TRABAJO DE GRADO
Opción Seminario-Diplomado

De lo monolítico a lo modular: Una experiencia de gestión de proyectos en One Tech Fury S.A.S.

Corporación Universitaria Remington
Facultad de Ingenierías
Especialización Seguridad de la información
Tecnología en desarrollo de software
Ingeniería de sistemas

Autores:
Mateo Bedoya Agudelo
Eyder Alejandro Payares
Sergio Ramírez Ramírez

Tutor: Johana Sepúlveda Jiménez
Seminario Gestión de proyectos y Habilidades Gerenciales
2026

Agradecimientos

Agradecemos a la Corporación Universitaria Remington por brindarnos las herramientas necesarias para nuestra formación. De manera especial, expresamos nuestra gratitud a la docente Johana Sepúlveda Jiménez por su orientación, paciencia y por compartir sus conocimientos en el Seminario de Gestión de Proyectos, los cuales fueron fundamentales para estructurar todo este entregable.

Tabla de Contenidos

Resumen

La empresa One Tech Fury S.A.S. dispone de una plataforma de comercio electrónico desarrollada internamente hace más de diez años, la cual soporta procesos críticos del negocio digital. Con el paso del tiempo, el sistema no ha tenido una evolución tecnológica adecuada, lo que ha generado un alto nivel de obsolescencia y una acumulación significativa de deuda técnica.

En la actualidad, la plataforma presenta una estructura monolítica, con componentes fuertemente acoplados y lógica rígida, lo que limita su mantenimiento y dificulta la incorporación de nuevas funcionalidades. A esto se suma el uso de tecnologías y dependencias desactualizadas, muchas de ellas sin soporte, lo que incrementa los riesgos asociados a la seguridad de la información y reduce la competitividad de la empresa frente a soluciones más modernas.

El equipo de desarrollo ha concluido que no es viable implementar mejoras relevantes sobre el sistema actual, debido a la ausencia de una arquitectura que permita su evolución. En el entorno operativo, se han identificado diversas fallas, tales como inconsistencias en transacciones, tiempos de respuesta elevados, errores en procesos clave —como órdenes, pagos y reembolsos—, así como un incremento en las incidencias reportadas por los usuarios. Estas condiciones impactan negativamente la experiencia del cliente y representan un riesgo para la conversión y la reputación de la organización.

En este contexto, la empresa enfrenta limitaciones importantes en términos de escalabilidad, seguridad y rendimiento, lo que hace necesaria una modernización profunda o la reconstrucción de la plataforma tecnológica.

Palabras clave: deuda técnica, monolito, obsolescencia, modernización, escalabilidad.

Marco Conceptual y Contextual

Este informe se desarrolla en el marco del seminario de Gestión de Proyectos y Habilidades Gerenciales, donde se trabajaron diferentes habilidades necesarias para el trabajo en equipo y la ejecución de proyectos. Durante las clases se abordaron temas como liderazgo, comunicación asertiva, manejo del tiempo, negociación y resolución de conflictos, aspectos que resultan clave en cualquier proyecto, especialmente en el desarrollo de software.

A lo largo del seminario se entendió que un proyecto no se trata únicamente de desarrollar un sistema, sino también de saber organizar el trabajo, tomar decisiones y enfrentar situaciones que pueden surgir dentro del equipo. En este sentido, se revisaron conceptos básicos de gestión de proyectos como la definición de roles, la asignación de responsabilidades y prácticas para priorizar tareas.

Por ejemplo, en este proyecto se trabajó con la matriz de Eisenhower, una herramienta de gestión del tiempo popularizada en la gestión del tiempo. Esta metodología permitió clasificar las tareas a desarrollar según su urgencia e importancia (Suárez, 2015). lo que facilita tomar decisiones más claras sobre qué se debe desarrollar primero, qué delegar y qué posponer por ser de baja prioridad.

En cuanto al liderazgo, se comprendió que no siempre depende de una sola persona con un cargo específico, sino que cualquier integrante del equipo puede asumir ese rol cuando la situación lo amerita. Se abordaron enfoques como el liderazgo situacional, el cual plantea que el estilo de liderazgo debe adaptarse según el contexto y las necesidades del equipo. Además, se resaltó la importancia de la inteligencia emocional, para manejar la presión y los conflictos sin afectar el ambiente de trabajo ni el desempeño del equipo (Editorial Kairós, 2021).

Por otro lado, en la parte técnica se tienen en cuenta conceptos fundamentales como la arquitectura de software, la deuda técnica, las pruebas (testing) y la seguridad. Estos temas son importantes porque un sistema no solo debe cumplir con su funcionalidad, sino también ser seguro, escalable y fácil de mantener en el tiempo.

Este trabajo se basa en el caso de la empresa One Tech Fury S.A.S., dedicada a la venta de productos tecnológicos a través de internet. La empresa cuenta con una plataforma propia que gestiona procesos como el catálogo de productos, las órdenes de compra, los pagos y la atención al cliente (módulo de tickets). El sistema actual presenta varios problemas debido a que fue desarrollado hace más de 10 años y no ha tenido actualizaciones constantes, lo que ha generado fallos, uso de tecnologías desactualizadas, vulnerabilidades de seguridad y dificultades para implementar nuevas funcionalidades. Además, al tratarse de un sistema monolítico y altamente acoplado, cualquier cambio se vuelve más complejo y riesgoso.

Teniendo en cuenta todo lo anterior, este informe busca aplicar lo visto en el seminario para analizar la situación del sistema y plantear una solución que permita mejorar tanto la parte técnica como la forma en que se gestiona el proyecto.

Necesidad Identificada

Se plantea la necesidad de una modernización completa del sistema, la cual incluye: (a) migración a arquitecturas modernas con un sistema desacoplado; (b) utilización de tecnologías más modernas y con soporte a largo plazo; (c) implementación de prácticas de desarrollo modernas como integración continua/despliegue continuo (CI/CD) y DevOps; y (d) mejora en la escalabilidad y mantenibilidad del sistema. Para esto se eligió Node.js como tecnología principal para la API REST, dado que fue concebido como un entorno de ejecución JavaScript asíncrono

basado en eventos, diseñado específicamente para construir aplicaciones de red escalables, capaz de manejar muchas conexiones de manera concurrente (Node.js, s.f.).

