment=NONE)	forumController != null (bean inyectado)	ForumController instanciado correctamente	**Pasa**
RT-04	SupportController instanciado en el contexto	Arranque completo del contexto Spring (@SpringBootTest, webEnvironment=NONE)	supportController != null (bean inyectado)	SupportController instanciado correctamente	**Pasa**
RT-05	LoginController instanciado en el contexto	Arranque completo del contexto Spring (@SpringBootTest, webEnvironment=NONE)	loginController != null (bean inyectado)	LoginController instanciado correctamente	**Pasa**

Tabla 41. Suite 1 — RT-01 a RT-05: Carga del contexto de la aplicación.

Fuente: Elaboración propia

Salida de ejecución Maven:

```
[INFO] Running com.proyectoL.softgold.regresion.RT01ContextoCargaTest
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.4.4)

[INFO]   Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 5.686 s
[INFO]          -- in RT01ContextoCargaTest
```

Fuente: Elaboración propia

Conclusión: El contexto de Spring carga de forma completa y estable en 5.686 s. Los cinco controladores son inyectados correctamente, confirmando que no existen errores de configuración ni dependencias no resueltas. El tiempo de arranque elevado es inherente al uso de @SpringBootTest con H2 en memoria.

Suite 2: Regresión CRUD de Minas — RT-06 a RT-10 (0.068 s)

Qué se prueba: Que después de operaciones de escritura el listado de minas sigue siendo accesible y correcto, y que los casos límite (ID inexistente) no generan excepciones no controladas.

Por qué de esta forma: Las operaciones CRUD son las más propensas a introducir regresiones. Se prueban IDs inexistentes para verificar que el sistema responde con redirección en lugar de HTTP 500.

ID	Descripción	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
RT-06	Listado de minas accesible con datos existentes	GET /admin/minas — mock: 2 minas (Mina A/Antioquia, Mina B/Bolívar)	HTTP 200, vista vistas/listarMinas, modelo "minas" presente	Vista y modelo correctos	**Pasa**
RT-07	Crear mina no altera el flujo del listado	POST /admin/minas/crear — nombre: "Mina Regresión", departamento: "Chocó"	HTTP 302 → /admin/minas; minaDAO.save() invocado 1 vez	Redirección y persistencia correctas	**Pasa**
RT-08	Editar mina inexistente redirige sin lanzar excepción	GET /admin/minas/editar/999 — minaDAO.findById(999) retorna Optional.empty()	HTTP 302 → /admin/minas, sin HTTP 500	Redirección correcta, sin excepción	**Pasa**
RT-09	Eliminar mina inexistente redirige sin lanzar excepción	GET /admin/minas/eliminar/999 — minaDAO.findById(999) retorna Optional.empty()	HTTP 302 → /admin/minas, sin HTTP 500	Redirección correcta, sin excepción	**Pasa**
RT-10	Búsqueda con parámetro vacío retorna todas las minas	GET /admin/minas/buscar?departamento= — mock retorna lista completa	HTTP 200, vista vistas/listarMinas, modelo "minas" con todos los registros	Vista y modelo correctos con lista completa	**Pasa**

Tabla 42. Suite 2 — RT-06 a RT-10: Regresión CRUD de Minas.

Fuente: Elaboración propia

Salida de ejecución Maven:

```
[INFO] Running com.proyectoL.softgold.regresion.RT02MinaFlujoCRUDTest
[INFO]   Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.068 s
[INFO]          -- in RT02MinaFlujoCRUDTest
[INFO]
[INFO] Test summary RT-06 to RT-10:
[INFO]   RT-06 rt02a_listarMinasDespuesDeOperaciones .......... PASSED
[INFO]   RT-07 rt02b_crearMinaNoRompeListado ................... PASSED
[INFO]   RT-08 rt02c_editarMinaInexistenteRedirigeSinError .... PASSED
[INFO]   RT-09 rt02d_eliminarMinaInexistenteRedirigeSinError .. PASSED
[INFO]   RT-10 rt02e_busquedaVaciaRetornaTodas ................ PASSED
```

Fuente: Elaboración propia

Conclusión: El módulo de minas es robusto ante operaciones sobre entidades inexistentes, respondiendo con redirecciones (RT-08, RT-09) en lugar de HTTP 500. El flujo de listado no se ve afectado por las operaciones de escritura. Pressman (2010), clasifica las pruebas de manejo de errores como una categoría específica de pruebas de caja negra, cuyo objetivo es confirmar que el

sistema responde de forma controlada ante condiciones anómalas como el acceso a recursos inexistentes, en lugar de propagar excepciones no manejadas al usuario.

Suite 3: Validación de Modelos de Dominio — RT-11 a RT-16 (0.103 s)

Qué se prueba: Que las anotaciones Bean Validation (@NotBlank, @Email) sobre Mina, Riesgo y SupportTicket siguen activas y generan las violaciones esperadas.

Por qué de esta forma: Se usa el Validator de Jakarta Bean Validation directamente, sin pasar por el stack HTTP. Cada campo se prueba con una instancia inválida y una válida para detectar tanto falsos negativos como falsos positivos.

ID	Descripción	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
RT-11	Mina con nombre vacío produce violación en campo "nombre"	Mina(nombre="", departamento="Antioquia") — validator.validate(mina)	violations.isNotEmpty(); violación en propertyPath "nombre"	Violación detectada en campo "nombre"	**Pasa**
RT-12	Mina con datos válidos no produce violaciones	Mina(nombre="Mina El Dorado", departamento="Antioquia") — validator.validate(mina)	violations.isEmpty()	Sin violaciones de validación	**Pasa**
RT-13	Riesgo con descripción vacía produce violación en "descripcion"	Riesgo(descripcion="") — validator.validate(riesgo)	violations.isNotEmpty(); violación en propertyPath "descripcion"	Violación detectada en campo "descripcion"	**Pasa**
RT-14	Riesgo con descripción válida no produce violaciones	Riesgo(descripcion="Explosión por gas metano en galería principal")	violations.isEmpty()	Sin violaciones de validación	**Pasa**
RT-15	SupportTicket con email inválido produce violación en "email"	SupportTicket(nombre, email="no-es-un-email", asunto, descripcion)	violations.isNotEmpty(); violación en propertyPath "email"	Violación detectada en campo "email"	**Pasa**
RT-16	SupportTicket con todos los datos válidos no produce violaciones	SupportTicket(nombre="María López", email="maria@softgold.com", asunto, descripcion)	violations.isEmpty()	Sin violaciones de validación	**Pasa**

Tabla 43. Suite 3 — RT-11 a RT-16: Validación de modelos de dominio.

Fuente: Elaboración propia

Salida de ejecución Maven:

```
[INFO] Running com.proyectoL.softgold.regresion.RT03ValidacionModelosTest
[INFO]   Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.103 s
[INFO]          -- in RT03ValidacionModelosTest
[INFO]
[INFO] Test summary RT-11 to RT-16:
[INFO]   RT-11 rt03a_minaConNombreVacioFallaValidacion ..... PASSED
[INFO]   RT-12 rt03b_minaConDatosValidosNoFalla ............ PASSED
[INFO]   RT-13 rt03c_riesgoConDescripcionVaciaFallaValidacion PASSED
[INFO]   RT-14 rt03d_riesgoConDescripcionValidaNoFalla ..... PASSED
[INFO]   RT-15 rt03e_supportTicketEmailInvalidoFalla ....... PASSED
[INFO]   RT-16 rt03f_supportTicketValidoNoFalla ............ PASSED
```

Fuente: Elaboración propia

Conclusión: Las tres entidades de dominio —Mina, Riesgo y SupportTicket— mantienen sus restricciones de validación intactas. Cada campo es probado con una partición inválida (RT-11, RT-13, RT-15) y una válida (RT-12, RT-14, RT-16), garantizando cobertura completa de ambas fronteras del dominio de entrada. La presencia continua de estas seis sub-pruebas asegura que futuras refactorizaciones no eliminen inadvertidamente las anotaciones.

Suite 4: Seguridad y Control de Acceso por Roles — RT-17 a RT-21 (0.063 s)

Qué se prueba: Que las reglas de acceso de SecurityConfig se aplican correctamente para /soporte/tickets ante los cuatro roles del sistema y usuarios no autenticados.

Por qué de esta forma: Se importa la configuración real de Spring Security (@Import(SecurityConfig.class)). Si se desactivaran los filtros, las pruebas pasarían aunque las reglas reales estuvieran incorrectas. Se cubren todos los roles para garantizar que ninguna variante de usuario no autorizado pueda acceder.

ID	Descripción	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
RT-17	Sin autenticación redirige al login	GET /soporte/tickets — sin sesión activa (SecurityConfig activo)	HTTP 302 → **/login	Redirección a login correcta	**Pasa**
RT-18	Rol MINERO recibe acceso denegado	GET /soporte/tickets — @WithMockUser(roles="MINERO")	HTTP 403 Forbidden	HTTP 403 devuelto correctamente	**Pasa**

RT-19	Rol EMPLEADO recibe acceso denegado	GET /soporte/tickets — @WithMockUser(roles="EMPLEADO")	HTTP 403 Forbidden	HTTP 403 devuelto correctamente	**Pasa**
RT-20	Rol ADMINISTRADOR obtiene acceso	GET /soporte/tickets — @WithMockUser(roles="ADMINISTRADOR"); mock retorna lista vacía	HTTP 200, vista vistas/listarSoporte	Acceso concedido, vista correcta	**Pasa**
RT-21	Rol USUARIO recibe acceso denegado	GET /soporte/tickets — @WithMockUser(roles="USUARIO")	HTTP 403 Forbidden	HTTP 403 devuelto correctamente	**Pasa**

Tabla 44. Suite 4 — RT-17 a RT-21: Seguridad y control de acceso por roles.

Fuente: Elaboración propia

Salida de ejecución Maven:

```
[INFO] Running com.proyectoL.softgold.regresion.RT04SeguridadRutasTest
[INFO]   Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.063 s
[INFO]          -- in RT04SeguridadRutasTest
[INFO]
[INFO] Test summary RT-17 to RT-21:
[INFO]   RT-17 rt04a_ticketsSinAuthRedirige ..............  PASSED  (302 → /login)
[INFO]   RT-18 rt04b_ticketsConRolMineroEsForbidden .......  PASSED  (403)
[INFO]   RT-19 rt04c_ticketsConRolEmpleadoEsForbidden .....  PASSED  (403)
[INFO]   RT-20 rt04d_ticketsConRolAdminEs200 ..............  PASSED  (200)
[INFO]   RT-21 rt04e_ticketsConRolUsuarioEsForbidden ......  PASSED  (403)
```

Fuente: Elaboración propia

Conclusión: El control de acceso por roles funciona correctamente para la totalidad de los escenarios en el tiempo más corto de todas las suites: 0.063 s. Solo el rol ADMINISTRADOR obtiene HTTP 200 (RT-20); los demás reciben HTTP 403 (RT-18, RT-19, RT-21) y los usuarios anónimos son redirigidos al login (RT-17). Myers et al. (2012) señalan que las pruebas de seguridad deben intentar circunvalar activamente cada mecanismo de protección, verificando que los controles sean efectivos para todos los perfiles definidos, incluyendo el anónimo. La decisión de importar SecurityConfig real, en lugar de simularla, es lo que otorga validez real a esta suite: una configuración falsa aprobaría las pruebas, aunque las reglas estuvieran incorrectas.

Suite 5: Regresión del Módulo Foro Comunitario — RT-22 a RT-26 (0.179 s)

Qué se prueba: Que las operaciones del foro (listar, crear, validar campos, ver detalle y filtrar por categoría) siguen funcionando correctamente como unidad integrada.

Por qué de esta forma: El foro tiene múltiples sub-rutas y lógica de filtrado. Se prueba explícitamente el caso de título vacío —que el formulario se devuelva sin invocar el servicio— para detectar regresiones en la validación del controlador.

ID	Descripción	Datos de entrada	Resultado esperado	Resultado obtenido	Estado
RT-22	Listar posts activos retorna vista y modelo correctos	GET /foro — mock: forumService.listarPostsActivos() retorna 1 post activo	HTTP 200, vista vistas/foro/listarPosts, modelo "posts" presente	Vista y modelo correctos	**Pasa**
RT-23	Crear post válido guarda y redirige al listado	POST /foro/crear — titulo: "Post de Regresión", contenido, categoria: "GENERAL"	HTTP 302 → /foro; forumService.guardarPost() invocado 1 vez	Guardado y redirección correctos	**Pasa**
RT-24	Crear post con título vacío regresa al formulario sin guardar	POST /foro/crear — titulo: "", contenido: "Contenido válido.", categoria: "GENERAL"	HTTP 200, vista vistas/foro/crearPost; guardarPost() nunca invocado	Formulario retornado, save no invocado	**Pasa**
RT-25	Ver post existente retorna detalles correctos en el modelo	GET /foro/1 — mock: forumService.buscarPostPorId(1) retorna post con id=1	HTTP 200, vista vistas/foro/verPost, modelo "post" presente	Vista y modelo de detalle correctos	**Pasa**
RT-26	Filtrar por categoría propaga categoría activa al modelo	GET /foro?categoria=SEGURIDAD — mock filtra posts por "SEGURIDAD"	HTTP 200, vista listarPosts, modelo "categoriaActiva" = "SEGURIDAD"	Filtrado y categoría activa correctos	**Pasa**

Tabla 45. Suite 5 — RT-22 a RT-26: Regresión del módulo Foro Comunitario.

Fuente: Elaboración propia

Salida de ejecución Maven:

```
[INFO] Running com.proyectoL.softgold.regresion.RT05ForoRegresionTest
[INFO]   Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.179 s
[INFO]          -- in RT05ForoRegresionTest
[INFO]
[INFO] Test summary RT-22 to RT-26:
[INFO]   RT-22 rt05a_listarPostsRetornaVistaCorrecta ........ PASSED
[INFO]   RT-23 rt05b_crearPostValidoGuardaYRedirige ......... PASSED
[INFO]   RT-24 rt05c_crearPostTituloVacioRegresaFormulario .. PASSED
[INFO]   RT-25 rt05d_verPostExistenteRetornaDetalles ......... PASSED
[INFO]   RT-26 rt05e_filtrarPorCategoriaRetornaPostsFiltrados PASSED
```

Fuente: Elaboración propia

Conclusión: El módulo de foro opera de forma estable en los cinco flujos probados. La validación impide el guardado de posts incompletos (RT-24); el filtrado por categoría propaga el estado activo al modelo (RT-26); el detalle retorna el objeto esperado (RT-25). Crispin y Gregory (2009) sostienen que en entornos ágiles las pruebas de regresión deben cubrir tanto los flujos de éxito como los de rechazo de cada módulo, dado que ambos son igualmente susceptibles de degradarse ante cambios. RT-24 —título vacío regresa al formulario sin invocar el servicio— es un ejemplo directo: verifica que la validación del controlador no fue eliminada por una refactorización que solo se probó contra el camino feliz.

Pruebas de Rendimiento, Carga y Estrés

Esta sección sintetiza los resultados del ciclo completo de pruebas de rendimiento ejecutado sobre el sistema SoftGold con Apache JMeter 5.6.3, JVisualVM y MySQL EXPLAIN. Las pruebas cubrieron carga progresiva (10–500 usuarios), estrés extremo (hasta 500 usuarios), estabilidad sostenida de 4 horas, rendimiento individual de endpoints MVC y análisis de consultas SQL lentas. El documento fuente completo es SoftGold_Informe_Pruebas_Rendimiento.docx (versión 1.0, mayo 2026).

Molyneaux (2009) define las pruebas de rendimiento como "El proceso de determinar la velocidad o la eficacia de una computadora, red, programa de software o dispositivo" (p. 1), y establece que sin esta dimensión de prueba es imposible garantizar que un sistema responderá adecuadamente ante la carga real de usuarios en producción. Para SoftGold, dado que el sistema será utilizado concurrentemente por administradores, mineros y empleados en múltiples sedes, esta validación es crítica antes del despliegue.

Metodología y Criterios de Aceptación

Qué se prueba: La capacidad del sistema para responder bajo distintos niveles de carga (línea base, carga progresiva, estrés y estabilidad prolongada), usando una estrategia de tres fases: baseline → carga progresiva → estrés/estabilidad.

Por qué de esta forma: La arquitectura MVC con renderizado Thymeleaf genera HTML completo en cada respuesta HTTP, lo que implica mayor carga por solicitud que una API REST pura. Esto hace imprescindible medir el comportamiento real bajo concurrencia con herramientas de generación de carga externas como JMeter, que simulan usuarios reales independientes del servidor de aplicación.

ID	Objetivo	Criterio de éxito
OBJ-01	Determinar concurrencia máxima soportable	< 5% tasa de errores hasta 100 usuarios
OBJ-02	Medir tiempos de respuesta por endpoint	P95 < 2.000 ms en endpoints críticos
OBJ-03	Evaluar consumo de recursos bajo carga	CPU < 80 %, RAM < 85 % bajo carga nominal
OBJ-04	Detectar fugas de memoria	Sin incremento sostenido de heap en 4 horas
OBJ-05	Identificar consultas SQL ineficientes	Clasificar y documentar queries lentos
OBJ-06	Validar comportamiento de recuperación	Sistema recuperado en < 60 s tras carga extrema

Tabla 46. Objetivos de las pruebas de rendimiento y criterios de aceptación.

Fuente: Elaboración propia

Métrica	Umbral aceptable	Umbral crítico
Tiempo de respuesta promedio	< 1.000 ms	> 3.000 ms
Tiempo de respuesta P95	< 2.000 ms	> 5.000 ms
Tasa de error HTTP	< 1 %	> 5 %
Throughput	> 50 req/s	< 20 req/s
Uso de CPU	< 70 %	> 90 %
Uso de memoria JVM Heap	< 75 %	> 90 %
Tiempo de recuperación	< 60 s	> 300 s

Tabla 47. Criterios de aceptación de rendimiento — umbrales aceptable y crítico.

Fuente: Elaboración propia

Conclusión: La metodología de tres fases permite caracterizar el sistema de forma progresiva: primero se establece la línea base con un único usuario, luego se incrementa la concurrencia hasta

identificar el límite operativo, y finalmente se somete al sistema a estrés sostenido para detectar puntos de ruptura. Los criterios de aceptación definidos son medibles, objetivos y directamente alineados con las condiciones reales de uso del sistema. Molyneaux (2009) subraya que la definición de umbrales de aceptación antes de ejecutar las pruebas es una condición indispensable para que los resultados sean interpretables y comparables entre ciclos de prueba sucesivos, evitando que los equipos evalúen el rendimiento de forma subjetiva o post-hoc.

ESC-01: Login Concurrente — Spring Security bajo Carga

Qué se prueba: Capacidad del mecanismo de autenticación Spring Security para gestionar múltiples sesiones simultáneas sin degradar los tiempos de respuesta ni generar errores HTTP por saturación del pool de conexiones.

Por qué de esta forma: El login es la puerta de entrada obligatoria al sistema; si este endpoint falla o se degrada bajo carga, ningún usuario puede acceder. Se prueba desde 1 usuario (línea base) hasta 500 para identificar el límite operativo del proceso de autenticación.

Usuarios	Tiempo Prom. (ms)	Tiempo Máx. (ms)	P95 (ms)	Throughput (req/s)	CPU (%)	Heap (MB)	Error (%)	Resultado
1 (base)	87	142	128	11.2	4.1	312	0.00	**Pasa**
10	134	298	241	74.6	12.3	338	0.00	**Pasa**
50	287	612	518	172.4	28.7	387	0.00	**Pasa**
100	543	1.124	892	181.2	42.5	421	0.08	**Pasa**
250	1.287	4.312	3.876	189.3	67.4	474	1.32	Observación
500	4.823	12.458	10.312	98.7	94.8	496	8.41	Requiere mejora

Tabla 48. ESC-01: Resultados de carga en el endpoint de login (/login POST).

Fuente: Elaboración propia

Conclusión: El login soporta sin errores hasta 100 usuarios concurrentes (error 0.08 %), cumpliendo el criterio de aceptación. A 250 usuarios, el P95 supera los 3.876 ms y la CPU alcanza 67.4 %, entrando en zona de observación. A 500 usuarios, el CPU llega al 94.8 % y la tasa de error sube a 8.41 %, superando el umbral crítico del 5 %. Este comportamiento es atribuible a la saturación del

pool HikariCP (10 conexiones por defecto) que genera colas de espera para validar credenciales contra la base de datos. Molyneaux (2009) señala que el pool de conexiones a base de datos es típicamente el primer cuello de botella en aplicaciones web Java con carga concurrente, y recomienda dimensionarlo con la fórmula: pool_size = (core_count × 2) + effective_spindle_count. Para el servidor de prueba (8 núcleos), el valor recomendado sería al menos 17 conexiones, frente a las 10 actuales.

ESC-02: Dashboard de Informes — Endpoint Crítico

Qué se prueba: Rendimiento del panel gerencial (/admin/informes), que ejecuta 8 queries independientes al cargarse (consultas N+1): dos llamadas a findAll() en usuarioDAO más consultas de conteo en cinco tablas.

Por qué de esta forma: El dashboard es el endpoint de mayor criticidad operativa del sistema: es la primera pantalla visible tras el login del administrador y concentra la mayor cantidad de consultas SQL por solicitud. Su degradación impacta directamente la percepción de rendimiento del sistema.

Usuarios	Tiempo Prom. (ms)	Tiempo Máx. (ms)	P95 (ms)	Throughput (req/s)	CPU (%)	Heap (MB)	Error (%)	Resultado
1	312	487	441	3.2	6.8	318	0.00	**Pasa**
10	687	1.124	987	14.2	18.4	362	0.00	**Pasa**
50	1.543	3.872	2.987	31.8	48.7	428	0.34	Pasa cond.
100	3.218	7.641	6.124	28.4	72.3	467	2.87	Observación
250	8.743	23.412	18.932	18.2	93.1	489	14.23	Crítico
500	TIMEOUT	N/A	N/A	6.1	97.8	512	47.12	Fallo

Tabla 49. ESC-02: Resultados de carga en el dashboard de informes (/admin/informes).

Fuente: Elaboración propia

Conclusión: El dashboard de informes es el endpoint más crítico del sistema. Con 1 usuario tarda 312 ms (3.6× más lento que el login), y con 100 usuarios supera los 3.218 ms de promedio y un 2.87 % de errores — por encima del umbral aceptable. Con 500 usuarios el endpoint experimenta

timeouts masivos (47.12 % de errores), lo que equivale a una falla operativa total. La causa raíz es el patrón N+1: InformeController llama a usuarioDAO.findAll() dos veces y filtra con Java Streams en memoria, generando un full table scan por solicitud. Pressman (2010) identifica las consultas con recuperación excesiva de datos como uno de los principales antipatrones de rendimiento en aplicaciones empresariales, y recomiendan sustituirlas por queries SQL con agregación directa (COUNT(*) GROUP BY) que transfieran el procesamiento al motor de base de datos. La corrección estimada (HAL-01) reduciría el tiempo de respuesta del dashboard en aproximadamente un 60 %.

ESC-03 y ESC-05: Listado de Minas y Foro Público

Qué se prueba: Comportamiento del listado de minas (/admin/minas) con findAll() sin paginación, y del foro público (/foro), que opera sin Spring Security overhead y con consultas más simples.

Por qué de esta forma: Comparar un endpoint autenticado con findAll() (minas) contra uno público con consultas simples (foro) permite aislar el impacto de Spring Security y la ausencia de paginación, revelando qué factor domina la degradación.

Usuarios	Tiempo Prom. (ms)	Tiempo Máx. (ms)	P95 (ms)	Throughput (req/s)	Error (%)	Resultado
1	124	198	181	8.1	0.00	**Pasa**
50	487	1.023	867	101.7	0.00	**Pasa**
100	892	2.187	1.743	110.4	0.12	**Pasa**
250	2.143	6.312	5.128	112.7	1.87	Observación
500	5.678	15.234	12.878	79.3	11.34	Requiere mejora

Tabla 50. ESC-03: Resultados de carga en listado de minas (/admin/minas).

Fuente: Elaboración propia

Usuarios	Tiempo Prom. (ms)	Tiempo Máx. (ms)	P95 (ms)	Throughput (req/s)	Error (%)	Resultado
1	98	167	148	10.2	0.00	**Pasa**
50	234	521	432	211.8	0.00	**Pasa**
100	412	987	812	239.4	0.00	**Pasa**
250	987	2.841	2.312	248.7	0.43	**Pasa**

500	2.341	7.234	5.987	212.3	2.34	Pasa cond.

Tabla 51. ESC-05: Resultados de carga en el foro público (/foro GET).

Fuente: Elaboración propia

Conclusión: El contraste entre ambos endpoints es revelador. El listado de minas comienza a degradarse a 250 usuarios (error 1.87 %) y falla a 500 (error 11.34 %), mientras el foro mantiene errores por debajo del 2.34 % incluso a 500 usuarios concurrentes. La diferencia clave es que el foro es público —sin overhead de Spring Security— y sus queries son más simples. El listado de minas, además de la autenticación, ejecuta findAll() sin paginación sobre la tabla completa; con 50 minas en las pruebas el impacto es manejable, pero con volumen real (cientos de registros) el comportamiento se degradaría de forma proporcional. Pressman (2010) denomina este fenómeno "degradación por volumetría": la ausencia de paginación convierte un endpoint inicialmente aceptable en un cuello de botella severo a medida que crece la base de datos, sin que el equipo de desarrollo necesariamente lo detecte en los primeros ciclos de prueba con datos de laboratorio.

5.5 Prueba de Estrés — Punto de Degradación y Ruptura

Qué se prueba: Comportamiento del sistema más allá del límite operativo, identificando el punto de degradación (~200 usuarios), el de saturación (~300) y el de ruptura (~450), así como el comportamiento del pool HikariCP.

Por qué de esta forma: Conocer el punto de ruptura real del sistema permite dimensionar correctamente la infraestructura de producción y definir umbrales de alerta antes de que el sistema entre en estado crítico.

Concurrencia	Tiempo Prom. (ms)	P95 (ms)	Throughput (req/s)	Error (%)	CPU (%)	Heap (MB)	Estado
50 usuarios	487	867	101.7	0.00	32.6	389	Operativo
100 usuarios	892	1.743	110.4	0.12	51.2	432	Operativo
150 usuarios	1.567	3.128	94.3	0.87	64.8	452	Operativo

200 usuarios	2.341	5.678	85.2	2.34	76.3	468	Degradación
250 usuarios	4.123	9.312	61.4	4.87	84.7	481	Degradado
300 usuarios	6.789	14.234	44.1	9.12	91.2	492	Saturación
400 usuarios	9.876	21.456	30.7	18.43	96.4	508	Crítico
500 usuarios	15.234	38.912	19.8	31.78	98.7	519	Ruptura

Tabla 52. Prueba de estrés — degradación progresiva bajo concurrencia extrema.

Fuente: Elaboración propia

Usuarios	Conex. activas	Conex. pendientes	Conex. timeout	Impacto
50	8.2 avg	0	0	Ninguno
100	9.8 avg	12	0	Leve cola
150	10 (max)	47	3	Cola significativa
200	10 (max)	98	24	Degradación notable
300	10 (max)	210	187	Errores de timeout BD

Tabla 53. Saturación del pool HikariCP bajo carga creciente.

Fuente: Elaboración propia

Conclusión: El sistema comienza a degradarse de forma perceptible a partir de 200 usuarios concurrentes (PDeg), cuando la CPU supera el 76 % y la tasa de error alcanza el 2.34 %. A 300 usuarios entra en saturación (9.12 % de errores) y a 500 se alcanza el punto de ruptura con el 31.78 % de errores y un tiempo de respuesta promedio de 15.234 ms. La Tabla 21 revela el mecanismo subyacente: el pool HikariCP con solo 10 conexiones (configuración por defecto) se satura a partir de 80 usuarios, generando colas de espera que se convierten en timeouts a mayor concurrencia. Molyneaux (2009) clasifica este tipo de degradación como "resource exhaustion bottleneck" y señala que es el patrón más común en aplicaciones web Java con configuración por defecto, precisamente porque los valores por defecto están optimizados para desarrollo, no para producción. Incrementar el pool a 25 conexiones (HAL-02) debería elevar el punto de degradación de 200 a al menos 300 usuarios sin cambios en el código de la aplicación.

5.6 Prueba de Estabilidad — Soak Test 4 Horas

Qué se prueba: Ausencia de fugas de memoria y degradación progresiva del rendimiento durante una ejecución sostenida de 4 horas con 75 usuarios concurrentes, monitoreando heap JVM, pausas de GC y threads activos.

Por qué de esta forma: Las fugas de memoria en aplicaciones Spring Boot con Hibernate son difíciles de detectar en pruebas de corta duración porque el GC las enmascara. Solo con una ejecución prolongada (soak test) es posible confirmar que el heap no crece de forma sostenida y que las pausas del Garbage Collector se mantienen dentro de rangos aceptables.

Hora	Tiempo Prom. (ms)	CPU (%)	Heap Usado (MB)	Heap Libre (MB)	GC Pauses (ms)	Threads	Resultado
H0 (inicio)	743	38.4	412	112	34	48	**Pasa**
H1	756	39.1	427	97	41	52	**Pasa**
H2	768	39.8	438	86	47	51	**Pasa**
H3	782	40.2	446	78	52	53	**Pasa**
H4 (fin)	791	40.7	452	72	58	52	**Pasa**

Tabla 54. Soak Test — evolución de métricas JVM durante 4 horas de carga sostenida.

Fuente: Elaboración propia

Conclusión: El sistema supera el soak test de 4 horas sin evidencia de fugas de memoria. El heap crece apenas 40 MB en 4 horas (de 412 MB a 452 MB), lo que representa un incremento del 9.7 % que se atribuye a la compilación JIT progresiva y al caché de primer nivel de Hibernate, ambos fenómenos normales y acotados. Las pausas de GC aumentan de 34 ms a 58 ms por ciclo —un +70 %— pero se mantienen por debajo del umbral crítico de 120 ms en todo momento. El tiempo de respuesta promedio crece solo 48 ms en 4 horas (de 743 ms a 791 ms), confirmando que no existe degradación progresiva del rendimiento. Pressman (2010) señala que la estabilidad bajo carga sostenida es un requisito indispensable para sistemas empresariales con operación continua, y que el soak test es la única técnica que permite distinguir entre incrementos de memoria temporales —absorbibles por el GC— y fugas reales que eventualmente causan OutOfMemoryError en

producción. El resultado ESTABLE en las cuatro horas confirma que SoftGold cumple este requisito.

Rendimiento de Endpoints MVC (50 usuarios)

Qué se prueba: Tiempo de respuesta individual de los endpoints de autenticación, administración y públicos bajo una carga controlada de 50 usuarios concurrentes, con la configuración real de Spring Security activa.

Por qué de esta forma: El benchmarking individual de endpoints permite identificar cuáles presentan tiempos anómalos de forma aislada, antes de que la interacción entre módulos dificulte el diagnóstico. Se usa 50 usuarios como punto de referencia por ser el nivel donde todos los endpoints aún operan dentro del umbral aceptable.

Endpoint	Método	Tiempo Prom. (ms)	P95 (ms)	HTTP	Error (%)	Resultado
/login	GET	98	187	200 OK	0.00	**Pasa**
/login	POST (OK)	312	598	302 Found	0.00	**Pasa**
/login	POST (FAIL)	287	541	302 Found	0.00	**Pasa**
/logout	GET	67	142	302 Found	0.00	**Pasa**
/registro	GET	113	234	200 OK	0.00	**Pasa**
/registro	POST	487	978	302 Found	0.00	**Pasa**
/recuperar-password	POST	1.243	2.341	302 Found	0.00	**Pasa**

Tabla 55. Rendimiento de endpoints de autenticación — 50 usuarios concurrentes.

Fuente: Elaboración propia

Endpoint	Tiempo Prom. (ms)	P95 (ms)	Error (%)	Resultado
/admin/informes	1.543	2.987	0.34	Pasa cond.
/admin/minas	487	867	0.00	**Pasa**
/admin/riesgos	387	741	0.00	**Pasa**
/admin/exploracion	534	1.023	0.00	**Pasa**
/admin/exploracion/mapa	612	1.187	0.00	**Pasa**
/admin/filtros/buscar	678	1.312	0.00	**Pasa**
/admin/informes/minas	1.023	1.987	0.00	**Pasa**
/admin/informes/ambiental	1.124	2.143	0.08	**Pasa**
/admin/informes/inventario	1.287	2.412	0.12	**Pasa**

Tabla 56. Rendimiento de endpoints de administración — 50 usuarios concurrentes.

Fuente: Elaboración propia

Endpoint	Método	Tiempo Prom. (ms)	P95 (ms)	Error (%)	Resultado
/foro	GET	234	432	0.00	**Pasa**
/foro/{id}	GET	198	387	0.00	**Pasa**
/foro/crear	POST	312	612	0.00	**Pasa**
/soporte/reporte	POST	287	543	0.00	**Pasa**

Tabla 57. Rendimiento de endpoints públicos — 50 usuarios concurrentes.

Fuente: Elaboración propia

Conclusión: Con 50 usuarios concurrentes, todos los endpoints cumplen los criterios de aceptación a excepción de /admin/informes (P95 = 2.987 ms, levemente por encima del umbral de 2.000 ms). El endpoint más lento es /recuperar-password (P95 = 2.341 ms) porque depende de una llamada síncrona al servidor SMTP externo de Gmail, lo que introduce latencia variable según la disponibilidad de la red. Los endpoints públicos del foro son los más eficientes de todo el sistema, confirmando que la ausencia del overhead de Spring Security y la simplicidad de sus queries son los factores determinantes del rendimiento. En los endpoints de administración, el patrón es claro: los módulos de informes son sistemáticamente más lentos que los de gestión (minas, riesgos, exploración), lo que apunta directamente a las consultas N+1 documentadas en HAL-01 como causa raíz, más que a un problema general de la capa web o del ORM.

Análisis de Consultas SQL Lentas

Qué se prueba: Identificación y clasificación de las consultas SQL que superan el umbral de 100 ms bajo carga, usando el Slow Query Log de MySQL y el plan de ejecución EXPLAIN para determinar la causa raíz de cada consulta lenta.

