



TRABAJO DE GRADO
Opción Seminario-Diplomado.

Seminario de Programación Mobile

Corporación Universitaria Remington.

Facultad de ingeniería.

Ingeniería de sistemas.

Nelson Felipe Murillo Quinayas

Tutor: Jonathan Stick Campos Núñez.

Trabajo para seminario como opción de grado.

2025.

Tabla de Contenidos

Resumen.....	3
Palabras clave.....	3
Objetivo general.....	4
Objetivos específicos	4
Metodología de búsqueda de la información.	5
Sustentación teórica de la pregunta.....	6
1. Flutter.....	6
2. Framework	7
3. Requerimiento para Flutter	7
3.1. Requerimientos de Hardware y Software.	7
4. Android Studio.....	8
Informe de la Actividad	9
1. Instalación	9
2. Especificación del proyecto y prototipado visual.	10
2.1. Interfaz gráfica.....	11
Conclusiones.	15
Anexos	16
Referencias.....	21

Resumen

El presente trabajo tiene como propósito desarrollar una aplicación móvil utilizando **Flutter**, un framework de código abierto creado por Google que permite la construcción de aplicaciones a partir de un código base. La investigación se centra en analizar y aplicar las ventajas que ofrece este entorno, tales como la rapidez en el desarrollo, la consistencia en el diseño de interfaces y la compatibilidad con sistemas operativos Android.

Como caso de aplicación se plantea el diseño de una herramienta orientada a la gestión de rutinas de entrenamiento. No obstante, el punto central del trabajo no recae en el ámbito del ejercicio físico, sino en el proceso técnico de construcción, implementación y validación de la aplicación basada en dicho framework.

Palabras clave

Framework, aplicación, Flutter, Aplicación móvil, Desarrollo de software, Código abierto, Widgets, Usabilidad, Rendimiento, Arquitectura de software, Gestión de rutinas de entrenamiento, Personalización, Actividad física, Salud digital

Pregunta orientadora de la búsqueda

Preguntas que fueron surgiendo con la participación y trabajo diario del seminario Flutter:

- ¿Qué es Flutter?
- ¿Qué es un Framework?
- ¿Cuáles son los requerimientos para poder programar?

Objetivo general

Desarrollar una aplicación móvil mediante el uso de Flutter, enfocada en la planificación de rutinas de entrenamiento en el gimnasio, con el fin de analizar el seguimiento de los ejercicios. Demostrando las capacidades técnicas de este framework en la construcción.

Objetivos específicos

- Desarrollar el backup de las vistas de la aplicación que se piensa desarrollar.
- Comprender y analizar los componentes que requiere para el desarrollo de la aplicación.
- Dar conocimiento de los fundamentos que requiere el framework para su instalación y su ejecución.
- Codificar el código para el desarrollo de la aplicación móvil que se tiene planeado hacer según el modelo backup.

Metodología de búsqueda de la información.

La metodología empleada para este proyecto fue mediante la búsqueda de artículos, foros, la página oficial de Flutter y otros medios adicionales.

Se hizo uso de herramientas como Google, Google Académico y videos de YouTube. La búsqueda se realizó para consultar temas relacionados con la aplicación y obtener información relevante que permitiera alcanzar los objetivos planteados con anterioridad, así como para la instalación de extensiones que faciliten un buen desarrollo.

Sustentación teórica de la pregunta

1. Flutter



Imagen 1 (Logo de Flutter).

Flutter es un framework de desarrollo de aplicaciones móviles multiplataforma creado por Google. Es de código abierto y permite construir aplicaciones tanto para Android como para iOS. Flutter es el kit de herramientas de interfaz de usuario portátil de Google para crear atractivas aplicaciones compiladas de forma nativa para dispositivos móviles, web y de escritorio a partir de una única base de código. Flutter funciona con código existente, lo utilizan desarrolladores y organizaciones de todo el mundo, y es gratuito y de código abierto.

Está construido por capas, estando el motor escrito en C/C++ y las librerías en Dart. También usa Skia para el renderizado 2D. (Rodríguez, 2019). El objetivo de esta herramienta es renderizar el componente central del motor de Flutter, encargándose de crear la interfaz de usuario en código (widgets) en píxeles visibles en la pantalla. Además, Skia es utilizado por el Flutter para dibujar sobre el proyecto. Cuando se crea un widget, Flutter emite la orden en comandos a Skia, como “dibuja este texto”, y el Skia se encarga de ejecutarlo y mostrar en la pantalla.