Riesgos

La realización del sistema desde cero trae varios riesgos: (a) pérdida de ciertos datos por la incompatibilidad entre estructuras de base de datos y (b) impacto en la experiencia del usuario final durante el período de transición.

Pregunta de Investigación

¿Cuál es la mejor estrategia de gestión de proyectos para desarrollar la modernización completa de un sistema legacy, garantizando la continuidad operativa y mejorando la calidad del sistema?

Metodología Utilizada

Para dar inicio, se debe definir qué es un proyecto. Para ello se utiliza el concepto definido por la metodología PRINCE2 (2001) como "un entorno de gestión que se crea con el propósito de entregar uno o más productos de acuerdo al caso de negocio especificado". Sin embargo, es necesario complementar esta definición con una característica fundamental: la temporalidad. Se define un proyecto cuando el entorno de gestión y planificación tiene establecido un periodo de tiempo y uno o varios entregables claramente definidos.

De acuerdo con la metodología PRINCE2, un proyecto no se limita a ser un entorno de gestión enfocado en la entrega de productos, sino que también se concibe como un proceso temporal, debidamente planificado, supervisado y respaldado por una justificación basada en el caso de negocio. En el contexto del proyecto desarrollado para la empresa One Tech Fury S.A.S.,

PRINCE2 facilita la organización y el control del desarrollo del nuevo sistema de información, garantizando que cada etapa se mantenga alineada con los objetivos empresariales y que los recursos sean aprovechados de forma adecuada.

En el desarrollo de este proyecto se hace necesario aplicar metodologías de trabajo ágiles que permitan organizar el equipo, mejorar la eficiencia y dar solución a la problemática actual del sistema de la empresa One Tech Fury S.A.S. Estas metodologías no solo ayudan a coordinar mejor el trabajo, sino que también permiten ajustar los cambios y mejorar continuamente.

En la práctica, PRINCE2 se vio reflejado en la forma en que se estructuró y controló cada fase del proyecto. Desde el inicio se definió claramente el caso de negocio: la plataforma de One Tech Fury S.A.S.

Durante la ejecución, el equipo realizó revisiones al final de cada fase para evaluar si los resultados obtenidos seguían alineados con los objetivos del proyecto, lo cual corresponde al principio de supervisión continua que propone PRINCE2. Además, la gestión de riesgos que se documentó en cada módulo con el fin de anticipar y mitigar los riesgos ya que es parte fundamental del proceso.

Fases

El proyecto se organiza en seis fases:

- Fase Autenticación
- Fase Módulo de Usuarios
- Fase Módulo de Productos
- Fase Módulo de Órdenes
- Fase Módulo de Tickets
- Fase Implementación

Definición de Roles para Cada Fase

Tabla 1

Definición de Roles para Cada Fase

Rol	Responsabilidad	Importancia
Project Manager	Control en general de la planificación y ejecución de las tareas en cada fase. Encargado de liderar el equipo, comunicar partes del equipo y gestionar los riesgos.	Alta
Líder técnico	Supervisión técnica de avances en el desarrollo del proyecto. Garantizar la calidad de cada módulo y estar pendiente del código y la arquitectura.	Media
Dev Backend	Desarrollar la lógica del negocio y endpoints de la API. Encargado de varias zonas del desarrollo y despliegue.	Baja
Dev Frontend	Desarrollar las interfaces del usuario y diseñar la experiencia del usuario.	Baja
QA Tester	Revisar las funcionalidades y realizar un control de calidad mediante pruebas unitarias y pruebas manuales.	Baja
Stakeholders	Financiar inicialmente el proyecto. Validación del producto y refinamiento, además de regular las acciones que se toman en el proyecto.	Alta

Nota. Elaboración propia.

Fase Autenticación

Descripción

El módulo de autenticación se encarga de controlar el acceso de los usuarios a la plataforma. Su función principal es verificar la identidad o rol de cada usuario mediante el uso de credenciales como correo electrónico y contraseña. Este módulo es muy importante dentro del sistema, ya que permite proteger la información personal de los usuarios y evitar accesos no autorizados. Básicamente, es la base que permite que funcionen otros módulos del sistema, como usuarios, órdenes y pagos. Debido a los problemas actuales del sistema (falta de seguridad y

tecnologías obsoletas), este módulo se plantea con un enfoque moderno que garantice mayor protección y control de acceso.

Planificación

Para el desarrollo del módulo se definieron las funcionalidades principales, así como las tecnologías que se utilizarán. Se propone una arquitectura basada en API REST, separando el frontend del backend para mejorar la escalabilidad del sistema. En cuanto a seguridad, se implementarán: (a) encriptación de contraseñas; (b) uso de tokens (JWT) para sesiones; y (c) validación de datos de entrada, dado que los JWT son el mecanismo estándar para verificar la identidad del usuario sin reenviar credenciales en cada solicitud (Auth0, s.f.).

Los endpoints definidos son: POST /auth/register (registro de usuario); POST /auth/login (inicio de sesión); PUT /auth/update (actualizar token de autenticación); DELETE /auth/delete (eliminar usuario). En cuanto a las vistas del frontend, se diseñaron: registro, login, perfil y edición de perfil.

Ejecución

Durante esta fase se desarrollaron las funcionalidades del módulo. Se implementó el registro de usuarios, donde la contraseña es encriptada antes de ser almacenada en la base de datos, lo que mejora la seguridad del sistema. Para el inicio de sesión, el sistema valida las credenciales ingresadas; si son correctas, se genera un token que permite al usuario mantenerse autenticado dentro de la plataforma. También se implementaron funciones para editar información del usuario y eliminar cuentas, siguiendo una estructura organizada que permite futuras mejoras.