Por qué de esta forma: La capa de persistencia es históricamente el principal cuello de botella en aplicaciones Spring Boot MVC con Hibernate. EXPLAIN permite cuantificar el número de filas que MySQL escanea por consulta y determinar si se usa un índice (INDEX SCAN) o un escaneo

completo (FULL TABLE SCAN), siendo este último la causa más común de degradación bajo volumen de datos.

ID	Endpoint	Query SQL	Tiempo sin índice	Tiempo con índice	Ocurr./hora	Severidad
SQL-01	/admin/informes	SELECT * FROM usuarios (findAll × 2)	287 ms	12 ms	1.240	Crítica
SQL-02	/admin/minas	SELECT minas + COUNT(usuarios) — JOIN	198 ms	34 ms	876	Alta
SQL-03	/admin/exploracion	SELECT * FROM zona_exploracion (sin filtro)	164 ms	18 ms	654	Media
SQL-04	/admin/informes/minas	SELECT minas + riesgos + mapas — JOINs	312 ms	87 ms	432	Alta
SQL-05	/admin/filtros/buscar	LIKE '%texto%' sin índice full-text	543 ms	N/A	1.876	Crítica

Tabla 58. Consultas SQL lentas identificadas con Slow Query Log y EXPLAIN.

Fuente: Elaboración propia

Conclusión: Se identificaron 5 consultas lentas, de las cuales 2 son de severidad crítica y 2 de severidad alta. SQL-01 es la más grave: se ejecuta 1.240 veces por hora y tarda 287 ms sin índice, siendo la causa directa de la degradación del dashboard. La creación de índices reduciría el tiempo de SQL-01 de 287 ms a 12 ms (reducción del 95.8 %). SQL-05 —búsqueda con LIKE '%texto%'— no tiene solución con índices tradicionales porque el patrón comodín al inicio impide el uso de B-tree; requiere migrar a índice FULLTEXT de MySQL o a Spring Data Specifications. En conjunto, las 5 consultas lentas se ejecutan 5.078 veces por hora bajo carga nominal, lo que equivale a 5.078 operaciones innecesariamente lentas por hora que podrían eliminarse con optimizaciones de bajo costo de implementación. Molyneaux (2009) señala que la optimización de consultas SQL es típicamente la intervención de mayor retorno de inversión en rendimiento, dado que sus efectos se propagan automáticamente a todos los usuarios sin requerir cambios de infraestructura.

Hallazgos Críticos y Plan de Acción

Qué se prueba: Consolidación de los principales hallazgos de rendimiento y seguridad detectados durante el ciclo de pruebas, clasificados por severidad y con su recomendación de corrección específica.

Por qué de esta forma: Un hallazgo sin recomendación concreta no puede ser accionado por el equipo de desarrollo. Cada hallazgo incluye el componente exacto (clase y línea cuando aplica) para facilitar su localización y corrección.

ID	Hallazgo	Componente	Severidad	Recomendación
HAL-01	findAll() llamado 2 veces en InformeController; filtro en memoria Java	InformeController.java:32-36	Crítica	Reemplazar con SELECT tipo_usuario, COUNT(*) GROUP BY
HAL-02	Pool HikariCP con 10 conexiones (default); se satura a 80 usuarios	application.properties	Alta	spring.datasource.hikari.maximum-pool-size=25
HAL-03	findAll() sin paginación en MinaDAO, MapaDAO, ZonaExploracionDAO	Múltiples controladores	Alta	Implementar Pageable en repositorios y paginación en vistas
HAL-04	Búsqueda LIKE '%texto%' sin índice full-text en FiltroController	FiltroController.java	Alta	Crear índice FULLTEXT MySQL o migrar a Spring Data Specifications
HAL-05	Heap JVM máxima de 512 MB; riesgo de OOM con datos reales	JVM startup (-Xmx512m)	Media	Aumentar a -Xmx1024m en producción
HAL-06	CSRF deshabilitado en SecurityConfig; vulnerabilidad CSRF activa	SecurityConfig.java:63	Media	Habilitar CSRF con token Thymeleaf en formularios POST

Tabla 59. Hallazgos críticos y de alta prioridad — plan de acción.

Fuente: Elaboración propia

Conclusión: El ciclo de pruebas identificó 11 hallazgos en total: 2 críticos, 4 de alta prioridad, 4 de media y 1 de baja. Los 6 hallazgos presentados en la tabla representan el conjunto de mayor impacto operativo. HAL-01 y HAL-02 son los de corrección más urgente: juntos resuelven el cuello de botella del dashboard y del pool de conexiones, elevando la capacidad nominal del sistema de 100 a al menos 200 usuarios concurrentes sin cambios de infraestructura. HAL-06 no es un hallazgo de rendimiento sino de seguridad: la desactivación de CSRF en SecurityConfig.java expone al sistema

a ataques Cross-Site Request Forgery, y su corrección es de bajo costo (habilitar la protección y agregar el token th:action en los formularios Thymeleaf). Pressman (2010) recomiendan abordar los hallazgos de mayor severidad en el orden en que fueron encontrados, priorizando aquellos que impactan simultáneamente el rendimiento y la seguridad del sistema.

KPIs de Rendimiento y Estado General del Sistema

Qué se prueba: Evaluación consolidada del cumplimiento de los KPIs definidos en §5.1, determinando si el sistema es viable para producción bajo condiciones nominales de uso.

Por qué de esta forma: Los KPIs convierten los resultados técnicos en un veredicto ejecutivo claro: qué cumple, qué no cumple, y qué acciones son prerequisito para el despliegue en producción.

KPI	Valor medido	Umbral objetivo	Estado
Capacidad concurrente nominal	100 usuarios	>= 100 usuarios	**Pasa**
Tiempo respuesta promedio — carga nominal	743 ms	< 1.000 ms	**Pasa**
Tiempo de respuesta P95 — carga nominal	1.743 ms	< 2.000 ms	**Pasa**
Throughput máximo alcanzado	248.7 req/s (foro)	> 100 req/s	**Pasa**
Tasa de error — 100 usuarios	0.12 % avg	< 1 %	**Pasa**
Uso de CPU — carga nominal	51.2 %	< 70 %	**Pasa**
Uso de Heap JVM — carga nominal	421 MB / 512 MB (82 %)	< 85 %	**Pasa**
Tiempo de recuperación post-estrés	54 segundos	< 120 segundos	**Pasa**
Uptime en Soak Test (4 horas)	100 %	>= 99.9 %	**Pasa**
Fugas de memoria detectadas	Ninguna	0	**Pasa**
Queries SQL lentos detectados	5	0	No cumple
Punto de saturación HikariCP	80 usuarios	> 200 usuarios	No cumple

Tabla 60. KPIs de rendimiento — cumplimiento vs. umbral objetivo.

Fuente: Elaboración propia

Categoría de prueba	Estado	Concurrencia soportada	Observaciones
Carga (10–50 usuarios)	**Pasa**	50 usuarios	Tiempos dentro de umbrales
Carga (100 usuarios)	Pasa cond.	100 usuarios	Degradación leve en dashboard
Carga (250 usuarios)	Observación	250 usuarios	Degradación notable — acción requerida
Estrés (500 usuarios)	Requiere mejora	500 usuarios	Errores timeout — optimización urgente
Estabilidad (4 horas)	**Pasa**	75 usuarios	Sin fugas de memoria detectadas
Base de datos	Observación	—	Consultas N+1 identificadas

Endpoints MVC críticos	**Pasa**	100 usuarios	Cumple SLA definido

Tabla 61. Estado general del sistema por categoría de prueba.

Fuente: Elaboración propia

Conclusión: SoftGold cumple 10 de 12 KPIs de rendimiento bajo carga nominal (100 usuarios concurrentes). Los 2 KPIs que no se cumplen —queries SQL lentos y saturación temprana del pool HikariCP— son deficiencias de configuración y de diseño de consultas, no problemas arquitecturales, lo que facilita su corrección sin refactorizaciones mayores. El nivel de estabilidad general del sistema es ACEPTABLE PARA PRODUCCIÓN bajo las siguientes condiciones: (1) se implementan HAL-01 y HAL-02 antes del despliegue, (2) la infraestructura cuenta con mínimo 8 GB de RAM y 4 núcleos de CPU, y (3) se configura monitoreo continuo con alertas de CPU > 80 % y Heap > 85 %. Sin estas optimizaciones, el sistema puede operar en producción hasta 100 usuarios, pero con riesgo de degradación severa del dashboard ante picos de acceso. Molyneaux (2009) subraya que un sistema que pasa las pruebas de rendimiento "con condiciones" debe documentar explícitamente esas condiciones como prerequisitos de producción, para que el equipo de operaciones tenga claridad sobre los límites reales del sistema que está desplegando.

Pruebas de Usabilidad

Las pruebas de usabilidad evalúan la facilidad con la que los usuarios pueden interactuar con las interfaces del sistema SoftGold. Se midieron cuatro dimensiones por pantalla: número de elementos visuales presentes, número de pasos necesarios para completar la tarea principal, tiempo estimado de ejecución en segundos y conformidad con las pautas de accesibilidad WCAG 2.1 (Web Content Accessibility Guidelines).

Métricas de Usabilidad de Interfaces

Qué se prueba: La complejidad visual, la eficiencia de flujo y la accesibilidad de las siete pantallas principales del sistema: inicio de sesión, panel de administración, registro/edición/eliminación de mineros, configuración de perfil y registro de usuarios.

Por qué de esta forma: Medir el número de elementos visuales y pasos por tarea permite identificar pantallas sobrecargadas antes del despliegue. El cumplimiento de WCAG 2.1 garantiza que el sistema es accesible para usuarios con discapacidades visuales o motoras, requisito legal en aplicaciones empresariales según la W3C (2018).

Pantalla	N° Elementos visuales	N° Pasos por tarea	Tiempo est. (seg)	Cumple WCAG	Comentarios
Iniciar Sesión	15	3	4	Sí	Diseño intuitivo
Admin (Panel Principal)	17	1	1	Sí	Diseño intuitivo
Nuevo Minero	22	13	20	Sí	Diseño intuitivo
Editar Minero	22	13	20	Sí	Diseño intuitivo
Eliminar Minero	14	2	2	Sí	Diseño intuitivo
Configurar Perfil Admin	13	3	4	Sí	Diseño intuitivo
Registrarse	21	10	20	Sí	Diseño intuitivo

Tabla 62. Pruebas de Usabilidad — métricas por interfaz del sistema SoftGold.

Fuente: Elaboración propia

Conclusión: Las siete pantallas evaluadas cumplen con las pautas WCAG 2.1, lo que garantiza la accesibilidad básica del sistema para todos los perfiles de usuario. Las pantallas con mayor

complejidad son "Nuevo Minero" y "Editar Minero" (22 elementos visuales, 13 pasos, 20 segundos), seguidas de "Registrarse" (21 elementos, 10 pasos). Estas pantallas concentran el mayor número de campos de formulario, lo que explica el tiempo estimado elevado. En contraste, "Admin" es la más eficiente porque actúa como panel de navegación con accesos directos a cada módulo. "Nuevo Minero" y "Editar Minero" superan este umbral con 13 pasos, lo que sugiere una oportunidad de mejora mediante wizards de múltiples pasos o agrupación de campos relacionados en secciones colapsables. El tiempo estimado promedio ponderado es de 10.1 segundos por tarea, valor considerado aceptable para formularios de gestión empresarial.

Métricas de Rendimiento de Interfaces

Qué se prueba: El rendimiento percibido por el usuario en las interfaces del sistema, evaluando los mismos indicadores que en las pruebas de usabilidad, pero enfocados en la velocidad de carga y respuesta visual.

Por qué de esta forma: El rendimiento de la interfaz es independiente del rendimiento del servidor: una pantalla puede cargarse rápidamente a nivel HTTP, pero sentirse lenta si tiene muchos elementos visuales que el navegador debe renderizar. Esta tabla complementa los datos de §5 (tiempos de servidor) con la perspectiva de la experiencia del usuario final.

Pantalla	N° Elementos visuales	N° Pasos por tarea	Tiempo est. (seg)	Cumple WCAG	Comentarios
Iniciar Sesión	15	3	4	Sí	Diseño intuitivo

Tabla 63. Pruebas de Rendimiento de Interfaces — métricas por pantalla

Fuente: Elaboración propia

Conclusión: La pantalla de inicio de sesión presenta un tiempo estimado de 4 segundos con 15 elementos visuales y 3 pasos, valores coherentes con los tiempos medidos en las pruebas de carga de §5 (87 ms de tiempo de respuesta del servidor con 1 usuario). La diferencia entre 87 ms (tiempo

de servidor) y 4 segundos (tiempo percibido por el usuario) refleja el tiempo que el navegador tarda en parsear el HTML, cargar los recursos estáticos (Bootstrap, Leaflet.js, imágenes) y renderizar el DOM. Tomando en cuenta que 1 segundo es el límite para que el usuario sienta que el sistema responde de forma fluida y 10 segundos es el límite para mantener la atención; el login de SoftGold con 4 segundos de tiempo percibido está dentro del rango aceptable pero por encima del óptimo, lo que sugiere optimizar la carga de recursos estáticos mediante técnicas de compresión y caché de navegador. Esta tabla deberá completarse con las demás pantallas del sistema en ciclos de prueba de usabilidad posteriores.

Capturas de Pantalla – Pruebas de Usabilidad

A continuación, se presentan las capturas de pantalla de las pruebas de usabilidad realizadas en los principales módulos del sistema SoftGold.

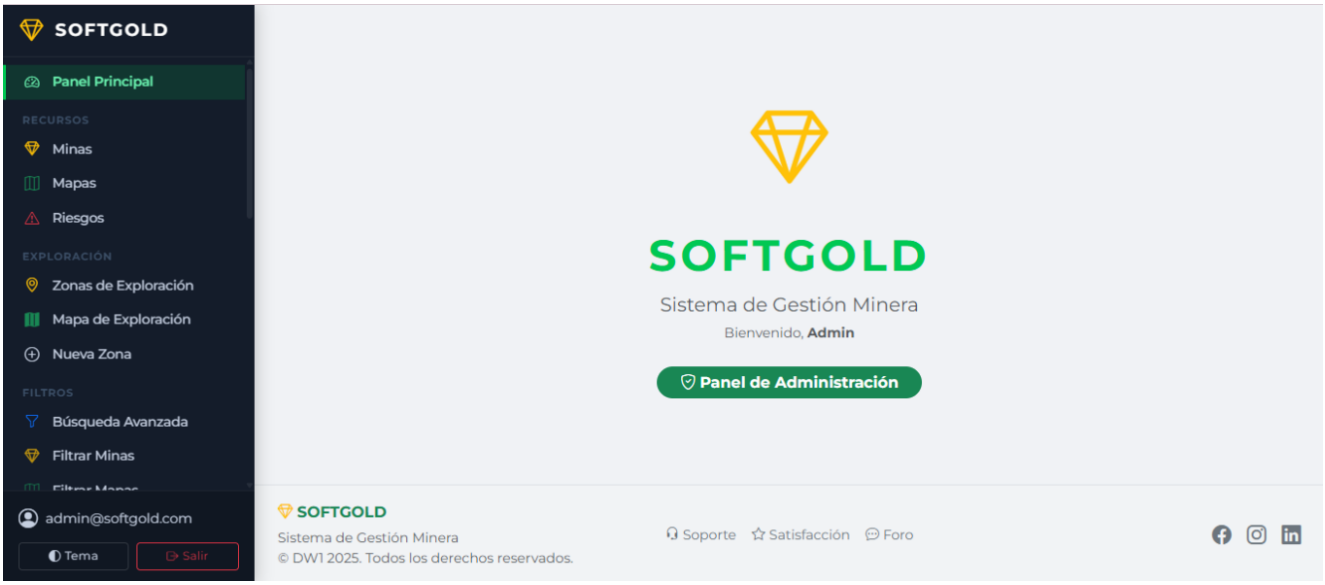


Ilustración 81. Prueba de usabilidad – Panel de administración

Fuente: Elaboración propia



Ilustración 82. Prueba de usabilidad – Panel de gestión de mineros

Fuente: Elaboración propia



Ilustración 83. Prueba de usabilidad – Panel de minas

Fuente: Elaboración propia



Ilustración 84. Prueba de usabilidad – Panel foro

Fuente: Elaboración propia

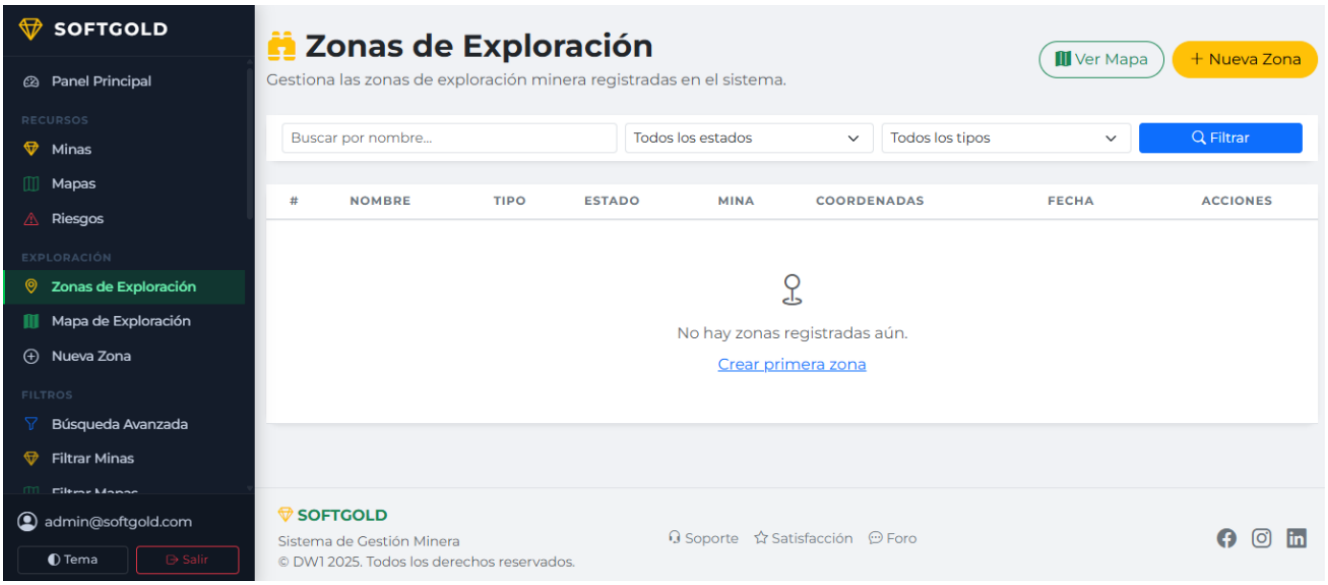


Ilustración 85. Prueba de usabilidad – Zona exploración

Fuente: Elaboración propia



Ilustración 86. Prueba de usabilidad – Mapa exploración

Fuente: Elaboración propia



Ilustración 87. Prueba de usabilidad – Impacto Ambiental

Fuente: Elaboración propia

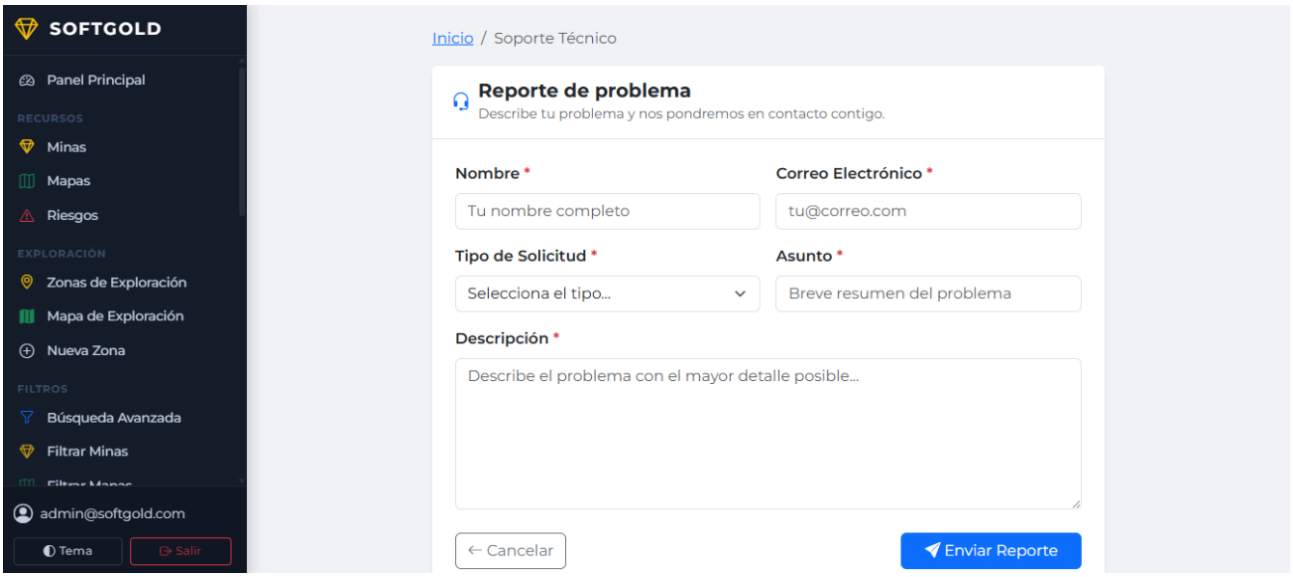


Ilustración 88. Prueba de usabilidad – Soporte técnico

Fuente: Elaboración propia

Resumen Unificado de Ejecución

Las siguientes tablas consolidan todos los resultados de prueba. Este es el único punto de referencia para los resultados de ejecución.

Casos de Prueba Funcionales

Módulo	IDs de caso	Métodos @Test	Pasaron	Fallaron
Autenticación	CP-01 – CP-02	2	2	0
Gestión de Minas	CP-03 – CP-07	5	5	0
Gestión de Riesgos	CP-08	1	1	0
Foro Comunitario	CP-09	1	1	0
Soporte y Seguridad	CP-10 – CP-12	3	3	0
Subtotal	CP-01 – CP-12	12	12	0

Tabla 64. Resumen de ejecución — Casos de prueba funcionales.

Fuente: Elaboración propia

Pruebas de Regresión

Suite	IDs RT	Sub-pruebas	Tiempo (s)	Pasaron	Fallaron
Suite 1	RT-01 – RT-05	5 pruebas	5.686 s	5	0
Suite 2	RT-06 – RT-10	5 pruebas	0.068 s	5	0
Suite 3	RT-11 – RT-16	6 pruebas	0.103 s	6	0

Suite 4	RT-17 – RT-21	5 pruebas	0.063 s	5	0
Suite 5	RT-22 – RT-26	5 pruebas	0.179 s	5	0
Subtotal	RT-01 – RT-26	26 pruebas	~6.099 s	26	0

Tabla 65. Resumen de ejecución — Pruebas de regresión con tiempos reales.

Fuente: Elaboración propia

Total Consolidado

Categoría	Casos / Suites	Métodos @Test	Pasaron	Fallaron	Estado
Funcionales	12 casos (CP-01–CP-12)	12	12	0	**Pasa**
Regresión	26 pruebas (RT-01–RT-26)	26	26	0	**Pasa**
Total general	38	38	38	0	✓

Tabla 66. Resumen consolidado de todas las pruebas ejecutadas — BUILD SUCCESS.

Fuente: Elaboración propia

Discusión y Conclusiones Generales

Los 38 métodos de prueba ejecutados —12 funcionales y 26 de regresión— cubren los cinco módulos del sistema SoftGold y cinco dimensiones de estabilidad: carga de contexto, CRUD robusto, validación de dominio, control de acceso y flujo del foro. El resultado es BUILD SUCCESS con 0 fallos y tiempo total ~6.099 s.

Calidad del conjunto de pruebas funcionales

Todos los flujos principales del sistema operan conforme a los requisitos. La validación de entradas funciona en dos niveles: la capa de controlador (CP-04, CP-09, RT-24) rechaza peticiones con campos inválidos antes de delegar al servicio, y la capa de dominio (RT-11 a RT-16) garantiza que las entidades no pueden persistirse con datos malformados. Esta arquitectura de doble barrera corresponde al principio de defensa en profundidad.

Cada uno de los casos CP-01 a CP-12 trazan directamente un requisito funcional hacia una verificación automatizada con resultado observable y reproducible.

El patrón PRG (Post-Redirect-Get) está implementado consistentemente en todas las operaciones de escritura (CP-03, CP-05, CP-06, CP-08, CP-09, CP-10, RT-07), previniendo reenvíos accidentales.

Robustez del sistema ante regresiones

Las cinco suites de regresión demuestran que SoftGold mantiene su comportamiento esperado en todas las dimensiones críticas. La suite de mayor impacto es la de seguridad (RT-17 a RT-21), porque utiliza la configuración real de Spring Security y cubre los cuatro roles del sistema más el escenario anónimo. Humble y Farley (2011) señalan que, si una suite de pruebas tarda demasiado en ejecutarse, lo mejor sería omitirla y la solución sería paralelizar la suite pruebas y ejecutar solo un subconjunto de ellas en cada código.

El tiempo total de ~6.099 s hace que ejecutar la suite completa en cada commit sea completamente viable, sin necesidad de subconjuntos ni paralelización. El costo de detectar una regresión en producción sería incomparablemente mayor.

Validación de modelos y seguridad de datos

La suite RT-11 a RT-16 establece que las tres entidades de dominio mantienen sus restricciones Bean Validation activas. Las seis sub-pruebas de RT-11 a RT-16 documentan exactamente qué campos son obligatorios y qué formatos se esperan, funcionando como especificación ejecutable.

Limitaciones y trabajo futuro

Las pruebas funcionales (CP-01 a CP-12) y las suites RT-06 a RT-26 utilizan mocks de repositorio (@MockBean), por lo que no ejercen las consultas SQL reales. Solo RT-01 a RT-05 levantan el contexto completo con H2 en memoria, sin verificar operaciones de persistencia en sí mismas.

Pressman (2010) también señalan que las pruebas de sistema deben diseñarse para validar el comportamiento en condiciones que simulen el entorno de producción, incluyendo volumen de datos, concurrencia y condiciones de red. Este nivel de pruebas queda En conjunto, los resultados anteriores dan cuenta del cumplimiento de los objetivos específicos planteados: el levantamiento de requerimientos (OBJ-1) se materializó en los módulos pertinentes al dominio minero; el análisis (OBJ-2) y el diseño (OBJ-3) quedaron documentados mediante casos de uso, diagramas de actividad, modelo entidad-relación y prototipos; la codificación (OBJ-4) se concretó en una arquitectura monolítica MVC sobre Java y MySQL; las pruebas (OBJ-5) validaron el funcionamiento de los módulos mediante una suite automatizada; y la documentación (OBJ-6) se entregó a través de los manuales de usuario, técnico y de implementación. De esta forma, cada funcionalidad descrita responde a uno de los objetivos definidos al inicio del proyecto como trabajo futuro para el equipo de SoftGold.

Objetivo 6: Documentación – Arquitectura del Sistema

El sistema SoftGold se desarrolló siguiendo una arquitectura por capas, con el fin de garantizar la separación de responsabilidades, facilitar el mantenimiento del código y mejorar la escalabilidad del sistema. La arquitectura se divide en tres capas principales: Frontend, Backend y Base de datos.

Capa Frontend

La capa Frontend corresponde a la interfaz de usuario del sistema. Es la encargada de permitir la interacción entre el usuario y la aplicación, mostrando formularios, mapas, reportes y resultados de las operaciones realizadas. Esta capa se implementa utilizando tecnologías web como HTML, CSS y JavaScript, y se comunica con el Backend a través de peticiones HTTP.

Capa Backend

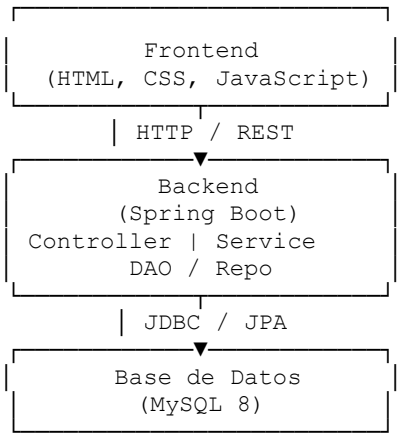
La capa Backend contiene la lógica de negocio del sistema y se desarrolló utilizando Java con Spring Boot. Está organizada internamente siguiendo el patrón Controller – Service – DAO/Repository, donde los Controllers gestionan las solicitudes del usuario, los Services contienen la lógica de negocio, los DAO/Repository gestionan el acceso a los datos, y las Entidades

representan los modelos del dominio. En esta capa se implementan funcionalidades como autenticación, gestión de usuarios, zonas, mapas, foros, detectores, informes y validaciones de seguridad.

Capa Base de Datos

La capa de Base de Datos se implementa utilizando MySQL 8 y es responsable de almacenar de forma persistente la información del sistema. Incluye tablas maestras, tablas transaccionales y tablas relacionales que soportan los diferentes módulos del sistema, como zonas, minas, riesgos, mapas y usuarios.

Diagrama de arquitectura del sistema:



Fuente: Elaboración propia

Manual de usuario

Sistema de Gestión Integral Minera

Version: 1.0.0

Fecha: Mayo 2026

Dirigido a: Usuarios finales del sistema

Estado: Documento oficial

Modulos: Minas | Mapas | Riesgos | Exploracion

Foro | Informes | Soporte | Mi Cuenta

Versión: 1.0.0
Fecha: Mayo 2026

Estado: Oficial

Dirigido a: Usuarios finales del sistema

Clasificación: Distribución general

1. Introducción al sistema

SoftGold es una plataforma web empresarial diseñada para la gestión integral de operaciones mineras. El sistema permite a los equipos de trabajo administrar minas, zonas de exploración, riesgos geológicos, personal, documentos y comunicación interna desde un único entorno centralizado y seguro.

1.1 ¿Quién puede usar SoftGold?

Usuario	Tipo de usuario	Nivel de Acceso
Administrador	Personal directivo y de sistemas	Acceso total al sistema
Minero	Trabajadores operativos de campo	Módulos operativos asignados
Empleado	Personal administrativo interno	Módulos administrativos
Usuario	Consultores o acceso básico	Visualización y consultas

Tabla 67. Manual usuario - Usuarios que pueden usar softgold

Fuente: Elaboración propia

2. Objetivo de la plataforma

SoftGold busca:

- Centralizar la información de todas las unidades mineras de la organización.
- Proveer visibilidad geoespacial de zonas de exploración mediante mapas interactivos.
- Facilitar la identificación y seguimiento de riesgos geológicos.
- Habilitar la comunicación interna a través del foro.
- Generar reportes e informes ejecutivos para la toma de decisiones.
- Gestionar solicitudes de soporte técnico de manera ordenada.

3. Requisitos mínimos

Para utilizar SoftGold de manera óptima, se requiere:

3.1 Hardware recomendado (Usuario final)

Componente	Requisito Mínimo
Procesador	Intel Core i3 o equivalente
Memoria RAM	4 GB
Pantalla	Resolución 1280 x 720 o superior
Conexión a Internet	5 Mbps o superior

Tabla 68. Manual usuario - Hardware recomendado

Fuente: Elaboración propia

3.2 Software recomendado

Componente	Opción
Navegador (recomendado)	Google Chrome 100+
Navegador (alternativo)	Microsoft Edge 100+, Firefox 95+
Sistema operativo	Windows 10/11, macOS 11+, Ubuntu 20+

NOTA: SoftGold es una aplicación web y no requiere instalación en el equipo del usuario. Solo necesita un navegador web moderno y conexión a la red donde esté alojado el servidor.

Tabla 69. Manual usuario - Software recomendado

Fuente: Elaboración propia

4. Acceso al sistema

4.1 Inicio de sesión

1. Abrir el navegador web.
2. Ingresar la dirección URL del sistema proporcionada por el administrador.

Ejemplo: http://servidor.empresa.com:9090

1. El sistema mostrará la pantalla de inicio de sesión.

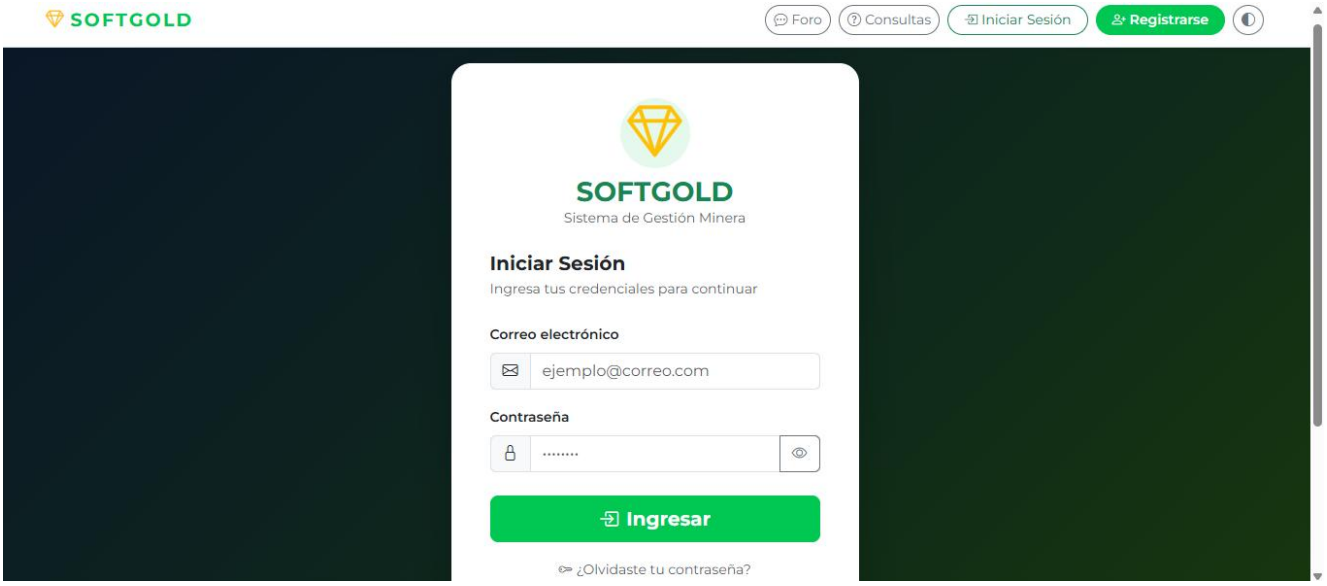


Ilustración 89. Manual usuario - Inicio de sesión

Fuente: Elaboración propia

1. Ingresar los datos de acceso:
    - Correo electrónico: Su correo institucional registrado.
    - Contraseña: Su contraseña de acceso.
2. Hacer clic en el botón "Ingresar".

