



**PLATAFORMA DE GESTIÓN PARA GIMNASIO CON REGISTRO DE
SUSCRIPCIONES, EN EL CANTON PINDAL DE LA PROVINCIA DE
LOJA EN EL PERIODO COMPRENDIDO DE 2022 -2023**

AUTOR: ALVARO ALEJANDRO ENRIQUEZ V.

DIRECTOR: LIC. ANDREA M. TAMBO MINGA

**INFORME DE TESIS PREVIO A LA
OBTENCIÓN DEL TÍTULO DE
TECNÓLOGO EN ANÁLISIS DE
SISTEMAS**

LOJA – ECUADOR

2022- 2023

CERTIFICACIÓN

CERTIFICA:

Que el presente trabajo de investigación, previo a obtener el título de Tecnólogo en ANÁLISIS DE SISTEMAS, cuyo tema es: **“PLATAFORMA DE GESTIÓN PARA GIMNASIO CON REGISTRO DE SUSCRIPCIONES, EN EL CANTON PINDAL DE LA PROVINCIA DE LOJA EN EL PERIODO COMPRENDIDO DE 2022 -2023”**, ha sido realizado bajo conducción, control y supervisión, por lo que autorizo su presentación, sustentación y defensa.

Loja, 27 septiembre de 2023

LIC. ANDREA MONSERRATH TAMBO MINGA
DIRECTOR

AUTORÍA

Yo, **ALVARO ALEJANDRO ENRIQUEZ VINCES**, con numero de cedula. Nro.**1104870173**, declaro ser el autor del presente trabajo de tesis y eximo expresamente al **INSTITUTO TECNOLOGICO SUPERIOR “LOS ANDES”** y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo al **INSTITUTO TECNOLÓGICO SUPERIOR “LOS ANDES”**, la publicación de mi tesis en el Repositorio – Biblioteca Virtual.



ALVARO ALEJANDRO ENRIQUEZ VINCES

AUTOR

C.I: 1104870173

**CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL
AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL
O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO
COMPLETO.**

Yo, **ALVARO ALEJANDRO ENRIQUEZ VINCES**, declaro ser el autor de la tesis titulada:
**“PLATAFORMA DE GESTIÓN PARA GIMNASIO CON REGISTRO DE
SUSCRIPCIONES”**. Como requisito para optar por el grado de:

TECNÓLOGO EN ANÁLISIS DE SISTEMAS, autorizo al Instituto Superior Tecnológico
Los “Andes”, para que, con fines académicos, muestre al mundo la producción intelectual del
Instituto Superior.

Para constancia de esta autorización, en la ciudad de Loja a los 27 días del mes Septiembre del
año 2023.



ALVARO ALEJANDRO ENRIQUEZ VINCES

AUTOR

C.I. 1104870173

AGRADECIMIENTO

El trabajo de investigación va dedicado principalmente a Dios, por darme la fortaleza y capacidad para seguir adelante en esta etapa de mi vida.

A mis padres por el apoyo brindado, a mi esposa e hija que han sido mi motivación principal a lo largo de esta carrera.

A mis docentes por sus sabias enseñanzas, consejos y amistad durante mi etapa profesional.

A las autoridades de la institución en especial a la Mgs. Mariel Minga por abrirme las puertas de tan prestigiosa institución y darme la oportunidad de cumplir una meta más en mi vida.

Infinitas gracias a todos quienes aportaron directa e indirectamente para alcanzar este logro dentro de mi vida profesional.



ALVARO ALEJANDRO ENRIQUEZ VINES

AUTOR

DEDICATORIA

Dedico este trabajo de investigación a nuestro padre creador: Dios.

Quien nunca permitió que decayera en momentos de debilidad, por mantenerme siempre firme y luchando para alcanzar este tan anhelado sueño.

A mis padres quienes han sido parte fundamental en este proceso de mi vida profesional.

A mi esposa e hija por su cariño apoyo y comprensión, a mi familia quienes de una u otra manera han sido parte de este logro importante en mi vida.

A mis compañeros que a pesar de no haber compartido mucho tiempo el aula en físico, en los últimos tiempos nos convertimos en una pequeña familia virtual.