2. Framework

Un framework, traducido habitualmente al español como «marco de trabajo», es una estructura o conjunto de herramientas y componentes que nos proporciona una base para desarrollar aplicaciones o crear páginas web de una manera más organizada, robusta y escalable. (Zúñiga, 2025). Sin embargo, se suele pensar que el framework es toda la serie de componentes y directrices sobre los cuales se va a crear todo el proyecto.

Un framework suele ser comparado con una librería, ya que son conceptos que suelen asociarse, sin embargo, aunque los dos sirven para resolver problemas de una manera más eficiente, se dice que el framework es el pilar base de donde se desarrolla la aplicación, mientras que una librería es un conjunto de funciones creadas y reutilizadas para facilitar herramienta de forma más simple y concretas.

Entre las ventajas clave de usar un framework esta la optimización de desarrollo, la rapidez de respuesta y la aceleración de procesos, al ofrécenos una base estructurada sin tener que preocuparnos por crear arquitectura y ciertas funcionalidades que se suelen repetir.

3. Requerimiento para Flutter

Para la instalación de Flutter en el sistema se requieren ciertos requerimientos mínimos puesto que el framework es un poco pesado, lo cual necesita que el hardware y software lo soporte y lo pueda correr para una facilidad de uso.

3.1. Requerimientos de Hardware y Software.

- Memoria RAM: Mínima 4 GB de RAM, recomendable 8GB o superior para un desarrollo fluido, siendo que el Flutter requiere mucha memoria es lo más recomendable.

- Sistema operativo:
 - Windows 10 o superior (64 bits).
 - MacOS (Versiones 10.14 Mojave o posteriores).
- Entorno de Desarrollo:
 - Android Studios (incluye emulador para Android y herramientas necesarias para el SDK).
 - Visual Studio Code, aunque se pueden emplear otros, el más recomendable es este por la facilidad de extensiones.
- Dependencia y SDK:
 - Dark SDK (viene incluido en el Flutter).
 - Android SDK (es para el desarrollo móvil).
 - Git (para clonar repositorios y manejar versiones del Flutter).

4. Android Studio.

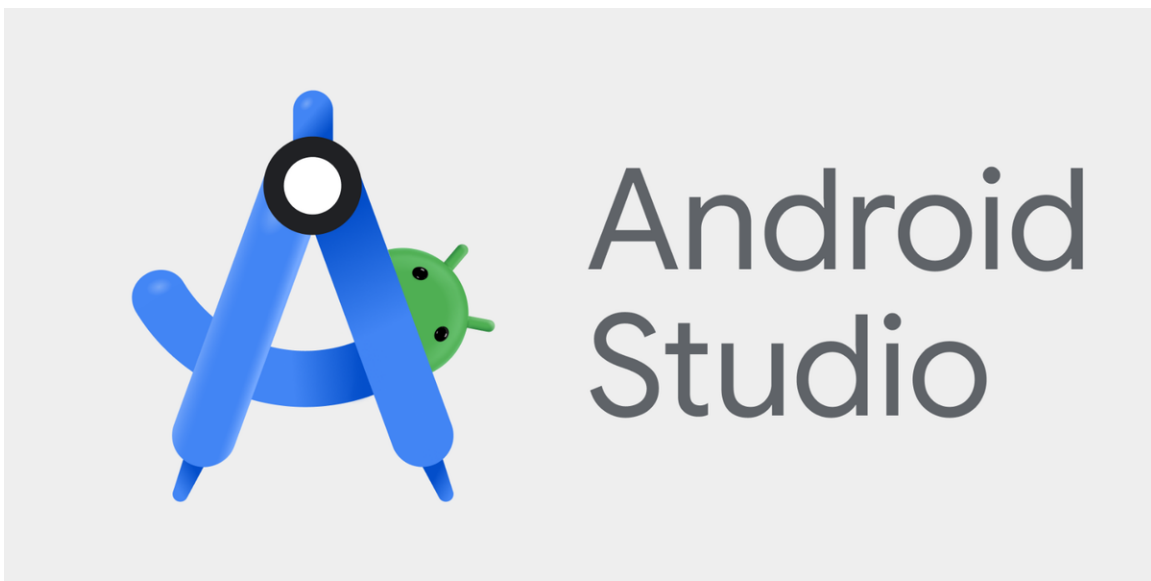


Imagen 2 (Logo de Android Studio).