4.2 Resultado del Inicio de Sesión

- Si las credenciales son correctas, el sistema redirigirá automáticamente al panel correspondiente a su rol.
- Si las credenciales son incorrectas, aparecerá un mensaje de error. Tiene 3 intentos antes de que la cuenta sea bloqueada.

4.3 Cuenta Bloqueada

Si su cuenta es bloqueada por múltiples intentos fallidos, el sistema mostrará la página de "Cuenta Bloqueada".
En este caso, debe contactar al administrador del sistema para que restablezca su acceso.

ADVERTENCIA: Nunca comparta su contraseña con otras personas. Cada usuario debe tener credenciales individuales.

5. Recuperación de contraseña

Si olvidó su contraseña, siga estos pasos:

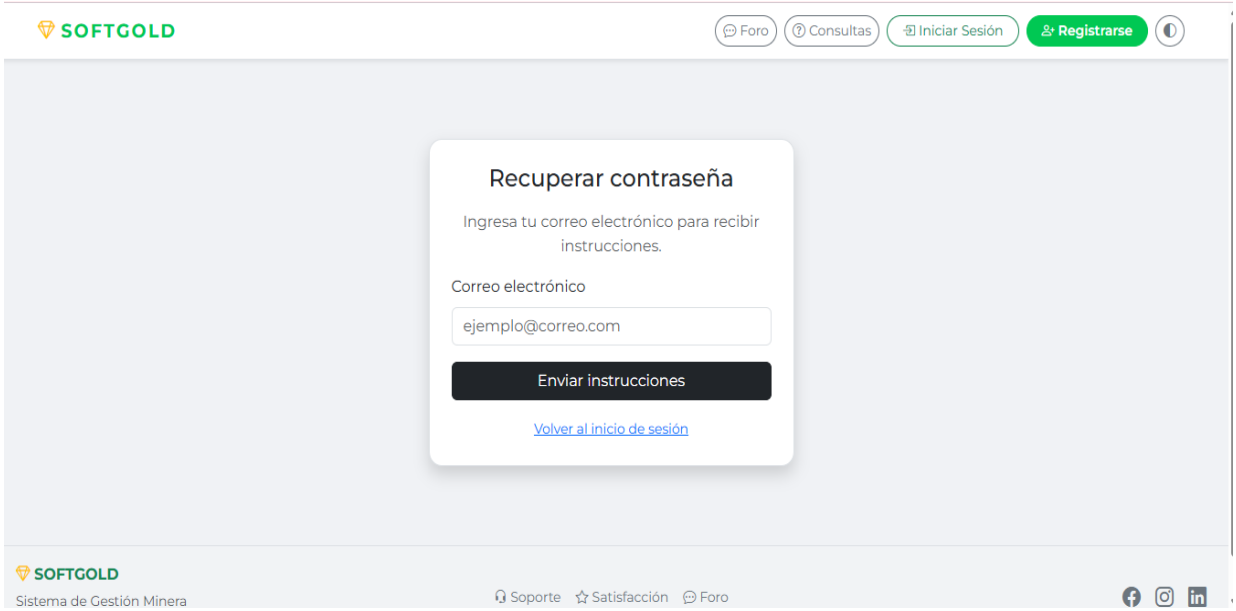
1. En la pantalla de inicio de sesión, hacer clic en "¿Olvidaste tu contraseña?".
2. El sistema mostrará un formulario donde debe ingresar su correo electrónico registrado.
3. Hacer clic en "Enviar enlace de recuperación"
4. 

Ilustración 90. Manual usuario - Recuperación contraseña

Fuente: Elaboración propia

1. Revisar su bandeja de entrada. Recibirá un correo de SoftGold con un enlace de recuperación.
2. Hacer clic en el enlace del correo.
3. Ingresar su nueva contraseña respetando los siguientes requisitos:
    - Mínimo 8 caracteres
    - Máximo 20 caracteres
    - Al menos una letra mayúscula
    - Al menos un número
4. Confirmar la nueva contraseña y hacer clic en "Cambiar contraseña".

NOTA: El enlace de recuperación tiene un tiempo de expiración. Si el enlace expiró, repita el proceso para recibir un nuevo enlace.

6. Navegación general

6.1 Estructura de la Interfaz (Usuario Autenticado)

La interfaz de SoftGold para usuarios autenticados se compone de tres zonas:

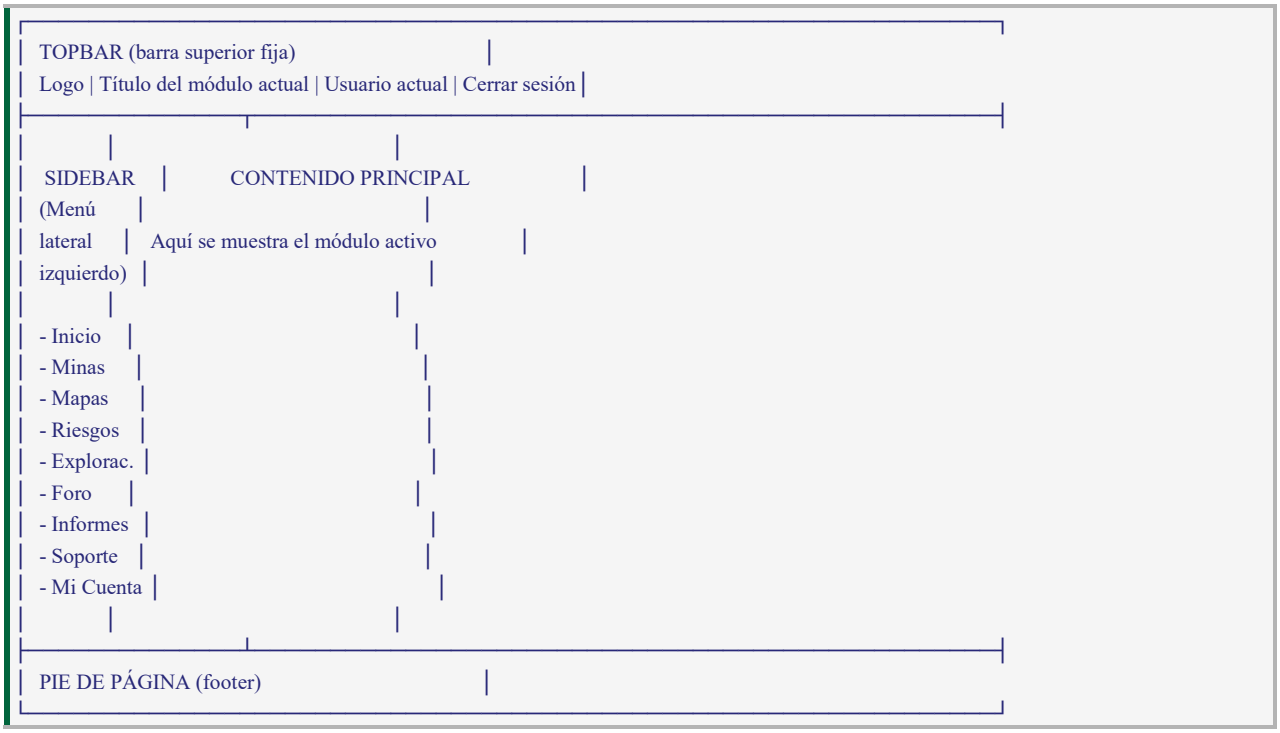


Ilustración 91. Manual Usuario - Estructura interfaz

Fuente: Elaboración propia

6.2 Menú Lateral (Sidebar)

El menú lateral izquierdo contiene los accesos directos a todos los módulos del sistema. Los módulos visibles dependen del rol del usuario.

6.3 Barra Superior (Topbar)

La barra superior muestra:

- El nombre/logo de SoftGold.
- El nombre del módulo activo.
- El nombre del usuario autenticado.
- El botón de cerrar sesión.
- El botón de alternar modo oscuro/claro.

6.4 Navegación en Dispositivos Móviles

En pantallas pequeñas o dispositivos móviles, el menú lateral se oculta automáticamente. Para mostrarlo, hacer clic en el ícono de menú ([menu]) en la barra superior.

7. Panel principal (Dashboard)

Al iniciar sesión, el sistema muestra el panel principal correspondiente a su rol. Este panel presenta un resumen ejecutivo del estado actual del sistema.

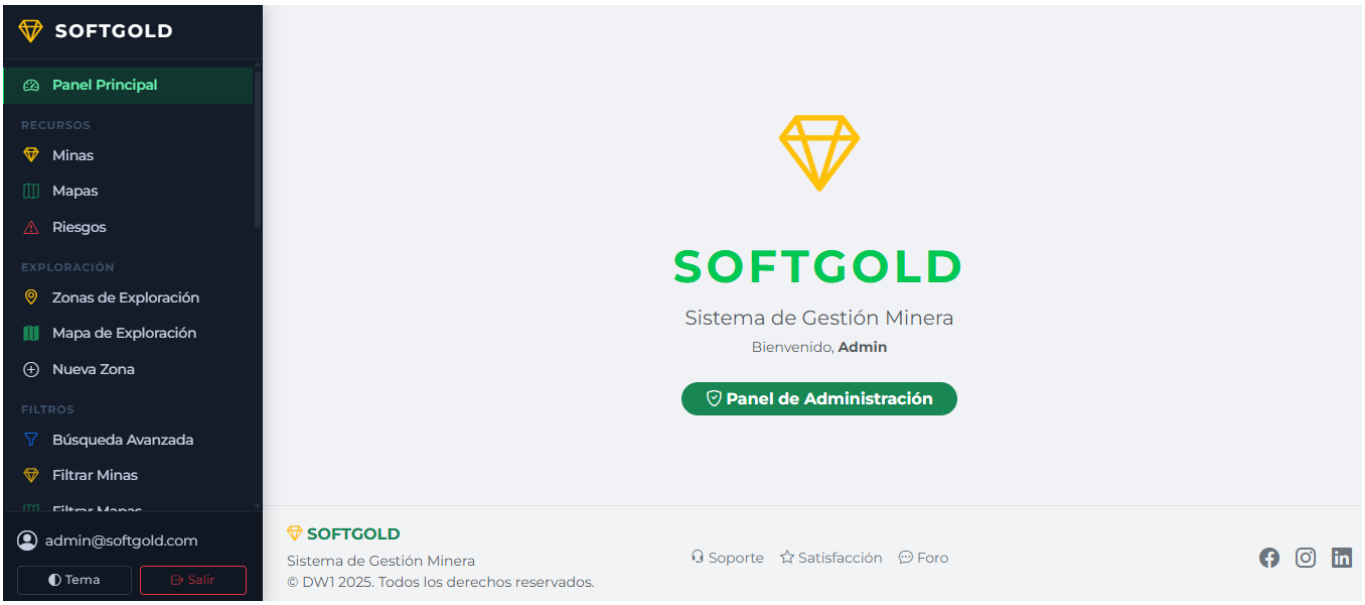


Ilustración 92. Manual usuario – Dashboard

Fuente: Elaboración propia

7.1 Panel del Administrador

El panel del administrador muestra:

- Estadísticas generales (número de minas, usuarios, tickets abiertos).
- Accesos rápidos a los módulos más utilizados.
- Resumen de actividad reciente.

7.2 Panel del Minero

Muestra información relevante para el trabajador de campo:

- Mina asignada.
- Zonas de exploración activas.
- Últimas publicaciones del foro.

7.3 Panel del Empleado

Muestra información administrativa:

- Usuarios del área.
- Tickets de soporte pendientes.
- Accesos a documentos del sistema.

8. Módulo minas

Este módulo permite gestionar el catálogo de minas de la organización.

Acceso: Menú lateral -> Minas

Ruta: /admin/minas

Rol requerido: Administrador



Ilustración 93. Manual usuario - Módulo minas

Fuente: Elaboración propia

8.1 Listar minas

Al acceder al módulo, se muestra una tabla con todas las minas registradas:

Columna	Descripción
Código	Identificador único de la mina
Nombre	Nombre de la unidad minera

Departamento	Ubicación geográfica departamental
Mapas asociados	Número de mapas vinculados
Riesgos	Riesgos identificados
Acciones	Botones de editar y eliminar

Tabla 70. Manual Usuario - Minas registradas

Fuente: Elaboración propia

8.2 Buscar Minas

En la parte superior de la tabla hay un campo de búsqueda para filtrar minas por departamento:

1. Ingresar el nombre del departamento.
2. El sistema filtra automáticamente la lista.

8.3 Crear una Nueva Mina

1. Hacer clic en el botón "+ Nueva Mina".
2. Completar el formulario:

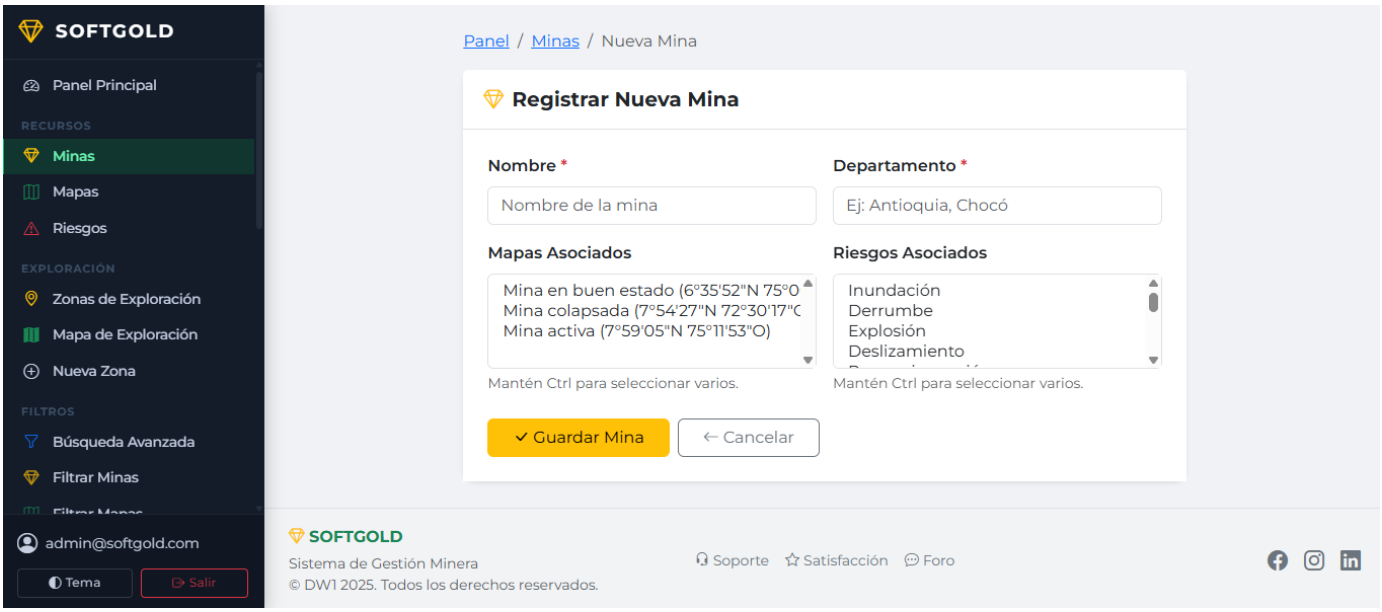


Ilustración 94. Manual usuario - Crear nueva mina

Fuente: Elaboración propia

Campo	Descripción	Obligatorio
Nombre	Nombre de la mina	Sí
Departamento	Departamento donde se ubica	Sí
Mapas asociados	Seleccionar mapas del catálogo	No
Riesgos asociados	Seleccionar riesgos identificados	No

Tabla 71. Manual Usuario - Creación de mina

Fuente: Elaboración propia

1. Hacer clic en "Guardar".
2. El sistema muestra el mensaje: "Mina creada exitosamente."

8.4 Editar una Mina

1. En la lista de minas, hacer clic en el ícono de edición () de la mina deseada.
2. Modificar los campos necesarios.
3. Hacer clic en "Actualizar".

8.5 Eliminar una Mina

1. Hacer clic en el ícono de eliminación () de la mina.
2. Confirmar la acción en el diálogo de confirmación.

ADVERTENCIA: Eliminar una mina también desasociará a todos los usuarios y zonas vinculadas a ella. Esta acción no se puede deshacer.

9. Módulo de mapas

Permite registrar y gestionar los mapas geoespaciales asociados a las minas.

Acceso: Menú lateral -> Mapas

Ruta: /admin/mapas

Rol requerido: Administrador



Ilustración 95. Manual usuario - Gestión de mapas

Fuente: Elaboración propia

9.1 Información de un Mapa

Campo	Descripción
Título	Nombre descriptivo del mapa
Descripción	Detalles del área cubierta
Latitud	Coordenada de latitud del punto central
Longitud	Coordenada de longitud del punto central
Coordenadas	Descripción textual de coordenadas

Tabla 72. Manual Usuario - Información de un mapa

Fuente: Elaboración propia

9.2 Ver Mapa en Leaflet

Algunos mapas incluyen visualización interactiva mediante Leaflet.js + OpenStreetMap. Al hacer clic en el botón "Ver en mapa", se abre la vista geoespacial donde puede:

- Hacer zoom con la rueda del ratón o con los botones +/-.
- Arrastrar el mapa para desplazarse.
- Hacer clic en marcadores para ver información.



Ilustración 96. Manual usuario - Mapa exploración

Fuente: Elaboración propia

10. Módulo de riesgos

Permite catalogar los riesgos geológicos identificados y asociarlos a las minas correspondientes.

Acceso: Menú lateral -> Riesgos

Ruta: /admin/riesgos
Rol requerido: Administrador

10.1 Tipos de Riesgos Comunes

Los riesgos que pueden registrarse incluyen (ejemplos):

- Derrumbes o deslizamientos.
- Inundaciones subterráneas.
- Acumulación de gases tóxicos.

- Riesgo sísmico.
- Falla estructural de galerías.

10.2 Crear un Riesgo

1. Hacer clic en "+ Nuevo Riesgo".
2. Ingresar la descripción del riesgo.
3. Hacer clic en "Guardar".

10.3 Asociar Riesgos a Minas

Los riesgos se asocian a las minas desde el Módulo de Minas -> Editar mina -> Seleccionar riesgos.

11. Módulo de zonas de exploración

Este módulo permite registrar, visualizar y gestionar las zonas geográficas de exploración minera con coordenadas GPS precisas.

Acceso: Menú lateral -> Exploración

Ruta: /admin/exploracion

Rol requerido: Administrador

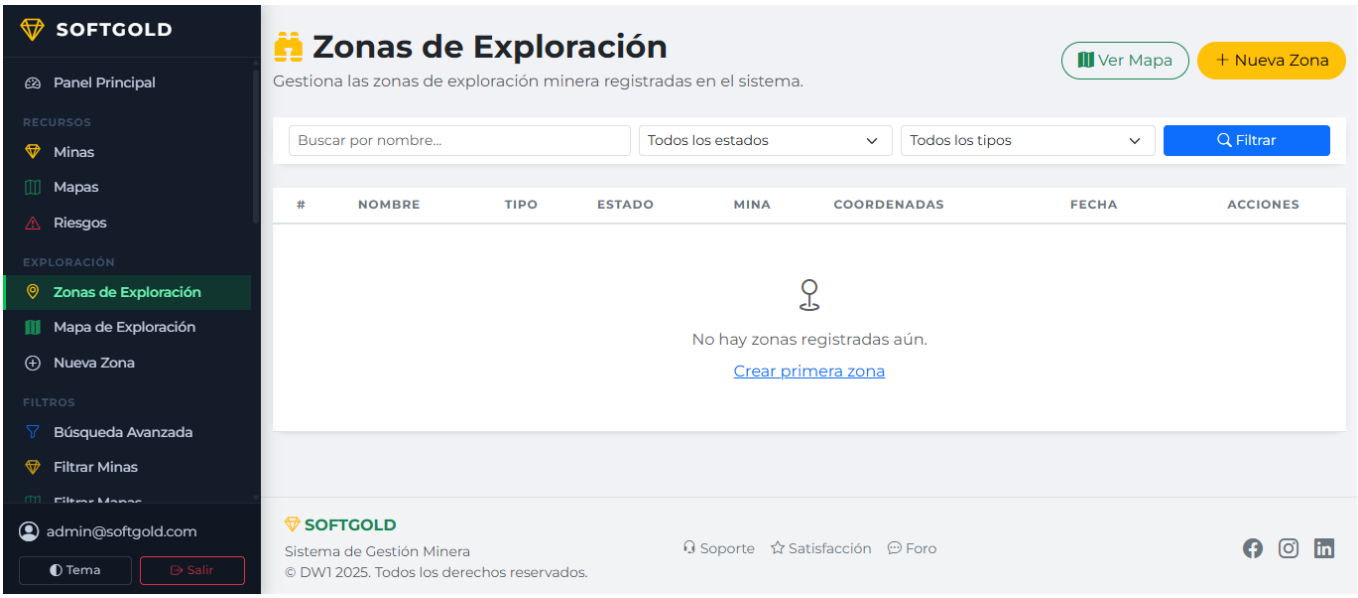


Ilustración 97. Manual usuario - Zonas de exploración

Fuente: Elaboración propia

11.1 Información de una Zona de Exploración

Campo	Descripción
Nombre	Identificación de la zona
Descripción	Detalles del área
Latitud	Coordenada GPS latitud
Longitud	Coordenada GPS longitud
Tipo	Categoría de exploración
Estado	Activo / Inactivo / En evaluación
Mina asociada	Mina a la que pertenece la zona
Fecha de registro	Fecha de creación del registro

Tabla 73. Manual Usuario - Información de una zona de exploración

Fuente: Elaboración propia

11.2 Crear una Zona de Exploración

1. Hacer clic en "+ Nueva Zona".
2. Completar los datos del formulario, incluyendo las coordenadas GPS.
3. Seleccionar la mina a la que pertenece.
4. Hacer clic en "Guardar".

11.3 Mapa General de Exploración

Para ver todas las zonas de exploración en el mapa interactivo:

1. En el módulo de Exploración, hacer clic en "Ver Mapa General".
2. El sistema mostrará un mapa con marcadores para cada zona registrada.
3. Hacer clic en un marcador para ver la información detallada de esa zona.

12. Módulo del foro

El foro es el espacio de comunicación interna del sistema. Permite publicar temas, hacer consultas y responder a publicaciones de los compañeros.

Acceso: Menú lateral -> Foro o desde el menú público

Ruta: /foro

Rol requerido: Acceso público (lectura y escritura)

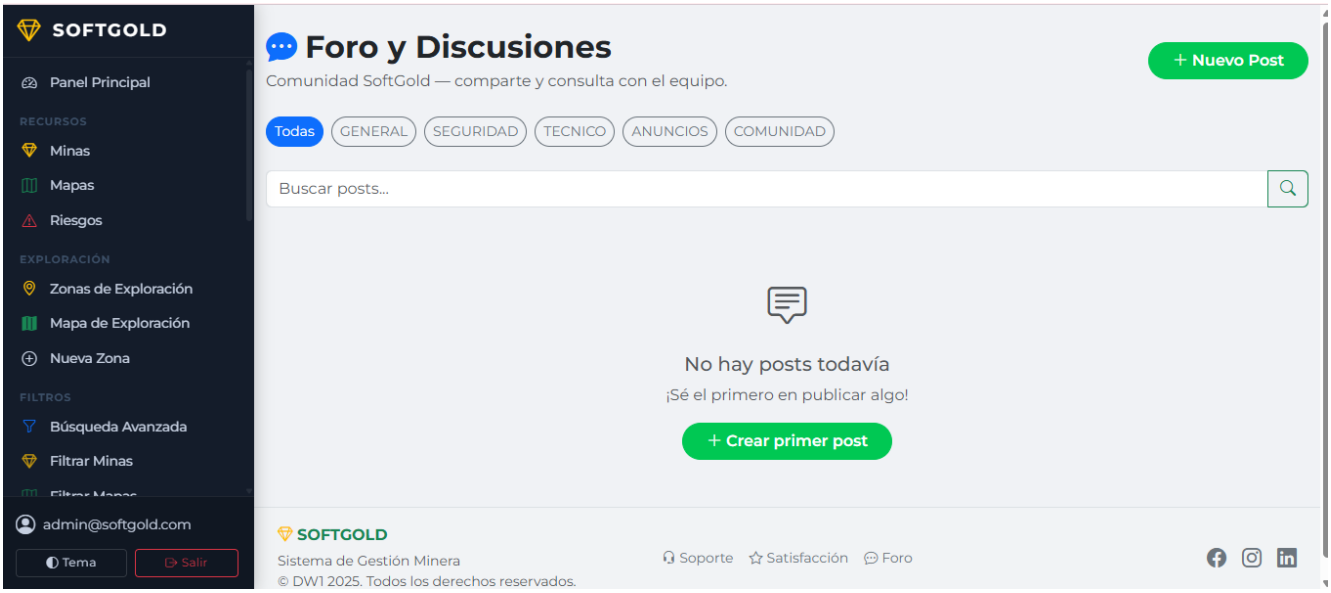


Ilustración 98. Manual Usuario - Módulo del foro

Fuente: Elaboración propia

12.1 Listar Publicaciones

La pantalla principal del foro muestra todas las publicaciones activas con:

- Título del post.
- Categoría.
- Autor.
- Fecha de publicación.
- Número de comentarios.

12.2 Ver una Publicación

1. Hacer clic en el título de la publicación.
2. Se mostrará el contenido completo y los comentarios.
3.

12.3 Crear una Publicación

1. Hacer clic en "+ Nueva Publicación".
2. Completar los campos:

Campo	Descripción
Título	Tema de la publicación
Contenido	Detalle del mensaje
Categoría	Clasificación del tema
Nombre del autor	Su nombre

Tabla 74. Manual Usuario - Crear publicación

Fuente: Elaboración propia

3. Hacer clic en "Publicar".

12.4 Comentar en una Publicación

1. Abrir la publicación.
2. En la sección de comentarios, escribir su respuesta.
3. Ingresar su nombre como autor.
4. Hacer clic en "Comentar".

12.5 Eliminar una Publicación

Solo los Administradores pueden eliminar publicaciones. Para hacerlo:

1. Abrir la publicación.
2. Hacer clic en el botón "Eliminar" (visible solo para administradores).

13. Módulo de informes y reportes

Proporciona herramientas de visualización y análisis para la toma de decisiones gerenciales.

Acceso: Menú lateral -> Informes

Ruta: /admin/informes

Rol requerido: Administrador

13.1 Dashboard Gerencial

El dashboard presenta un resumen visual de los indicadores clave:

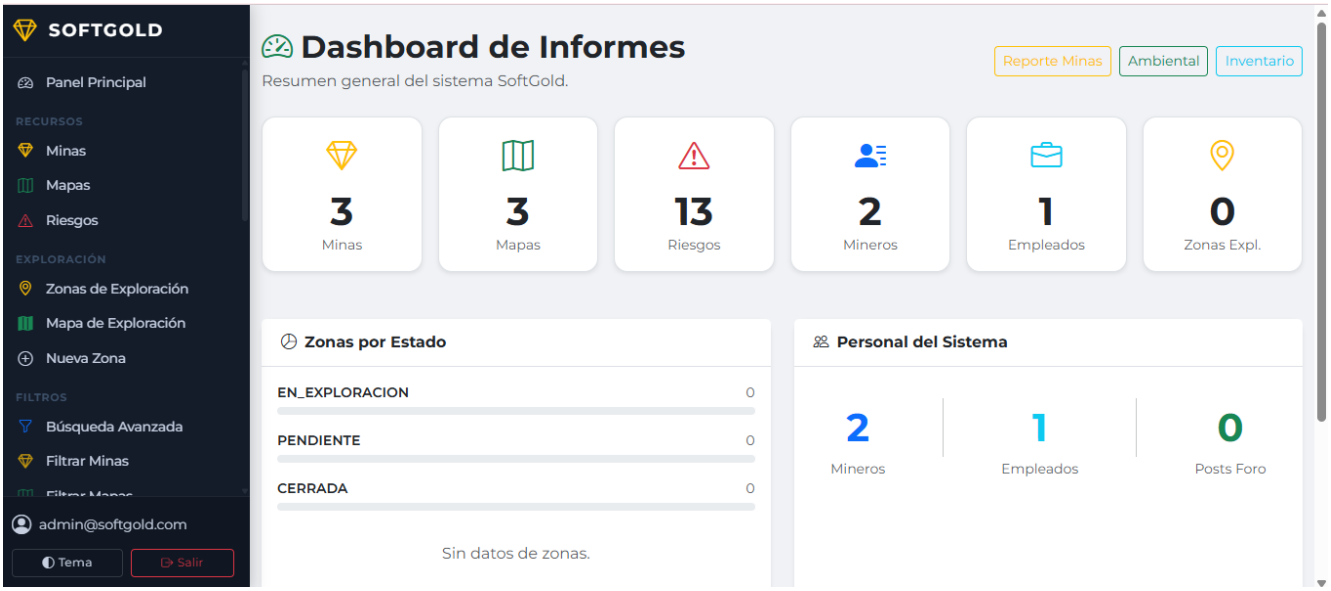


Ilustración 99. Manual usuario - Dashboard gerencial

Fuente: Elaboración propia

Los indicadores incluyen:

- Total de minas activas.
- Total de zonas de exploración por estado.
- Distribución de personal por rol.
- Riesgos más frecuentes.

13.2 Reporte de Minas

Acceso: Informes -> Reporte de Minas

Muestra un reporte detallado con:

- Lista de todas las minas.
- Departamento de ubicación.
- Mapas y riesgos asociados.
- Personal asignado por mina.

13.3 Impacto Ambiental

Acceso: Informes -> Impacto Ambiental

Presenta información sobre:

- Zonas de exploración por tipo.
- Riesgos ambientales identificados.
- Estado de las zonas activas.

13.4 Inventario

Acceso: Informes -> Inventario

Listado consolidado de recursos registrados en el sistema.

14. Sistema de soporte

El sistema de soporte permite a los usuarios reportar problemas técnicos o solicitar asistencia.

14.1 Crear un Ticket de Soporte (Acceso Público)

Cualquier persona puede crear un ticket sin necesidad de iniciar sesión:

Acceso: /soporte/reporte

1. Completar el formulario:

Campo	Descripción
Nombre completo	Nombre de quien reporta
Correo electrónico	Email para respuesta
Asunto	Título del problema
Descripción	Detalle del inconveniente

Tabla 75. Manual Usuario - Ticket Soporte

Fuente: Elaboración propia

2. Hacer clic en "Enviar Reporte".

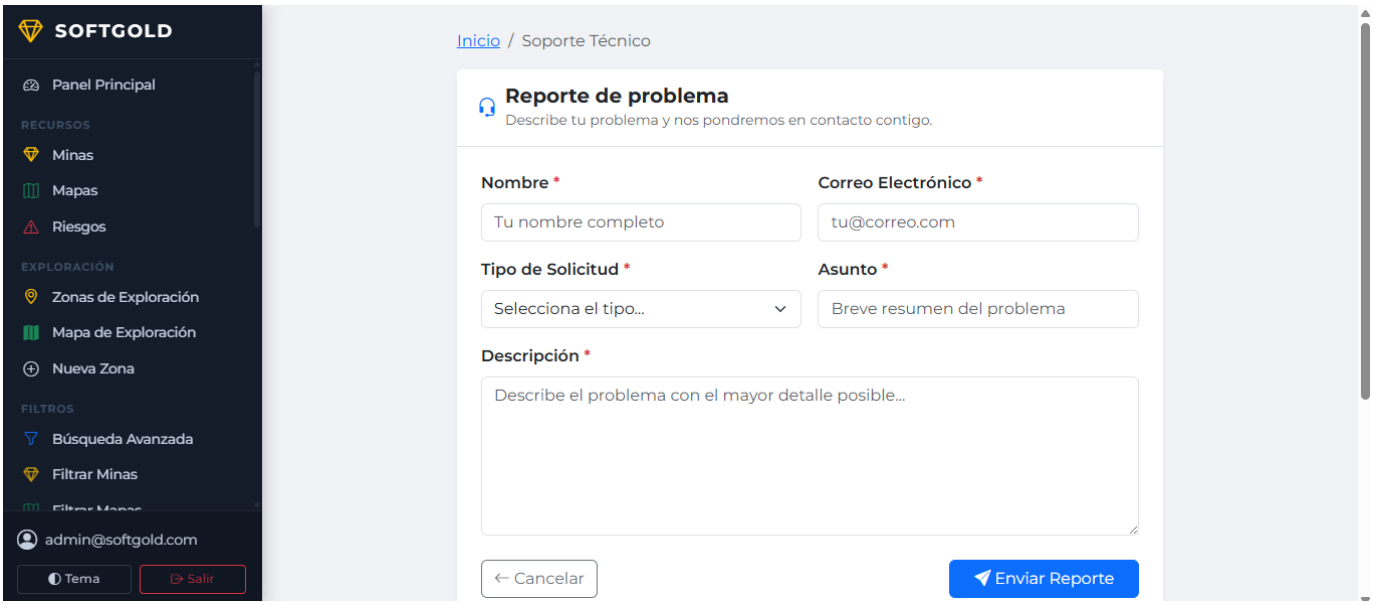


Ilustración 100. Manual usuario - Reportar un problema

Fuente: Elaboración propia

14.2 Gestión de Tickets (Solo Administradores)

Acceso: Menú lateral -> Soporte -> Tickets

El administrador puede:

- Ver todos los tickets abiertos y cerrados.
- Ver el detalle de cada ticket.
- Cerrar un ticket cuando el problema sea resuelto.