**GRACIAS A TODOS POR COMPARTIR ESTOS BELLOS MOMENTOS EN MI
VIDA PROFESIONAL.**

ALVARO ALEJANDRO ENRIQUEZ VINCES

AUTOR

INDICE

CERTIFICACIÓN	ii
AUTORÍA.....	iii
CARTA DE AUTORIZACIÓN DE TESIS.....	iv
DEDICATORIA.....	vi
INDICE	vii
Resumen	ix
Abstract	2
Introducción	3
Objetivos	4
Objetivo general	4
Objetivos Específicos:	4
CAPITULO I : Metodología	5
Método Histórico:.....	5
Método Descriptivo:.....	5
Técnicas de investigación	6
Entrevista	6
Revisión de la Literatura	7
Django.....	7
Términos y definiciones.....	8
Análisis de la situación actual	10
Tipos de usuarios	11
Limitaciones.....	12
Identificación de Requisitos Específicos	12
Autenticación de Usuarios	13
Registro de Usuarios	13
Cierre de Sesión	13
Consulta de Actividades	13
Consulta de Productos en la Tienda.....	13
Adición de Productos a la Tienda	13
Modificación de Productos en la Tienda	13
Eliminación de Productos en la Tienda	14
Generación de Notificaciones	14

Interfases de usuario	14
Interfaz de Software	15
Interface de hardware.....	15
Interfaz de Programación de Aplicaciones	16
CAPITULO II : Desarrollo	17
Requerimientos funcionales.....	17
Atributos del sistema	26
Otros requerimientos.....	26
Plan de desarrollo	27
Metodología de desarrollo de software XP.....	28
Planificación.....	29
Plan de Iteración para el Desarrollo de la Aplicación de Gimnasio.....	30
Diseño.....	34
Diagrama de clases	34
Diagrama de casos	35
Casos de Uso.....	36
Diagramas de Flujo	38
Codificación	42
Estructura.....	42
Modelo	43
Vista	44
Controlador	44
Entorno de desarrollo.....	46
Python 3.10:	47
Django:.....	48
Proyecto de Django.....	48
<i>Creación del Proyecto:</i>	48
Estructura del proyecto	50
Desarrollo de Componentes Esenciales:	52
Estilos.....	60
Pruebas de desarrollo	62
Pruebas de despliegue	63
DISCUSIÓN.....	68
Hallazgo 1.....	69
Hallazgo 2.....	69

Hallazgo 3.....	70
Hallazgo 4:.....	71
CONCLUSIONES	72
RECOMENDACIONES	74
REFERENCIAS BIBLIOGRÁFICAS	75
ANEXOS.....	79
ANEXO N° 1.....	79
ANEXO N° 2.....	80
Ficha informativa de la Empresa	80
ANEXO N° 3.....	81
Entrevista de involucrados.....	81
ANEXO N° 4.....	82
ANEXO N° 5.....	83
Imágenes	83
ANEXO N° 6.....	85
ANEXO N° 7.....	88

INDICE DE TABLAS

Tabla 1. Autenticación de usuarios.....	17
Tabla 2.Registro de suscriptores.....	18
Tabla 3. Cierre de sesión	19
Tabla 4. Visualización de actividades.....	20
Tabla 5. Visualización de Productos.....	21
Tabla 6. Creación de productos	22
Tabla 7. Modificación de productos	25
Tabla 8. Eliminación de Productos.....	26
Tabla 9. Notificaciones.....	28
Tabla 10. Iteración 1	31
Tabla 11. Iteración 2	31
Tabla 12. Iteración 3	32
Tabla 13. Iteración 4.....	32
Tabla 14. Iteración 5.....	33
Tabla 15. Iteración 6.....	33

INDICE DE FIGURAS

Figura 1. Diagrama de Clases.	34
Figura 2. Diagrama de Casos.	36
Figura 3. Diagrama de Casos.	36
Figura 5. Caso de uso.	37
Figura 6. Diagrama de flujo 1.	38
Figura 7. Diagrama de flujo 2.	39
Figura 8. Diagrama de flujo 3.	39
Figura 9. Diagrama de flujo 4.	40
Figura 10. Diagrama de flujo 5.	40
Figura 11. Diagrama de flujo 6.	41
Figura 12. Estructura de Django.	51
Figura 13. Creación de modelo en Django.	52
Figura 14. Creación de formulario de Django.....	53
Figura 15. Plantillas Bootstrap.	54
Figura 16. Modelo miembro.....	55
Figura 17. Modelo producto.....	56
Figura 18. Administración de Django.	57
Figura 19. Gestionar a través de Administración de Django.....	58
Figura 20. Interfaz de suscripciones.....	59
Figura 21. Aplicaciones instaladas de proyecto Django.	61
Figura 22. Prueba de Autenticación de usuarios.	62
Figura 23. Prueba de creación de productos.	62
Figura 24. Prueba de despliegue.	63
Figura 25. Prueba de Despliegue.	64
Figura 26. Prueba de Despliegue.	65
Figura 27. Prueba de Despliegue.	65
Figura 28. Prueba de Despliegue.	66
Figura 29. Prueba de Despliegue.	67
Figura 30. Prueba de Despliegue.....	67
Figura 31. Anexo.	80
Figura 32. Anexo.....	84
Figura 33. Anexo.....	84

Resumen

Este proyecto tiene como finalidad el desarrollo de una aplicación web utilizando el framework Django, con el propósito de mejorar la administración y la consulta de información en un gimnasio. La aplicación ofrece a los usuarios la capacidad de registrarse, autenticarse y acceder a características como la visualización de servicios disponibles en el gimnasio y la exploración de productos en su tienda.