Android Studio es el entorno de desarrollo integrado (IDE) oficial que se usa en el desarrollo de aplicaciones para Android. Basado en el potente editor de código y las herramientas para desarrolladores de IntelliJ IDEA.

Además, en la instalación de Android Studio se incluirá también el SDK de Android y sus herramientas. Esto es necesario para poder ejecutar una APK en los dispositivos Android, también se hará el uso del Android Studio para otras herramientas de incorpora el IDE como lo es el simulador.

Informe de la Actividad

Durante los periodos de clases se estuvo viendo vario temas referentes a la programación, instalación y ejecución de cómo funciona Flutter, a continuación, veremos los resultados de como instalar Flutter, algunas funciones de programación, el mockup del proyecto que se quiere obtener como primera versión.

1. Instalación

Para la instalación del Flutter no iremos al sitio web de Flutter (<https://docs.flutter.dev/>), vamos a empezar y seleccionamos nuestro dispositivo en el cual lo instalaremos (Android, MacOs, Linux) y en Android ya que se programará aplicaciones móviles.

Primer paso es la instalación del Git estos se hace para instalar los SDK del Flutter que se encuentran alojados ahí. Segundo paso es la instalación de Android Studio para poder ejecutar las aplicaciones como se mencionó con anterioridad (4. Android Studio). Como tercer paso instalaremos un editor de código o entorno de desarrollo. En este caso elegimos a Visual Studio Code ya que este nos brindara facilidad con las extensiones del Flutter y algunos componentes.

Una vez con el Visual Studio Code se va a ir a la parte de extensión y buscaremos el Flutter y lo instalamos. Una vez instalado volvemos a la página del Flutter y se descargara el Flutter para Windows, el archivo que se descarga la recomendación es almacenar desde una carpeta dev creada en el disco local para evitar errores de permisos.

Cuando se obtenga la carpeta se debe ir a la parte de variables de entorno y agregar las rutas de la carpeta del Flutter de igual manera en el path de las variables de entorno.

Después de haber hecho esas instalaciones, vamos a configurar el entorno para el desarrollo de Android, descargando las herramientas del SDK conforme la guía de la página de Flutter.

Como último paso es aceptar las licencias de Android con el comando “C:> flutter doctor --android-licenses”, verificando que todo salió correctamente se revisara que la configuración de desarrollo del Flutter salió todo correctamente consultando en consola con “flutter doctor”, después de verificar que todo fue instalado correctamente, se puede empezar con el desarrollo.

2. Especificación del proyecto y prototipado visual.

La presente propuesta de proyecto tiene como finalidad el diseño y desarrollo de una aplicación móvil, que permita a los usuarios acceder a unas rutinas de gimnasio, seleccionando aquellas que se adapten mejor a sus objetivos personales. Esto con el fin que, en la actualidad, la práctica de actividades físicas se ha vuelto más recurrentes. Sin embargo, a pesar de que cada día es más frecuente las personas que van a los gimnasios se encuentran con la dificultad de organizar las rutinas y como se debe realizar los ejercicios.

La propuesta busca inicialmente construir una primera versión con unas características base que es el seguimiento de los días que se entrenó con el calendario y mostrar cuales son los ejercicios más recomendables para trabajar en el musculo.

2.1. Interfaz gráfica.

La aplicación contará con una interfaz de vista de bienvenida y en primera vista se podrá visualizar el calendario y los diferentes músculos que se va a poder trabajar.