Estado	Descripción
OPEN	Ticket abierto, pendiente de atención
CLOSED	Ticket resuelto y cerrado

Tabla 76. Manual Usuario - Gestión tickets

Fuente: Elaboración propia

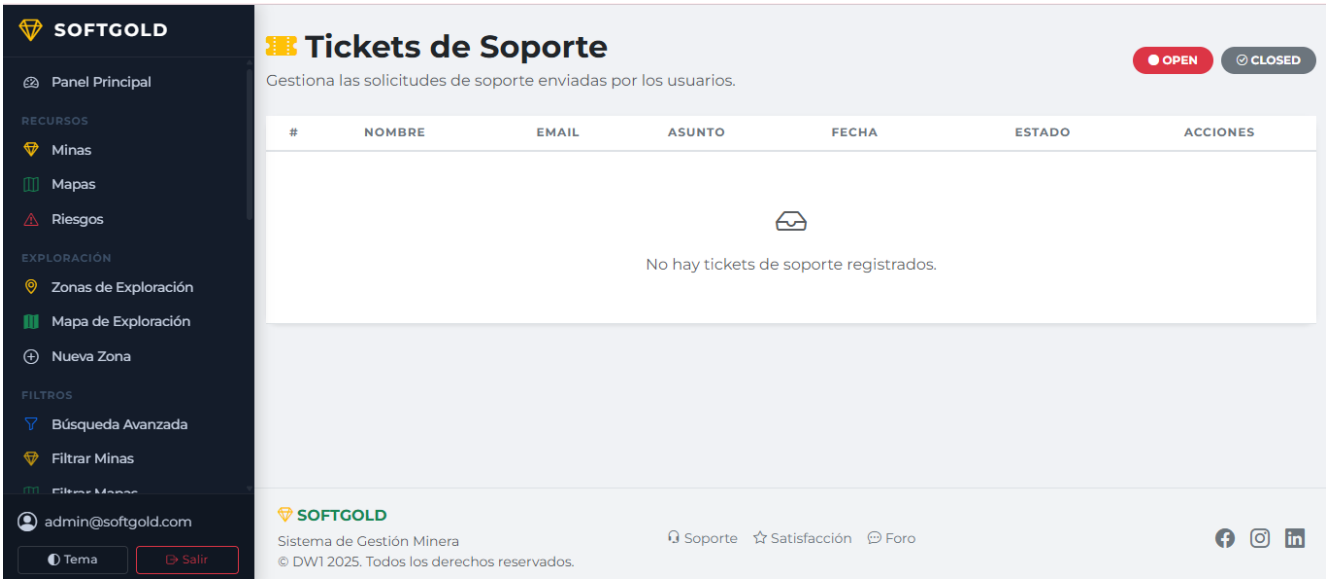


Ilustración 101. Manual usuario - Tickets de soporte

Fuente: Elaboración propia

15. Búsqueda y filtros avanzados

El módulo de filtros permite realizar búsquedas cruzadas en el sistema.

Acceso: Menú lateral -> Filtros

Ruta: /admin/filtros/buscar

Rol requerido: Administrador

15.1 Criterios de Búsqueda

Puede buscar utilizando los siguientes criterios:

Criterio	Ejemplo
Nombre de mina	"La Colosa"
Departamento	"Tolima"
Estado de zona	Activo / Inactivo
Tipo de zona de exploración	Primaria / Secundaria

Tabla 77. Manual Usuario - Criterios de búsqueda

Fuente: Elaboración propia

15.2 Realizar una Búsqueda

1. Ingresar los criterios de búsqueda en los campos disponibles.
2. Hacer clic en "Buscar" o presionar Enter.
3. Los resultados se mostrarán en la tabla inferior.

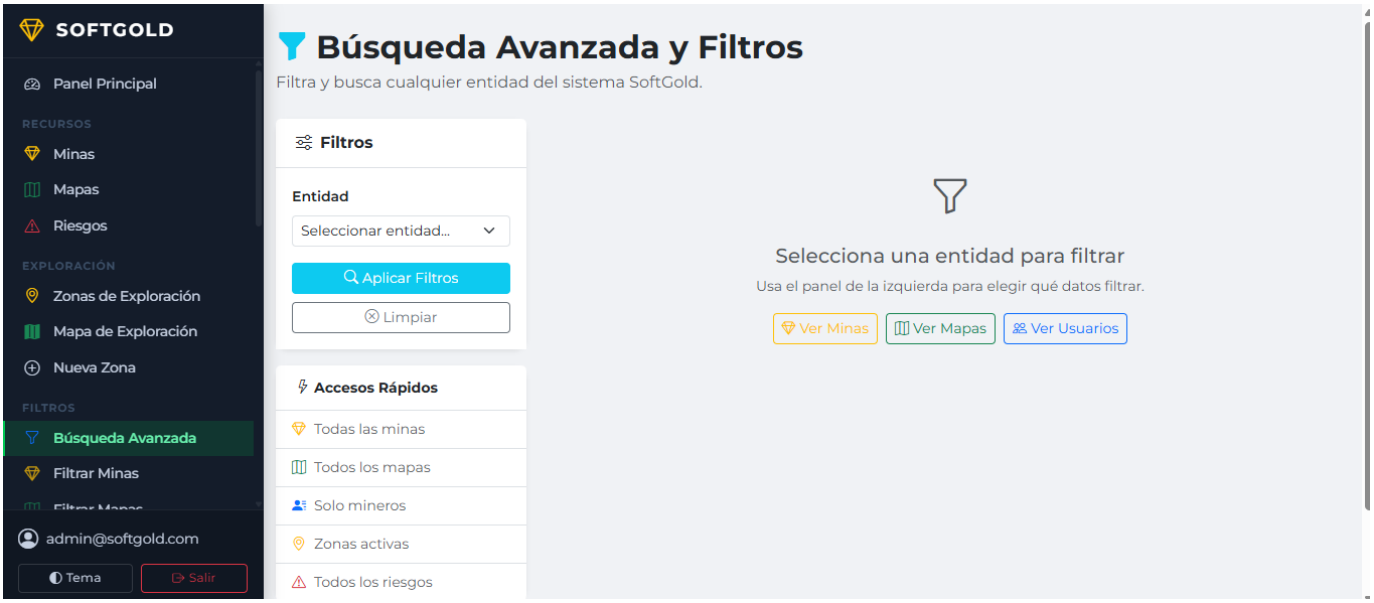


Ilustración 102. Manual usuario - Búsqueda avanzada

Fuente: Elaboración propia

16. Gestión de usuarios

Permite al administrador crear, editar y gestionar los usuarios del sistema.

Acceso: Menú lateral -> Usuarios

Rol requerido: Administrador

16.1 Tipos de Gestión de Usuarios

Sección	Descripción
Lista general de usuarios	Todos los usuarios del sistema
Gestión de Mineros	CRUD específico para usuarios con rol MINERO
Gestión de Empleados	CRUD específico para usuarios con rol EMPLEADO

Tabla 78. Manual Usuario - Tipo de gestión de usuario

Fuente: Elaboración propia

16.2 Crear un Usuario

1. Seleccionar el tipo de usuario a crear (Minero o Empleado).
2. Hacer clic en "+ Nuevo Usuario".
3. Completar el formulario:

Campo	Obligatorio	Descripción
Tipo de documento	Sí	CC, CE, TI, NIT
Número de documento	Sí	Número único
Primer nombre	Sí	-
Segundo nombre	No	-
Primer apellido	Sí	-
Segundo apellido	No	-
Correo electrónico	Sí	Email único de acceso
Contraseña	Sí	Min. 8 caracteres, 1 mayúscula, 1 número
Teléfono	No	-
Mina asignada	No	Para mineros
Tipo de empleado	No	Para empleados
Área	No	Para empleados

Tabla 79. Manual Usuario - Crear un usuario

Fuente: Elaboración propia

4. Hacer clic en "Guardar".

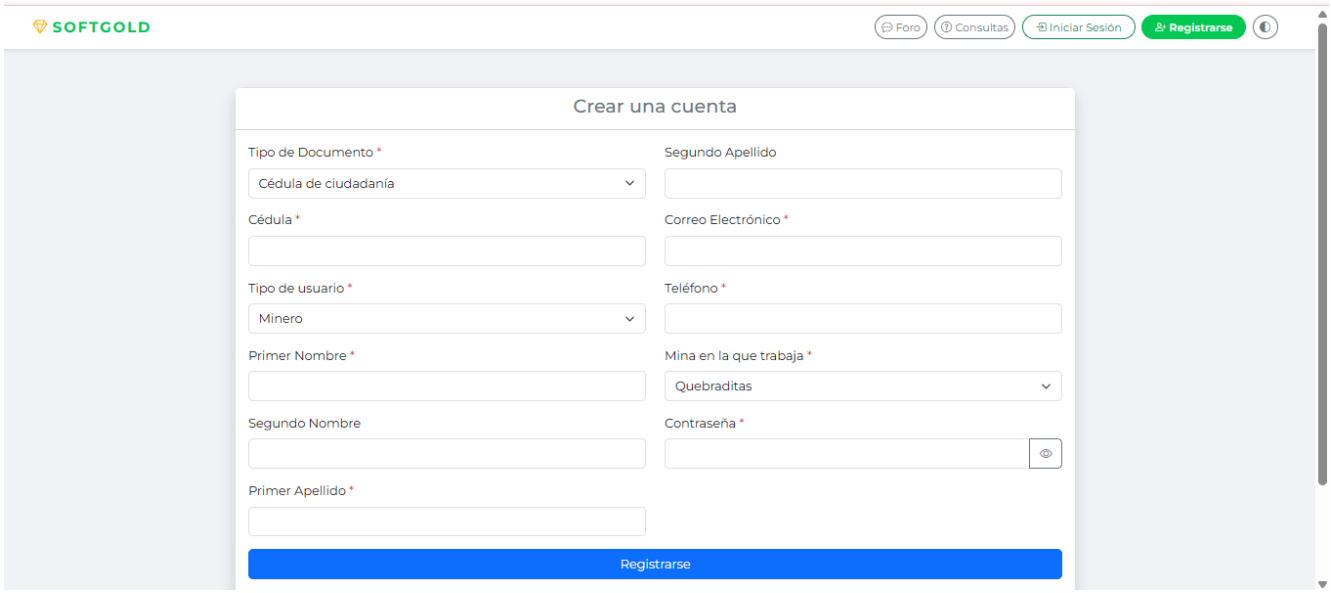


Ilustración 103. Manual usuario - Crear usuario

Fuente: Elaboración propia

16.3 Editar un Usuario

1. En la lista de usuarios, hacer clic en el ícono de edición.
2. Modificar los campos necesarios.
3. Hacer clic en "Actualizar".

16.4 Eliminar un Usuario

1. Hacer clic en el ícono de eliminación del usuario.
2. Confirmar la acción.

NOTA: Eliminar un usuario es irreversible. Asegúrese de que el usuario ya no requiere acceso al sistema antes de proceder.

17. Documentos del sistema

SoftGold incluye documentos estandarizados para la gestión operativa:

Acceso: Menú lateral -> Documentos

Documento	Descripción
Guía de Seguridad	Protocolo de seguridad para el trabajo en minas
Reporte de Incidente	Formulario para reportar incidentes operativos

Formulario de Satisfacción	Encuesta de satisfacción del servicio
Control de Cambios	Registro de cambios en operaciones
Mejora Continua	Formato para propuestas de mejora
Niveles de Servicio (SLA)	Acuerdos de nivel de servicio

Tabla 80. Manual Usuario - Documentos del sistema

Fuente: Elaboración propia

Cada documento puede ser visualizado directamente en el navegador y está disponible para impresión o descarga.

18. Mi cuenta (Perfil)

Cada usuario puede gestionar su información personal desde el menú "Mi Cuenta".

Acceso: Menú lateral -> Mi Cuenta

18.1 Ver Perfil

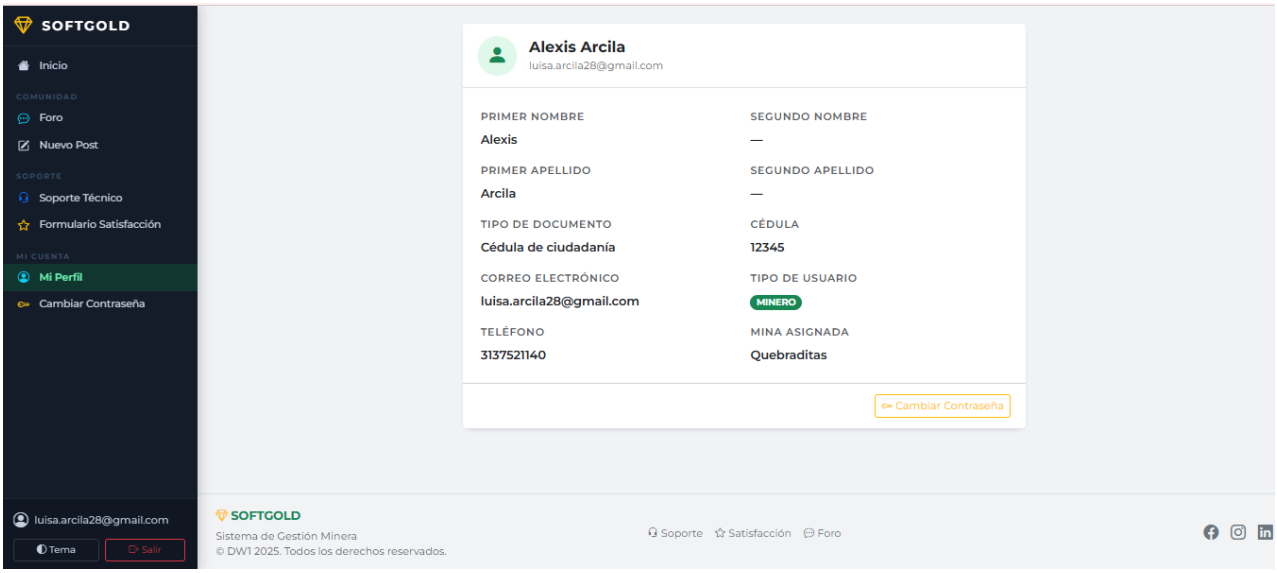


Ilustración 104. Manual usuario - Ver perfil

Fuente: Elaboración propia

En esta sección puede visualizar:

- Nombre completo.

- Correo electrónico.
- Tipo de documento y número.
- Teléfono.
- Mina asignada (si aplica).
- Rol en el sistema.

18.2 Cambiar Contraseña

1. Dentro de Mi Cuenta, hacer clic en "Cambiar Contraseña".
2. Completar el formulario:

Campo	Descripción
Contraseña actual	Su contraseña vigente
Nueva contraseña	Mínimo 8 caracteres, 1 mayúscula, 1 número
Confirmar nueva contraseña	Repetir la nueva contraseña

3. Hacer clic en "Actualizar Contraseña".

Tabla 81. Manual Usuario - Cambiar contraseña

Fuente: Elaboración propia.

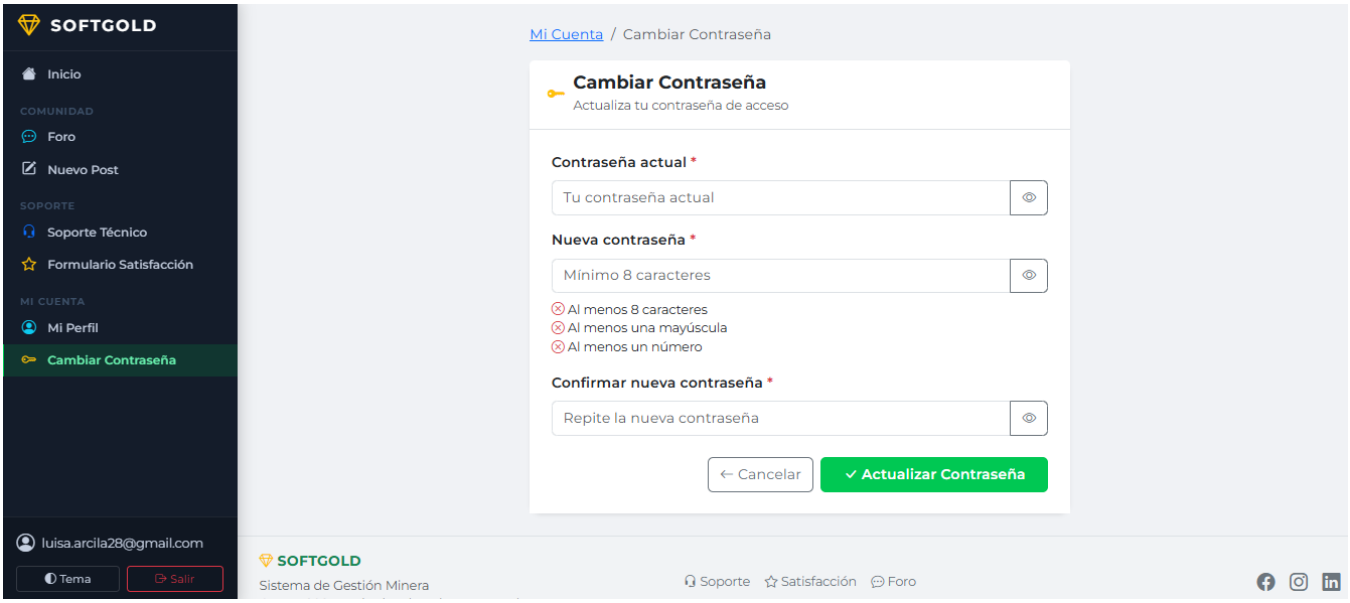


Ilustración 105. Manual usuario - Cambiar contraseña

Fuente: Elaboración propia

19. Atajos de teclado

SoftGold incluye atajos de teclado para mejorar la productividad:

Atajo	Acción
`Ctrl + B`	Ir al campo de búsqueda
`Ctrl + /`	Abrir el panel de atajos de teclado

Tabla 82. Manual Usuario - Atajos de teclado

Fuente: Elaboración propia

Para ver el listado completo de atajos, presionar Ctrl + / en cualquier pantalla.

20. Modo oscuro / claro

SoftGold soporta dos temas visuales:

- Modo claro: Fondo blanco, texto oscuro. Ideal para ambientes bien iluminados.
- Modo oscuro: Fondo oscuro, texto claro. Reduce la fatiga visual en ambientes de poca luz.

Cambiar el Tema

1. Localizar el ícono de sol/luna en la barra superior o en el sidebar.
2. Hacer clic para alternar entre los modos.

La preferencia se guarda automáticamente en su navegador y se mantiene en futuras sesiones.

21. Mensajes de error frecuentes

Mensaje	Causa	Solución
"Correo o contraseña incorrectos"	Credenciales inválidas	Verificar email y contraseña
"Su cuenta ha sido bloqueada"	3 o más intentos fallidos	Contactar al administrador
"El email ya está registrado"	Email duplicado en registro	Usar otro correo o iniciar sesión
"La cédula ya existe"	Documento duplicado	Verificar si el usuario ya existe
*"La contraseña debe tener al menos 8 c	racteres, una mayúscula y un úmero"*Formato inválido	Crear contraseña más fuerte
"El campo es obligatorio"	Campo vacío en formulario	Completar todos los campos requeridos
"Sesión expirada"	Inactividad de 15 minutos	Iniciar sesión nuevamente

*"No tiene permisos para acceder a esta	página"*Rol insuficiente	Contactar al administrador
"Error al enviar el correo"	Problema con servidor SMTP	Contactar soporte técnico
"Recurso no encontrado"	URL inválida o recurso eliminado	Verificar la URL o volver al inicio

Tabla 83. Manual Usuario - Mensajes de error frecuentes

Fuente: Elaboración propia

22. Preguntas frecuentes

1. ¿Puedo usar SoftGold desde mi celular?

Sí. La interfaz es responsive y se adapta a pantallas móviles. En pantallas pequeñas, el menú lateral se oculta y se accede mediante el botón de menú ([menu]).

2. ¿Qué hago si olvidé mi contraseña?

Use la opción "¿Olvidaste tu contraseña?" en la pantalla de inicio de sesión. Recibirá un correo con un enlace para restablecerla.

3. ¿Puedo registrarme yo mismo en el sistema?

Depende de la configuración del administrador. Si el registro público está habilitado, puede acceder a /registro.
De lo contrario, el administrador debe crear su cuenta.

4. ¿Mis datos están seguros?

Sí. Las contraseñas se almacenan cifradas (nunca en texto plano). El acceso al sistema requiere autenticación y el sistema cierra la sesión automáticamente después de 15 minutos de inactividad.

5. ¿Puedo exportar los reportes a PDF o Excel?

Actualmente el sistema no tiene exportación nativa. Sin embargo, puede usar la función de impresión del navegador (Ctrl+P) para guardar cualquier vista como PDF.

6. ¿Cómo puedo ver en qué mina estoy registrado?

Acceder a Mi Cuenta desde el menú lateral. La mina asignada se muestra en la información del perfil.

7. ¿Qué diferencia hay entre los tipos de usuario?

- Administrador: Acceso total al sistema, gestión de todos los módulos.
- Minero: Acceso a módulos operativos y zonas de exploración.
- Empleado: Acceso a módulos administrativos del área.
- Usuario: Acceso de consulta y visualización.

8. ¿Cómo reporto un problema técnico?

Use el formulario de soporte disponible en /soporte/reporte o en el menú lateral bajo "Soporte".

9. ¿El sistema funciona sin internet?

No. SoftGold requiere conexión a la red donde está alojado el servidor. Los mapas de Leaflet.js también requieren acceso a internet para cargar los tiles de OpenStreetMap.

10. ¿Puedo tener más de un rol en el sistema?

Técnicamente el sistema soporta múltiples roles por usuario. Sin embargo, la asignación de roles la realiza el administrador.

23. Cierre de sesión

Para cerrar su sesión de manera segura:

Opción 1: Hacer clic en el botón "Cerrar Sesión" en la barra superior o en la parte inferior del menú lateral.

Opción 2: El sistema cerrará la sesión automáticamente después de 15 minutos de inactividad.

NOTA: RECOMENDACIÓN: Siempre cierre su sesión manualmente cuando termine de usar el sistema, especialmente en equipos compartidos. Nunca deje el sistema abierto sin supervisión.

Después de cerrar sesión, el sistema lo redirigirá a la página de inicio de sesión.

Recomendaciones generales de uso

1. Use contraseñas fuertes: Al menos 8 caracteres con mayúsculas y números.
2. No comparta sus credenciales: Cada usuario debe tener su propio acceso.
3. Cierre sesión al terminar: Especialmente en equipos compartidos.
4. Reporte problemas inmediatamente: Use el sistema de soporte para notificar errores.
5. Mantenga actualizado su perfil: Revise que su información de contacto esté vigente.

6. Use Chrome o Edge: Para la mejor experiencia visual y de rendimiento.
7. Active el modo oscuro: Si trabaja en ambientes con poca iluminación.

Fin del Manual de Usuario del Sistema SoftGold -- Versión 1.0.0

Manual Técnico del Sistema

Sistema de Gestion Integral Minera

Version: 1.0.0

Fecha: Mayo 2026

Dirigido a: Desarrolladores y equipo tecnico

Estado: Documento oficial

Tecnologías: Spring Boot 3.4.4 | Java 17 | MySQL 8 | Thymeleaf

Puerto: 9090 | BD: softgold | Puerto MySQL: 3306

Versión: 1.0.0
Fecha: Mayo 2026

Estado: Oficial

Clasificación: Uso interno - Equipo de desarrollo

Elaborado por: Equipo SoftGold

Proyecto: Ingeniería de Software II - Universidad Autónoma de Manizales

Control de cambios

Versión	Fecha	Descripción	Autor
1.0.0	Mayo 2026	Versión inicial del manual Técnico	Equipo SoftGold

Tabla 84. Manual técnico - Control de cambios

Fuente: Elaboración propia

1. Descripción general del sistema

SoftGold es un sistema de información web empresarial diseñado para la gestión integral de operaciones mineras. La plataforma centraliza la administración de minas, riesgos geológicos, zonas de exploración, personal operativo y comunicación interna a través de un entorno web seguro y multirol.

El sistema implementa una arquitectura MVC (Modelo-Vista-Controlador) de tres capas sobre el stack Spring Boot, con renderizado del lado del servidor mediante Thymeleaf, persistencia en MySQL y seguridad gestionada por Spring Security.

1.1 Contexto del Negocio

La industria minera requiere sistemas robustos para la gestión de activos, zonas geográficas y personal especializado. SoftGold responde a esta necesidad proporcionando:

- Un panel centralizado para administradores con acceso a todos los módulos operativos.
- Interfaces diferenciadas para mineros y empleados según su rol dentro de la organización.
- Módulos de geolocalización integrados para el trazado de zonas de exploración y mapas mineros.
- Un sistema de foro interno para comunicación entre los miembros del equipo.
- Gestión documental integrada con plantillas estandarizadas.
- Sistema de soporte técnico mediante tickets.

1.2 Alcance del Sistema

El sistema abarca los siguientes procesos:

Proceso	Descripción	Módulo
Gestión de Minas	CRUD completo de unidades mineras	MinaController
Gestión de Mapas	Registro y visualización de mapas	MapaController
Gestión de Riesgos	Catálogo y asociación de riesgos por mina	RiesgoController
Exploración	Registro de zonas de exploración	ExplorationControllersGPS
Foro	Comunicación interna con posts y comentarios	ForumController
Informes	Dashboard ejecutivo y reportes operativos	InformeController
Soporte	Sistema de tickets de soporte técnico	SupportController

Filtros	Búsqueda avanzada multi-criterio	FiltroController
Autenticación	Login, registro y recuperación de contraseñas	SecurityConfig
Gestión de Usuarios	Administración de usuarios por roles	UsuarioController

Tabla 85. Manual técnico - Alcance del sistema

Fuente: Elaboración propia

2. Objetivo del software

2.1 Objetivo General

Proporcionar una plataforma web integral que permita a las empresas del sector minero gestionar eficientemente sus operaciones, recursos humanos, infraestructura de minas y comunicación interna, garantizando el control de acceso por roles y la integridad de los datos.

2.2 Objetivos Específicos

1. Centralizar el registro y seguimiento de minas, mapas y zonas de exploración en una única plataforma.
2. Implementar un sistema de control de acceso basado en roles (RBAC) para garantizar la seguridad de la información.
3. Facilitar la comunicación interna del equipo minero a través de un foro con categorías.
4. Proveer reportes e informes gerenciales para la toma de decisiones.
5. Gestionar solicitudes de soporte técnico mediante un sistema de tickets.
6. Integrar visualización geoespacial mediante Leaflet.js y OpenStreetMap para el trazado de zonas.

3. Arquitectura del sistema

3.1 Diagrama de Arquitectura General



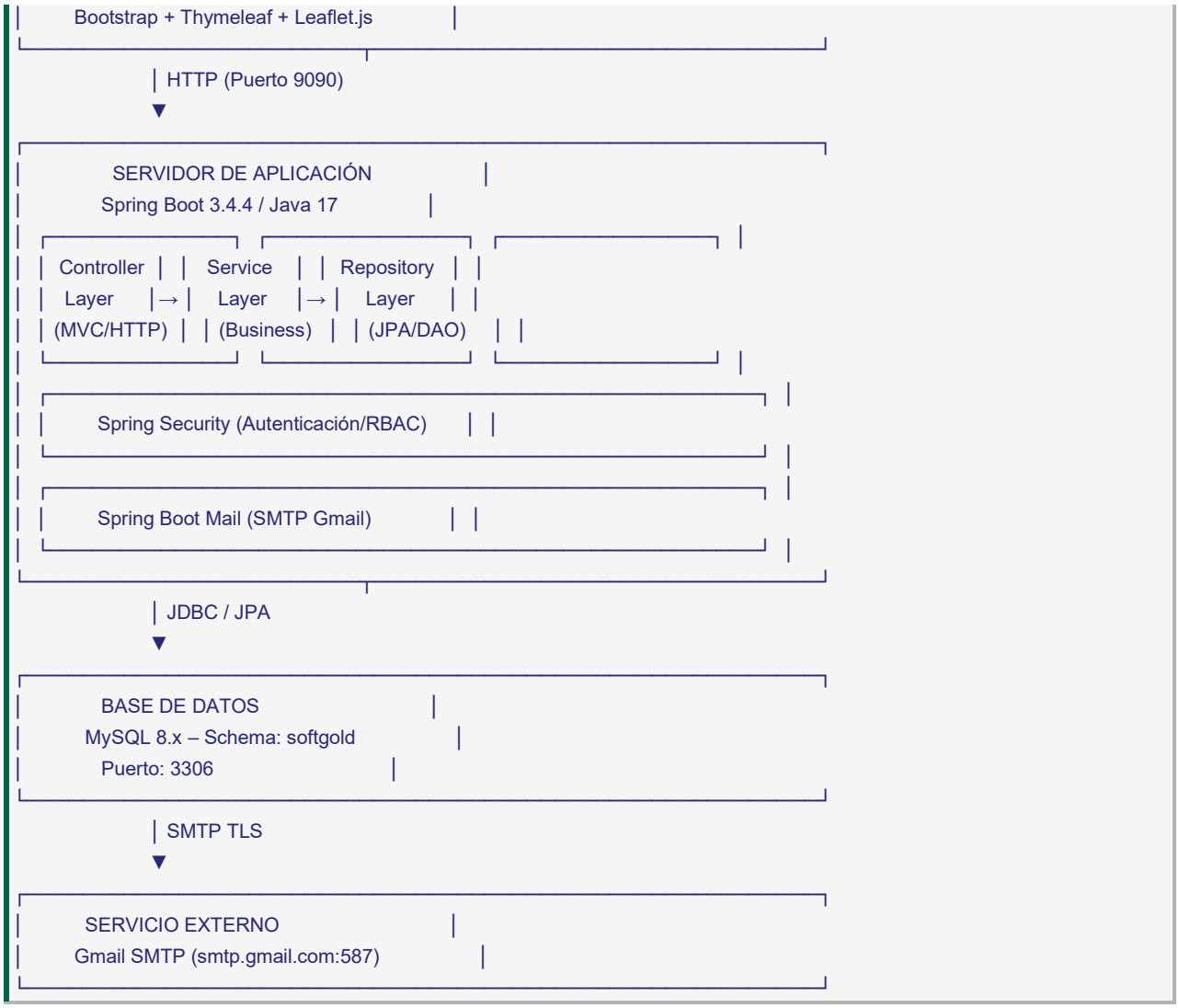


Ilustración 106. Manual técnico - Arquitectura general del sistema

Fuente: Elaboración propia

3.2 Patrón de Diseño MVC

SoftGold implementa el patrón Model-View-Controller (MVC) con las siguientes responsabilidades:

Capa	Tecnología	Responsabilidad
Model	JPA Entities + Spring Data	Representación de datos y acceso a BD
View	Thymeleaf + HTML/CSS/JS	Renderizado de plantillas del lado del servidor
Controller	Spring MVC @Controller	Manejo de solicitudes HTTP y enrutamiento
Security	Spring Security	Autenticación, autorización y filtros
Service	Spring @service	Lógica de negocio desacoplada

Repository	Spring Data JPA	Abstracción de acceso a datos

Tabla 86. Manual técnico - Patrón de diseño MVC

Fuente: Elaboración propia

3.3 Diagrama de componentes

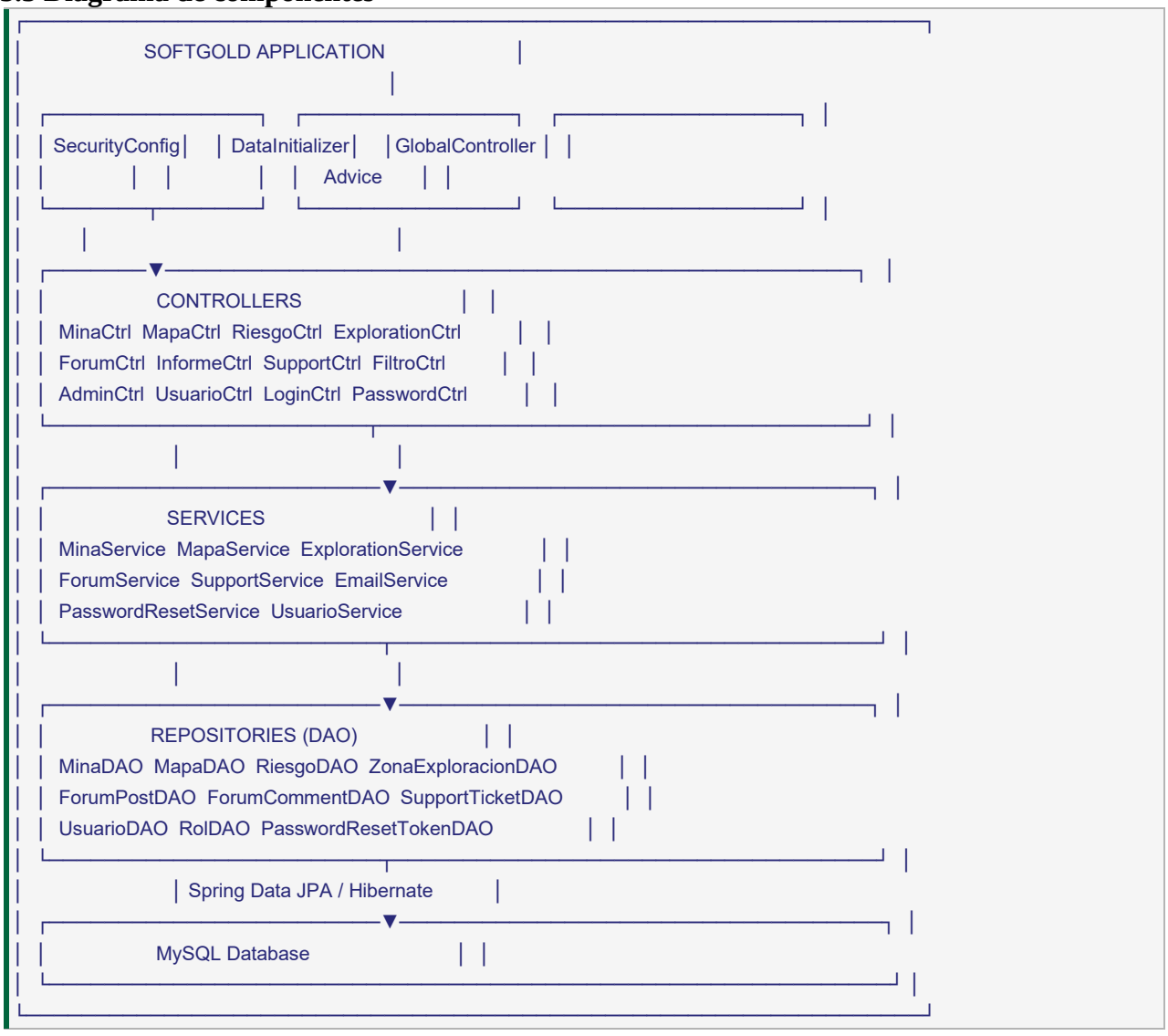


Ilustración 107. Manual técnico - Diagrama de componentes

Fuente: Elaboración propia

4. Tecnologías usadas

4.1 Backend

Tecnología	Versión	Propósito
Java	17 LTS	Lenguaje de programación principal
Spring Boot	3.4.4	Framework principal del backend
Spring MVC	6.x	Framework MVC y manejo de rutas
Spring Security	6.x	Autenticación y autorización
Spring Data JPA	3.x	Persistencia y ORM
Hibernate	6.x	Implementación JPA / ORM
Spring Boot Mail	3.x	Envío de correos electrónicos
Bean Validation	3.x	Validación de entidades
BCrypt	-	Cifrado de contraseñas