La plataforma proporciona una interfaz de usuario intuitiva que permite a los usuarios registrarse y autenticarse para acceder a sus perfiles personalizados. Una vez autenticados, los usuarios pueden explorar los diferentes servicios ofrecidos por el gimnasio, como clases y entrenamientos especializados. Además, tienen la oportunidad de navegar por los productos disponibles en la tienda del gimnasio, incluyendo accesorios deportivos y productos relacionados con la salud y el bienestar.

La aplicación también cuenta con un panel de administración que facilita al personal del gimnasio la gestión eficiente de los servicios y productos. Este panel permite agregar, modificar y eliminar servicios y productos, así como administrar las cuentas de usuario de manera efectiva.

Utilizando tecnología Django, esta aplicación web ofrece una experiencia fluida y amigable para los usuarios, al tiempo que brinda al gimnasio una herramienta poderosa para administrar su oferta de servicios y productos de manera efectiva. El enfoque central del proyecto es crear una solución que simplifique tanto la experiencia del usuario como la gestión interna del gimnasio.

Abstract

This project aims to develop a web application using the Django framework with the purpose of enhancing the administration and information retrieval in a gym. The application provides users with the ability to register, authenticate, and access features such as viewing available services at the gym and exploring products in its store.

The platform offers an intuitive user interface that allows users to register and authenticate to access their personalized profiles. Once authenticated, users can explore the different services offered by the gym, such as classes and specialized workouts. Additionally, they have the opportunity to browse products available in the gym's store, including sports accessories and health-related products.

The application also includes an administration panel that facilitates efficient management of services and products for gym staff. This panel allows for adding, modifying, and deleting services and products, as well as effectively managing user accounts.

Using Django technology, this web application offers a smooth and user-friendly experience while providing the gym with a powerful tool to effectively manage its service and product offerings. The central focus of the project is to create a solution that simplifies both the user experience and internal gym management.

Introducción

En este apartado, exploraremos los objetivos y las motivaciones que han dado origen al desarrollo de este proyecto como parte del Trabajo de Fin de Grado.

La génesis de nuestra aplicación web surge de la imperante necesidad de gestionar de manera eficiente tanto las actividades como los productos en venta en el contexto de un centro deportivo. Esta aplicación está diseñada para servir como un medio eficaz para notificar a los usuarios sobre información relevante y resolver sus consultas de manera oportuna y efectiva.

La motivación personal que impulsa la creación de esta aplicación web proviene de una petición directa por parte de los dueños del centro deportivo mencionado. Además, la decisión de utilizar el entorno de programación Django como plataforma principal para el desarrollo fue influenciada por mi experiencia en una empresa donde se exploraba la implementación de este entorno. Estos factores determinaron tanto el enfoque temático del proyecto como la elección del marco de programación.

En las próximas secciones, profundizaremos en cómo nuestra aplicación aborda estas motivaciones y cómo contribuye a satisfacer las necesidades específicas identificadas en el contexto del centro deportivo.

Objetivos

Objetivo general

Desarrollar e implementar una aplicación que facilite la administración del centro deportivo **GYM DOMAIN**, la gestión integral de suscriptores y productos, creación, modificación, eliminación y consulta de actividades. En el periodo comprendido en el año 2023.

Objetivos Específicos:

- Llevar a cabo un exhaustivo análisis y diseño de software con el propósito de desarrollar una aplicación robusta, eficiente y altamente funcional que satisfaga las necesidades específicas de nuestros usuarios finales.
- Definir los requisitos funcionales y no funcionales de la aplicación, incluyendo la capacidad de agregar, modificar o eliminar entradas para la gestión eficiente de suscriptores y productos del centro deportivo.
- Realizar un diseño de interfaz centrado en el usuario que facilite la difusión de información sobre actividades y productos, asegurando que los usuarios puedan conocer fácilmente las ofertas actuales del centro deportivo.
- Implementar un sistema de notificaciones que permita informar a los usuarios registrados sobre novedades, actualizaciones y datos relevantes relacionados con el centro deportivo.

CAPITULO I : Metodología

Método Histórico:

Dentro del enfoque del método histórico, se procedió a examinar la evolución de la gestión de actividades y productos en centros deportivos a lo largo del tiempo. A través de una investigación retrospectiva, se analizaron sistemas y métodos previos empleados para la administración de servicios y productos en el gimnasio en cuestión. Se identificaron desafíos históricos y soluciones implementadas en el pasado para satisfacer las necesidades de gestión. El desarrollo de la aplicación web con Django se consideró como una respuesta a las demandas históricas de optimización en la gestión de actividades y productos. Este enfoque permitió situar el proyecto en el contexto de una progresión histórica y resaltar su contribución como una solución innovadora.