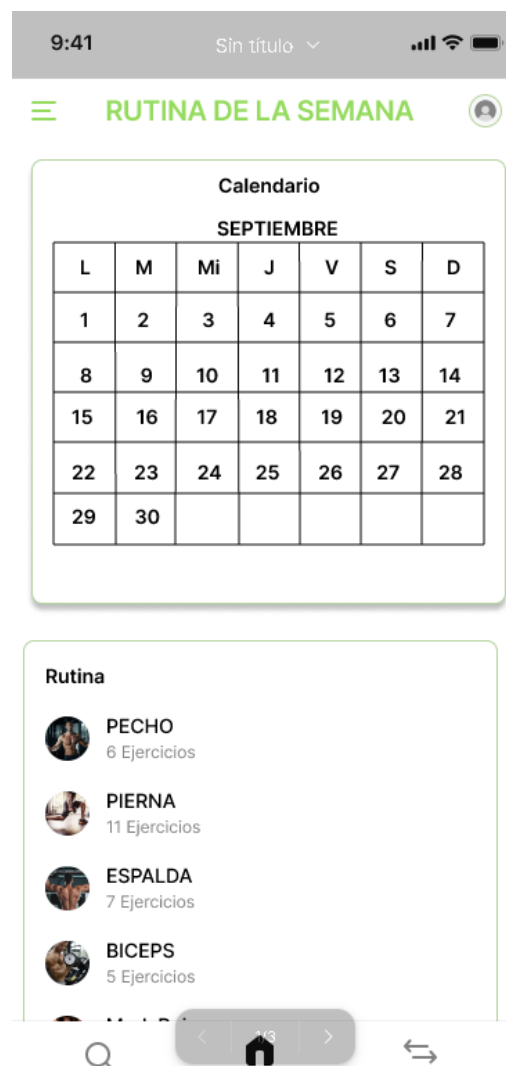


Imagen 3 (Vista de principal).

Como podemos apreciar en la imagen 3, esa es la vista principal de la aplicación, vamos a poder ver el calendario el cual colocara una notificación de los días que se entrenó. En la parte inferior veremos unas tarjetas las cuales al seleccionarlas no llevara a todas las rutinas que están alojadas en cada grupo muscular.

En la parte superior derecha encontramos el botón que nos llevar al login que cual nos permite registrarnos para obtener notificaciones, actualizaciones y demás cosas.

9:41

RUTINAS GYM

Crea una cuenta
Ingresa tu correo electrónico
para registrarte en esta aplicación

correoelectrónico@dominio.com

Continuar

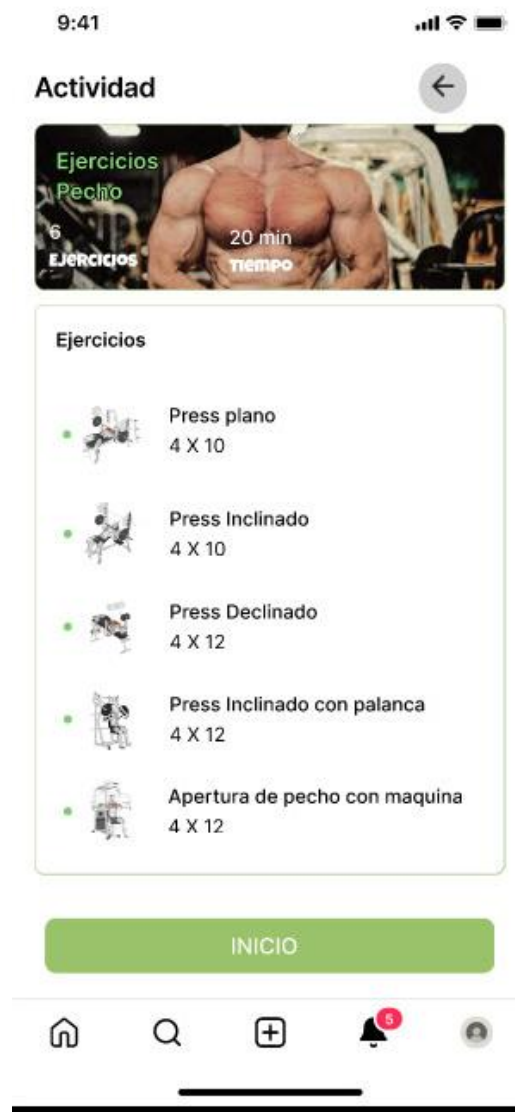
Continuar con Google

Continuar con Apple

Al hacer clic en continuar, aceptas nuestros [Términos de servicio](#) y [Política de privacidad](#)

Imagen 4 (Vista Login).

Esta es la vista en la que crearemos el registro del usuario para obtener las notificaciones, recomendaciones y actualizaciones. Además, se podrá hacer el registro con google y con Apple, esto para que el usuario si desea pueda ser notificado al correo de su seguimiento.

**Imagen 4 (Vista rutina).**

Después de seleccionar el grupo muscular que se va a entrenar, se dirigirá a la vista (imagen 4) la cual como se puede ver nos da un tiempo aproximado y nos referencia de cuantos ejercicios se van a realizar. Al iniciar el entrenamiento nos dará una guía de cómo hacer el ejercicio, cuántas repeticiones y series, Igualmente, contabilizará el tiempo de descanso para poder hacer la pausa de recuperación y seguir con el siguiente ejercicio.