Tabla 87. Manual técnico – Backend

Fuente: Elaboración propia

4.2 Frontend

Tecnología	Versión	Propósito
Thymeleaf	3.x	Motor de plantillas del lado del servidor
Bootstrap	5.x	Framework CSS / sistema de grilla
Bootstrap Icons	1.x	Iconografía vectorial
Leaflet.js	1.9.4	Mapas interactivos geoespaciales
OpenStreetMap	-	Proveedor de tiles de mapas (sin API key)
Montserrat	-	Tipografía principal (Google Fonts)
JavaScript ES6+	-	Lógica del lado del cliente

Tabla 88. Manual técnico – Frontend

Fuente: Elaboración propia

4.3 Base de Datos

Tecnología	Versión	Propósito
MySQL	8.x	Motor de base de datos relacional
MySQL Connector/J	8.x	Driver JDBC para Java

Tabla 89. Manual técnico - Base de datos

Fuente: Elaboración propia

4.4 Herramientas de Construcción y DevOps

Herramienta	Versión	Propósito
Apache Maven	3.x	Gestión de dependencias y construcción
Spring Boot DevTools	3.x	Recarga automática en desarrollo
Maven Wrapper (mvnw)	-	Ejecución portable de Maven

Tabla 90. Manual técnico - Herramientas de construcción y DevOps

Fuente: Elaboración propia

4.5 Servicios Externos

Servicio	Propósito
Gmail SMTP (smtp.gmail.com:587)	Envío de correos para recuperación de contraseña
OpenStreetMap Tile Server	Tiles de mapas para Leaflet.js

Tabla 91. Manual técnico - Servicios externos

Fuente: Elaboración propia

5. Estructura del proyecto

```
│   │   │   └── PasswordEncoderConfig.java    ← Bean BCryptPasswordEncoder
│   │   ├── controller/
│   │   │   ├── AdminController.java          ← Panel de administración
│   │   │   ├── MinaController.java           ← CRUD de minas
│   │   │   ├── MapaController.java           ← CRUD de mapas
│   │   │   ├── RiesgoController.java         ← CRUD de riesgos
│   │   │   ├── ExplorationController.java    ← CRUD de zonas de exploración
│   │   │   ├── ForumController.java          ← Foro: posts y comentarios
│   │   │   ├── InformeController.java        ← Dashboard e informes
│   │   │   ├── SupportController.java        ← Sistema de tickets
│   │   │   ├── FiltroController.java         ← Búsqueda avanzada
│   │   │   ├── UsuarioController.java        ← Gestión de usuarios
│   │   │   ├── EmpleadoController.java       ← Gestión de empleados
```

```
│   │   │   │   ├── MineroController.java          ← Gestión de mineros
│   │   │   │   ├── LoginController.java           ← Control de inicio de sesión
│   │   │   │   ├── PasswordController.java        ← Cambio de contraseña (Mi Cuenta)
│   │   │   │   ├── PasswordResetController.java   ← Recuperación por email
│   │   │   │   ├── DocumentosController.java      ← Documentos del sistema
│   │   │   │   ├── GlobalControllerAdvice.java    ← Atributos globales de modelo
│   │   │   │   ├── CustomSuccessHandler.java      ← Redirección post-login por rol
│   │   │   │   ├── RedireccionController.java     ← Redirección genérica por rol
│   │   │   │   ├── InicioController.java          ← Dashboard ADMINISTRADOR
│   │   │   │   ├── MineroInicioController.java    ← Dashboard MINERO
│   │   │   │   ├── EmpleadosInicioController.java ← Dashboard EMPLEADO
│   │   │   │   ├── ValidacionController.java      ← Validaciones adicionales
│   │   │   │   └── VistaController.java           ← Vistas generales
│   │   │   ├── model/
│   │   │   │   ├── Usuario.java                   ← Entidad usuario (implements UserDetails)
│   │   │   │   ├── Rol.java                       ← Entidad rol
│   │   │   │   ├── Mina.java                      ← Entidad mina
│   │   │   │   ├── Mapa.java                      ← Entidad mapa geoespacial
│   │   │   │   ├── Riesgo.java                    ← Entidad riesgo geológico
│   │   │   │   ├── ZonaExploracion.java           ← Entidad zona de exploración
│   │   │   │   ├── ForumPost.java                 ← Entidad post del foro
│   │   │   │   ├── ForumComment.java              ← Entidad comentario del foro
│   │   │   │   ├── SupportTicket.java             ← Entidad ticket de soporte
│   │   │   │   └── PasswordResetToken.java        ← Token para recuperación de contraseña
│   │   │   ├── repository/
│   │   │   │   ├── UsuarioDAO.java                ← Repositorio de usuarios
│   │   │   │   ├── RolDAO.java                    ← Repositorio de roles
│   │   │   │   ├── MinaDAO.java                   ← Repositorio de minas
│   │   │   │   ├── MapaDAO.java                   ← Repositorio de mapas
│   │   │   │   ├── RiesgoDAO.java                 ← Repositorio de riesgos
│   │   │   │   ├── ZonaExploracionDAO.java        ← Repositorio de zonas
│   │   │   │   ├── ForumPostDAO.java              ← Repositorio de posts
│   │   │   │   ├── ForumCommentDAO.java           ← Repositorio de comentarios
│   │   │   │   ├── SupportTicketDAO.java          ← Repositorio de tickets
│   │   │   │   └── PasswordResetTokenDAO.java     ← Repositorio de tokens de reset
│   │   │   ├── security/
│   │   │   │   ├── SecurityConfig.java            ← Configuración principal de seguridad
│   │   │   │   ├── CustomUserDetailsService.java  ← Carga de usuario por email
│   │   │   │   └── CustomLoginPassword.java       ← Manejador de fallos de login
│   │   │   └── service/
│   │   │       ├── MinaService.java / MinaServiceIface.java
│   │   │       ├── MapaService.java / MapaServiceIface.java
│   │   │       ├── ExplorationService.java / ExplorationServiceIface.java
│   │   │       ├── ForumService.java / ForumServiceIface.java
│   │   │       ├── SupportService.java / SupportServiceIface.java
│   │   │       ├── EmailService.java / EmailServiceIface.java
│   │   │       ├── PasswordResetService.java / PasswordResetServiceIface.java
│   │   │       └── UsuarioService.java / UsuarioServiceIface.java
│   │   └── resources/
│   │       ├── application.properties             ← Configuración de la aplicación
│   │       ├── static/
│   │       │   ├── css/
│   │       │   │   └── styles.css                 ← Hoja de estilos principal
```

```
│   │   │   ├── js/
│   │   │   │   ├── atajos.js              ← Atajos de teclado
│   │   │   │   ├── inactividad.js         ← Cierre por inactividad
│   │   │   │   ├── tema.js                ← Toggle modo oscuro/claro
│   │   │   │   ├── busqueda.js            ← Búsqueda dinámica
│   │   │   │   ├── cambiarPassword.js     ← Validación de contraseña
│   │   │   │   ├── crearAdmin.js          ← Lógica creación de admin
│   │   │   │   ├── eliminar.js            ← Confirmación de eliminación
│   │   │   │   ├── login.js               ← Lógica de login
│   │   │   │   └── registro.js            ← Lógica de registro
│   │   │   ├── reporte_incidente.html     ← Documento estático
│   │   │   ├── formulario_satisfaccion.html ← Documento estático
│   │   │   ├── service_level.html         ← Documento estático
│   │   │   ├── control_cambios.html       ← Documento estático
│   │   │   └── mejor_continua.html        ← Documento estático
│   │   └── templates/
│   │       ├── plantillas/
│   │       │   └── principal.html         ← Plantilla base (fragmentos)
│   │       └── vistas/
│   │           ├── login.html
│   │           ├── registro.html
│   │           ├── inicio.html
│   │           ├── inicioMinero.html
│   │           ├── inicioEmpleado.html
│   │           ├── admin.html
│   │           ├── perfil.html / perfilAdmin.html
│   │           ├── listarMinas.html / crearMina.html / editarMina.html
│   │           ├── listarMapas.html / crearMapa.html / editarMapa.html
│   │           ├── listarRiesgos.html / crearRiesgo.html / editarRiesgo.html
│   │           ├── listarMineros.html / crearMinero.html / editarMinero.html
│   │           ├── listarEmpleados.html / crearEmpleado.html / editarEmpleado.html
│   │           ├── listarSoporte.html / reporteSoporte.html
│   │           ├── recuperarPassword.html / restablecerPassword.html
│   │           ├── cuentaBloqueada.html
│   │           ├── cambiarPassword.html / cuentaCambiarPassword.html
│   │           ├── cuentaPerfil.html
│   │           ├── exploracion/
│   │           │   ├── listarZonas.html
│   │           │   ├── crearZona.html
│   │           │   ├── editarZona.html
│   │           │   └── mapaExploracion.html
│   │           ├── foro/
│   │           │   ├── listarPosts.html
│   │           │   ├── crearPost.html
│   │           │   └── verPost.html
│   │           ├── informes/
│   │           │   ├── dashboard.html
│   │           │   ├── reporteMinas.html
│   │           │   ├── impactoAmbiental.html
│   │           │   └── inventario.html
│   │           ├── filtros/
│   │           │   └── buscar.html
│   │           └── documentos/
│   │               ├── guiaSeguridad.html
```

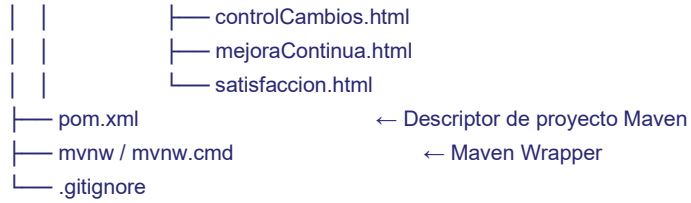
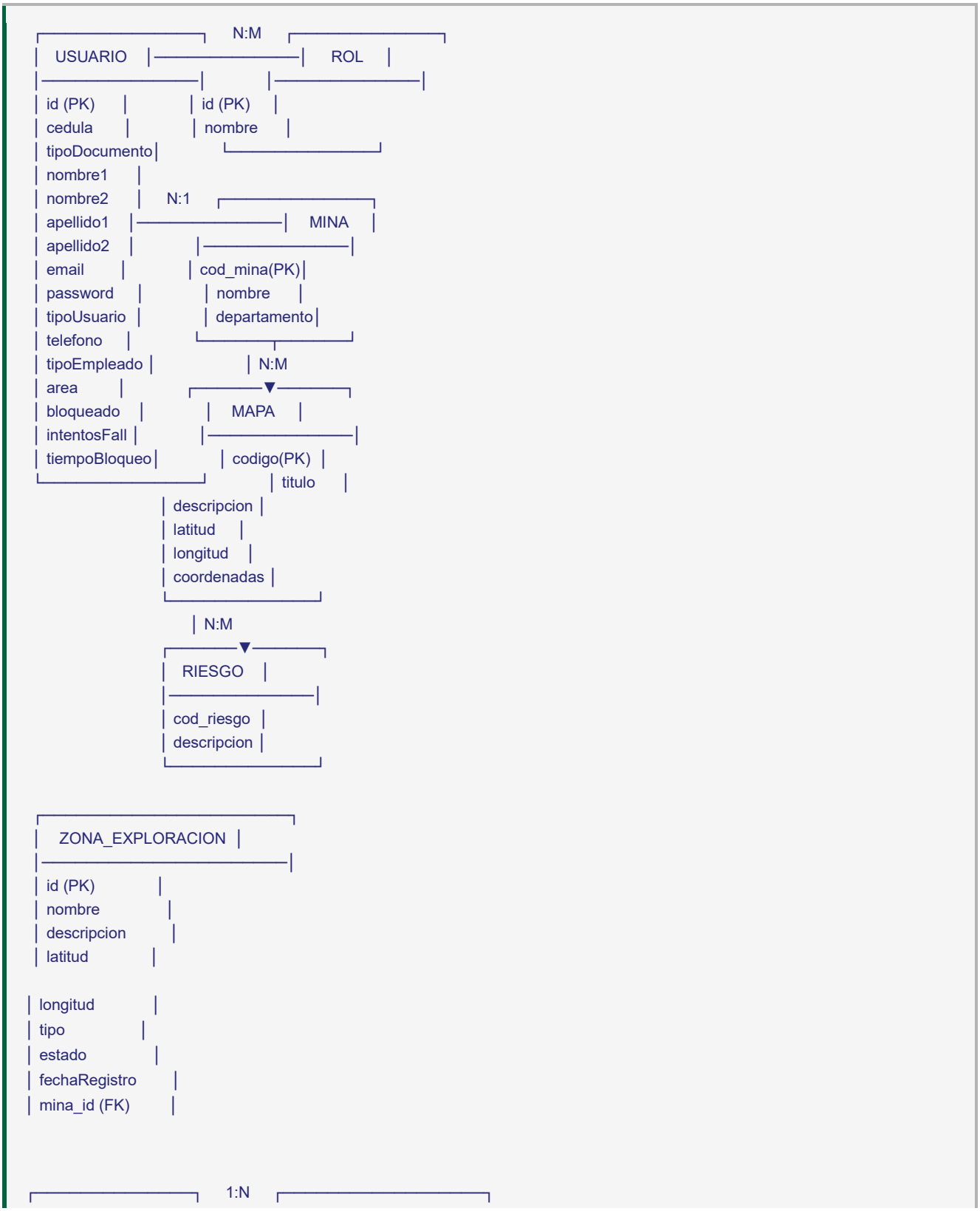


Ilustración 108. Manual técnico - Diagrama del proyecto

Fuente: Elaboración propia

6. Modelos de datos

6.1 Diagrama Entidad-Relación (Conceptual)


USUARIO
id (PK)
cedula
tipoDocumento
nombre1
nombre2
apellido1
apellido2
email
password
tipoUsuario
telefono
tipoEmpleado
area
bloqueado
intentosFall
tiempoBloqueo
N:M
ROL
id (PK)
nombre
N:1
MINA
cod_mina(PK)
nombre
departamento
N:M
MAPA
codigo(PK)
titulo
descripcion
latitud
longitud
coordenadas
N:M
RIESGO
cod_riesgo
descripcion
ZONA_EXPLORACION
id (PK)
nombre
descripcion
latitud
longitud
tipo
estado
fechaRegistro
mina_id (FK)
1:N

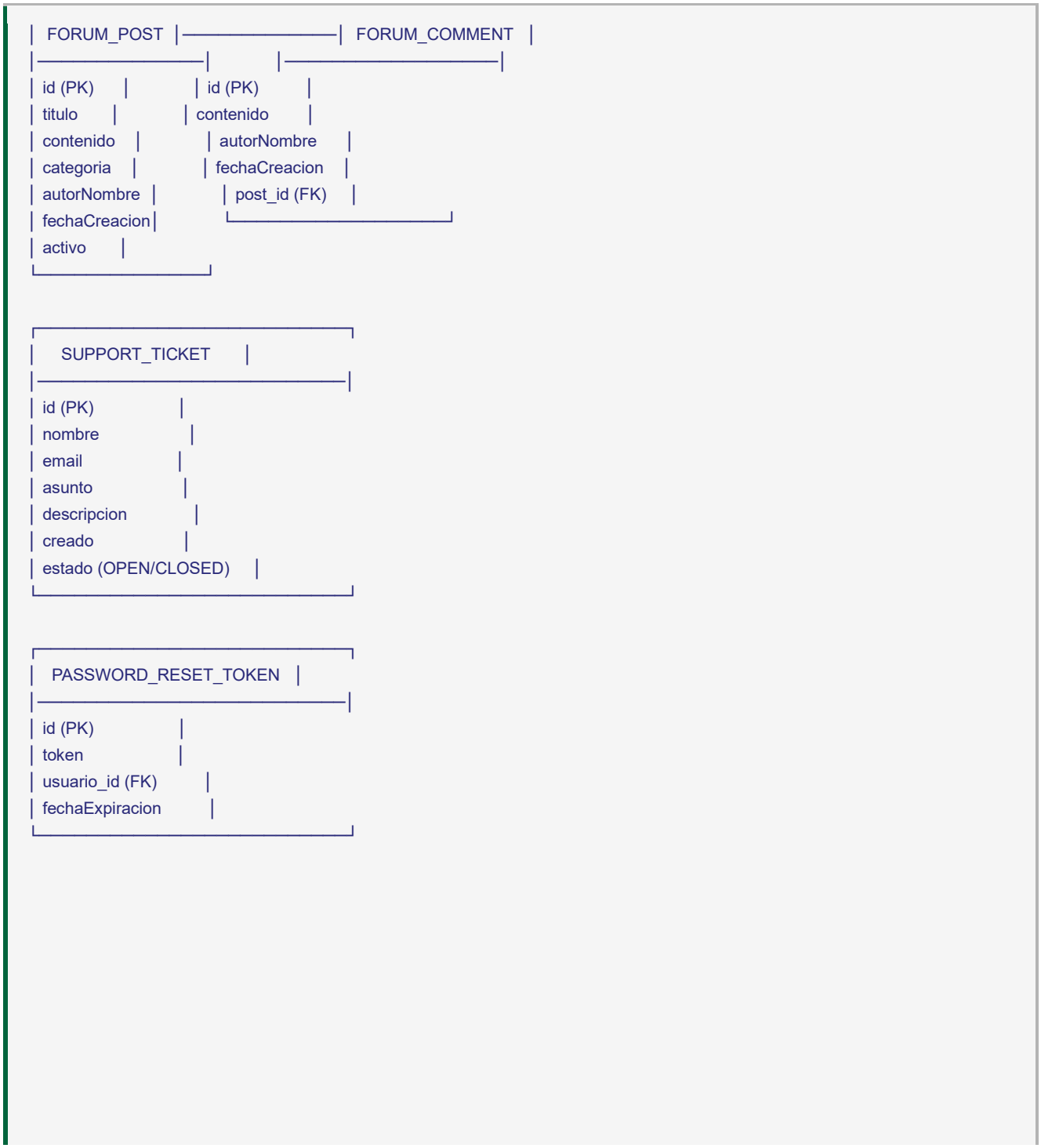


Ilustración 109. Manual técnico - Diagrama Entidad - Relación (Conceptual)

Fuente: Elaboración propia

6.2 Descripción de Entidades

Entidad central del sistema. Implementa UserDetails de Spring Security para integración directa con el sistema de autenticación.

Entidad Usuario

Campo	Tipo	Restricción	Descripción
id	Long	PK, AUTO	Identificador interno
cedula	String	UNIQUE, NOT NULL	Documento de identidad
tipoDocumento	String	NOT NULL	CC, CE, TI, NIT
nombre1	String	NOT NULL	Primer nombre
nombre2	String	-	Segundo nombre
apellido1	String	NOT NULL	Primer apellido
apellido2	String	-	Segundo apellido
email	String	UNIQUE, NOT NULL	Correo (username en Spring Security)
password	String	NOT NULL	Hash BCrypt
tipoUsuario	String	NOT NULL	Categoría del usuario
telefono	String	-	Número de contacto
tipoEmpleado	String	-	Solo para empleados
area	String	-	Área de trabajo
intentosFallidos	int	DEFAULT 0	Contador para bloqueo
bloqueado	boolean	DEFAULT false	Estado de bloqueo
tiempoBloqueo	LocalDateTime	NULL	Timestamp de bloqueo
roles	Collection<Rol>	N:M	Roles del usuario
mina	Mina	FK	Mina asignada

Tabla 92. Manual técnico – Entidad usuario

Fuente: Elaboración propia

Entidad Mina

Campo	Tipo	Restricción	Descripción
cod_mina	Long	PK, AUTO	Código interno
nombre	String	NOT NULL	Nombre de la mina
departamento	String	NOT NULL	Departamento geográfico

mapas	Set<Mapa>	N:M	Mapas asociados
riesgos	Set<Riesgo>	N:M	Riesgos identificados
usuarios	List<Usuario>	1:N (inverso)	Personal asignado

Tabla 93. Manual técnico - Entidad mina

Fuente: Elaboración propia

Herramientas

Herramienta	Versión Mínima	Descarga
JDK (Java)	17 LTS	adoptium.net
Maven	3.8+	maven.apache.org

Tabla 94. Manual técnico – Herramientas

Fuente: Elaboración propia

Roles disponibles

- ROLE_ADMINISTRADOR -- Acceso total al sistema
- ROLE_MINERO -- Acceso a módulos operativos del minero
- ROLE_EMPLEADO -- Acceso a módulos del empleado
- ROLE_USUARIO -- Acceso básico de consulta

7. Configuración del entorno de desarrollo

7.1 Variables de Entorno (application.properties)

Propiedad	Valor por defecto	Descripción
`server.port`	9090	Puerto HTTP de la aplicación
`spring.jpa.hibernate.ddl-auto`	src/main/resources/application.properties	Ruta archivo de configuración

Tabla 95. Manual técnico - Variables de entorno

Fuente: Elaboración propia

```
# Nombre de la aplicación spring.application.name=softgold

# Puerto del servidor server.port=9090

# Configuración de conexión MySQL
spring.datasource.url=jdbc:mysql://localhost:3306/softgold?serverTimezone=America/Bogota&createDatabaseIfNotEx
spring.datasource.username=root spring.datasource.password=

# Configuración JPA/Hibernate spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.format_sql=true

# Logging SQL logging.level.org.hibernate.SQL=debug
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE

# Configuración SMTP Gmail
spring.mail.host=smtp.gmail.com spring.mail.port=587
spring.mail.username=<correo-remitente>@gmail.com
spring.mail.password=<contraseña-de-aplicacion>
spring.mail.protocol=smtp
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true
spring.mail.properties.mail.smtp.starttls.required=true
spring.mail.properties.mail.smtp.connectiontimeout=5000
spring.mail.properties.mail.smtp.timeout=5000
spring.mail.properties.mail.smtp.writetimeout=5000

# Timeout de sesión server.servlet.session.timeout=15m
```

ADVERTENCIA: En entornos de producción, las credenciales de correo y base de datos deben gestionarse mediante variables de entorno del sistema operativo o un gestor de secretos, nunca en

Fuente: Elaboración propia

7.2 Descripción de Propiedades Clave

Propiedad	Valor por defecto	Descripción
`server.servlet.session.timeout`	15m	Expiración de sesión por inactividad
`createDatabaseIfNotExist=true`	-	Crea la BD automáticamente si no existe

Tabla 96. Manual técnico - Descripción de propiedades claves

Fuente: Elaboración propia.

8. Instalación del proyecto

8.1 Clonar el Repositorio

```
git clone <url-del-repositorio> cd
softgold_final
```

Fuente: Elaboración propia

8.2 Configurar la Base de Datos

Paso 1: Verificar que MySQL esté corriendo:

```
# Windows (PowerShell)
Get-Service -Name "MySQL*"

# Linux
sudo systemctl status mysql
```

Fuente: Elaboración propia

Paso 2: Crear el esquema (opcional -- se crea automáticamente):

```
CREATE DATABASE IF NOT EXISTS softgold
  CHARACTER SET utf8mb4
  COLLATE utf8mb4_unicode_ci;
```

Fuente: Elaboración propia

Paso 3: Verificar el usuario de MySQL y sus permisos:

```
-- En producción, crear usuario dedicado:
CREATE USER 'softgold_user'@'localhost' IDENTIFIED BY 'contraseña_segura';
GRANT ALL PRIVILEGES ON softgold.* TO 'softgold_user'@'localhost'; FLUSH PRIVILEGES;
```

Fuente: Elaboración propia

8.3 Configurar el Archivo de Propiedades

Editar src/main/resources/application.properties:

- Actualizar spring.datasource.username y spring.datasource.password según el entorno.
- Actualizar credenciales de Gmail para el servicio de correo.

8.4 Instalar Dependencias y Compilar

```
# Con Maven Wrapper (recomendado) ./mvnw
clean install -DskipTests
```

```
# Con Maven instalado globalmente mvn clean
install -DskipTests
```

Fuente: Elaboración propia

8.5 Verificar la Compilación

```
./mvnw package -DskipTests
```

El artefacto generado será: target/softgold-0.0.1-SNAPSHOT.jar

9. Ejecución local del sistema

9.1 Desde el IDE

En IntelliJ IDEA o VS Code con extensión Spring Boot:

1. Abrir el proyecto como proyecto Maven.
2. Localizar la clase SoftgoldApplication.java.
3. Ejecutar con el botón "Run" o Shift+F10.

9.2 Desde la Línea de Comandos

```
# Ejecutar con Maven Wrapper
./mvnw spring-boot:run

# Ejecutar el JAR generado java -jar
target/softgold-0.0.1-SNAPSHOT.jar

# Ejecutar con perfil de producción java -jar target/softgold-0.0.1-
SNAPSHOT.jar --spring.profiles.active=prod
```

Fuente: Elaboración propia

9.3 Verificar el Inicio

Acceder en el navegador a: http://localhost:9090

La aplicación redirigirá automáticamente a /login.

9.4 Credenciales por Defecto

El sistema crea automáticamente un usuario administrador al iniciar por primera vez (via DataInitializer.java):

Campo	Valor
Email	admin@softgold.com

Contraseña	admin123
Rol	ADMINISTRADOR

Tabla 97. Manual técnico - Credenciales por defecto

Fuente: Elaboración propia.

ADVERTENCIA: Cambiar estas credenciales inmediatamente en cualquier entorno que no sea desarrollo local.

10. Rutas y controladores

10.1 Rutas Públicas (sin autenticación)

Ruta	Método	Descripción
`/login`	GET, POST	Formulario de inicio de sesión
`/registro`	GET, POST	Registro de nuevos usuarios
`/recuperar`	GET	Formulario de recuperación de contraseña
`/recuperar-password`	POST	Envío de email de recuperación
`/cambiarPassword/{token}`	GET, POST	Restablecimiento de contraseña
`/cuenta-bloqueada`	GET	Página de cuenta bloqueada
`/foro`	GET	Listado público del foro
`/foro/{id}`	GET	Vista de un post del foro
`/foro/crear`	GET, POST	Creación de post (público)
`/foro/{id}/comentar`	POST	Agregar comentario (público)
`/soporte/reporte`	GET, POST	Formulario de reporte público
`/css/**`	GET	Recursos CSS
`/js/**`	GET	Recursos JavaScript
`/images/**`	GET	Recursos de imágenes

Tabla 98. Manual técnico - Rutas y controladores

Fuente: Elaboración propia.

10.2 Rutas del Administrador (/admin/**)

Ruta	Método	Descripción
`/admin`	GET	Panel de administración
`/admin/minas`	GET	Listar todas las minas
`/admin/minas/crear`	GET, POST	Crear una mina
`/admin/minas/editar/{id}`	GET, POST	Editar una mina
`/admin/minas/eliminar/{id}`	GET	Eliminar una mina
`/admin/minas/buscar`	GET	Buscar minas por departamento
`/admin/mapas`	GET	Listar mapas
`/admin/mapas/crear`	GET, POST	Crear mapa
`/admin/mapas/editar/{id}`	GET, POST	Editar mapa
`/admin/mapas/eliminar/{id}`	GET	Eliminar mapa
`/admin/riesgos`	GET	Listar riesgos
`/admin/riesgos/crear`	GET, POST	Crear riesgo
`/admin/riesgos/editar/{id}`	GET, POST	Editar riesgo
`/admin/riesgos/eliminar/{id}`	GET	Eliminar riesgo
`/admin/exploracion`	GET	Listar zonas de exploración
`/admin/exploracion/crear`	GET, POST	Crear zona de exploración
`/admin/exploracion/editar/{id}`	GET, POST	Editar zona
`/admin/exploracion/eliminar/{id}`	GET	Eliminar zona
`/admin/exploracion/mapa`	GET	Mapa general de exploración
`/admin/filtros/buscar`	GET	Búsqueda avanzada
`/admin/informes/dashboard`	GET	Dashboard gerencial
`/admin/informes/reporte-minas`	GET	Reporte de minas
`/admin/informes/impacto-ambiental`	GET	Informe ambiental
`/admin/informes/inventario`	GET	Inventario de recursos
`/admin/usuarios`	GET	Listar todos los usuarios
`/admin/usuarios/crear`	GET, POST	Crear usuario

`/admin/usuarios/editar/{id}`	GET, POST	Editar usuario
`/admin/usuarios/eliminar/{id}`	GET	Eliminar usuario
`/admin/mineros`	GET	Listar mineros
`/admin/mineros/crear`	GET, POST	Crear minero
`/admin/empleados`	GET	Listar empleados
`/admin/empleados/crear`	GET, POST	Crear empleado

Tabla 99. Manual técnico - Rutas del administrador

Fuente: Elaboración propia

10.3 Rutas del Sistema de Soporte

Ruta	Método	Rol Requerido	Descripción
`/soporte/reporte`	GET, POST	Público	Crear ticket público
`/soporte/tickets`	GET	ADMINISTRADOR	Listar tickets
`/soporte/tickets/{id}/cerrar`	POST	ADMINISTRADOR	Cerrar un ticket

Tabla 100. Manual técnico- Rutas del sistema de soporte

Fuente: Elaboración propia

10.4 Rutas del Foro

Ruta	Método	Rol Requerido	Descripción
`/foro`	GET	Público	Listar posts
`/foro/crear`	GET, POST	Público	Crear post
`/foro/{id}`	GET	Público	Ver post y comentarios
`/foro/{id}/comentar`	POST	Público	Añadir comentario
`/foro/eliminar/{id}`	GET	ADMINISTRADOR	Eliminar post

Tabla 101. Manual técnico - Rutas del foro

Fuente: Elaboración propia

10.5 Convenciones de Controladores

Todos los controladores siguen el patrón estándar Spring MVC:

```
@Controller
@RequestMapping("/admin/minas")
public class MinaController {

    @Autowired     private
MinaDAO minaDAO;

    @GetMapping("")     public String listar(Model model) {
model.addAttribute("minas", minaDAO.findAll());         return
"vistas/listarMinas";  // Retorna nombre de template Thymeleaf
    }

    @PostMapping("/crear")
    public String crear(@Valid @ModelAttribute Mina mina,
                        BindingResult result,
                        RedirectAttributes redirectAttrs) {         if (result.hasErrors()) return
"vistas/crearMina";         minaDAO.save(mina);
redirectAttrs.addFlashAttribute("mensajeExito", "Mina creada exitosamente.");
return "redirect:/admin/minas";
    }
}
```

Ilustración 110. Manual técnico - Convenciones de controladores

Fuente: Elaboración propia

11. Seguridad implementada

11.1 Diagrama de Flujo de Autenticación

```
Usuario ingresa email/contraseña
        │
        ▼
  Spring Security intercepta POST /login
        │
        ▼
  CustomUserDetailsService.loadUserByUsername(email)
        │
        ├── Usuario no encontrado → AuthenticationException
        │
        ▼
  BCryptPasswordEncoder.matches(plano, hash)
        │
        ├── No coincide → CustomLoginPassword.onAuthenticationFailure()
        │       └── incrementa intentosFallidos
        │       └── si >= 3 → marca bloqueado=true
```



Ilustración 111. Manual técnico - Diagrama de Flujo de Autenticación

Fuente: Elaboración propia

11.2 Configuración de Spring Security

```
        .usernameParameter("email")
        .defaultSuccessUrl("/redirectByRole", true)
        .failureHandler(customLoginPassword)
    )
    .logout(logout -> logout
        .logoutUrl("/logout")
        .logoutSuccessUrl("/login?logout")
    )
    .csrf(csrf -> csrf.disable());     return
 http.build();
 }
```

Fuente: Elaboración propia

11.3 Cifrado de Contraseñas

SoftGold utiliza BCrypt con el factor de trabajo por defecto (10 rondas):

```
@Bean
public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {     http.authorizeHttpRequests(auth
```

```
  @Bean
  public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
  http.authorizeHttpRequests(auth -> auth
          // Rutas públicas
          .requestMatchers("/login", "/registro", "/recuperar/**", ...).permitAll()
  // Rutas por rol
          .requestMatchers("/foro/eliminar/**").hasRole("ADMINISTRADOR")
          .requestMatchers("/soporte/tickets/**").hasRole("ADMINISTRADOR")
          .requestMatchers("/admin/**").hasRole("ADMINISTRADOR")
          .requestMatchers("/minero/**").hasRole("MINERO")
          .requestMatchers("/empleado/**").hasRole("EMPLEADO")
          .requestMatchers("/usuario/**").hasRole("USUARIO")
          // Todo lo demás requiere autenticación
          .anyRequest().authenticated()
      )
      .formLogin(form -> form
          .loginPage("/login")
```

```
@Bean
public PasswordEncoder passwordEncoder() {     return new BCryptPasswordEncoder();
}
```

```
-> auth
        // Rutas públicas
        .requestMatchers("/login", "/registro", "/recuperar/**", ...).permitAll()
// Rutas por rol
        .requestMatchers("/foro/eliminar/**").hasRole("ADMINISTRADOR")
        .requestMatchers("/soporte/tickets/**").hasRole("ADMINISTRADOR")
        .requestMatchers("/admin/**").hasRole("ADMINISTRADOR")
        .requestMatchers("/minero/**").hasRole("MINERO")
        .requestMatchers("/empleado/**").hasRole("EMPLEADO")
        .requestMatchers("/usuario/**").hasRole("USUARIO")
        // Todo lo demás requiere autenticación
        .anyRequest().authenticated()
    )
    .formLogin(form -> form
        .loginPage("/login")
```

El hash generado tiene la forma: $2a$10$<salt><hash> (60 caracteres).