Método Descriptivo:

Desde la perspectiva del método descriptivo, se proporcionó una descripción minuciosa y objetiva de la aplicación web desarrollada para la gestión y consulta en el centro deportivo. Se detallaron exhaustivamente las características de la aplicación, destacando cómo se realiza la gestión de actividades y productos, y cómo los usuarios interactúan con la plataforma. Se presentó en detalle la interfaz de usuario, se describieron las funcionalidades clave y se delinearon los procesos dentro de la aplicación. Se llevó a cabo un análisis detallado de cómo la aplicación cumple con los objetivos establecidos, se describió su integración con los flujos de trabajo del centro deportivo y se resaltaron los beneficios para el propietario y usuarios. Este enfoque proporcionó una visión completa y comprensible del desarrollo y la implementación exitosa de la solución en cuestión.

Técnicas de investigación

Entrevista

La técnica de entrevista es una herramienta esencial en el proceso de definición de requerimientos de un proyecto de desarrollo de software. En el contexto de tu proyecto de aplicación para un centro deportivo, la técnica de entrevista se utilizó para recopilar información valiosa de diversas partes interesadas, como propietario del centro deportivo, administradores, empleados y posiblemente usuarios finales, con el fin de comprender sus necesidades, expectativas y requisitos específicos.

Revisión de la Literatura

En el contexto del presente proyecto, se lleva a cabo una revisión literaria que explora y analiza el papel esencial del entorno de programación Django en el desarrollo de una aplicación web destinada a la gestión y consulta en un centro deportivo. Esta revisión tiene como objetivo principal ofrecer una comprensión profunda y detallada de las características y ventajas que Django aporta a la creación de soluciones tecnológicas en el ámbito deportivo. (*Estrategias Para El Desarrollo de Videojuegos Libres Para La Educación Formal de Colombia*, n.d.)

Django, un marco de programación en Python especialmente diseñado para aplicaciones web, ha ganado reconocimiento por su simplicidad de instalación, documentación exhaustiva y la incorporación de un Sistema Gestor de Bases de Datos. A través de su enfoque en modelos y la independencia de los Gestores de Bases de Datos, Django proporciona una estructura versátil que facilita la administración de información en un entorno deportivo. Su compatibilidad con Python 3 garantiza la actualidad y longevidad de las soluciones creadas.

Django

Django es un framework de programación en Python especialmente diseñado para construir aplicaciones web. Su proceso de instalación es sencillo y cuenta con una documentación detallada y completa. Integra un Sistema Gestor de Bases de Datos (sqlite3), lo que simplifica la configuración inicial. La estructura basada en modelos asegura una separación efectiva entre la base de datos y la aplicación, lo que proporciona flexibilidad. Su compatibilidad con Python 3 garantiza que las librerías y métodos se mantengan actualizados, evitando la obsolescencia. (Cristina & Buriticá, 2015)

Django permite importar Bootstrap para el diseño de la interfaz de usuario y admite el uso de andamiaje (scaffolding) para generar código a partir de plantillas predefinidas. Ejemplos notables de aplicaciones construidas con Django incluyen Instagram, Pinterest y National Geographic.

La especificación de requisitos del proyecto tiene como objetivo mejorar la comunicación entre el cliente y los usuarios, permitiendo consultas y mostrando actualizaciones de actividades y productos en el centro deportivo. La aplicación propuesta busca optimizar la gestión y consulta de información, con funciones esenciales como crear, modificar y eliminar actividades y productos.

Términos y definiciones

Framework: Marco de trabajo utilizado por desarrolladores para construir y desarrollar software de manera más eficiente. Utilizar un framework beneficia al evitar la repetición de código, lo que a su vez contribuye a mantener la consistencia del código.(Merangin et al., 2018)

Base de datos: Conjunto organizado de datos relacionados que se almacenan en la memoria de un ordenador para su posterior uso. Estos datos se organizan en campos y registros, donde los campos representan atributos y los registros contienen información específica de cada atributo.

Almacenamiento: Capacidad de un sistema informático para retener información utilizando diferentes tipos de dispositivos de almacenamiento.(León Soberón, 2020)

Sistema Gestor de Bases de Datos (DBMS): Conjunto de programas que permiten la extracción, modificación y almacenamiento de información en una base de datos. Además, los DBMS incluyen herramientas que posibilitan el análisis de los datos contenidos en la base.

Interfaz: Medio de conexión que posibilita a los usuarios comunicarse con un sistema, permitiendo la interacción entre el usuario y la tecnología.

Internet: Red global de comunicación que interconecta múltiples ordenadores y dispositivos, permitiendo la conexión y comunicación entre ellos a través de diferentes protocolos y servicios.