Control y Seguimiento

Se realizaron pruebas básicas para asegurar el correcto funcionamiento del módulo. Se verificaron casos como: (a) registro con datos válidos e inválidos; (b) inicio de sesión correcto e incorrecto; (c) acceso a funciones protegidas; y (d) edición y eliminación de usuarios. También se revisó que las contraseñas estén encriptadas, que no haya accesos sin autenticación y que el sistema responda correctamente ante errores.

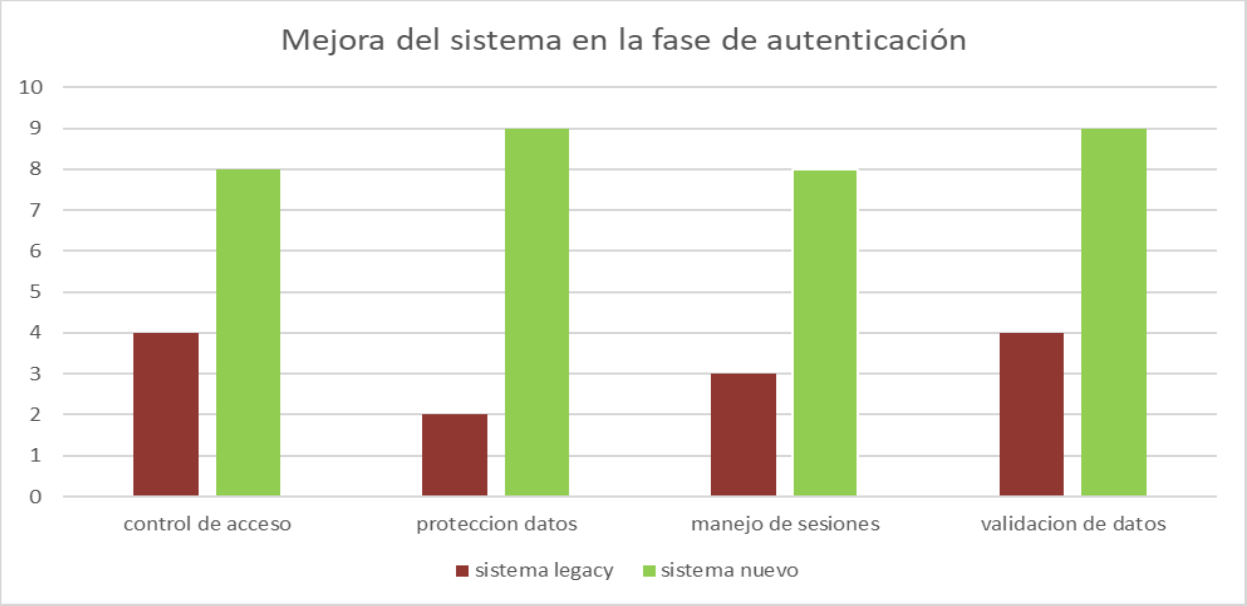
Resultados Obtenidos

Con este módulo se logra mejorar el control de acceso al sistema, aumentando la seguridad de la plataforma. Se obtiene una estructura más organizada y preparada para integrarse con otros módulos del sistema. Además, se reducen los riesgos asociados al manejo de datos sensibles y se facilita la implementación de nuevas funcionalidades en el futuro. La importancia de la gestión de proyectos en esta fase radica en que se puede dividir el desarrollo de todo lo que conlleva el módulo de autenticación por partes, el modelado de la tabla de usuarios, la parte de la autenticación propiamente (end point de la api + frontend), después la parte del registro y finalmente el testing. Todo este desarrollo se guía y se separa para no confundir tareas.

La aplicación de PRINCE2 en esta fase de autenticación ayudó desde un inicio tener todo claro de manera que se pudo controlar la calidad gracias a las pruebas específicas que se realizaron antes de continuar con la fase siguiente.

Figura 1

Diferencia entre el login anterior y el nuevo



Nota. Elaboración propia. La figura muestra que el nuevo sistema mejora todos los aspectos evaluados frente al anterior. Se evidencia mayor seguridad y mejor funcionamiento general.

Fase Módulo de Usuarios

Descripción

Esta fase corresponde al desarrollo del módulo de gestión de clientes y usuarios basados en roles y permisos. El objetivo es desarrollar un módulo de usuarios que incluya la gestión completa con barra de búsqueda, total de órdenes y total de tickets que tenga el usuario. El módulo debe ser escalable, fácil de utilizar y proveer herramientas para la gestión completa de clientes. La primera versión de este módulo solo incluirá dos tipos de usuario: administrador y cliente ya que son los roles principales.

Las funcionalidades esperadas son: (a) listar usuarios; (b) crear usuarios; (c) editar usuarios; (d) suspender o activar usuarios; (e) eliminar usuarios; (f) mostrar columna con total de órdenes; (g) mostrar columna con total de tickets; y (h) mostrar columna con estado de usuario (Activo/Suspendido).

Planificación Basada en Entregables

Tabla 2

Planificación Basada en Entregables – Módulo de Usuarios

Producto	Descripción
Endpoint de usuarios	Servicio backend con endpoints: GET /api/users, GET /api/users/{id}, POST /api/users, PUT/PATCH /api/users/{id}, DELETE /api/users/{id}.
Interfaz de usuario en área administrativa	Panel administrativo que incluya en el menú lateral la opción Usuarios.
Base de datos	Modelo de usuarios con estructura base: {id, name, lastname, email, status, type, verified_at, created_at}.
Documentación	Comentarios bien estructurados en el código y manual de uso para la creación y administración de usuarios.
Casos de prueba	Pruebas unitarias y escenarios funcionales.

Nota. Elaboración propia.

Ejecución

Arquitectura Implementada

Se implementó un enfoque desacoplado: (a) API REST para gestión de usuarios; (b) tabla general de usuarios y relaciones entre usuarios-órdenes y usuarios-tickets; (c) separación frontend/backend para tener un producto totalmente desacoplado que cuente con la facilidad de escalado.