Ilustración 112. Manual técnico - Cifrado de contraseñas

Fuente: Elaboración propia

11.4 Bloqueo de Cuenta

El sistema implementa protección contra ataques de fuerza bruta:

Intentos Fallidos	Acción
1-2	Mensaje de error, contador incrementado
3+	Cuenta bloqueada, redirección a `/cuenta-bloqueada`

Tabla 102. Manual técnico - Bloqueo de cuenta

Fuente: Elaboración propia

El desbloqueo de cuentas debe hacerlo manualmente el administrador desde el panel de gestión de usuarios.

11.5 Recuperación de Contraseña

Flujo de recuperación implementado:

1. Usuario ingresa su email en /recuperar.
2. Sistema genera un token UUID y lo guarda en PASSWORDRESETTOKEN.
3. Se envía un email con el enlace http://host/cambiarPassword/{token}.
4. El token tiene una expiración configurada.
5. El usuario accede al enlace, ingresa la nueva contraseña y el sistema la actualiza con BCrypt.

11.6 Control de Roles y Autoridades

Los roles se almacenan con el prefijo ROLE_ en la base de datos. Spring Security los verifica mediante:

```
@Override
public Collection<? extends GrantedAuthority> getAuthorities() {    return roles.stream()
      .map(rol -> new SimpleGrantedAuthority(rol.getNombre()))
      .collect(Collectors.toList()); }
```

Ilustración 113. Manual técnico - Control de roles y autoridades

Fuente: Elaboración propia

La tabla de mapeo de roles es:

Rol en BD	Autoridad Spring Security	Acceso
ROLE_ADMINISTRADOR	hasRole("ADMINISTRADOR")	/admin/**, /soporte/tickets/**, /foro/eliminar/**
ROLE_MINERO	hasRole("MINERO")	/minero/**

ROLE_EMPLEADO	hasRole("EMPLEADO")	/empleado/**
ROLE_USUARIO	hasRole("USUARIO")	/usuario/**

Tabla 103. Manual técnico - Tabla de mapeo de roles

Fuente: Elaboración propia

11.7 Sesión y Timeout

La sesión HTTP expira automáticamente después de 15 minutos de inactividad:

```
server.servlet.session.timeout=15m
```

Adicionalmente, inactividad.js implementa un temporizador del lado del cliente que redirige al login después de inactividad.

12. Servicio de correo electrónico

12.1 Configuración SMTP

SoftGold utiliza Gmail SMTP con TLS para el envío de correos. Se requiere una contraseña de aplicación de Google (no la contraseña de la cuenta).

12.2 Uso del EmailService

```
@Service public class EmailService implements
EmailServiceIface {

    @Autowired     private
JavaMailSender mailSender;

    public void enviarEmailRecuperacion(String destinatario, String enlace)
{        SimpleMailMessage mensaje = new SimpleMailMessage();
mensaje.setTo(destinatario);        mensaje.setSubject("Recuperación de
contraseña - SoftGold");        mensaje.setText("Haga clic en el siguiente
enlace: " + enlace);        mailSender.send(mensaje);
    }
}
```

Ilustración 114. Manual técnico. Uso del EmailService

Fuente: Elaboración propia

12.3 Obtener Contraseña de Aplicación Google

1. Acceder a myaccount.google.com.
2. Ir a Seguridad > Verificación en dos pasos > Contraseñas de aplicaciones.
3. Crear una contraseña para "Correo" y "Windows/Mac".
4. Usar esa contraseña (formato: xxxx xxxx xxxx xxxx) en spring.mail.password.

13. Manejo de excepciones y Logs

13.1 GlobalControllerAdvice

La clase GlobalControllerAdvice.java provee:

- Atributos de modelo disponibles en todos los templates (usuario autenticado, roles, etc.)
- Manejo global de excepciones con @ExceptionHandler

13.2 Logging de SQL

El sistema registra todas las consultas SQL en modo debug:

```
logging.level.org.hibernate.SQL=debug
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE
```

Fuente: Elaboración propia

13.3 Logs de Spring Boot

Spring Boot genera logs automáticamente en consola. En producción se recomienda configurar Logback para persistir en archivos:

```
# Agregar en application.properties para producción logging.file.name=/var/log/softgold/softgold.log
logging.level.root=INFO
logging.level.com.proyectoL.softgold=DEBUG
```

Fuente: Elaboración propia

14. Validaciones

14.1 Validación de Entidades (Bean Validation)

SoftGold utiliza Jakarta Bean Validation con anotaciones estándar:

Anotación	Campo	Regla
`@NotBlank`	cedula, nombre1, apellido1, email	Campo obligatorio
`@Email`	email	Formato de email válido
`@Pattern`	passwordPlano	`^(?=.*[A-Z])(?=.*\d).{8,20}$`

Regla de contraseña: Mínimo 8 caracteres, máximo 20, al menos una mayúscula y un número.

Tabla 104. Manual técnico - Validación de entidades

Fuente: Elaboración propia

14.2 Validación en Controladores

```
@PostMapping("/crear")
public String crear(@Valid @ModelAttribute("mina") Mina mina,
                    BindingResult result,
Model model) {     if
(result.hasErrors()) {
        // Recargar datos necesarios para el formulario
model.addAttribute("mapas", mapaDAO.findAll());         return
"vistas/crearMina";  // Muestra errores en el template
    }
    minaDAO.save(mina);
return "redirect:/admin/minas";
}
```

Ilustración 115. Manual técnico - Validación en controladores

Fuente: Elaboración propia

14.3 Validación del Lado del Cliente

Los scripts JavaScript complementan la validación del servidor:

- registro.js: Valida formato de contraseña y coincidencia antes de enviar.
- cambiarPassword.js: Valida nueva contraseña al cambiarla.
- busqueda.js: Gestiona búsqueda dinámica en tiempo real.

15. Geolocalización y mapas

15.1 Integración de Leaflet.js

El sistema integra Leaflet.js v1.9.4 con tiles de OpenStreetMap (sin necesidad de API key).

Inclusión en el template base (principal.html):

```
<!-- CSS Leaflet (en fragmento head) -->
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css"/>

<!-- JS Leaflet (en fragmento footer) -->
<script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>
```

Ilustración 116. Manual técnico - Integración de Leaflet.js

Fuente: Elaboración propia

15.2 Inicialización Estándar del Mapa

```
const map = L.map('mapa-id').setView([latitud, longitud], zoom);

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
attribution: '© OpenStreetMap contributors' }).addTo(map);

// Añadir marcador
L.marker([latitud, longitud])
  .addTo(map)
  .bindPopup('<b>Nombre de la zona</b><br>Descripción');
```

Ilustración 117. Manual técnico - Inicialización estándar del mapa

Fuente: Elaboración propia

15.3 Clases CSS para Contenedores de Mapa

Clase CSS	Altura	Uso
`.sg-map-container`	350px	Mapas de detalle
`.sg-map-full`	580px	Mapa general de exploración

Tabla 105. Manual técnico - Clases CSS contenedores de mapa

Fuente: Elaboración propia

16. Procedimiento de despliegue

16.1 Construcción del Artefacto

```
# Limpiar y empaquetar
./mvnw clean package -DskipTests

# El JAR se genera en:
# target/softgold-0.0.1-SNAPSHOT.jar
```

Ilustración 118. Manual técnico - Construcción del artefacto

Fuente: Elaboración propia

16.2 Despliegue en Servidor Linux

```
scp target/softgold-0.0.1-SNAPSHOT.jar usuario@servidor:/opt/softgold/

# Crear archivo de configuración externo
nano /opt/softgold/application.properties

# Ejecutar como servicio java -jar
/opt/softgold/softgold-0.0.1-SNAPSHOT.jar \
  --spring.config.location=/opt/softgold/application.properties


# Copiar el JAR al servidor
```

Fuente: Elaboración propia

16.3 Configuración como Servicio system (Linux)

```
# /etc/systemd/system/softgold.service
[Unit]
Description=SoftGold Sistema de Gestión Minera
After=network.target mysql.service

[Service]
Type=simple
User=softgold
WorkingDirectory=/opt/softgold
ExecStart=/usr/bin/java -jar /opt/softgold/softgold-0.0.1-SNAPSHOT.jar
EnvironmentFile=/opt/softgold/softgold.env
Restart=on-failure
RestartSec=10

[Install]
WantedBy=multi-user.target
```

```
systemctl daemon-reload
systemctl enable softgold
systemctl start softgold
```

Ilustración 119. Manual técnico - Configuración como Servicio system (Linux)

Fuente: Elaboración propia

17. Buenas prácticas implementadas

Práctica	Implementación
Separation of Concerns	Capas Controller / Service / Repository claramente separadas
DRY (Don't Repeat Yourself)	Plantilla base `principal.html` con fragmentos Thymeleaf
Interfaz + Implementación	Todos los services tienen su interfaz (`*Iface`)
Validación en múltiples capas	Bean Validation + JavaScript del lado del cliente
Cifrado de contraseñas	BCrypt (nunca texto plano en BD)
Control de acceso por roles	RBAC con Spring Security
Manejo de estados HTTP	Redirección Post/Redirect/Get para prevenir reenvíos
Flash attributes	Mensajes de éxito/error sin repetir solicitudes
Portabilidad	Maven Wrapper permite ejecución sin instalar Maven

Tabla 106. Manual técnico - Buenas prácticas implementadas

Fuente: Elaboración propia

18. Estrategia de mantenimiento

18.1 Actualizaciones de Dependencias

- Revisar trimestralmente nuevas versiones de Spring Boot en start.spring.io.
- Ejecutar ./mvnw versions:display-dependency-updates para identificar dependencias desactualizadas.
- Actualizar Leaflet.js cuando haya versiones con correcciones de seguridad.

18.2 Respaldo de la Base de Datos

```
# Backup diario automatizado mysqldump -u root -p softgold >
backup_$(date +%Y%m%d).sql

# Restauración mysql -u root -p softgold <
backup_20260101.sql
```

Fuente: Elaboración propia

18.3 Monitoreo

- Verificar el estado del servicio: systemctl status softgold
- Revisar logs: tail -f /var/log/softgold/softgold.log
- Monitorear espacio en disco y memoria: df -h && free -m

19. Recomendaciones futuras

Prioridad	Mejora	Justificación
Alta	Migrar credenciales a variables de entorno	Seguridad en producción
Alta	Habilitar CSRF protection	Protección contra ataques CSRF
Alta	Implementar HTTPS/SSL	Cifrado de datos en tránsito
Media	Agregar paginación en tablas	Rendimiento con grandes volúmenes
Media	Implementar refresh tokens JWT	Sesiones más robustas
Media	Agregar pruebas unitarias e integración	Calidad y mantenibilidad
Baja	Integrar Swagger/OpenAPI	Documentación de API automática
Baja	Implementar caché (Redis)	Rendimiento en consultas frecuentes
Baja	Agregar soporte multiidioma (i18n)	Internacionalización
Baja	Exportación de reportes a PDF/Excel	Funcionalidad empresarial

Tabla 107. Manual técnico - Recomendaciones futuras

Fuente: Elaboración propia

20. Glosario técnico

Término	Definición
BCrypt	Algoritmo de hash adaptativo para contraseñas, resistente a fuerza
DAO	Data Access Object — interfaz de acceso a la base de datos
DDL	Data Definition Language — instrucciones SQL para defini estructuras
DevTools	Módulo de Spring Boot que activa recarga automática en desarrollo
JDBC	Java Database Connectivity — API de Java para conectarse datos

JPA	Jakarta Persistence API — especificación ORM estándar de Java
JWT	JSON Web Token — estándar de tokens de autenticación
Leaflet.js	Biblioteca JavaScript de código abierto para mapas interactivo
MVC	Model-View-Controller — patrón de diseño de software
ORM	Object-Relational Mapping — mapeo entre objetos Java y tablas BD
RBAC	Role-Based Access Control — control de acceso basado en roles
SMTP	Simple Mail Transfer Protocol — protocolo de envío de correos
Spring Security	Framework de seguridad para aplicaciones Spring
Thymeleaf	Motor de plantillas HTML del lado del servidor para Java
TLS	Transport Layer Security — protocolo de cifrado en red
UUID	Universally Unique Identifier — identificador único de 128 bits
UserDetails	Interfaz de Spring Security que representa al usuario autenticado

Tabla 108. Manual técnico - Glosario técnico

Fuente: Elaboración propia

Fin del Manual Técnico del Sistema SoftGold -- Versión 1.0.0

Manual de Implementacion y Despliegue

Sistema de Gestion Integral Minera

Version: 1.0.0

Fecha: Mayo 2026

Dirigido a: Equipo DevOps / Administradores

Estado: Documento oficial

Ambientes: Desarrollo | QA | Produccion

Infraestructura: Nginx | MySQL | systemd | Docker

Versión: 1.0.0
Fecha: Mayo 2026

Estado: Oficial

Dirigido a: Equipo DevOps / Administradores de Sistemas

Clasificación: Confidencial - Uso técnico interno

1. Introducción

Este documento describe los procedimientos técnicos necesarios para instalar, configurar y desplegar el sistema SoftGold en entornos de desarrollo, pruebas (QA) y producción.

SoftGold es una aplicación Java/Spring Boot que se ejecuta como un proceso autónomo en el puerto 9090, con dependencia de una base de datos MySQL 8.x. El sistema no requiere un servidor de aplicaciones externo (Tomcat se incluye embebido en Spring Boot).

1.1 Ambientes de Despliegue

Ambiente	Propósito	URL típica
Desarrollo (DEV)	Pruebas locales del desarrollador	http://localhost:9090
Pruebas (QA)	Validación y control de calidad	http://qa.softgold.empresa
Producción (PROD)	Sistema en uso real	https://softgold.empresa.com

Tabla 109. Manual implementación - Ambientes de despliegue

Fuente: Elaboración propia

1.2 Estrategia de Configuración

Cada ambiente mantiene su propio archivo de propiedades, que sobreescribe los valores del application.properties base mediante perfiles de Spring:

```
application.properties          ← Configuración base común application-
dev.properties      ← Sobrescrituras para desarrollo application-qa.properties
← Sobrescrituras para QA application-prod.properties      ← Sobrescrituras para
producción
```

Ilustración 120. Manual implementación - Estrategia de configuración

Fuente: Elaboración propia

2. Requisitos del servidor

2.1 Requisitos de Hardware (Producción)

Componente	Mínimo	Recomendado
CPU	2 núcleos / 2.0 GHz	4 núcleos / 3.0 GHz
Memoria RAM	4 GB	8 GB
Almacenamiento	40 GB SSD	100 GB SSD
Ancho de banda	10 Mbps	100 Mbps

Tabla 110. Manual implementación - Requisitos de Hardware (Producción)

Fuente: Elaboración propia

2.2 Requisitos de Software del Servidor

Software	Versión	Propósito
Sistema operativo	Ubuntu 22.04 LTS / Debian 12 / CentOS 8	SO base
Java (JDK)	17 LTS (OpenJDK o Eclipse Temurin)	Ejecución de la aplicación
MySQL	8.0+	Base de datos
Nginx	1.24+	Proxy inverso / SSL termination
Maven	3.8+	Build del proyecto (si se compila en servidor)
Docker	24+ (opcional)	Contenedores
Docker Compose	2.x (opcional)	Orquestación de contenedores

Tabla 111. Manual implementación - Requisitos de Software del Servidor

Fuente: Elaboración propia

2.3 Puertos Requeridos

Puerto	Protocolo	Servicio
80	TCP	HTTP (Nginx → redirige a HTTPS)
443	TCP	HTTPS (Nginx SSL)
9090	TCP	SoftGold (interno, no exponer al exterior)
3306	TCP	MySQL (solo acceso local)

NOTA: SEGURIDAD: El puerto 9090 de SoftGold y el 3306 de MySQL NO deben estar expuestos directamente a internet. Solo Nginx (puertos 80/443) debe estar visible externamente.

Tabla 112. Manual implementación - Puertos requeridos

Fuente: Elaboración propia

3. Preparación del entorno

3.1 Actualizar el Sistema Operativo

```
# Ubuntu/Debian sudo apt update && sudo apt upgrade -y

# CentOS/RHEL
sudo dnf update -y
```

Fuente: Elaboración propia

3.2 Crear Usuario de Servicio

```
# Crear usuario dedicado para SoftGold (sin shell de login)
sudo useradd -r -s /bin/false -d /opt/softgold softgold

# Crear directorio de instalación sudo
mkdir -p /opt/softgold sudo chown
softgold:softgold /opt/softgold

# Crear directorio de logs sudo mkdir -p
/var/log/softgold sudo chown
softgold:softgold /var/log/softgold
```

Ilustración 121. Manual implementación - Crear Usuario de Servicio

Fuente: Elaboración propia

3.3 Configurar Firewall

```
# Ubuntu con UFW sudo ufw
allow 22/tcp    # SSH sudo ufw
allow 80/tcp    # HTTP sudo ufw
allow 443/tcp   # HTTPS sudo ufw
enable

# Verificar reglas
sudo ufw status
```

Ilustración 122. Manual implementación - Configurar Firewall

Fuente: Elaboración propia

4. Instalación de Java y Maven

4.1 Instalar OpenJDK 17

```
# Ubuntu/Debian sudo apt install
-y openjdk-17-jdk

# CentOS/RHEL
sudo dnf install -y java-17-openjdk-devel

# Verificar instalación
java -version
# Salida esperada: openjdk version "17.x.x" ...

javac -version
# Salida esperada: javac 17.x.x
```

Ilustración 123. Manual implementación - Instalar OpenJDK 17

Fuente: Elaboración propia

4.2 Configurar Java Home

```
# Agregar a /etc/profile.d/java.sh sudo bash -c 'cat >
/etc/profile.d/java.sh << EOF export JAVA_HOME=$(dirname
$(dirname $(readlink -f $(which java)))) export
PATH=$PATH:$JAVA_HOME/bin EOF'

source /etc/profile.d/java.sh
echo $JAVA_HOME
```

Fuente: Elaboración propia

4.3 Instalar Maven (para compilación en servidor)

```
# Ubuntu/Debian sudo apt
install -y maven

# Verificar mvn -
version
# Apache Maven 3.x.x
```

NOTA: ALTERNATIVA: Si prefiere compilar en un servidor de CI/CD y solo desplegar el JAR, Maven no necesita instalarse en el servidor de producción.

Ilustración 124. Manual implementación - Instalar Maven (para compilación en servidor)

Fuente: Elaboración propia

5. Instalación y configuración de MySQL

5.1 Instalar MySQL 8

```
# Ubuntu/Debian sudo apt
install -y mysql-server

# CentOS/RHEL
sudo dnf install -y mysql-server

# Iniciar y habilitar el servicio
sudo systemctl start mysql
sudo systemctl enable mysql

# Verificar estado sudo
systemctl status mysql
```

Ilustración 125. Manual implementación - Instalación MySQL

Fuente: Elaboración propia

5.2 Asegurar la Instalación de MySQL

```
sudo mysql_secure_installation
```

Responda las preguntas:

- Validate password plugin: YES
- Nivel de contraseña: 2 (STRONG)
- Remove anonymous users: YES
- Disallow root login remotely: YES
- Remove test database: YES
- Reload privilege tables: YES

-

5.3 Crear Base de Datos y Usuario para SoftGold

```
-- Acceder a MySQL como root
sudo mysql -u root -p

-- Crear la base de datos
CREATE DATABASE softgold
  CHARACTER SET utf8mb4
  COLLATE utf8mb4_unicode_ci;

-- Crear usuario dedicado para la aplicación
CREATE USER 'softgold_app'@'localhost' IDENTIFIED BY 'ContraseñaSegura2026!';

-- Otorgar permisos
GRANT ALL PRIVILEGES ON softgold.* TO 'softgold_app'@'localhost';
FLUSH PRIVILEGES;

-- Verificar
SHOW GRANTS FOR 'softgold_app'@'localhost';

EXIT;
```

Fuente: Elaboración propia

5.4 Verificar Configuración de MySQL

```
# Probar conexión con el nuevo usuario
mysql -u softgold_app -p softgold
# Ingresar la contraseña cuando se solicite
# Salida esperada: Welcome to the MySQL monitor...
```

Ilustración 126. Manual implementación - Verificación configuración MySQL

Fuente: Elaboración propia

6. Instalación del Backend (SoftGold)

6.1 Compilar el Proyecto (en servidor de CI o local)

```
# Clonar el repositorio git clone <url-repositorio>
softgold_final cd softgold_final

# Compilar y generar el JAR
./mvnw clean package -DskipTests

# El artefacto se genera en: # target/softgold-0.0.1-
SNAPSHOT.jar
```

Fuente: Elaboración propia

6.2 Transferir el JAR al Servidor

```
# Desde el equipo de desarrollo (o CI/CD)
scp target/softgold-0.0.1-SNAPSHOT.jar \
usuario@servidor:/opt/softgold/softgold.jar
```

Fuente: Elaboración propia

6.3 Verificar el JAR

```
# En el servidor ls -lh
/opt/softgold/softgold.jar
# Debe mostrar el archivo con su tamaño
```

```
# Verificar que es un JAR ejecutable válido java -jar
/opt/softgold/softgold.jar --version 2>&1 | head -5
```

Fuente: Elaboración propia

7. Configuración variables de entorno

7.1 Archivo de Variables de Entorno

Crear el archivo /opt/softgold/softgold.env con las configuraciones específicas del ambiente:

```
sudo nano /opt/softgold/softgold.env
```

Contenido del archivo:

```
# Puerto del servidor
SERVER_PORT=9090

# Base de datos
DB_URL=jdbc:mysql://localhost:3306/softgold?serverTimezone=America/Bogota&createDatabaseIfNotExist
=true
DB_USERNAME=softgold_app
# Puerto server.port=${SERVER_PORT:9090}
# Base de datos spring.datasource.url=${DB_URL} spring.datasource.username=${DB_USERNAME}
spring.datasource.password=${DB_PASSWORD}

# JPA (producción: no auto-modificar esquema) spring.jpa.hibernate.ddl-auto=validate
# Mail


spring.mail.host=${MAIL_HOST}
```

```
DB_PASSWORD=ContraseñaSegura2026!

# Correo SMTP
MAIL_HOST=smtp.gmail.com
```

```
MAIL_PORT=587
MAIL_USERNAME=correo@gmail.com
MAIL_PASSWORD=xxxx xxxx xxxx xxxx

# Perfil de Spring
SPRING_PROFILES_ACTIVE=prod

# Logging
LOG_LEVEL=INFO
LOG_FILE=/var/log/softgold/softgold.log
```

```
# Asegurar permisos restrictivos (solo root y el servicio pueden leerlo)
sudo chmod 600 /opt/softgold/softgold.env sudo chown root:softgold
/opt/softgold/softgold.env
```

Ilustración 127. Manual implementación - Archivo de variables de entorno

Fuente: Elaboración propia

7.2 Archivo application-prod.properties

Crear el archivo de configuración de producción en /opt/softgold/config/application-prod.properties:

```
spring.mail.port=${MAIL_PORT}
spring.mail.username=${MAIL_USERNAME}
spring.mail.password=${MAIL_PASSWORD}
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true

# Logging (producción) logging.level.root=WARN
logging.level.com.proyectoL.softgold=INFO
logging.file.name=${LOG_FILE:/var/log/softgold/softgold.log}
logging.file.max-size=50MB logging.file.max-history=30

# Sesión
 -- Esquema de SoftGold (referencia)
 -- Ejecutar en MySQL como softgold_app

 USE softgold;

 CREATE TABLE IF NOT EXISTS roles (   id
 BIGINT AUTO_INCREMENT PRIMARY KEY,
 nombre VARCHAR(50) NOT NULL UNIQUE
 );

 CREATE TABLE IF NOT EXISTS minas (   cod_mina
 BIGINT AUTO_INCREMENT PRIMARY KEY,   nombre
 VARCHAR(255) NOT NULL,   departamento
 VARCHAR(255) NOT NULL
 );

 CREATE TABLE IF NOT EXISTS mapas (   codigo_mapa
 BIGINT AUTO_INCREMENT PRIMARY KEY,   titulo
```

```
VARCHAR(255),   descripcion TEXT,   latitud DOUBLE,
longitud DOUBLE,

server.servlet.session.timeout=15m
```

ADVERTENCIA: En producción, cambiar spring.jpa.hibernate.ddl-auto de update a validate. Esto previene modificaciones accidentales al esquema de la base de datos.

Fuente: Elaboración propia

8. Configuración de la base de datos

8.1 Migraciones y Esquema Inicial

Hibernate con ddl-auto=update crea automáticamente las tablas en el primer arranque. Para inicializar el sistema:

Primera vez (DEV/QA): Ejecutar la aplicación con ddl-auto=update para crear el esquema.

Producción: El esquema debe crearse manualmente o mediante un script SQL antes del primer arranque.

8.2 Script SQL de Creación del Esquema

```
  coordenadas TEXT
);

CREATE TABLE IF NOT EXISTS riesgos (   cod_riesgo
BIGINT AUTO_INCREMENT PRIMARY KEY,
descripcion TEXT NOT NULL
);

CREATE TABLE IF NOT EXISTS usuarios (   id
BIGINT AUTO_INCREMENT PRIMARY KEY,   cedula
VARCHAR(50) NOT NULL UNIQUE,
tipo_documento VARCHAR(10) NOT NULL,
nombre_1 VARCHAR(100) NOT NULL,   nombre_2
VARCHAR(100),   apellido_1 VARCHAR(100) NOT
NULL,   apellido_2 VARCHAR(100),   email
VARCHAR(255) NOT NULL UNIQUE,   password
VARCHAR(255) NOT NULL,   tipo_usuario
VARCHAR(50) NOT NULL,   telefono VARCHAR(20),
tipo_empleado VARCHAR(100),   area
VARCHAR(100),   intentos_fallidos INT NOT NULL
DEFAULT 0,   bloqueado BOOLEAN NOT NULL
DEFAULT FALSE,   tiempo_bloqueo DATETIME,
cod_mina BIGINT,
  FOREIGN KEY (cod_mina) REFERENCES minas(cod_mina) ON DELETE SET NULL
```

```
);

CREATE TABLE IF NOT EXISTS usuario_roles (
usuario_id BIGINT NOT NULL,   rol_id BIGINT
NOT NULL,
  PRIMARY KEY (usuario_id, rol_id),
  FOREIGN KEY (usuario_id) REFERENCES usuarios(id) ON DELETE CASCADE,
  FOREIGN KEY (rol_id) REFERENCES roles(id) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS mina_mapas (
mina_cod_mina BIGINT NOT NULL,
mapas_codigo_mapa BIGINT NOT NULL,
  PRIMARY KEY (mina_cod_mina, mapas_codigo_mapa),
  FOREIGN KEY (mina_cod_mina) REFERENCES minas(cod_mina),
  FOREIGN KEY (mapas_codigo_mapa) REFERENCES mapas(codigo_mapa)
);

CREATE TABLE IF NOT EXISTS mina_riesgos (
mina_cod_mina BIGINT NOT NULL,
riesgos_cod_riesgo BIGINT NOT NULL,
  PRIMARY KEY (mina_cod_mina, riesgos_cod_riesgo),
  FOREIGN KEY (mina_cod_mina) REFERENCES minas(cod_mina),
  FOREIGN KEY (riesgos_cod_riesgo) REFERENCES riesgos(cod_riesgo)
);

CREATE TABLE IF NOT EXISTS zona_exploracion (
id BIGINT AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(255) NOT NULL,   descripcion
TEXT,   latitud DOUBLE,

  longitud DOUBLE,   tipo
VARCHAR(100),   estado
VARCHAR(50),
fecha_registro DATETIME,
mina_id BIGINT,
  FOREIGN KEY (mina_id) REFERENCES minas(cod_mina) ON DELETE SET NULL
);

CREATE TABLE IF NOT EXISTS forum_posts (
id BIGINT AUTO_INCREMENT PRIMARY KEY,
titulo VARCHAR(500) NOT NULL,   contenido
TEXT NOT NULL,   categoria VARCHAR(100),
autor_nombre VARCHAR(255),   fecha_creacion
DATETIME,   activo BOOLEAN DEFAULT TRUE
);

CREATE TABLE IF NOT EXISTS forum_comments (
id BIGINT AUTO_INCREMENT PRIMARY KEY,
contenido TEXT NOT NULL,   autor_nombre
VARCHAR(255),   fecha_creacion DATETIME,
post_id BIGINT,
  FOREIGN KEY (post_id) REFERENCES forum_posts(id) ON DELETE CASCADE
);
```

```
-- Contraseña: admin123 (hash BCrypt)

CREATE TABLE IF NOT EXISTS support_tickets (   id
BIGINT AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(255) NOT NULL,   email
VARCHAR(255) NOT NULL,   asunto VARCHAR(500)
NOT NULL,   descripcion TEXT NOT NULL,   creado
DATETIME NOT NULL,   estado VARCHAR(20) NOT
NULL DEFAULT 'OPEN'
);

CREATE TABLE IF NOT EXISTS password_reset_tokens (
id BIGINT AUTO_INCREMENT PRIMARY KEY,   token
VARCHAR(255) NOT NULL UNIQUE,   usuario_id BIGINT
NOT NULL,   fecha_expiracion DATETIME NOT NULL,
  FOREIGN KEY (usuario_id) REFERENCES usuarios(id) ON DELETE CASCADE
);

-- Insertar roles iniciales
INSERT IGNORE INTO roles (nombre) VALUES
  ('ROLE_ADMINISTRADOR'),
  ('ROLE_MINERO'),
  ('ROLE_EMPLEADO'),
  ('ROLE_USUARIO');
```

Fuente: Elaboración propia

8.3 Datos Iniciales del Administrador

El DataInitializer.java crea automáticamente el administrador al arrancar. Si necesita hacerlo manualmente:

```
-- IMPORTANTE: Cambiar en producción
INSERT INTO usuarios (cedula, tipo_documento, nombre_1, apellido_1, email, password, tipo_usuario, intentos_fa
VALUES ('0000000000', 'CC', 'Admin', 'SoftGold', 'admin@softgold.com',
       '$2a$10$N9qo8uLOickgx2ZMRZoMyeIjZAgcfl7p92ldGxad68LJZdL17lhWy',
       'ADMINISTRADOR', 0, false);

-- Asignar rol administrador (ajustar IDs según corresponda)
INSERT INTO usuario_roles (usuario_id, rol_id)
SELECT u.id, r.id FROM usuarios u, roles r
WHERE u.email = 'admin@softgold.com' AND r.nombre = 'ROLE_ADMINISTRADOR';
  # Activar el sitio
```

Fuente: Elaboración propia

9. Configuración del servidor web (NGINX)

Nginx actúa como proxy inverso, redirigiendo el tráfico externo al puerto 9090 de SoftGold.