Inicio de sesión (Login): Proceso de autenticación mediante el cual un usuario accede a un sistema después de proporcionar las credenciales adecuadas.(López Herrera, 2016)

Acrónimos:

URL (Uniform Resource Locator): Dirección única y específica asignada a cada página web, facilitando la ubicación y acceso a la misma por parte de los usuarios.

Análisis de la situación actual

Dentro del marco de este proyecto, se ha proyectado el desarrollo de un sistema integral dividido en distintos módulos, cada uno con funcionalidades específicas dentro de la aplicación. Estos módulos engloban la gestión de usuarios, actividades, tienda y contacto, diseñados para abordar necesidades concretas de la empresa.

El propósito fundamental del producto a desarrollar es satisfacer diversas necesidades identificadas en la empresa:

Gestión Integral de Actividades y Productos: La aplicación tiene como objetivo centralizar y simplificar la gestión de actividades y productos ofrecidos por el centro deportivo. A través de la pestaña de actividades, los usuarios podrán acceder a información detallada sobre cada actividad, incluyendo horarios y fechas de realización. De igual forma, la pestaña de tienda permitirá a los usuarios explorar los productos disponibles, presentando imágenes y precios. Esta funcionalidad se concibe para mejorar la accesibilidad y visibilidad de la oferta del centro.

Interacción y Comunicación Específica: La implementación de un módulo de administración permitirá a los usuarios con el rol de administrador gestionar de manera eficiente todos los aspectos de la aplicación. Los administradores serán responsables de ingresar nuevos usuarios al gimnasio, registrar su participación en actividades, definir la duración y tarifas de sus membresías, así como gestionar los pagos correspondientes. Esta funcionalidad garantiza un control preciso y facilita la administración integral del gimnasio.

Acceso Simplificado para los Usuarios: La experiencia de los usuarios se simplifica considerablemente al no requerir registro. Los usuarios no administradores podrán acceder directamente a una página pública donde encontrarán información detallada sobre todas las actividades y productos ofrecidos por el gimnasio. Esto brinda un acceso transparente y cómodo a las ofertas disponibles, sin la necesidad de un proceso de registro adicional.

Roles y Privilegios Diferenciados: Se establecerá una jerarquía de roles en la aplicación, otorgando a los administradores un conjunto completo de privilegios para gestionar y supervisar el gimnasio. Por otro lado, los usuarios no administradores podrán explorar la variedad de actividades y productos sin necesidad de registro. Esta diferenciación de roles asegura un manejo efectivo de la plataforma y garantiza que cada usuario tenga acceso solo a las funciones relevantes para su interacción.

Gestión Integral de Actividades y Productos: El sistema de administración brindará a los usuarios con rol de administrador las herramientas necesarias para gestionar eficientemente todas las actividades y productos del gimnasio. Desde el panel de administración, los usuarios autorizados podrán agregar nuevas actividades, actualizar horarios, definir tarifas y plazos de pago, y supervisar la asistencia de los usuarios. Esta funcionalidad facilita la administración de la oferta del gimnasio y mejora la interacción con los miembros. (Pizarro Castro, 2014)

En resumen, la implementación de este sistema con roles diferenciados permite una gestión completa por parte de los administradores, mientras que los usuarios no administradores disfrutan de un acceso sencillo y directo a la información relevante. Esta estructura optimiza la experiencia de los usuarios y garantiza una administración eficaz de todas las actividades y productos ofrecidos por el gimnasio.

Tipos de usuarios

La base de usuarios de esta aplicación debería ser categorizada en 2 grupos distintos:

El primer grupo está conformado por usuarios ocasionales de la aplicación, es decir, personas que la utilizan para buscar información o explorar los productos disponibles en la tienda. Para pertenecer a este grupo, no se requiere un conocimiento profundo, solo nociones básicas sobre el funcionamiento general de aplicaciones web. (Unzurrunzaga, 2013)

El segundo grupo de usuarios está formado por el personal del gimnasio, específicamente aquellos con permisos de administrador. Esto les habilita a acceder a un conjunto adicional de funcionalidades, como agregar o eliminar actividades del centro y gestionar los productos de la tienda. Si bien no es necesario un conocimiento exhaustivo, estos usuarios deben poseer una comprensión básica del uso de aplicaciones web y, además, tener familiaridad con el funcionamiento y la administración del centro. Esto se debe a que una de sus responsabilidades incluye responder a preguntas y dudas planteadas por los usuarios, y para hacerlo de manera precisa, necesitan estar informados sobre el funcionamiento del centro.

Limitaciones

El desarrollo de este proyecto se llevará a cabo en Django, un marco de trabajo de código abierto diseñado para la creación de aplicaciones web utilizando el lenguaje de programación Python. En esta iniciativa, se empleará Python, un lenguaje orientado a objetos, para construir la capa lógica de la aplicación. Además, se utilizará HTML para establecer la estructura y el diseño de la aplicación web, y CSS se empleará para la apariencia gráfica.