El flujo esperado del módulo incluye: (a) página principal con un listado de usuarios incluyendo paginador, campo de búsqueda, columnas con estado, tipo de usuario, número de órdenes y número de tickets; (b) creación de nuevos usuarios con información básica; (c) edición de información existente del usuario; (d) eliminación de usuarios del sistema sin afectar órdenes y tickets, cuyo historial se debe mantener; y (e) visualización del historial de órdenes y tickets por parte de los usuarios.

Control y Seguimiento

Tabla 3

Gestión de Riesgos – Módulo de Usuarios

Riesgo	**Mitigación**
Uso de un algoritmo de encriptación diferente para las contraseñas de usuario	Desarrollar una vista que exija la actualización de contraseña al usuario.
Mala experiencia de usuario	Confusión por tener que actualizar contraseña y pérdida de preferencias personales.

Nota. Elaboración propia.

El control de calidad se realizó mediante pruebas unitarias y pruebas de integración para comprobar que todas las partes del módulo funcionen correctamente.

Resultados Obtenidos

Al finalizar esta fase se lograron los siguientes resultados: (a) implementación de módulo independiente; (b) mejora en la administración de usuarios; (c) mejora en la seguridad de la información de los usuarios y encriptación; y (d) módulo escalable a largo plazo.

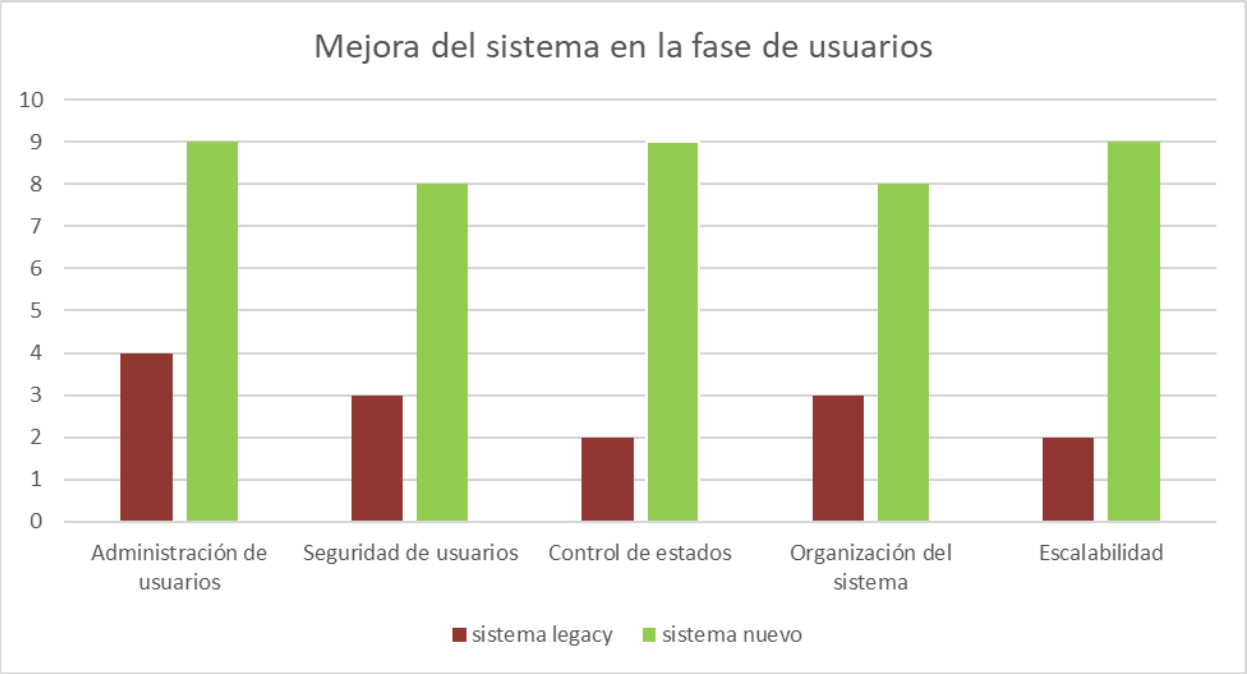
La gestión de proyectos en esta fase fue importante porque el líder de proyectos, detalló al máximo cada funcionalidad del módulo de usuarios, supo gestionar la parte del desarrollo backend y frontend para que primeramente la parte del backend entregará los endpoints con la

documentación y como segundo la parte del frontend pudiera entender como funcionaba cada endpoint y hacer su parte respectivamente.

La aplicación de PRINCE2 que estructura los entregables por fases facilitó planificar y organizar el seguimiento de este módulo de usuarios resultando en una fase con mejoras en concreto, siendo estas la de seguridad y administración de usuarios.

Figura 2

Estadísticas comparativas entre el módulo anterior de usuarios y el nuevo



Nota. Elaboración propia. La figura presenta una mejora considerable en el nuevo módulo de usuarios mucho mejor que el anterior. Se evidencia mejora en administración, seguridad y gestión de la información.

Fase Módulo de Productos

Descripción

En esta fase de desarrollo se enfoca en la construcción del módulo de gestión de productos y en la edición de información base de cada producto. El objetivo es desarrollar un módulo sencillo, moderno y escalable que permita la administración fácil de cada uno de los productos. Las funcionalidades esperadas son: (a) listar productos; (b) crear productos con nombre, descripción, precio y stock; (c) editar productos; (d) eliminar productos; y (e) gestionar las imágenes en una relación uno a muchos, dado que un producto puede tener muchas imágenes.

Planificación Basada en Entregables

Tabla 4

Planificación Basada en Entregables – Módulo de Productos

Producto	Descripción
Endpoint de productos	Servicio backend en Node.js con endpoints: GET /api/products, GET /api/products/{id}, POST /api/products, PUT/PATCH /api/products/{id}, DELETE /api/products/{id}.
Interfaz de usuario en área administrativa	Panel administrativo con opción Productos en el menú lateral, que despliega: Todos y Crear Producto.
Base de datos	Modelo de productos, modelo de imágenes con relaciones y restricciones para que cuando se elimine un producto, las imágenes también se eliminen.
Documentación	Comentarios bien estructurados en el código y manual de uso.
Casos de prueba	Escenarios funcionales.