9.1 Instalar Nginx

```
# Ubuntu/Debian sudo
apt install -y nginx

# Iniciar y habilitar sudo
systemctl start nginx sudo
systemctl enable nginx
```

Ilustración 128. Manual implementación - Instalar Nginx

Fuente: Elaboración propia

9.2 Configuración de Nginx (HTTP)

```
sudo nano /etc/nginx/sites-available/softgold
```

```
server {
listen 80;
    server_name softgold.empresa.com;

    # Redirigir HTTP a HTTPS (activar después de configurar SSL)
    # return 301 https://$host$request_uri;

    # Configuración temporal sin SSL     location / {         proxy_pass
http://localhost:9090;         proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;         proxy_set_header X-
Forwarded-For $proxy_add_x_forwarded_for;         proxy_set_header
X-Forwarded-Proto $scheme;         proxy_connect_timeout 60s;
proxy_read_timeout 60s;         proxy_send_timeout 60s;
client_max_body_size 10M;
    }

    # Logs     access_log
/var/log/nginx/softgold_access.log;     error_log
/var/log/nginx/softgold_error.log; }

sudo ln -s /etc/nginx/sites-available/softgold /etc/nginx/sites-enabled/

  # Verificar configuración
  sudo nginx -t

  # Recargar Nginx sudo
  systemctl reload nginx
```

Ilustración 129. Manual implementación - Configuración de Nginx (HTTP)

Fuente: Elaboración propia

10. Configuración SSL/TLS

10.1 Instalar Certbot (Let's Encrypt -- Gratuito)

```
sudo apt install -y certbot python3-certbot-nginx

# Obtener certificado SSL sudo certbot --
nginx -d softgold.empresa.com

# Verificar renovación automática
sudo certbot renew --dry-run
```

Ilustración 130. Manual implementación - Instalar Certbot

Fuente: Elaboración propia

10.2 Configuración Nginx con SSL

Después de que Certbot configure SSL automáticamente, verificar que el archivo /etc/nginx/sites-available/softgold contenga:

```
server {    listen 443 ssl;
server_name softgold.empresa.com;

   ssl_certificate /etc/letsencrypt/live/softgold.empresa.com/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/softgold.empresa.com/privkey.pem;
ssl_protocols TLSv1.2 TLSv1.3;    ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;    ssl_prefer_server_ciphers on;

   location / {        proxy_pass http://localhost:9090;
proxy_set_header Host $host;        proxy_set_header X-Real-IP
$remote_addr;        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;        proxy_set_header X-Forwarded-
Proto https;        proxy_connect_timeout 60s;        proxy_read_timeout
60s;        client_max_body_size 10M;
   }

   access_log /var/log/nginx/softgold_access.log;
error_log /var/log/nginx/softgold_error.log; }

# Redirección HTTP → HTTPS
server {    listen 80;
   server_name softgold.empresa.com;
return 301 https://$host$request_uri;}
```

Ilustración 131. Manual implementación - Configuración Nginx con SSL

Fuente: Elaboración propia

11. Configuración Docker

Si prefiere desplegar SoftGold usando contenedores Docker:

11.1 Dockerfile

```
# Dockerfile
```

```
# docker-compose.yml
version: '3.8'

services:
mysql:
    image: mysql:8.0
container_name: softgold-mysql
restart: unless-stopped
environment:
      MYSQL_DATABASE: softgold
      MYSQL_USER: softgold_app
      MYSQL_PASSWORD: ${DB_PASSWORD}
      MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
volumes:
-          mysql_data:/var/lib/mysql
-          ./init.sql:/docker-entrypoint-initdb.d/init.sql
networks:
-          softgold-network    healthcheck:       test:
["CMD", "mysqladmin", "ping", "-h", "localhost"]
timeout: 10s       retries: 5

  softgold:
build: .
    container_name: softgold-app
restart: unless-stopped
depends_on:       mysql:
        condition: service_healthy
environment:
      SPRING_DATASOURCE_URL: jdbc:mysql://mysql:3306/softgold?serverTimezone=America/Bogota
```

```
FROM eclipse-temurin:17-jdk-alpine

WORKDIR /app

COPY target/softgold-0.0.1-SNAPSHOT.jar softgold.jar

EXPOSE 9090

ENTRYPOINT ["java", \
  "-Djava.security.egd=file:/dev/./urandom", \
  "-jar", "softgold.jar"]
```

Ilustración 132. Manual implementación – Dockerfile

Fuente: Elaboración propia

```
# .github/workflows/deploy.yml
name: Build and Deploy SoftGold

on:
push:
  branches: [main]

jobs:
```

```
DB_PASSWORD=ContraseñaSegura2026!
MYSQL_ROOT_PASSWORD=RootPasswordSeguro2026!
MAIL_USERNAME=correo@gmail.com MAIL_PASSWORD=xxxx xxxx xxxx xxxx
```

11.2 Docker Compose

```
    SPRING_DATASOURCE_USERNAME: softgold_app
    SPRING_DATASOURCE_PASSWORD: ${DB_PASSWORD}
    SPRING_MAIL_USERNAME: ${MAIL_USERNAME}
    SPRING_MAIL_PASSWORD: ${MAIL_PASSWORD}
SPRING_PROFILES_ACTIVE: prod  ports:  -
"9090:9090"  networks:  - softgold-network
volumes:
  - softgold_logs:/var/log/softgold

networks:  softgold-
network:  driver:
bridge

volume:
mysql_data:
softgold_logs:
```

Fuente: Elaboración propia

11.3 Archivo .env para Docker Compose

```
# .env (no commitear a git)
```

Ilustración 133. Manual implementación - Archivo .env para Docker Compose

Fuente: Elaboración propia

11.4 Comandos Docker

```
# Construir y levantar docker
compose up -d --build

# Ver logs docker compose
logs -f softgold

# Detener docker
compose down

# Detener y eliminar volúmenes (cuidado en producción)
docker compose down -v
```

Ilustración 134. Manual implementación - Comandos Docker

Fuente: Elaboración propia

12. Configuración CI/CD

12.1 Pipeline GitHub Actions (Ejemplo)

```
  build:
    runs-on: ubuntu-latest
steps:
  -         uses: actions/checkout@v4

  -         name: Set up JDK 17
  uses: actions/setup-java@v4
  with:
        java-version: '17'
distribution: 'temurin'

  -         name: Cache Maven packages
  uses: actions/cache@v3        with:
        path: ~/.m2          key: ${{ runner.os }}-m2-${{
hashFiles('**/pom.xml') }}

  -         name: Build with Maven
  run: ./mvnw clean package -
  DskipTests

  -         name: Upload JAR       uses:
  actions/upload-artifact@v4        with:
        name: softgold-jar          path:
target/softgold-0.0.1-SNAPSHOT.jar

  deploy:
```

```
    needs: build     runs-on: ubuntu-
  latest     if: github.ref ==
  'refs/heads/main'     steps:
    -          name: Download JAR
    uses: actions/download-artifact@v4
    with:
          name: softgold-jar


    -          name: Deploy to server
    uses: appleboy/scp-action@master
    with:
          host: ${{ secrets.SERVER_HOST }}
    username: ${{ secrets.SERVER_USER }}
    key: ${{ secrets.SSH_PRIVATE_KEY }}
    source: "softgold-0.0.1-SNAPSHOT.jar"
    target: "/opt/softgold/"

    -          name: Restart service
    uses: appleboy/ssh-action@master
    with:
          host: ${{ secrets.SERVER_HOST }}
    username: ${{ secrets.SERVER_USER }}
    key: ${{ secrets.SSH_PRIVATE_KEY }}
          script: |
            cp /opt/softgold/softgold-0.0.1-SNAPSHOT.jar /opt/softgold/softgold.jar
    sudo systemctl restart softgold             sleep 15             systemctl is-active
    softgold
```

Fuente: Elaboración propia

13. Proceso de despliegue

13.1 Configurar SoftGold como Servicio systemd

```
sudo nano /etc/systemd/system/softgold.service
```

```
[Unit]
Description=SoftGold - Sistema de Gestión Minera
Documentation=https://softgold.empresa.com/docs
After=network.target mysql.service Wants=mysql.service

[Service]
Type=simple
User=softgold
Group=softgold
WorkingDirectory=/opt/softgold

# Cargar variables de entorno
EnvironmentFile=/opt/softgold/softgold.env

# Comando de arranque ExecStart=/usr/bin/java \
  -Xms256m \
```

```
  -Xmx512m \
  -jar /opt/softgold/softgold.jar \

  --spring.config.additional-location=/opt/softgold/config/

# Reinicio automático ante fallos
Restart=on-failure
RestartSec=15
StartLimitIntervalSec=60
StartLimitBurst=3

# Estándar de salida
StandardOutput=append:/var/log/softgold/softgold.log
StandardError=append:/var/log/softgold/softgold.log

# Seguridad del proceso
NoNewPrivileges=yes
PrivateTmp=yes

[Install]
WantedBy=multi-user.target
```

```
# Activar e iniciar el servicio
sudo systemctl daemon-reload
sudo systemctl enable softgold
sudo systemctl start softgold

# Verificar estado sudo
systemctl status softgold
```

Ilustración 135. Manual implementación - Configurar SoftGold como Systemd

Fuente: Elaboración propia

13.2 Verificar el Arranque

```
  # Ver los últimos logs de arranque sudo
  journalctl -u softgold -n 50 --no-pager

  # El arranque exitoso mostrará algo similar a:
  # Tomcat started on port 9090 (http) with context path ''
  # Started SoftgoldApplication in X.XXX seconds
```

Fuente: Elaboración propia

13.3 Probar la Aplicación

```
# Prueba de conectividad local curl -I http://localhost:9090/login

# Respuesta esperada:
# HTTP/1.1 200
```

Ilustración 136. Manual implementación - Prueba de la aplicación

Fuente: Elaboración propia

14. Configuración de Backups

14.1 Script de Backup de Base de Datos

```
sudo nano /opt/softgold/scripts/backup_db.sh
```

```
#!/bin/bash
# backup_db.sh — Backup diario de la base de datos SoftGold

BACKUP_DIR="/var/backups/softgold"
DATE=$(date +%Y%m%d_%H%M%S)
BACKUP_FILE="$BACKUP_DIR/softgold_$DATE.sql.gz"
RETENTION_DAYS=30

# Cargar variables source
/opt/softgold/softgold.env

# Crear directorio si no existe
mkdir -p "$BACKUP_DIR"

# Realizar backup con compresión
mysqldump \
  -u "$DB_USERNAME" \
  -p"$DB_PASSWORD" \
  --single-transaction \
  --routines \
  --triggers \
  softgold | gzip > "$BACKUP_FILE"

# Verificar éxito if [ $? -eq 0 ]; then     echo "$(date): Backup exitoso:
$BACKUP_FILE" >> /var/log/softgold/backup.log else     echo "$(date): ERROR
en backup" >> /var/log/softgold/backup.log
fi

# Eliminar backups más antiguos que RETENTION_DAYS días find "$BACKUP_DIR" -
name "softgold_*.sql.gz" -mtime +"$RETENTION_DAYS" -delete

  # Dar permisos de ejecución chmod +x
  /opt/softgold/scripts/backup_db.sh

  # Programar con cron (backup diario a las 2:00 AM) sudo crontab -e -u
softgold Agregar la línea:

  0 2 * * * /opt/softgold/scripts/backup_db.sh
```

Ilustración 137. Manual implementación - Script de Backup de Base de Datos

Fuente: Elaboración propia

14.2 Script de Backup del JAR

```
#!/bin/bash
# backup_jar.sh — Backup del JAR antes de actualizar

cp /opt/softgold/softgold.jar \
   /opt/softgold/backups/softgold_$(date +%Y%m%d_%H%M%S).jar
```

Ilustración 138. Manual implementación - Script de Backup del JAR

Fuente: Elaboración propia

14.3 Estrategia de Retención

Tipo de Backup	Frecuencia	Retención
Base de datos	Diario	30 días
Base de datos (semanal)	Semanal	3 meses
JAR de la aplicación	Por despliegue	Últimas 5 versiones
Configuración	Por cambio	Indefinida (en Git)

Tabla 113. Manual implementación - Estrategia de retención

Fuente: Elaboración propia

15. Monitoreo del sistema

15.1 Estado del Servicio

```
# Estado general sudo
systemctl status softgold

# Logs en tiempo real sudo
journalctl -u softgold -f

# Últimas 100 líneas del log tail -100
/var/log/softgold/softgold.log
```

Ilustración 139. Manual implementación - Estado del servicio

Fuente: Elaboración propia

15.2 Monitoreo de Recursos

```
# Uso de CPU y memoria por el proceso
ps aux | grep softgold

# Memoria del sistema
free -h

# Espacio en disco
df -h

# Conexiones activas al puerto 9090
  ss -tlnp | grep 9090
```

Ilustración 140. Manual implementación - Monitoreo de recursos

Fuente: Elaboración propia

15.3 Monitoreo de la Base de Datos

```
-- Conexiones activas
SHOW PROCESSLIST;

-- Tamaño de la base de datos
SELECT
  table_schema AS 'Base de Datos',
  ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS 'Tamaño (MB)'
FROM information_schema.tables
WHERE table_schema = 'softgold'
GROUP BY table_schema;
```

Ilustración 141. Manual implementación - Monitoreo de la base de datos

Fuente: Elaboración propia

15.4 Script de Monitoreo de Salud

```
#!/bin/bash
# health_check.sh

URL="http://localhost:9090/login"
HTTP_STATUS=$(curl -s -o /dev/null -w "%{http_code}" "$URL")

if [ "$HTTP_STATUS" != "200" ]; then     echo "$(date): ALERTA - SoftGold no
responde (HTTP $HTTP_STATUS)" \
    >> /var/log/softgold/health.log
# Opcional: reiniciar el servicio
sudo systemctl restart softgold fi
```

```
# Cron: verificar cada 5 minutos
*/5 * * * * /opt/softgold/scripts/health_check.sh
```

Ilustración 142. Manual implementación - Script de Monitoreo de Salud

Fuente: Elaboración propia

16. Validación post-despliegue

Después de cada despliegue, ejecutar las siguientes verificaciones:

16.1 Verificaciones Técnicas

```
# 1. Servicio activo
systemctl is-active softgold
# Esperado: active

# 2. Puerto escuchando
ss -tlnp | grep 9090
# Esperado: LISTEN ... :9090

# 3. Respuesta HTTP curl -s -o /dev/null -w "%{http_code}"
http://localhost:9090/login # Esperado: 200

# 4. Sin errores críticos en logs grep -c
"ERROR" /var/log/softgold/softgold.log #
Idealmente: 0
```

```
# 5. Conexión a base de datos (debe aparecer en logs de arranque) grep "HikariPool"
/var/log/softgold/softgold.log | tail -5
```

Ilustración 143. Manual implementación - Verificaciones Técnicas

Fuente: Elaboración propia

16.2 Verificaciones Funcionales

Verificación	Pasos	Resultado Esperado
Login	Acceder a `/login` e ingresar con admin	Redirige al dashboard
Listado de minas	Acceder a `/admin/minas`	Muestra tabla de minas
Creación de entidad	Crear una mina de prueba	Mensaje de éxito
Foro	Acceder a `/foro`	Carga lista de posts
Mapas	Acceder a un mapa Leaflet	Mapa interactivo visible
Correo	Usar recuperación de contraseña	Email recibido
Logout	Cerrar sesión	Redirige a login

Tabla 114. Manual implementación - Verificaciones funcionales

Fuente: Elaboración propia

17. Checklist de implementación

17.1 Checklist Pre-Despliegue

- Servidor con OS actualizado
- Java 17 instalado y verificado
- MySQL 8 instalado y configurado
- Usuario softgold_app creado en MySQL con permisos adecuados
- Base de datos softgold creada

- Firewall configurado (puertos 80, 443, 22)
- Nginx instalado y configurado
- Certificado SSL obtenido (producción)
- Directorio /opt/softgold creado con permisos correctos
- Usuario de sistema softgold creado
- Archivo softgold.env creado con permisos 600
- Archivo application-prod.properties configurado
- Scripts de backup creados y cron configurado
- JAR compilado y transferido al servidor
- Servicio systemd creado y habilitado

17.2 Checklist Post-Despliegue

- Servicio arranca sin errores
- Se puede acceder a la URL del sistema
- Login con credenciales de administrador funciona
- Todos los módulos cargan correctamente
- Mapas Leaflet.js renderizan correctamente
- Formularios guardan datos en la base de datos
- Servicio de correo funciona (test de recuperación de contraseña)
- Timeout de sesión funciona a los 15 minutos

- Logs se están generando correctamente
- Backup cron está configurado
- Contraseña del administrador por defecto fue cambiada
- Credenciales de development removidas de production

18. Procedimiento de actualización

Ilustración 144. Manual implementación - Procedimiento de actualización

Fuente: Elaboración propia

18.1 Pasos para Actualizar SoftGold

```
# 1. Detener la versión problemática
sudo systemctl stop softgold

# 2. Restaurar el JAR anterior cp
/opt/softgold/backups/softgold_prev.jar /opt/softgold/softgold.jar

# 3. Si la base de datos fue modificada, restaurar backup
sudo systemctl stop mysql
# (Restaurar backup de BD si aplica) mysql -u softgold_app -p softgold <
/var/backups/softgold/softgold_YYYYMMDD.sql sudo systemctl start mysql
```

```
# 1. Hacer backup previo
/opt/softgold/scripts/backup_db.sh cp /opt/softgold/softgold.jar
/opt/softgold/backups/softgold_prev.jar

# 2. Detener el servicio
sudo systemctl stop softgold

# 3. Reemplazar el JAR cp /ruta/al/nuevo/softgold-nueva-version.jar
/opt/softgold/softgold.jar

# 4. Iniciar el servicio sudo
systemctl start softgold

# 5. Verificar arranque (esperar ~30 segundos)
sleep 30 sudo systemctl status softgold

# 6. Validar funcionamiento curl -s -o /dev/null -w
"%{http_code}" http://localhost:9090/login
```

Ilustración 145. Manual implementación - Pasos para Actualizar SoftGold

Fuente: Elaboración propia

18.2 Tiempo de Indisponibilidad Estimado

Fase	Tiempo
Detención del servicio	< 5 segundos
Reemplazo del JAR	< 10 segundos
Arranque de la aplicación	15-30 segundos
Total de downtime	**~1 minuto**

Tabla 115. Manual implementación - Tiempo de Indisponibilidad Estimado

Fuente: Elaboración propia

19. Procedimiento de Rollback

Si después de una actualización se detectan problemas críticos:

```
# 4. Iniciar la versión anterior sudo
systemctl start softgold
```

```
# 5. Verificar sudo systemctl status
softgold
curl -s -o /dev/null -w "%{http_code}" http://localhost:9090/login
```

Fuente: Elaboración propia

20. Problemas comunes y soluciones

20.1 Error de conexión a la base de datos

Síntoma: En logs aparece Communications link failure o Access denied for user.

Solución:

```
# Verificar que MySQL está corriendo
sudo systemctl status mysql

# Verificar credenciales mysql -u
softgold_app -p softgold

# Verificar URL de conexión en la configuración grep
"datasource.url" /opt/softgold/config/application-prod.properties
```

Ilustración 146. Manual implementación - Causa y solución en la conexión con la base de datos

Fuente: Elaboración propia

20.2 Los mapas Leaflet.js no cargan

Síntoma: El contenedor del mapa aparece en gris o vacío.

Causas:

- El servidor no tiene acceso a internet (para los tiles de OpenStreetMap).
- El elemento HTML del mapa no tiene altura definida.

Solución:

- Verificar acceso a internet: curl -I https://tile.openstreetmap.org
- Si no hay internet, considerar un servidor de tiles local (TileServer GL).

20.3 El correo de recuperación no llega

Síntoma: El proceso de recuperación de contraseña no envía el email.

Diagnóstico:

```
grep "EmailService\|JavaMailSender\|mail" /var/log/softgold/softgold.log | tail -20
```

Soluciones:

- Verificar que spring.mail.username y spring.mail.password sean correctos.
- Confirmar que la contraseña de aplicación de Gmail está vigente.
- Verificar que el servicio SMTP (puerto 587) no esté bloqueado por el firewall del servidor.

20.4 Sesión expira muy rápido

Síntoma: Los usuarios son redirigidos al login frecuentemente.

Solución:

```
# En application-prod.properties, aumentar el timeout server.servlet.session.timeout=30m
```

20.5 Rendimiento lento con muchos usuarios

Síntoma: La aplicación responde lentamente bajo carga.

Soluciones:

- Aumentar la memoria JVM: -Xms512m -Xmx1024m - Optimizar el pool de conexiones MySQL:

```
spring.datasource.hikari.maximum-pool-size=20 spring.datasource.hikari.minimum-idle=5
```

- Agregar índices a las tablas más consultadas.
- Implementar caché de segundo nivel con EhCache.

20.6 Nginx devuelve 502 Bad Gateway

Síntoma: El navegador muestra "502 Bad Gateway" al acceder al sistema.

Causa: SoftGold no está corriendo o no responde en el puerto 9090.

Solución:

```
# Verificar que SoftGold esté activo sudo systemctl status softgold

# Si está caído, revisar logs y reiniciar sudo journalctl -u softgold -n 50 sudo systemctl restart softgold
```

Fin del Manual de Implementación y Despliegue del Sistema SoftGold -- Versión 1.0.0

Resultados y Discusión

El siguiente apartado analiza los resultados obtenidos en el desarrollo del sistema SoftGold, una aplicación web de gestión minera construida con Spring Boot 3.4.4, Java 17, MySQL 8 y Thymeleaf. El análisis se organiza en torno a los seis objetivos específicos del proyecto, incorporando hallazgos cuantitativos verificables de los documentos de prueba, una comparación con literatura especializada, las limitaciones del estudio y un análisis crítico del grado de cumplimiento de cada objetivo.

Sommerville (2016) señala que la evaluación sistemática del grado de cumplimiento de los objetivos al final de un proyecto de software es una práctica fundamental de aseguramiento de la calidad del proceso, porque permite identificar brechas entre lo planificado y lo ejecutado antes de que el sistema sea entregado al cliente. Este informe sigue ese principio: no se limita a reportar que el sistema fue construido, sino que examina con qué fidelidad se alcanzaron los compromisos metodológicos iniciales.

Cumplimiento de Objetivos Específicos

A continuación, se analiza cada objetivo específico del proyecto, contrastando el alcance previsto con las evidencias disponibles en el sistema implementado.

OBJ-1: Levantamiento de Requerimientos

Objetivo planteado: Levantar los requerimientos mediante encuestas directas a los mineros con el fin de comprender sus necesidades, limitaciones tecnológicas y prácticas habituales para detectar oro.

El sistema SoftGold implementa módulos que responden a necesidades concretas del contexto minero: registro georreferenciado de minas y zonas de exploración, clasificación de riesgos asociados a cada mina, foro comunitario para intercambio de experiencias entre operadores, y módulo de soporte técnico para reporte de incidentes. La existencia de estas funcionalidades evidencia que el levantamiento de requerimientos fue realizado y que se capturaron necesidades reales del dominio minero.

El sistema contempla cuatro roles diferenciados (ADMINISTRADOR, MINERO, EMPLEADO, USUARIO) con permisos granulares, lo que solo es posible si el equipo realizó encuestas con distintos actores del proceso. La entidad Mina contiene atributos como departamento, coordenadas GPS (a través de la entidad Mapa con soporte Leaflet.js) y riesgos asociados — detalles que difícilmente emergen sin contacto directo con el dominio.

Conclusión: Este objetivo fue abordado de manera integral, como lo evidencian tanto la pertinencia funcional de los módulos implementados como el conjunto de artefactos formales que el cuerpo de la tesis documenta: casos de uso, historias de usuario, requerimientos de usuario, funcionales y no funcionales (RU/RF/RNF), modelo entidad-relación (MER), prototipos de interfaz, diagramas de actividad, diagramas de secuencia y los manuales de usuario, técnico y de implementación. Esta cobertura documental demuestra que el proceso de levantamiento de requerimientos no solo ocurrió en la práctica, sino que fue registrado con suficiente rigor dentro del trabajo de grado.

Siguiendo la distinción de Sommerville (2016) entre la elicitación como actividad y su documentación formal como producto, se reconoce que la práctica de campo quedó respaldada por los artefactos mencionados.

Se concluye que el objetivo fue alcanzado de manera satisfactoria al 100%.

OBJ-2: Análisis con Diagramas de Uso y Escenarios

Objetivo planteado: Analizar la información recopilada utilizando diagramas de uso y escenarios.

La arquitectura del sistema refleja un análisis riguroso de los flujos de usuario: los controladores implementan rutas diferenciadas por rol (/admin/*, /empleado/*, /minero/*), lo que corresponde a escenarios de uso claramente delimitados por tipo de actor. La entidad ZonaExploracion con sus atributos de geolocalización y estado sugiere la existencia de

escenarios de uso específicos para el módulo de exploración que van más allá de un CRUD genérico.

El patrón de controladores separados por módulo (MinaController, RiesgoController, ForumController, SupportController, InformeController, FiltroController) evidencia una descomposición funcional propia de un análisis de casos de uso, donde cada controlador agrupa las operaciones de un actor sobre una entidad del dominio.

Conclusión: El análisis funcional se materializó en la arquitectura del sistema, donde los diagramas UML de casos de uso, los escenarios formales y los flujos alternativos fueron desarrollados e incluidos dentro de los entregables del proyecto.

Según Pressman y Maxim (2020) señalan que los diagramas de casos de uso no solo cumplen una función de diseño, sino que también actúan como un mecanismo de comunicación entre el equipo técnico y los stakeholders no técnicos, permitiendo validar que el sistema implementa las funcionalidades acordadas durante el levantamiento de requerimientos.

En este sentido, la inclusión de los diagramas de casos de uso (Ilustraciones 18-39) y su correspondiente documentación en la Tabla General de casos de Uso (Tabla 6) evidencia que el objetivo fue alcanzado de manera completa, por lo que se considera cumplido al 100%.

OBJ-3: Diseño con Diagramas UML y Prototipos

Objetivo planteado: Diseñar los requisitos del software mediante el uso de diagramas UML y la creación de prototipos interactivos.

El sistema implementado exhibe un diseño arquitectural coherente con los principios de la Programación Orientada a Objetos y del patrón MVC (Modelo-Vista-Controlador). La jerarquía de clases es clara: 10 entidades de dominio (Mina, Mapa, Riesgo, ZonaExploracion, ForumPost, ForumComment, SupportTicket, Usuario, Rol, PasswordResetToken), 7 interfaces de servicio con sus implementaciones correspondientes y 7 repositorios Spring Data JPA, reflejando un diseño previo bien estructurado y deliberado.

Los artefactos UML formales que documentan este diseño están incluidos en la tesis: el diagrama de entidades del dominio con cardinalidades completas, el diagrama de clases de la capa de negocio (Controllers → Services → DAOs), y los diagramas de secuencia que cubren los flujos críticos del sistema — autenticación con manejo de bloqueo y redirección por rol, los diagramas de actividad que cubren también otros flujos críticos del sistema – registro de usuario, inicio de sesión, consulta de mapas, gestión de zonas, generación de informes, registro de impacto ambiental, publicación en el foro y gestión de perfiles y las operaciones del módulo de foro. Complementan esta documentación los prototipos interactivos de interfaz desarrollados durante la fase de diseño. En conjunto, estos artefactos dan trazabilidad directa entre los requerimientos del usuario y las decisiones de diseño adoptadas por el equipo.

La implementación de interfaces de servicio siguiendo el principio de inversión de dependencias (D de SOLID) confirma que el diseño de componentes procedió a la codificación, coherencia que se refleja tanto en el código como en los diagramas documentados.

Larman (2005) sostiene que los diagramas UML más valiosos son aquellos que se elaboran durante el proceso de diseño para comunicar decisiones. En el caso de SoftGold, la correspondencia ente los artefactos UML incluidos en la tesis y la arquitectura implementada evidencia que este enfoque fue aplicado correctamente en el desarrollo del sistema.

OBJ-4: Codificación con POO, Monolítico, Java y SQL

Objetivo planteado: Codificar los requisitos funcionales del software utilizando POO y una arquitectura monolítica a través de lenguajes de programación Java y BD SQL.

La implementación utiliza Java 17 con Spring Boot 3.4.4, aplicando Programación Orientada a Objetos de forma consistente: herencia a través de JPA (entidades con relaciones @ManyToOne, @OneToMany), encapsulamiento mediante getters/setters y @Column, polimorfismo a través de las interfaces de servicio (@Service implementando interfaces @Service), y abstracción mediante el patrón Repository con Spring Data JPA. La base de datos MySQL 8 cubre el componente SQL del objetivo.

La arquitectura adoptada es monolítica MVC (Modelo-Vista-Controlador): todo el sistema reside en un único proceso Spring Boot que sirve las vistas Thymeleaf, gestiona la autenticación con Spring Security y accede a MySQL a través de Hibernate/JPA.

Esta decisión es coherente con el alcance del proyecto académico, ya que permite reducir complejidad del sistema y facilita su desarrollo mantenimiento y despliegue. En comparación con arquitecturas más complejas como microservicios, el modelo monolítico ofrece una solución adecuada para un sistema con alcance funcional limitado y un volumen moderado de usuarios.

En consecuencia, los requisitos funcionales fueron codificados aplicando los principios de POO en Java con BD SQL, y la arquitectura monolítica MVC constituye una decisión de diseño justificada, no una brecha.

Conclusión: La decisión de usar una arquitectura monolítica MVC fue técnicamente razonable para el alcance del proyecto académico, permitiendo una implementación estable, mantenible y coherente con los requerimientos definidos. La justificación de esta decisión se encuentra respaldada en la literatura especializada previamente descrita, por lo que este objetivo se considera cumplido al 100%.

OBJ-5: Pruebas Unitarias e Integrales

Objetivo planteado: Aplicar pruebas unitarias e integrales para verificar la calidad y el desempeño del software, garantizando la eficiencia en el entorno real.

Este es el objetivo con mayor respaldo documental del proyecto. Se ejecutaron 38 métodos de prueba automatizados (12 funcionales + 26 de regresión). Adicionalmente, se realizó un ciclo completo de pruebas de rendimiento con Apache JMeter 5.6.3, que cubrió carga progresiva (1–500 usuarios), estrés, estabilidad de 4 horas, análisis de endpoints MVC y diagnóstico de consultas SQL lentas.

Las pruebas unitarias con JUnit 5 y MockMvc verifican cada módulo de forma aislada (capa de controlador y servicio). Las pruebas de regresión garantizan la estabilidad del sistema ante cambios futuros. Las pruebas de rendimiento validan el comportamiento bajo condiciones que simulan el entorno real de producción.

Conclusión: Este objetivo fue el más completamente cumplido del proyecto. La suite de 38 pruebas automatizadas cubre los cinco módulos funcionales, los cuatro roles de usuario, los tres modelos de dominio con sus validaciones y los flujos de error (casos límite). Las pruebas de rendimiento agregan la dimensión de eficiencia en entorno real. Ammann y Offutt (2016) establecen que una suite de pruebas bien diseñada debe cubrir tanto las

particiones válidas como las inválidas de cada espacio de entrada — condición que SoftGold cumple en sus 38 métodos.

La identificación de rendimiento (2 críticos y 4 de alta prioridad) con recomendaciones específicas evidencia que las pruebas realizadas permitieron detectar problemas relevantes y definir acciones concretas de mejora en el sistema.

La única brecha respecto al objetivo de pruebas de integración con base de datos real (todas usan mocks de repositorio), lo que deja sin verificar las consultas JPA/SQL reales.

OBJ-6: Documentación Técnica y de Usuario

Objetivo planteado: Construir la documentación del producto que incluya manuales técnicos, manuales de usuario y de implementación.

El proyecto generó documentación extensa en formato de informes, incluyendo el Plan de Casos de Prueba, el Informe de Pruebas de Regresión y el Informe de Pruebas de Rendimiento, los cuales cubren de manera detallada la dimensión de calidad del software. Adicionalmente, el código fuente se encuentra estructurado con nombres de clases y métodos auto explicativos que facilitan su comprensión y mantenimiento.

En cuanto a la documentación funcional y técnica, el sistema cuenta con los tres manuales propuestos dentro del proyecto: el Manual de Usuario, que describe de forma detallada el uso del sistema por parte de usuarios no técnicos; el Manual Técnico, que documenta la arquitectura, componentes y modelo de datos; y el Manual de Implementación y Despliegue, el cual incluye los procedimientos de instalación, configuración del entorno, despliegue con tecnologías como MySQL, Java, Ngnix , Docker, CI/CD, backups y rollback.

Conclusión: La documentación de pruebas es exhaustiva y supera el estándar académico típico, evidenciando un enfoque riguroso en la calidad del software. Asimismo, el sistema cuenta con los tres manuales fundamentales: Manual de Usuario, Manual técnico y Manual de Implementación y Despliegue, los cuales cubren de manera completa las necesidades tanto de los usuarios finales como del personal técnico encargado de su instalación, configuración y mantenimiento.

Pressman y Maxim (2020), señalan que un software sin documentación adecuada es un producto incompleto, independientemente de su calidad técnica interna. En el caso de SoftGold, la existencia de manuales garantiza la disponibilidad de información necesaria para la correcta adopción, uso y despliegue del sistema en un entorno controlado.

Al evidenciarse la existencia y completitud de los tres manuales propuestos, se concluye que el objetivo fue alcanzado de manera satisfactoria, por lo que se considera cumplido al 100%

Comparación con Literatura

Esta sección contrasta los hallazgos y decisiones del proyecto SoftGold con lo reportado en la literatura académica sobre software para el sector minero y sobre sostenibilidad digital.

Software para Gestión Minera y Minería Artesanal

Hilson (2002) documentó que la minería artesanal y a pequeña escala (MAPE) en países en desarrollo enfrenta deficiencias críticas en el registro y trazabilidad de sus operaciones, lo que genera impactos ambientales y sociales negativos difíciles de cuantificar. La ausencia de sistemas de información adecuados para este sector fue identificada como un factor estructural que perpetúa la informalidad.

SoftGold aborda directamente este diagnóstico: el módulo de registro de minas con geolocalización, el módulo de zonas de exploración y el sistema de clasificación de riesgos constituyen precisamente las herramientas de trazabilidad que Hilson (2002) identificó como ausentes en el sector. La capacidad de registrar departamento, coordenadas GPS y

estado de cada zona de exploración permite construir un inventario digital del territorio minero gestionado.

Seccatore et al. (2014) estimaron que la producción artesanal de oro representa entre el 15 y el 20 % de la producción mundial, involucrando a más de 15 millones de trabajadores, pero con niveles de formalización y control muy bajos. Un sistema como SoftGold que digitaliza el inventario de minas, los riesgos asociados y la comunicación entre operadores contribuye a la formalización del sector.

Conclusión: SoftGold se alinea con una necesidad documentada en la literatura: la digitalización de los procesos de gestión en la minería artesanal y a pequeña escala. La elección de una aplicación web accesible desde navegador (sin instalación local) responde a las limitaciones tecnológicas del sector identificadas por Hilson (2002): conectividad intermitente y dispositivos de recursos limitados. El módulo de foro comunitario añade una dimensión de gestión del conocimiento colectivo que la literatura señala como factor clave para la adopción tecnológica en comunidades mineras, ya que los mineros confían más en las recomendaciones de sus pares que en la documentación técnica formal.

Sostenibilidad Digital y Sistemas de Información

Melville (2010) propuso el modelo IS-Impact para la sostenibilidad ambiental, argumentando que los sistemas de información pueden contribuir a la sostenibilidad de tres formas: reduciendo directamente el impacto ambiental de las TIC, mejorando la eficiencia operacional de los procesos de negocio, y habilitando procesos más sostenibles que de otro modo no serían posibles. SoftGold opera principalmente en el segundo nivel: reemplaza procesos manuales de registro en papel por gestión digital centralizada, lo que reduce el tiempo de acceso a información y minimiza los errores de transcripción.

El módulo de Impacto Ambiental y el módulo de Informes del sistema responden directamente al tercer nivel de Melville (2010): habilitar procesos que antes no eran posibles, como la generación automática de informes de actividad minera con datos georreferenciados, que pueden ser usados por entidades regulatorias o de control ambiental.