A lo largo del proceso de desarrollo, se llevarán a cabo reuniones con el cliente para recibir sus aportes y sugerencias, especialmente mientras la aplicación aún se encuentre en su fase de desarrollo inicial. Esto permitirá evitar cambios estructurales significativos en etapas posteriores del proyecto, cuando la aplicación ya esté más avanzada.

El Sistema Gestor de Bases de Datos (SGBD) seleccionado para este proyecto es SQLite3, dado que es el gestor predeterminado utilizado por Django. Se integrará también Bootstrap, una biblioteca que desempeñará un papel fundamental en la definición del diseño de la aplicación.(Mahapsari, 2013)

Identificación de Requisitos Específicos

A continuación, se presentan los requisitos específicos del proyecto:

Autenticación de Usuarios

R001: Implementar un sistema de autenticación para permitir que los usuarios administradores inicien sesión en la aplicación.

Registro de Usuarios

R002: Facilitar un proceso de registro para que los suscriptores puedan mantener control de los procesos

Cierre de Sesión

R003: Proporcionar a los usuarios autenticados la opción de cerrar sesión cuando sea necesario.

Consulta de Actividades

R004: Habilitar la funcionalidad que permita a los usuarios revisar y obtener información sobre las distintas actividades ofrecidas.

Consulta de Productos en la Tienda

R005: Proporcionar a los usuarios la capacidad de explorar y obtener detalles acerca de los productos disponibles en la tienda.

Adición de Productos a la Tienda

R006: Habilitar la opción para que usuarios con los permisos apropiados puedan incorporar nuevos productos al catálogo de la tienda.

Modificación de Productos en la Tienda

R007: Permitir a usuarios con los permisos necesarios actualizar la información de los productos disponibles en la tienda.

Eliminación de Productos en la Tienda

R008: Proporcionar la capacidad para que usuarios autorizados puedan eliminar productos del inventario de la tienda.

Generación de Notificaciones

R009: Generar notificaciones para el usuario administrador, con el objetivo de garantizar un control de suscripciones.

Interfaces de usuario

Cuando se trata de las interfaces de usuario, nos referimos a las ventanas a través de las cuales los usuarios interactúan para llevar a cabo las diversas funcionalidades proporcionadas por la aplicación. En nuestro proyecto, los usuarios se relacionarán con la aplicación exclusivamente utilizando el teclado y el ratón. Las interfaces que facilitarán la interacción del usuario final con la aplicación se componen principalmente de los siguientes elementos:

Formularios: Utilizaremos formularios para agregar y eliminar tanto actividades como productos. Estos formularios ofrecerán una estructura organizada que permitirá a los usuarios ingresar y gestionar datos de manera efectiva.

Botones: Se implementarán botones como elementos gráficos que permitirán a los usuarios llevar a cabo acciones específicas y funciones dentro de la aplicación.

Mensajes Guía: La aplicación mostrará mensajes en momentos clave para orientar a los usuarios en el uso correcto de las diversas funciones disponibles. Estos mensajes brindarán instrucciones claras y concisas para garantizar una experiencia fluida.

Cuadros de Diálogo: Los cuadros de diálogo entrarán en juego al establecer una comunicación directa entre el usuario y la aplicación. Proporcionarán información esencial o solicitarán confirmación antes de ejecutar acciones importantes. (Severo Malaspina & Tonarelli Alvarez, 2021)

Interfaz de Software

Las interfaces de software tienen la función de proporcionar a los usuarios información sobre los procesos en curso o las herramientas de control en ejecución. Estos elementos son visualizables en la pantalla del ordenador y permiten a los usuarios obtener datos sobre las operaciones en marcha.

En nuestra aplicación, no se incluyen elementos que informen a los usuarios sobre el estado de los procesos. Esto se debe a que las funcionalidades de la aplicación no son computacionalmente intensivas ni requieren un tiempo prolongado para ejecutarse, lo que evita la necesidad de proporcionar información sobre su estado.

Interface de hardware

En términos de interfaces hardware, hacemos referencia al conjunto de dispositivos que permiten a los usuarios interactuar con la aplicación y comunicarse con la computadora. En el contexto de nuestra aplicación, se requerirá el uso de diversas interfaces hardware para facilitar esta interacción (Vásquez Rudas, 2014):

La pantalla del monitor se utilizará para mostrar la información relevante al usuario, brindando una visualización clara y accesible de los datos generados por la aplicación.

El ratón desempeñará un papel importante al registrar los movimientos y clics del usuario. Estas acciones se traducirán en interacciones visuales en la pantalla, permitiendo al usuario seleccionar opciones, interactuar con botones y cuadros de diálogo, y navegar por la aplicación de manera intuitiva.

El teclado, por su parte, servirá como medio de entrada de datos. Los usuarios podrán introducir información, realizar búsquedas y realizar consultas mediante las pulsaciones en el teclado, lo que ampliará la capacidad de interacción y personalización de la experiencia.