Nota. Elaboración propia.

Ejecución

Arquitectura Implementada

Se implementó un enfoque desacoplado basado en servicios con un frontend en React y backend en Node.js con Express: (a) API REST para gestión de productos; (b) base de datos independiente; (c) relación uno a muchos entre productos e imágenes; y (d) separación frontend/backend para facilitar el escalado.

El flujo esperado del módulo incluye: (a) página principal con listado de productos con paginador y campo de búsqueda; (b) creación de nuevos productos con información básica; (c) asociación de múltiples imágenes a cada producto; (d) edición de información existente; (e) eliminación de productos del sistema; y (f) visualización de productos actualizados en el catálogo.

Control y Seguimiento

Tabla 5

Gestión de Riesgos – Módulo de Productos

Riesgo	**Mitigación**
Pérdida de órdenes con productos relacionados	Para esta parte no hay mucha mitigación dado que la estructura nueva será diferente.
Inconsistencia en datos	Realizar scripts que se ejecuten en segundo plano para la migración y adaptación de datos si es posible.
Retrasos	División en tareas pequeñas y compensaciones económicas adicionales para los desarrolladores.

Nota. Elaboración propia.

El control de calidad se realizó mediante pruebas unitarias y pruebas de integración para verificar que todos los endpoints retornen y acepten la información esperada.

Resultados Obtenidos

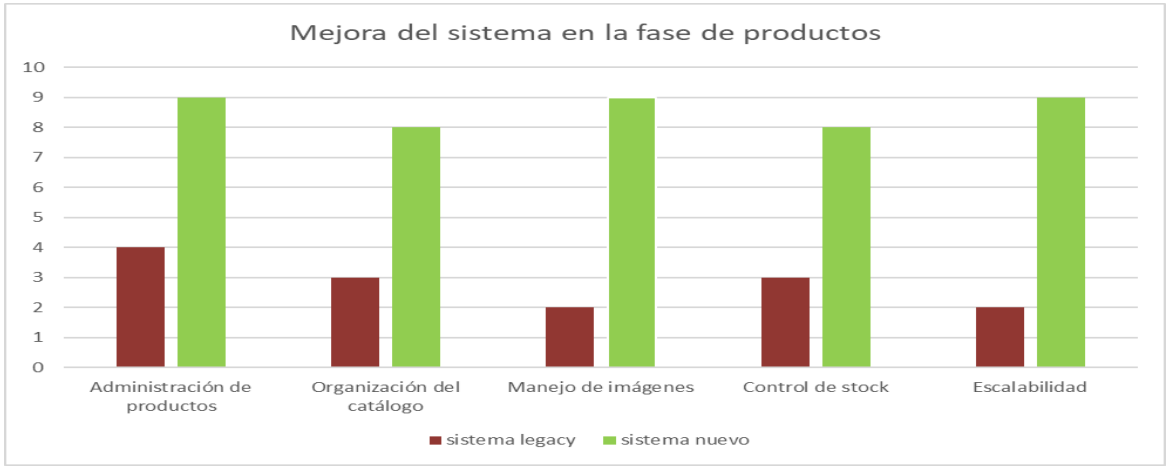
Al finalizar esta fase se logró implementar un módulo de productos totalmente funcional con sus respectivas pruebas. Ahora se pueden administrar los productos, información básica, stock e imágenes sin ningún problema. Se obtuvieron como resultados principales: (a) implementación de módulo independiente y (b) mejora en la administración de productos.

En esta fase la gestión fue bastante importante, ya que el módulo de productos tenía que integrarse con un "submódulo" de imágenes, entonces el equipo coordinó con el líder de proyectos y el líder técnico de manera que se desarrollará primeramente los endpoints de productos e imágenes. Una vez ya se tuvieran esos endpoints se tenía que hacer una revisión en la relación de las tablas para ver que todo estuviera bien relacionado y con las restricciones. Finalmente se coordinó con los encargados del frontend para que realizaran todas las vistas. Esto nos da a entender que todos los procesos deben de ir coordinados y priorizados.

PRINCE2 en esta fase logró anticipar problemas en el cambio de datos durante la migración esto gracias a la entrega completa del módulo anterior.

Figura 3

Estadísticas comparativas entre el módulo anterior de productos y el nuevo



Nota. Elaboración propia. La figura refleja que el nuevo módulo de productos supera al anterior por mucho, en funcionalidad, gestión, mejor organización y administración de los productos.

Fase Módulo de Órdenes

Descripción

Esta fase se centra en desarrollar un módulo encargado de gestionar las órdenes de los usuarios desde la parte administrativa. Los objetivos son: (a) desarrollar un módulo que permita la gestión de órdenes, además de ver sus detalles; y (b) garantizar la sincronización de stock de productos.

Las funcionalidades esperadas son: (a) gestión completa y detallada de órdenes; (b) listado de órdenes con filtros por estado (Pendiente, Pagado, Enviado, Cancelado, Reembolsado); (c) visualización detallada de la orden con productos comprados, cantidades, precios y datos de envío; (d) actualización de estados por parte del administrador; y (e) historial de órdenes accesible para el cliente.

Planificación Basada en Entregables

Tabla 6

Planificación Basada en Entregables – Módulo de Órdenes

Producto	Descripción
Endpoint de órdenes	Servicio backend en Node.js con endpoints: GET /api/orders, GET /api/orders/{id}, POST /api/orders, PUT /api/orders/{id}/status.
Interfaz administrativa	Sección Órdenes en el panel lateral que permite gestionar despachos y devoluciones.
Base de datos	Modelo con estructura: {id, user_id, total_price, status, payment_method, shipping_address, created_at} y tabla pivote para productos.
Documentación	Comentarios técnicos y manual de procesos para la gestión de reembolsos.
Casos de prueba	Pruebas de transaccionalidad y validación de stock.