Se pretende emplear también un botón para editar la rutina y seleccionar los ejercicios que mejor le convengan al usuario, para así organizar su rutina y su tiempo de entrenamiento.

Conclusiones.

A lo largo del desarrollo de este trabajo se comprendió lo que era el desarrollo para aplicaciones móviles, que era un framework y el Flutter. Se pudo evidenciar los requerimientos que se necesitan para la instalación, al igual que como se debe hacer la instalación.

Por otro lado lo visto en el seminario dio a conocer los temas bases para un buen desarrollo con Flutter y permitió así poder desarrollar una aplicación móvil básica para poder seguir implementándola. Se logro avanzar la implementación de las vistas para el login y la vista principal con los ejercicios, se espera que sus próximas actualizaciones tenga un entorno mas favorable para el usuario, una mejor organización con los ejercicios que se muestran, se podrá ampliar la funcionalidad con notificaciones, integración de servicios en la nube.

Anexos

Anexo 1. Código de la página principal.

```

52 //Calendario gym
53 TableCalendar(
54   firstDay: DateTime.utc(2000, 1, 1),
55   lastDay: DateTime.utc(2100, 12, 31),
56   focusedDay: _focusedDay,
57   selectedDayPredicate: (day) => isSameDay(_selectedDay, day),
58   onDaySelected: (selectedDay, focusedDay) {
59     setState(() {
60       _selectedDay = selectedDay;
61       _focusedDay = focusedDay;
62     });
63   },
64   calendarStyle: const CalendarStyle(
65     todayDecoration: BoxDecoration(
66       color: Colors.green,
67       shape: BoxShape.circle,
68     ), // BoxDecoration
69     selectedDecoration: BoxDecoration(
70       color: Colors.black87,
71       shape: BoxShape.circle,
72     ), // BoxDecoration
73   ), // CalendarStyle
74 ), // TableCalendar
75
76 const SizedBox(height: 20),
77 const Divider(thickness: 1),
78 const SizedBox(height: 10),
79

```

Este es parte del código del calendario. Tiene dependencia en el yaml

```

dependencies:
  flutter:
    sdk: flutter

  table_calendar: ^3.0.9

```

```

86 // distribucion de las rutinas
87 ListTile(
88   leading: Image.asset("assets/img/pecho.png", width: 40),
89   title: const Text("PECHO"),
90   subtitle: const Text("6 Ejercicios"),
91   onTap: () {
92     Navigator.push(
93       context,
94       MaterialPageRoute(builder: (context) => const PechoScreen()),
95     );
96   }, // ListTile
97 ), // ListTile
98 ListTile(
99   leading: Image.asset("assets/img/pierna.png", width: 40),
100  title: const Text("PIERNA"),
101  subtitle: const Text("11 Ejercicios"),
102  onTap: () {
103    Navigator.push(
104      context,
105      MaterialPageRoute(builder: (context) => const PechoScreen()),
106    );
107  },

```

Anexo 2. Login.

```

20 Row(
21   children: [
22     IconButton(
23       onPressed: () {
24         Navigator.pop(context);
25       },
26       icon: const Icon(Icons.arrow_back),
27       color: Colors.black87,
28     ), // IconButton
29     const Expanded(
30       child: Text(
31         "RUTINAS GYM",
32         textAlign: TextAlign.center,
33         style: TextStyle(
34           fontSize: 24,
35           fontWeight: FontWeight.bold,
36           color: Colors.green,
37         ), // TextStyle
38       ), // Text
39     ), // Expanded
40     const SizedBox(width: 48),
41   ],
42 ), // Row

```

```

46  ✓      const Text(
47          "Crea una cuenta",
48          style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
49      ), // Text
50      const SizedBox(height: 10),
51
52  ✓      const Text(
53          "Ingresa tu correo electrónico\npara registrarte en esta aplicación",
54          textAlign: TextAlign.start,
55          style: TextStyle(fontSize: 14, color: Colors.black54),
56      ), // Text
57      const SizedBox(height: 20),
58
59      // Campo de texto
60  ✓      TextField(
61          controller: emailController,
62          keyboardType: TextInputType.emailAddress,
63  ✓      decoration: InputDecoration(
64          hintText: "correoelectrónico@dominio.com",
65  ✓      border: OutlineInputBorder(
66          borderRadius: BorderRadius.circular(8),
67          ), // OutlineInputBorder
68          ), // InputDecoration
69      ), // TextField
70      const SizedBox(height: 15),