En conjunto, estas interfaces hardware crearán un entorno propicio para la comunicación fluida entre el usuario y la aplicación. Al aprovechar las capacidades de la pantalla, el ratón y el teclado, los usuarios podrán interactuar de manera efectiva con la aplicación, contribuyendo a una experiencia de usuario enriquecedora y satisfactoria.

Interfaz de Programación de Aplicaciones

En cuanto a las interfaces de comunicación, nos referimos a los componentes que posibilitan la transmisión de información desde un equipo de datos a un módem u otro dispositivo que convierte señales. Este proceso facilita la comunicación entre computadoras a través de una línea o mediante el medio seleccionado para transmitir datos a otro equipo.

En el contexto de nuestra aplicación, las interfaces de comunicación desempeñan un papel fundamental en la conexión entre la base de datos y la aplicación desarrollada en Python. En este proceso, la función de la interfaz de comunicación es ejecutada por el Sistema de Gestión de Bases de Datos predeterminado que hemos seleccionado: SQLite3.

Esta interfaz de comunicación permite que la aplicación acceda y maneje eficazmente los datos almacenados en la base de datos. A través del uso del SQLite3, la aplicación puede realizar operaciones de lectura, escritura y modificación en la base de datos, lo que garantiza una interacción fluida y segura entre los datos y la funcionalidad de la aplicación.

La elección de una interfaz de comunicación eficaz y confiable es esencial para el rendimiento óptimo de la aplicación y la integridad de los datos. Al utilizar el Sistema de Gestión de Bases de Datos SQLite3 como interfaz de comunicación, estamos asegurando una conexión sólida entre los componentes de la aplicación, lo que contribuirá al éxito y la eficiencia general de nuestro proyecto.(Crespo, 2014)

CAPITULO II : Desarrollo

Requerimientos funcionales

Tabla 1. Autenticación de usuarios

R001: Autenticación de Usuarios
Tipo: Necesario
Estado: En Producción
Prioridad: Alta
Documentos: Ver [3]
Descripción: Permitir a los usuarios autenticarse en la aplicación mediante un nombre de usuario y contraseña para acceder a funcionalidades específicas destinadas a usuarios registrados o con permisos de administrador. Los usuarios deberán rellenar los siguientes campos de entrada:
- Nombre de Usuario (username)
- Contraseña (password)
Entradas: Nombre de usuario, Contraseña
Salidas: Acceso a la página principal con un mensaje de bienvenida que incluye el nombre de usuario (o "admin" si es un usuario administrador).
Precondición: El usuario accede a la ventana de autenticación.
Postcondición: El usuario autenticado es redirigido a la página principal y se muestra un mensaje de bienvenida.
Manejo de Situaciones Anormales:
- Si algún campo de entrada no está relleno, el sistema informará al usuario que ambos campos deben ser completados.
- Si el nombre de usuario o contraseña ingresados no coinciden con registros en la base de datos, se notificará al usuario que las credenciales son incorrectas.
Criterios de Aceptación:
- Ambos campos de entrada deben estar completados para realizar un acceso exitoso.
- Se debe proporcionar una retroalimentación clara en caso de campos no completados o credenciales incorrectas.
Destinatarios: Usuarios con permisos de administrador (empleados del gimnasio).
<i>Autor: Alvaro Enríquez</i>
<i>Fuente: Del sistema aplicado</i>

Tabla 2.Registro de suscriptores

R002: Registro de nuevos suscriptores
Tipo: Necesario
Estado: En Producción
Prioridad: Alta
Documentos:
<p>Descripción: Permitir el registro de nuevos usuarios en la aplicación por parte del administrador. Los usuarios interesados en suscribirse sean parte de un control adecuado para gestión del administrador. Los campos de entrada que los usuarios deberán completar para el registro incluyen:</p> <ul style="list-style-type: none"> - Cédula - Nombres - Apellidos - Fecha de nacimiento - Número de celular - Correo electrónico - Dirección - Historial médico - Periodo de suscripción - Cantidad del pago - Estado de la transacción - Fecha de registro - Horario
<p>Entradas: Cédula, Nombres, Apellidos, Fecha de nacimiento, Número de celular, Correo electrónico, dirección, Historial médico, Periodo de suscripción, Cantidad de pago, Estado de transacción , Fecha de registro , Horario.</p>
<p>Salidas: Acceso a la ventana de inicio de la aplicación, o mensaje de error en caso de fallo en el registro.</p>
<p>Precondición: El usuario administrador accede a la ventana de registro.</p>
<p>Postcondición: Si el registro es exitoso, el nuevo usuario será redirigido a la ventana de inicio de la aplicación.</p>
<p>Manejo de Situaciones Anormales:</p> <ul style="list-style-type: none"> - Notificación de error si se ingresan caracteres incorrectos según las directrices proporcionadas. - Información al usuario si no se completan todos los campos requeridos para el registro. - Alerta al usuario si los datos ingresados no cumplen con las condiciones, formatos y necesidades para el registro
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> - El registro se completará exitosamente solo si se llenan todos los campos requeridos y se comprueban las condiciones de registro. - Se proporcionará orientación sobre los requisitos de los campos durante el proceso de registro.