Nota. Elaboración propia.

Ejecución

Arquitectura Implementada

Se mantiene el enfoque desacoplado (frontend en React, backend en Node.js) para asegurar la escalabilidad. Se establece una relación muchos a muchos entre órdenes y productos a través de una tabla de detalles, y una relación uno a muchos entre usuarios y órdenes. Al generar una orden exitosa, el sistema dispara automáticamente una actualización en el módulo de productos para descontar el stock y de esta manera tener un control estricto sobre el inventario.

El flujo esperado del módulo es: (a) creación, donde el cliente finaliza su compra y se genera una orden con estado inicial en Pendiente; (b) validación que es donde el sistema verifica el pago a través de integración con pasarelas y cambia el estado a Pagado; (c) gestión donde el

administrador visualiza la orden, los detalles de la orden como productos, cantidades, prepara el envío y actualiza a Enviado; y (d) completado de la orden, donde se realiza una trazabilidad de la orden, confirmando que esta fue entregada y se actualiza el estado de la orden a Completa.

Control y Seguimiento

Se monitorea la correcta transición de estados para evitar fallos transaccionales, uno de los problemas críticos del sistema legacy.

Tabla 7

Gestión de Riesgos – Módulo de Órdenes

Riesgo	**Mitigación**
Fallas en la pasarela de pago	Implementación de webhooks para confirmaciones asíncronas de pago.
Inconsistencia de stock	Uso de transacciones de base de datos para asegurar que el stock solo se descuente si la orden se crea correctamente.

Nota. Elaboración propia.

Resultados Obtenidos

Los resultados obtenidos en esta fase fueron: (a) capacidad de rastrear cada orden completamente y productos desde la venta hasta la entrega; (b) reducción de deuda técnica ya que se eliminan procesos hardcodeados y con malas prácticas que se tenían el en antiguo sistema de pago por una estructura modular mucho más segura; y (c) el sistema ya es más escalable dado que el sistema ahora soporta altos volúmenes de transacciones sin los bloqueos del monolito antiguo.

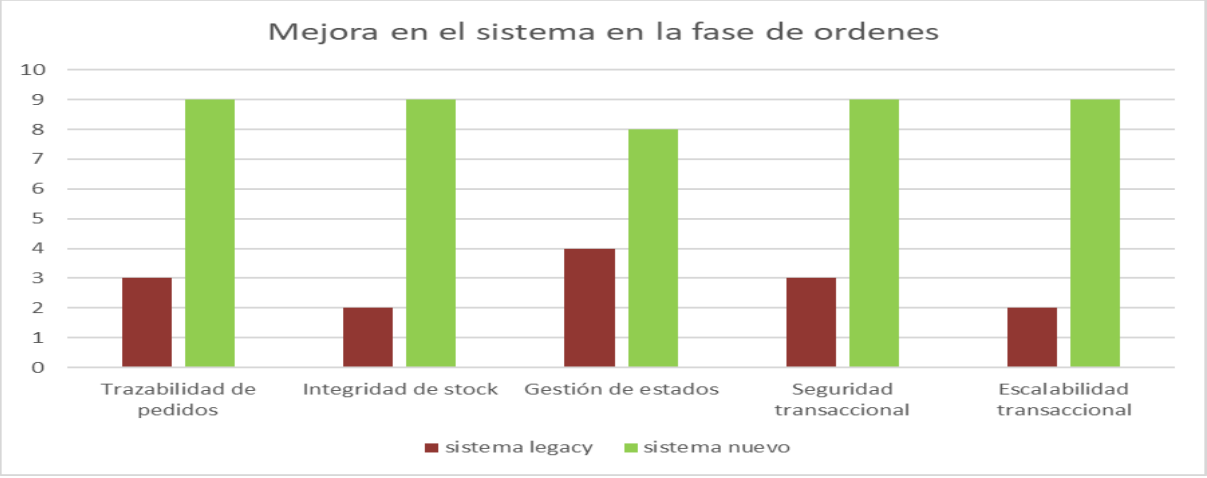
La gestión de proyectos en esta fase fue fundamental, dado que se trata de un módulo crítico, es decir, un módulo de alta importancia en la gestión de un ecommerce. El líder de proyectos desempeñó un papel crucial al planificar, coordinar controlar cada etapa del desarrollo,

ya que el líder teniendo las prioridades claras tuvo que agendar la implementación secuencial de este módulo, en primer lugar la integración de endpoints generales (detalles generales de la orden, productos, etc), segundo la creación de los webhooks para la confirmación de pagos automatica y como tercero la parte del frontend con una excelente UX.

El control de PRINCE2 fue muy importante en esta fase ya que al tener claro lo que fallaba en el sistema anterior permitió revisar el avance y cierre de cada iteración para detectar las fallas, una por una y de esta forma tener un sistema más seguro y completo

Figura 4

Comparativas entre funciones del módulo anterior de pedidos y el nuevo



Nota. Elaboración propia. La figura detalla una optimización del control y seguimiento frente al sistema anterior, entre estos esta mayor eficiencia en la gestión y procesamiento de pedidos.

Fase Módulo de Tickets

Descripción

En esta fase se desarrolla el módulo de tickets. El objetivo es construir un módulo que tenga un listado general con paginador, filtrado por estado, que permita editar el estado del ticket

(cerrado, en espera, en proceso, solucionado), responder a los mensajes del cliente y administrar los archivos que el cliente suba. Las funcionalidades esperadas son: (a) listar tickets; (b) editar datos generales del ticket; (c) responder mensajes dentro del ticket; (d) gestionar los archivos que se suben al ticket; y (e) eliminar tickets.