```

```

96      // Boton Google
97  ✓      SizedBox(
98          width: double.infinity,
99  ✓      child: OutlinedButton.icon(
100  ✓      onPressed: () {
101          print("Continuar con Google");
102      },
103  ✓      icon: Image.network(
104          "https://img.icons8.com/color/48/google-logo.png",
105          height: 20,
106      ), // Image.network
107          label: const Text("Continuar con Google"),
108      ), // OutlinedButton.icon
109      ), // SizedBox
110
111      const SizedBox(height: 10),

```

Anexo 2. Página de los ejercicios.

```
7  Widget build(BuildContext context) {
8    return Scaffold(
9      appBar: AppBar(
10       title: const Text("Actividad"),
11       backgroundColor: Colors.white,
12       elevation: 0,
13       leading: IconButton(
14         icon: const Icon(Icons.arrow_back, color: Colors.black),
15         onPressed: () {
16           Navigator.pop(context); // regresa a la pagina principal
17         },
18       ), // IconButton
19     ), // AppBar
20
```

```
21     body: SingleChildScrollView(
22       padding: const EdgeInsets.all(12.0),
23       child: Column(
24         crossAxisAlignment: CrossAxisAlignment.start,
25         children: [
26           Container(
27             decoration: BoxDecoration(
28               borderRadius: BorderRadius.circular(12),
29               image: const DecorationImage(
30                 image: AssetImage("assets/img/pecho_1.png"),
31                 fit: BoxFit.cover,
32               ), // DecorationImage
33             ), // BoxDecoration
```

```

41 padding: const EdgeInsets.all(16),
42 child: Column(
43   crossAxisAlignment: CrossAxisAlignment.start,
44   children: const [
45     Text(
46       "Ejercicios Pecho",
47       style: TextStyle(
48         color: Colors.green,
49         fontSize: 18,
50         fontWeight: FontWeight.bold,
51       ), // TextStyle
52     ), // Text
53     Spacer(),
54     Text(
55       "6 Ejercicios",
56       style: TextStyle(color: Colors.white, fontSize: 14),
57     ), // Text
58     SizedBox(height: 4),
59     Text(
60       "20 min TIEMPO",
61       style: TextStyle(color: Colors.white, fontSize: 14),
62     ), // Text

```

```

122   SizedBox(
123     width: double.infinity,
124     child: ElevatedButton(
125       style: ElevatedButton.styleFrom(
126         backgroundColor: Colors.green,
127         shape: RoundedRectangleBorder(
128           borderRadius: BorderRadius.circular(8),
129         ), // RoundedRectangleBorder
130         padding: const EdgeInsets.symmetric(vertical: 14),
131       ),
132       onPressed: () {
133         Navigator.pop(context); // boton de inicio del ejercicio
134       },
135       child: const Text(
136         "INICIO",
137         style: TextStyle(color: Colors.white, fontSize: 16),
138       ), // Text
139     ), // ElevatedButton
140   ), // SizedBox

```

Referencias

- Amazon Web Services (AWS). (2025). *¿Que es Flutter?* Obtenido de <https://aws.amazon.com/es/what-is/flutter/>
- Google. (2025). *Develop with Android Studio*. Obtenido de <https://developer.android.com/studio/intro?hl=es-419>
- Google. (2025). *Flutter*. Obtenido de <https://flutter.dev/>
- (s.f.). Obtenido de <https://docs.flutter.dev/>
- Rodríguez, V. V. (2019). *Desarrollo de aplicaciones móviles multiplataforma con Flutter*. Obtenido de https://repositorio.ual.es/bitstream/handle/10835/8010/TFG_VAZQUEZ%20RODRIGUEZ,%20VICTOR.pdf?sequence=1
- Zúñiga, F. G. (03 de 02 de 2025). *arsys*. Obtenido de <https://www.arsys.es/blog/que-es-un-framework-en-programacion-y-para-que-sirve#:~:text=Un%20framework%2C%20traducido%20habitualmente%20al,m%C3%A1s%20organizada%2C%20robusta%20y%20escalable.>