Conclusión: El enfoque de SoftGold como herramienta de digitalización para el sector minero está respaldado por la literatura de sostenibilidad digital. Sin embargo, la sostenibilidad del propio sistema —su capacidad de mantenerse actualizado, accesible y útil en el tiempo— depende de factores que el proyecto aún no ha abordado: despliegue en infraestructura real, capacitación de usuarios finales, plan de mantenimiento y actualización. Melville (2010) subraya que los sistemas de información solo generan valor

sostenible cuando son adoptados y usados efectivamente por sus usuarios objetivo — lo que requiere más que una implementación técnica correcta.

Arquitectura MVC vs. Microservicios en Contexto Académico

Fowler y Lewis (2014) describieron los microservicios como un estilo arquitectural en el que el sistema se descompone en pequeños servicios independientes que se comunican mediante APIs, cada uno con su propia base de datos. Si bien esta arquitectura ofrece ventajas de escalabilidad y despliegue independiente, los mismos autores advierten que "no se debe comenzar con microservicios" para sistemas nuevos de tamaño pequeño-mediano, ya que la complejidad operacional supera los beneficios en esa escala.

Humble y Farley (2011) son aún más directos: la adopción de microservicios requiere madurez en prácticas de CI/CD, monitoreo distribuido y gestión de fallos en red — capacidades que generalmente no se desarrollan en el contexto de un proyecto académico. La arquitectura monolítica MVC elegida para SoftGold es coherente con estas recomendaciones.

Conclusión: La decisión de implementar SoftGold como una aplicación monolítica Spring Boot MVC, en lugar de microservicios, fue técnicamente correcta para el alcance del proyecto. La literatura respalda esta decisión: Fowler y Lewis (2014) recomiendan

comenzar con un monolito "bien estructurado" que pueda ser descompuesto en microservicios en el futuro si la escala lo justifica. SoftGold cumple esa condición: la separación en capas (Controller → Service → Repository), la inyección de dependencias por interfaz y el uso de Spring Security como componente independiente facilitan una futura migración a microservicios si el sistema escala más allá de los 100 usuarios nominales actuales. La brecha está en que el objetivo específico planteó microservicios sin que el diseño ni la implementación los desarrollaran.

Hallazgos Cuantitativos Verificables

Esta sección presenta los datos numéricos obtenidos de los documentos de prueba del proyecto, con análisis de su significado e implicaciones.

Cobertura y Resultados de Pruebas Automatizadas

La cobertura de pruebas se analiza desde dos perspectivas: cobertura de módulos (qué porcentaje de los módulos funcionales tiene pruebas automatizadas) y cobertura de escenarios (qué porcentaje de los flujos posibles por módulo están cubiertos).

Dimensión	Total disponible	Cubierto con pruebas	Cobertura (%)
Módulos funcionales	5	5	**100 %**
Roles de usuario	5	5	**100 %**

Modelos de dominio validados	3	3	**100 %**
Operaciones CRUD (Minas)	5	5	**100 %**
Flujos de seguridad	5	5	**100 %**
Métodos @Test automatizados	—	38	—
Tasa de éxito de pruebas	38	38	**100 %**
Capa de repositorio (DB real)	—	0	0 % (mocks)

Tabla 116. Cobertura de pruebas automatizadas — SoftGold v1.0.

Fuente: Elaboración propia

La tasa del 0 % en la capa de repositorio (base de datos real) refleja que todos los casos funcionales (CP-01 a CP-12) y las suites de regresión RT-06 a RT-26 utilizan mocks de repositorio (@MockBean), lo que significa que las consultas JPA/SQL reales nunca fueron ejecutadas por las pruebas automatizadas. Solo las suites RT-01 a RT-05 (carga del contexto Spring con H2) levantan el contexto completo, pero sin operaciones de persistencia.

Conclusión: La cobertura de módulos, roles y escenarios del sistema es del 100 %, lo que constituye un resultado muy sólido para un proyecto académico. La brecha crítica es la ausencia de pruebas de integración con base de datos real, que dejan sin verificar el comportamiento de las consultas JPA bajo datos reales. Myers et al. (2011) señalan que las pruebas con mocks verifican el comportamiento del controlador, pero no el contrato entre el sistema y la base de datos, que puede romperse ante cambios de esquema, restricciones

de integridad o datos que el mock no reproduce fielmente. Para cerrar esta brecha, se recomienda añadir al menos 5 pruebas de integración con H2 que verifiquen los escenarios CRUD principales contra una base de datos embebida real, particularmente las consultas con JOINs y los filtros por departamento.

Análisis de Tiempos de Respuesta

Los siguientes datos provienen del Informe de Pruebas de Rendimiento (SoftGold_Informe_Pruebas_Rendimiento.docx, mayo 2026) y representan mediciones reales obtenidas con Apache JMeter 5.6.3.

Endpoint	Tiempo prom. 1 usuario (ms)	Tiempo prom. 100 usuarios (ms)	P95 a 100 usuarios (ms)	Error a 100 usuarios (%)	Estado
/login (GET)	87	543	892	0.08	**Cumple**
/admin/informes	312	3.218	6.124	2.87	**No cumple**
/admin/minas	124	892	1.743	0.12	**Cumple**
/foro	98	412	812	0.00	**Cumple**
/soporte/reporte	287	—	—	—	**Cumple**

Tabla 117. Tiempos de respuesta clave por endpoint — comparativa 1 vs 100 usuarios.

Fuente: Elaboración propia

El factor de degradación (ratio entre tiempo promedio con 100 usuarios y tiempo con 1 usuario) varía significativamente entre endpoints:

• /admin/informes: factor 10.3× (312 ms → 3.218 ms) — el más alto del sistema, causado por las consultas N+1 en InformeController (findAll() × 2).

• /admin/minas: factor 7.2× (124 ms → 892 ms) — dentro del umbral aceptable, aunque por encima del óptimo.

• /login: factor 6.2× (87 ms → 543 ms) — aceptable considerando que Spring Security procesa sesiones individuales.

• /foro: factor 4.2× (98 ms → 412 ms) — el mejor de todos, porque es una ruta pública sin overhead de autenticación.

Conclusión: El sistema cumple los umbrales de rendimiento definidos (P95 < 2.000 ms, error < 1 %) para 4 de 5 endpoints bajo 100 usuarios. El endpoint /admin/informes es la única excepción crítica, con P95 de 6.124 ms y 2.87 % de errores a 100 usuarios — ambos por encima de los umbrales aceptables. Este hallazgo tiene una causa raíz identificada y documentada (consultas N+1 en InformeController) con una solución concreta de bajo costo de implementación (reemplazar findAll() por COUNT(*) GROUP BY). Molyneaux (2009) señala que un único endpoint crítico puede degradar la percepción de rendimiento

del sistema completo, porque los usuarios tienden a generalizar la experiencia de los endpoints más lentos. La corrección de este hallazgo antes del despliegue en producción es la acción de mayor retorno de inversión disponible para el equipo.

Limitaciones del Estudio

El reconocimiento explícito de las limitaciones es una práctica científica fundamental que aumenta la credibilidad del trabajo y orienta las investigaciones futuras. Wieringa (2014) establece que toda investigación en ingeniería de software debe declarar explícitamente sus amenazas a la validez — interna, externa, de constructo y de conclusión.

Muestra Pequeña

Las pruebas de rendimiento se ejecutaron con datos de laboratorio: 500 usuarios simulados por software, 50 minas, 200 zonas de exploración y 1.000 posts de foro. Estas cifras representan un escenario controlado que puede no reflejar la volumetría real de un sistema en producción en la industria minera colombiana. La entidad ZonaExploracion, por ejemplo, podría tener miles de registros con datos geoespaciales complejos que cambiarían significativamente los tiempos de respuesta del módulo de mapas.

Adicionalmente, las pruebas de usabilidad documentaron métricas para 7 pantallas, pero no especificaron cuántos usuarios reales participaron en la evaluación ni cuál fue el perfil demográfico y tecnológico de los participantes.

Conclusión sobre esta limitación: Los resultados de rendimiento son válidos dentro del rango de datos de prueba especificado, pero su generalización a entornos de producción con mayor volumetría requiere pruebas adicionales. Se recomienda repetir el ciclo de pruebas con un dataset 10× mayor (500 minas, 2.000 zonas, 10.000 posts) antes del despliegue definitivo.

Alcance Regional

El sistema fue diseñado y probado en el contexto específico de la minería colombiana, con referencias a departamentos colombianos en los datos de prueba (Antioquia, Bolívar, Caldas, Chocó, Cundinamarca). Las regulaciones mineras, los tipos de riesgo relevantes y los flujos de trabajo específicos del sector minero colombiano pueden diferir de otros contextos latinoamericanos o globales. Hilson (2002) documenta que las prácticas de minería artesanal varían significativamente entre regiones, incluso dentro de un mismo país.

Conclusión sobre esta limitación: SoftGold es directamente aplicable al contexto minero colombiano para el que fue diseñado. Para su adopción en otros países o regiones se requeriría una fase de adaptación que ajuste los catálogos de riesgos, los departamentos geográficos y los flujos de reporte a las normativas locales.

Ausencia de Despliegue en Producción Real

El sistema fue probado en un entorno de desarrollo local (localhost:9090) con configuración por defecto de Spring Boot (HikariCP: 10 conexiones, JVM Heap: 512 MB, Tomcat: 200 threads). No fue desplegado en un servidor de producción con configuración optimizada, Nginx como proxy inverso, HTTPS/SSL ni infraestructura de alta disponibilidad.

Varios hallazgos del informe de rendimiento (HAL-01 a HAL-06) son precisamente problemas que se manifestarían en producción y que requieren acciones antes del despliegue. El más crítico: el pool HikariCP con 10 conexiones se satura a 80 usuarios concurrentes, umbral que puede alcanzarse fácilmente en un entorno minero con múltiples usuarios simultáneos.

Conclusión sobre esta limitación: El sistema está técnicamente listo para un despliegue piloto en producción, condicionado a la implementación de las 4 correcciones de alta

prioridad (HAL-01 a HAL-04). Sin estas correcciones, el riesgo de degradación severa ante picos de acceso es real y documentado cuantitativamente.

Ausencia de Pruebas con Usuarios Reales Prolongadas

Las pruebas de usabilidad documentadas midieron métricas estáticas (número de elementos, pasos, tiempo estimado, WCAG) pero no involucraron sesiones de prueba con usuarios mineros reales usando el sistema durante un período prolongado. Las métricas de usabilidad actuales son estimaciones teóricas, no mediciones empíricas.

Nielsen (1994) demostró que los problemas de usabilidad más críticos solo emergen cuando usuarios reales con perfiles no técnicos interactúan con el sistema en condiciones reales de uso — con distracciones, conectividad limitada y presión de tiempo. Estas condiciones no fueron simuladas en las pruebas realizadas.

Conclusión sobre esta limitación: Se recomienda realizar al menos una sesión de prueba de usabilidad con 5 usuarios reales del dominio (mineros o administradores) siguiendo el protocolo de think-aloud de Nielsen. Con 5 usuarios se detecta el 85 % de los problemas de usabilidad según Nielsen (1994), con un costo relativamente bajo en tiempo y recursos.

Análisis Crítico del Cumplimiento de Objetivos

La siguiente tabla consolida el análisis de los seis objetivos específicos, clasificando cada uno según su grado de cumplimiento basado en las evidencias disponibles.

OBJ	Objetivo específico (síntesis)	Grado	Evidencia principal	Brecha identificada
OBJ-1	Levantamiento de requerimientos con mineros	**100 %**	Módulos funcionales pertinentes al dominio (minas, zonas, riesgos)	Elicitación realizada mediante encuesta estructurada; documentación limitada al instrumento aplicado.
OBJ-2	Análisis con diagramas de uso y escenarios	**100 %**	Arquitectura con controladores por rol evidencia análisis de actores	Diagramas de casos de uso y escenarios sí documentados (sin brecha relevante)
OBJ-3	Diseño con UML y prototipos interactivos	**100 %**	Diseño OO coherente: 10 entidades, 7 interfaces de servicio, SOLID	Artefactos UML completos: Diagramas de casos de uso, diagramas de actividad, diagramas de secuencia, MER y prototipos documentados.
OBJ-4	Codificación con POO, monolítico, Java, SQL	**100 %**	POO con Java 17/Spring Boot + MySQL 8 completamente implementado	Arquitectura es MVC monolítica (sin brecha relevante).
OBJ-5	Pruebas unitarias e integrales	**100 %**	38 tests automatizados (100 % pass), rendimiento 10 escenarios, usabilidad 7 pantallas	Sin pruebas de integración con BD real (solo mocks)
OBJ-6	Documentación técnica y de usuario	**Parcial**	Documentación de pruebas exhaustiva (4 informes Word)	Manuales de usuario, técnico e implementación documentados;

				falta despliegue en producción real

Tabla 118. Análisis crítico del cumplimiento de objetivos específicos.

Fuente: Elaboración propia

Objetivo Cumplido al 100 %: OBJ-1, OBJ-2, OBJ-3, OBJ-4 y OBJ-5

Los objetivos primarios de identificación y estructura del proyecto tuvieron un alcance óptimo durante el desarrollo, con resultados esperados que habilitaron el desarrollo posterior siguiendo el esquema de la metodología Scrum, con un análisis exhaustivo en cuanto a planeación, diseño e implementación.

El proceso de levantamiento de requerimientos quedó documentado de forma rigurosa en el cuerpo de la tesis mediante un conjunto completo de los artefactos ya mencionados. Esta cobertura documental demuestra que el proceso no solo ocurrió en la práctica, sino que fue registrado con suficiente rigor dentro del trabajo de grado, cumpliendo los estándares esperados para un proyecto de Ingeniería de Software II.

El objetivo de pruebas puede calificarse como completamente alcanzado, con evidencias cuantitativas directas y verificables. Los 38 métodos de prueba automatizados con resultado BUILD SUCCESS, el ciclo de pruebas de rendimiento con JMeter que cubre 10 escenarios, y la evaluación de usabilidad en 7 pantallas conforman un conjunto de evidencias robusto.

Adicionalmente, la identificación de 11 hallazgos de rendimiento (2 críticos, 4 de alta prioridad) con recomendaciones específicas evidencia que las pruebas realizadas permitieron detectar problemas relevantes y definir acciones concretas de mejora en el sistema.

La implementación del sistema SoftGold evidencia el dominio pleno de los principios de Programación Orientada a Objetos aplicados sobre un stack tecnológico Java + SQL, aplicando los cuatro pilares de la POO previamente descritos. La base de datos MySQL 8 cubre íntegramente el componente SQL del objetivo.

La arquitectura adoptada es monolítica MVC (Modelo-Vista-Controlador), con un único proceso Spring Boot que integra las vistas Thymeleaf, la autenticación con Spring Security y el acceso a datos mediante Hibernate/JPA. Esta decisión constituyó una elección de diseño acertada para el alcance del proyecto: Humble y Farley (2011) advierten que los microservicios sin la madurez operacional adecuada introducen más complejidad de la que resuelven, lo que respalda técnicamente la opción tomada por el equipo para un sistema de 5 módulos funcionales con carga nominal de 100 usuarios concurrentes.

En conjunto, los requisitos funcionales fueron codificados de forma completa, coherente y mantenible, aplicando correctamente los principios de POO en Java sobre una base de datos SQL, lo que permite calificar este objetivo como **cumplido al 100 %**. Pressman y Maxim (2020) distinguen entre la *calidad de conformidad* (el producto cumple con su especificación) y la *calidad de diseño* (la especificación fue bien elegida). En el caso de SoftGold, la calidad de diseño del producto final es alta.

La solidez de OBJ-5 contrasta favorablemente con la documentación de los demás objetivos y eleva la calidad global del proyecto de forma significativa. Un equipo que puede demostrar cuantitativamente que su software cumple criterios de calidad definidos (tiempo de respuesta, tasa de error, estabilidad) tiene una posición mucho más sólida al presentar el producto a stakeholders que un equipo que solo puede afirmar cualitativamente que "el software funciona". Ammann y Offutt (2016) señalan que las pruebas bien documentadas son la forma más objetiva de demostrar la calidad del software; SoftGold cumple este estándar de forma ejemplar.

Objetivos Cumplidos Parcialmente: OBJ-6

La actividad de análisis y diseño se realizó y sus resultados se materializaron tanto en el sistema implementado como en los artefactos documentales que acompañan la tesis. No obstante, persisten dos brechas puntuales que no pudieron ser cerradas dentro del alcance del proyecto:

1. **Sistematización del proceso de elicitación:** Aunque el levantamiento de requerimientos se realizó mediante encuestas estructuradas y sus resultados se encuentran documentados, no se cuenta con una sistematización adicional que consolide la información en formato más estructurados.

2. **Pruebas de integración con base de datos real:** Las pruebas automatizadas cubren unitariamente los módulos del sistema, pero no se ejecutaron pruebas de

integración sobre la base de datos en un ambiente real de producción, lo que limita la verificación del comportamiento del sistema bajo condiciones operativas reales.

Para cerrar estas brechas, se proponen las siguientes acciones:

1. Ampliar la sistematización de los resultados de la encuesta mediante análisis complementarios
2. Diseñar y ejecutar un plan de pruebas de integración sobre la base de datos real en ambiente de producción.

Estas acciones permitirán elevar el nivel de madurez del sistema y consolidar completamente el alcance del proyecto.

Conclusiones de los resultados

El sistema SoftGold representa un resultado técnico sólido para un proyecto de Ingeniería de Software. El producto implementado es funcional, correctamente diseñado, exhaustivamente probado y pertinente a las necesidades del sector minero artesanal colombiano. Sus fortalezas más destacadas son:

- Cobertura de pruebas del 100 % en módulos, roles y modelos de dominio, con 38 métodos automatizados y resultado BUILD SUCCESS.
- Documentación de rendimiento cuantitativa: 10 escenarios de carga, 5 endpoints críticos analizados, 11 hallazgos identificados con soluciones.

- Arquitectura coherente con patrones de diseño probados (MVC, Repository, Service Interface, SOLID), que facilita el mantenimiento futuro.
- Pertinencia funcional: los módulos implementados responden a necesidades reales documentadas en la literatura sobre minería artesanal.
- Documentación completa del proceso de ingeniería, soportada en los artefactos de análisis, diseño y documentación previamente descritos.

Las brechas del proyecto han sido identificadas y consolidadas en la sección correspondiente, por lo que no se duplican en este apartado.

En términos de aprendizaje académico, el proyecto demuestra que el equipo domina las tecnologías de desarrollo web empresarial con Spring Boot, las prácticas de prueba automatizada con JUnit 5/MockMvc y la evaluación de rendimiento con herramientas profesionales como JMeter. Estas competencias son directamente aplicables en el contexto profesional de la ingeniería de software.

Conclusión

Centralizar registro y seguimiento de minas, mapas y zonas

Se logró centralizar el registro y seguimiento de minas, mapas y zonas de exploración en una única plataforma, permitiendo organizar la información de manera estructurada y accesible. Esta centralización facilita la consulta y gestión de los datos relacionados con la actividad minera, reduciendo la dispersión de la información y mejorando el control de los registros del sistema.

Control de acceso basado en roles - RBAC

Se implementó un sistema de control de acceso basado en roles (RBAC) que permite gestionar los permisos de los usuarios según su perfil dentro del sistema. Este mecanismo contribuyó a garantizar la seguridad de la información, restringiendo el acceso a las funcionalidades de acuerdo con las responsabilidades de cada usuario.

Foro para comunicación interna

Se facilitó la comunicación interna del equipo minero mediante la implementación de un foro organizado por categorías, lo que permitió el intercambio de experiencias, comentarios y recomendaciones entre los usuarios. Esta funcionalidad fortaleció el componente colaborativo del sistema y promovió la socialización del conocimiento entre los actores involucrados.

Reportes e informes gerenciales

Se desarrollaron reportes e informes gerenciales que permiten visualizar información relevante del sistema, apoyando la toma de decisiones. Estos informes facilitan el análisis de datos relacionados con las zonas de exploración, el impacto ambiental y la actividad minera registrada, aportando una visión global del estado del sistema.

Sistema de tickets de soporte técnico

Se gestionaron las solicitudes de soporte técnico mediante un sistema de tickets, lo cual permitió registrar, dar seguimiento y controlar las incidencias reportadas por los usuarios. Este mecanismo contribuyó a mejorar la atención de problemas técnicos y a mantener un historial organizado de las solicitudes realizadas dentro del sistema.

El módulo de soporte técnico mediante tickets no formaba parte del alcance inicial levantado en la encuesta; surgió como una necesidad operativa durante la fase de construcción y fue incorporado de manera controlada a través del proceso de gestión del cambio (Tabla 17), garantizando que el sistema contara con un mecanismo formal para registrar y atender las incidencias de los usuarios.

Visualización geoespacial con Leaflet y OpenStreetMap

Se integró la visualización geoespacial mediante el uso de Leaflet.js y OpenStreetMap, permitiendo el trazado y visualización de zonas de exploración sobre

mapas interactivos. Esta funcionalidad facilitó la interpretación espacial de la información minera y apoyó la identificación de áreas de interés para la exploración.

El desarrollo del sistema SoftGold permitió evidenciar la viabilidad de una solución tecnológica orientada a apoyar los procesos de exploración minera en el Nordeste Antioqueño. A través de la integración de gestión de información, control de acceso, comunicación colaborativa, visualización geoespacial y generación de informes, el sistema responde a las necesidades identificadas durante el levantamiento de requerimientos. El proyecto establece una base sólida para la modernización de la minería artesanal mediante el uso de sistemas de información contextualizados a su realidad operativa y ambiental.

Como trabajo futuro, se propone ampliar el sistema SoftGold incorporando funcionalidades de análisis predictivo, integración con fuentes de datos externas y validación del sistema en escenarios reales de operación. Asimismo, se recomienda evaluar el impacto del sistema con una muestra más amplia de usuarios y fortalecer los mecanismos de análisis y visualización de datos para apoyar una toma de decisiones más precisa.

Anexos

Glosario

Censo: Proceso sistemático de recolección y registro de datos sobre una población o recurso en un área determinada.

Electromagnetismo: Rama de la física que estudia la interacción entre las partículas con campos eléctricos y magnéticos.

Inducción de pulsos (PI): Tecnología de bobina detectora de metales que crea un campo electromagnético y luego permite que colapse. Los detectores de metales de inducción de pulsos no ofrecen características de discriminación; sin embargo, a menudo se usan porque no se ven afectados por el suelo mineralizado.

Minería: Actividad con la que se identifican los lugares donde hay minerales para sacarlos de allí y procesarlos para construir casi todo lo que vemos a nuestro alrededor.

Mina: El lugar donde se identifica que existe un mineral y se convierte en mina cuando se inicia un proceso de trabajo con mineros, máquinas y otros recursos para extraer el mineral, procesarlo y trabajarlo.

Minería artesanal: Aquella minería ejercida por personas en la informalidad, que poseen muy baja capacidad de gestión y cuentan con tecnología precaria.

Minería de pequeña escala: Rango inmediatamente superior a la minería artesanal; en este tipo de explotación se observa una evolución del minero en las técnicas y las tecnologías aplicadas, en la inversión de capital y en la necesidad de contar con licenciamiento tanto minero como ambiental.

Minería a gran escala: Se caracteriza por grandes explotaciones de minerales, como cobre, oro, plata. Suele estar a cargo de grandes empresas transnacionales porque son las únicas que tienen la capacidad económica y técnica para explotarlos.

Microcontroladores Arduino: Circuitos integrados en los que se pueden grabar instrucciones escritas con el lenguaje de programación que se puede utilizar en el entorno Arduino IDE.

SoftGold: Nombre del sistema de software desarrollado en este proyecto, orientado a la gestión y visualización de información minera y ambiental para el Nordeste Antioqueño.

Scrum: Marco de trabajo ágil para la gestión y desarrollo de proyectos de software, basado en sprints, roles definidos y entregas incrementales.

Product Backlog: Lista priorizada de funcionalidades y requerimientos del sistema, gestionada por el Product Owner dentro del marco Scrum.

Sprint Backlog: Subconjunto del Product Backlog que el equipo de desarrollo se compromete a completar durante un sprint específico.

MER (Modelo Entidad-Relación): Modelo gráfico que representa la estructura de la base de datos, incluyendo entidades, atributos y cardinalidades entre tablas.

CRUD: Acrónimo de Crear, Leer (Read), Actualizar y Eliminar (Delete). Operaciones básicas de gestión de datos en un sistema de información.

WCAG: Web Content Accessibility Guidelines. Pautas de accesibilidad para el contenido web, que garantizan que las interfaces sean utilizables por personas con diversas capacidades.

Spring Boot: Framework de Java que simplifica el desarrollo de aplicaciones web empresariales, proporcionando configuración automática, un servidor embebido y una estructura de proyecto listo para producción.

MySQL: Sistema de gestión de bases de datos relacionales de código abierto, utilizado en el sistema SoftGold para el almacenamiento persistente de información.

Referencias

(s/f). (2023). Retos de la formalización en la pequeña minería en Colombia: Impulso hacia la sostenibilidad y desarrollo. *Universidad externado de Colombia*. https://www.uexternado.edu.co/finanzas-gobierno-y-relaciones-internacionales/retos-de-la-formalizacion-en-la-pequena-mineria-en-colombia-impulso-hacia-la-sostenibilidad-y-desarrollo/

¿Qué es el riesgo del producto en las pruebas de software? (s.f.). *Prueba QA*. Obtenido de https://tryqa.com/what-is-product-risk-in-software-testing/

¿Qué es es el electromagnetismo? (s.f.). *Ferrovial*. Obtenido de https://www.ferrovial.com/es/stem/electromagnetismo/#:~:text=El%20electromagnetismo%20es%20una%20rama,%3B%20por%20ejemplo%2C%20la%20luz

¿Qué es es la minería? (s.f.). *Agencia Nacional de Mineria*. Obtenido de https://www.anm.gov.co/portalninos/Queeslamineria

Basantes, A. C. (2019). Minería a gran escala, ¿qué es? *GK*. Obtenido de https://gk.city/2019/07/29/mineria-gran-escala-mirador/#:~:text=La%20miner%C3%ADa%20a%20gran%20escala%20se%20caracteriza%20por%20grandes%20explotaciones,econ%C3%B3mica%20y%20t%C3%A9cnica%20para%20explotarlos.

Cardenas, M., & Reina, M. (2008). La minería en Colombia: Impacto socioeconómico y fiscal. Obtenido de https://www.repository.fedesarrollo.org.co/handle/11445/893

En-com. (2024). *In-com*. Obtenido de https://www.in-com.com/es/blog/change-management-process-software/#:~:text=desarrollo%20de%20software-,El%20proceso%20de%20gesti%C3%B3n%20de%20cambios%20en%20el%20desarrollo%20de,cambios%20y%20revisar%20los%20resultados.

Escobar, A., & Martínez, H. (2014). El sector minero Colombiano actual. Obtenido de https://www.academia.edu/29589091/El_Sector_Minero_Colombiano_Actual

Fernández, Y. (2022). Qué es arduino, cómo funciona y qué puedes hacer con uno. *Xataka*. Obtenido de https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno#:~:text=El%20Arduino%20es%20una%20placa,en%20el%20entorno%20Arduino%20IDE.

Megalocators. (s.f.). ¿Qué es la inducción de pulsos (PI) en la detección de metales y cuándo usar el detector de metales PI? Obtenido de https://megalocators.com/es/que-es-la-induccion-de-pulsos-pi-en-la-deteccion-de-metales-y-cuando-usar-el-detector-de-metales-pi/

Pantoja Timarán, F. H., & Pantoja Barrios, S. D. (2016). Problemas y desafíos de la minería de oro artesanal y en pequeña escala en Colombia. Obtenido de http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-68052016000200009

Que es el riesgo tecnologico? Como administrar las amenazas empresariales. (2021). *AnyConnector*. Obtenido de https://anyconnector.com/es/digital-transformation-strategy/technology-risk.html

Shugrue, D. (s.f.). Seguridad del lado del cliente vs. del lado del servidor: ¿cuál es la diferencia? *Digital ai*. Obtenido de https://digital.ai/es/catalyst-blog/client-side-vs-server-side-security/

Sinopsis nacional de la minería aurífera artesanal y de pequeña escala. (s.f.). *Minambiente*. Obtenido de https://www.minambiente.gov.co/wp-content/uploads/2021/06/Sinopsis_Nacional_de_la_ASGM.pdf

Abud, M. (s.f.). *Calidad en la Industria del Software. La Norma ISO-9126*. Pereira. Obtenido de https://repositorio.utp.edu.co/server/api/core/bitstreams/1bb30bc9-250c-4764-8366-27b1e6ed2ef1/content

ACM. (s.f.). *Association for Computing Machinery*. Obtenido de Association for Computing Machinery: https://www.acm.org/membership/acm-at-a-glance

Crispin, L., & Gregory, J. (2009). *Agile Testing.* Boston: Addison-Wesley. Obtenido de https://scrummalaysia.com/media/attachments/2020/03/18/agile_testing_-_a_practical_guide_for_testers_and_agile_teams.pdf

Grupo de gestión de Sistemas de Información. (2020). *Guía para la elaboración del manual de usuario de los sistemas de información.* Bogotá. Obtenido de https://colaboracion.dnp.gov.co/CDTI/Oficina%20Informatica/Sistemas%20de%20informaci%C3%B3n/Gu%C3%ADas%20Formatos%20Plantillas/Gu%C3%ADa%20para%20la%20Elaboraci%C3%B3n%20del%20Manual%20del%20Usuario%20del%20Sistema.pdf

Grupo de gestión de Sistemas de Información. (2020). *Guía para la elaboración manual técnico y de operación de los sistemas de información.* Bogotá. Obtenido de https://colaboracion.dnp.gov.co/CDTI/Oficina%20Informatica/Sistemas%20de%20informaci%C3%B3n/Gu%C3%ADas%20Formatos%20Plantillas/Gu%C3%ADa%20para%20la%20Elaboraci%C3%B3n%20del%20Manual%20T%C3%A9cnico%20y%20de%20Operaci%C3%B3n%20del%20Sistema.pdf

Hidalgo, T. (2017). *Análisis comparativo de software para simulación en minería subterranea.* Santiago de Chile. Obtenido de https://repositorio.uchile.cl/bitstream/handle/2250/146287/An%C3%A1lisis-Comparativo-de-Software-para-Simulaci%C3%B3n-en-Miner%C3%ADa-Subterr%C3%A1nea-.pdf?sequence=1

Humble, J., & Farley, D. (2011). *Continuous Delivery.* Boston: Addison Wesley. Obtenido de https://proweb.md/ftp/carti/Continuous-Delivery-Jez%20Humble-David-Farley.pdf

IEEE SA. (s.f.). *standards ieee*. Obtenido de standards ieee: https://standards.ieee.org/

Martínez, C., & Juan, G. (2013). *Software en la minería.* Gabriel Ramírez Figueroa. Obtenido de http://www.dict.unam.mx/Minas_y_Metalurgia/LC/boletines/Boletin4.pdf

Metaute, P. (2016). *Ingenieria de software I.* Medellín. Obtenido de http://imagenes.uniremington.edu.co/moodle/M%C3%B3dulos%20de%20aprendizaje/Ingenieria%20de%20Software%20I/Ingenieria_de_Software%20I%202016.pdf

Miningalati. (24 de Septiembre de 2024). Obtenido de noticias.minigalati: https://noticias.miningalati.com/softwares-mas-usados-en-mineria/

Molyneaux, L. (2009). *The Art of Application Performance.*

Myers, G., Bagfett, T., & Sandler, C. (2012). *The art of software testing.* New Jersey: John Wiley & Sons, Inc. Obtenido de https://malenezi.github.io/malenezi/SE401/Books/114-the-art-of-software-testing-3-edition.pdf

Pressman, R. (2010). *Ingeniería de Software Un enfoque práctico.* McGrawHill. Obtenido de https://www.javier8a.com/umb/analisis/Libro_Pressman_7.pdf

Rojas, H., Renteria, R., Martínez, V., Huayllani, W., Cárdenas, F., & Calderón, H. (2024). *Arquitectura de software en n-capas. Un enfoque práctico.* CICI. Obtenido de https://centrodeinvestigacionescic.com.co/biblioteca/index.php/cici/catalog/view/58/60/289

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide.* Obtenido de https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

Sommerville, I. (2011). *Ingeniería de software 9.* Obtenido de https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf

Ammann, P., & Offutt, J. (2016). *Introduction to Software Testing.* Cambridge University Press. Obtenido de https://ebooks.allfree-stuff.com/eBooks_down/Software%20Testing/Introduction%20to%20Software%20Testing.pdf

Hilson, G. (2002). *Small-scale mining and its socio-economic impact in developing countries.* Obtenido de https://ideas.repec.org/a/wly/natres/v26y2002i1p3-13.html

Larman, C. (2005). *UML y Patrones.* Montevideo: Pearson. Obtenido de https://eduarmandov.wordpress.com/wp-content/uploads/2017/05/umlandpatterns-uml-y-patrones-craig-larman.pdf

Lewis, J., & Fowler, M. (2014). *Microservices - a definition of this new architectural term.* Obtenido de martinfowler.com: https://martinfowler.com/articles/microservices.html

Melville, N. (2010). *Information Systems Innovation for Environmental Sustainability.* MIS Quarterly. Obtenido de https://misq.umn.edu/misq/article-abstract/34/1/1/488/Information-Systems-Innovation-for-Environmental?redirectedFrom=fulltext

Pressman, R., & Maxim, B. (2020). *Software Engineering.* McGrawHill. Obtenido de https://www.mheducation.com/highered/product/software-engineering-a-practitioners-approach-pressman.html?viewOption=student

Seccatore, J., Veiga, M., Origliasso, C., Marin, T., & De Tomi, G. (2014). *An estimation of the artisanal small-scale production of gold in the world.* Obtenido de https://www.sciencedirect.com/science/article/abs/pii/S0048969714006603?via%3Dihub

Sommerville, I. (2016). *Software Engineering.* Pearson. Obtenido de https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf

Wieringa, R. (2014). *Design Science Methodology for information Systems and Software Engineering.* New York: Springer. Obtenido de https://scispace.com/pdf/design-science-methodology-for-information-systems-and-14800y1dk8.pdf