Planificación Basada en Entregables

Tabla 8

Planificación Basada en Entregables – Módulo de Tickets

Producto	Descripción
Endpoint de tickets y archivos	Servicio backend en Node.js con endpoints para tickets: GET /api/ticket, GET /api/ticket/{id}, POST /api/ticket, PUT/PATCH /api/ticket/{id}, DELETE /api/ticket/{id}. Endpoints para mensajes: GET /api/ticket/{id}/messages, POST api/ticket/{id}/messages Endpoints para archivos: GET /api/ticket/{id}/files, POST /api/ticket/{id}/files, DELETE /api/ticket/{id}/files/{file_id}.
Interfaz de usuario en área administrativa	Panel administrativo con pestaña Soporte que lleva al listado de tickets.
Base de datos	Modelo de tickets y modelo de archivos con relaciones y restricciones para que cuando se elimine un ticket, los archivos también se eliminen.
Documentación	Comentarios bien estructurados en el código y manual de uso.
Casos de prueba	Escenarios funcionales.

Nota. Elaboración propia.

Ejecución

Arquitectura Implementada

Se implementó un enfoque desacoplado: (a) API REST para la gestión de tickets; (b) API REST para la gestión de archivos vinculados a un ticket; y (c) tabla de tickets con relaciones entre usuarios-tickets y tickets-archivos.

El flujo esperado del módulo establece que cuando se abre un ticket, el administrador puede: (a) leer el mensaje del usuario; (b) responder dentro del mismo hilo; (c) cambiar el estado (en proceso, solucionado, cerrado); y (d) revisar o gestionar los archivos adjuntados. El proceso normalmente sigue así: el ticket pasa de en espera a en proceso, luego a solucionado y finalmente a cerrado, dependiendo de cómo avance el caso.

Control y Seguimiento

Se realizaron pruebas básicas para asegurar el correcto funcionamiento del módulo. Se verificaron casos como: (a) creación de tickets; (b) subida de archivos a cada ticket; (c) edición del estado del ticket; (d) responder a tickets; y (e) eliminación de tickets y verificación de que los archivos también se eliminen. También se revisó que los archivos de los tickets queden bien relacionados, que no haya accesos sin autenticación y que los tickets cerrados no puedan recibir más respuestas hasta que se reabra el caso.

Resultados Obtenidos

Al finalizar esta fase se logró implementar un módulo de tickets funcional que permite gestionar de manera más ordenada las solicitudes de los usuarios referentes a problemas con pedidos o dudas en general. Ahora el sistema cuenta con un espacio administrativo para gestionar los tickets, lo que facilita la comunicación y permite llevar un seguimiento más claro de cada caso.

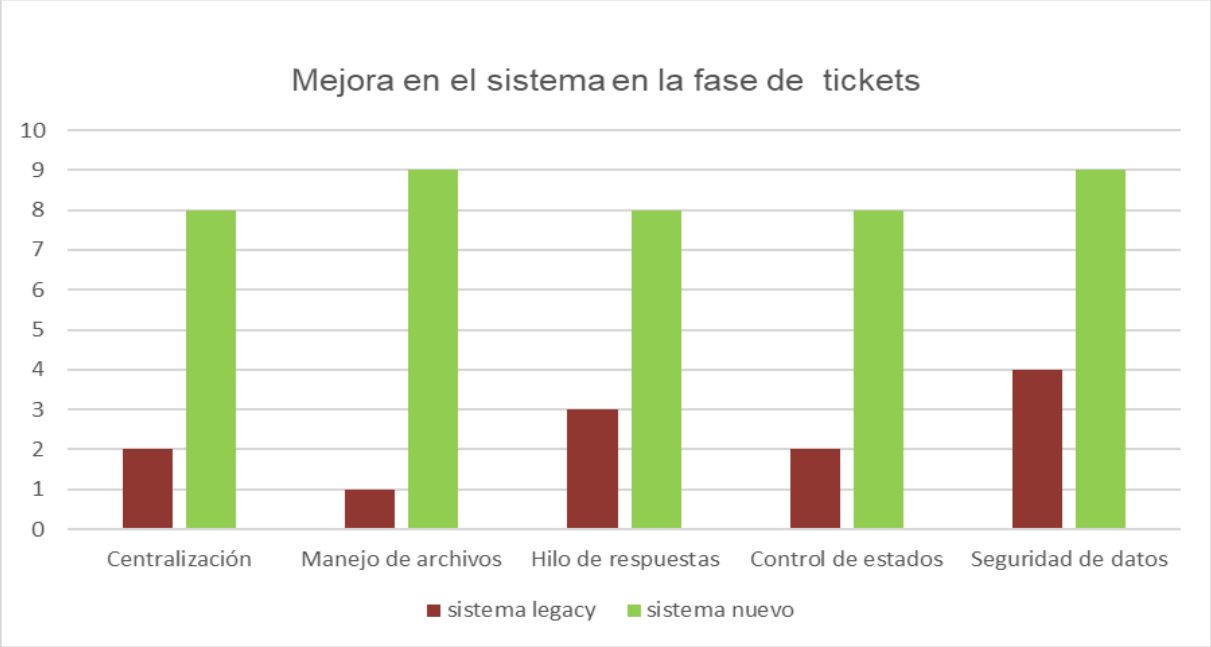
La gestión de proyectos en esta fase fue importante porque el módulo de tickets no era solo una funcionalidad simple, sino que involucra tickets, mensajes entre el agente y el usuario, archivos y estados. Con el líder de proyectos la planificación clara de entregables entre la parte del backend y frontend, era fácil que el equipo desarrollará partes desconectadas que luego no

estuvieran sincronizadas ya que primeramente el líder técnico tuvo que gestionar la parte del backend los endpoints para el ticket, mensajes y archivos. Son tres endpoints diferentes y deben tener estándares de respuestas en la API con formato json. Ya teniendo esta parte lista, el líder técnico se encargó de asignar las tareas a los del frontend para posteriormente pasar todo el módulo al área de QA.

Gracias a PRINCE2 se construyó un módulo más organizado y funcional manteniendo al equipo informado sobre las funcionalidades prioritarias que se debían realizar, evitando de esta manera más trabajo innecesario.

Figura 5

Comparativas entre funciones y rendimiento del módulo anterior de tickets vs vieja versión



Nota. Elaboración propia. Se evidencia que el nuevo módulo de tickets ha mejorado en la gestión de solicitudes frente al anterior, un mejor seguimiento, organización y atención de los casos.

Fase de Implementación

Descripción

Esta fase corresponde a la implementación del sistema en el servidor de producción, donde se integran todos los módulos desarrollados (autenticación, usuarios, productos, órdenes y tickets). El objetivo es reemplazar el sistema anterior sin afectar la operación del negocio y asegurar que el nuevo sistema funcione correctamente en un entorno real con usuarios reales, donde los usuarios puedan iniciar con la realización de pedidos.

Planificación Basada en Entregables

Tabla 9

Planificación Basada en Entregables – Fase de Implementación

Producto	Descripción
Entorno de producción	Configuración de servidores y base de datos para el despliegue.
Migración de datos	Traslado de la información del sistema anterior al nuevo.
CI/CD	Automatización de procesos de despliegue.
Pruebas finales	Validación del sistema antes de producción.
Monitoreo	Seguimiento del sistema en funcionamiento.
Documentación	Manual técnico y guía básica de uso.

Nota. Elaboración propia.

Ejecución

Arquitectura Implementada

Se mantiene el enfoque desacoplado: (a) frontend en React; (b) backend en Node.js; (c) API REST; y (d) base de datos.

Tabla 10

Flujo de Implementación

Actividad	Descripción
Despliegue	Instalación del sistema en servidores.
Base de datos	Configuración del entorno de datos.
Migración	Transferencia de información del sistema antiguo.
Integración	Unión de todos los módulos.
Validación	Pruebas de funcionamiento general.

Nota. Elaboración propia.

Control y Seguimiento

El seguimiento del sistema en producción incluyó: (a) verificación de funcionamiento; (b) control de errores; y (c) revisión de rendimiento.

Tabla 11

Gestión de Riesgos – Fase de Implementación

Riesgo	Mitigación
Pérdida de datos	Copias de seguridad constantes.
Fallos en producción	Plan de rollback (retorno al sistema anterior).
Errores de integración	Pruebas exhaustivas previas al lanzamiento.

Nota. Elaboración propia.

El control de calidad incluyó pruebas de integración y validación general del sistema.

Resultados obtenidos

El sistema quedó funcionando correctamente, cumpliendo con los objetivos de modernización de One Tech Fury S.A.S.

Todos los módulos se realizaron efectivamente gracias a la planificación estructurada de PRINCE2 permitiendo gestionar todo de manera sencilla, sin margen de error por la gestión y pruebas que se realizaron cada módulo entregado.

Resultados obtenidos

El sistema quedó funcionando correctamente, cumpliendo con los objetivos de modernización de One Tech Fury S.A.S.

Todos los módulos se realizaron efectivamente gracias a la planificación estructurada de PRINCE2 permitiendo gestionar todo de manera sencilla, sin margen de error por la gestión y pruebas que se realizaron cada módulo entregado.

Tabla 12

Resultados Obtenidos – Fase de Implementación

Resultado	Descripción
Sistema en producción	Plataforma funcionando correctamente.
Integración	Todos los módulos conectados.
Rendimiento	Mejora notable en tiempos de respuesta.
Seguridad	Mayor protección de datos sensibles.
Escalabilidad	Sistema preparado para crecer según la demanda.

Nota. Elaboración propia.

Importancia de La Gestión de Proyectos

La gestión de proyectos en la refactorización del sistema de la empresa, es de mucha importancia ya que se tenía que planear todo desde cero, se planeó en fases gracias a la metodología de PRINCE2, se repartieron los roles esto ayudó a entender que debía hacer cada

integrante del equipo evitando conflictos internos, se realizó una planificación basada en entregables en cada fase ayudando a tener una serie de pasos donde se registra, que se hizo, cómo y qué resultado, también se realizó un seguimiento detallado de calidad y testing en cada fase. Así que la gestión de proyectos fue fundamental para mantener un orden, estructurar cada fase, mantener un seguimiento y retroalimentación a largo plazo.

Lista de Tablas

Conclusiones

Tras la implementación final, el sistema mostró una mejora notable en los tiempos de respuesta y una estabilidad superior en un entorno de producción real con pruebas en escenarios reales. Se logró reemplazar un sistema monolítico de más de 10 años por una arquitectura desacoplada (Frontend en React y Backend en Node.js), eliminando la deuda técnica y la obsolescencia del sistema anterior. Dividir el proyecto en fases independientes por módulo, gracias a la metodología PRINCE2 fue clave para mantener el control del desarrollo.

Cada entrega permitió detectar errores a tiempo y ajustar el rumbo sin afectar los demás componentes del sistema. La separación entre frontend y backend no solo nos da un sistema desacoplado y escalable, sino que también permitió que cada parte del equipo trabajara de forma más independiente y sin depender constantemente de la otra. Además de que trabajar por fases facilitó muchos resultados en poco tiempo.

Gestionar la seguridad desde el módulo de autenticación establece en el sistema una base sólida. Decisiones como el uso de JWT y la encriptación de contraseñas con un nuevo algoritmo evitaron vulnerabilidades que en el sistema anterior eran un riesgo constante. A lo largo del proyecto se comprendió que la deuda técnica no es solo un problema de código, sino también de organización y toma de decisiones. Mantener reuniones de seguimiento y documentar cada fase evitó que los errores del sistema anterior se repitieran.

Referencias

Auth0. (s.f.). *Introduction to JSON Web Tokens*. JWT.io. https://jwt.io/introduction

Editorial Kairós. (2021, 16 de marzo). *Inteligencia emocional.* https://www.editorialkairos.com/catalogo/p/inteligencia-emocional

Node.js. (s.f.). *Sobre Node.js*. https://nodejs.org/es/about

PRINCE2. (2001). *PRINCE2: Metodología de gestión de proyectos.* Prince.

Suárez, N. R. (2015). *La gestión del tiempo* [Trabajo de grado, Universidad de La Laguna].