Destinatarios: Usuarios administradores y nuevos trabajadores del centro que necesitan registros con permisos de administrador.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Tabla 3. Cierre de sesión

R003: Cierre de Sesión de Usuarios Registrados

Tipo: Funcional

Estado: En Producción

Prioridad: Alta

Documentos:

Descripción: Permitir a los usuarios registrados cerrar sesión en sus cuentas autenticadas. Los usuarios podrán cerrar sesión haciendo clic en un botón designado para esta funcionalidad. El botón estará disponible una vez que el usuario se haya autenticado. No se requiere rellenar campos de entrada, solo pulsar el botón de cierre de sesión.

Entradas: Ninguna

Salidas: Cierre exitoso de sesión

Precondición: El usuario ha iniciado sesión en la aplicación.

Postcondición: El usuario cierra sesión y es redirigido a la página de inicio o pantalla de autenticación.

Manejo de Situaciones Anormales:

- Ninguna

Criterios de Aceptación:

-
- Los usuarios autenticados pueden cerrar sesión de manera sencilla y sin problemas.
 - El botón de cierre de sesión aparece de manera adecuada después de la autenticación.

Destinatarios: Usuarios registrados y usuarios con permisos de administrador que deseen cerrar sesión en sus cuentas autenticadas.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Tabla 4. Visualización de actividades

R004: Visualización de Actividades
Tipo: Funcional
Estado: En Producción
Prioridad: Alta
Documentos:
Descripción: Permitir a los usuarios consultar las actividades actuales ofrecidas por el centro. Los usuarios podrán acceder a la ventana de consulta de actividades sin dificultades. No se requiere rellenar campos de entrada, solo seleccionar el botón correspondiente para acceder a la ventana de consulta.
Entradas: Ninguna
Salidas: Ventana de consulta de actividades
Precondición: El usuario accede a la aplicación.
Postcondición: El usuario accede a la ventana de consulta de actividades.
Manejo de Situaciones Anormales:
- Ninguna
Criterios de Aceptación:
- El acceso a la ventana de consulta de actividades debe ser fluido y sin problemas.
- Los usuarios pueden acceder a la ventana sin necesidad de rellenar campos de entrada.
Destinatarios: Todos los usuarios, incluyendo usuarios casuales y usuarios registrados con permisos de administrador, que deseen consultar las actividades de la aplicación.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Tabla 5. Visualización de Productos

R005: Visualización de Productos en la Tienda
Tipo: Funcional
Estado: En Producción
Prioridad: Alta
Documentos:
Descripción: Permitir a los usuarios consultar los productos disponibles en la aplicación. Los usuarios podrán acceder a la sección de la tienda y visualizar los productos disponibles sin dificultad. No se requiere rellenar campos de entrada, solo seleccionar el botón designado para acceder a la pestaña de consulta de productos de la tienda.
Entradas: Ninguna
Salidas: Lista de productos disponibles en la tienda
Precondición: El usuario ha iniciado sesión en la aplicación (opcional).
Postcondición: El usuario puede visualizar la lista de productos disponibles en la tienda.
Manejo de Situaciones Anormales:
- Ninguna
Criterios de Aceptación:
- Los usuarios pueden acceder a la ventana de consulta de productos de la tienda sin problemas.
- La lista de productos disponibles se muestra correctamente en la ventana de consulta.
Destinatarios: Todos los usuarios, incluyendo usuarios casuales, usuarios registrados y usuarios con permisos de administrador, que deseen consultar los productos disponibles en la tienda.
<i>Autor: Alvaro Enríquez</i>
<i>Fuente: Del sistema aplicado</i>

Tabla 6. Creación de productos

R006: Adición de Productos por Usuarios Administradores en la Tienda

Tipo: Funcional

Estado: En Producción

Prioridad: Alta

Documentos:

Descripción: Permitir a los usuarios registrados que posean permisos de administrador añadir nuevos productos en la ventana de tienda de la aplicación. Estos usuarios deben tener permisos otorgados para acceder a esta funcionalidad. En la ventana de adición de productos en la tienda, se proporcionarán campos para ingresar información del producto, incluyendo su nombre, una imagen representativa y el precio. Los administradores deberán completar estos campos para agregar un nuevo producto. Una vez realizado el proceso, el usuario será redirigido a la ventana de la tienda, donde podrá verificar la correcta adición del producto con su nombre, imagen y precio correspondientes.

Entradas:

- Nombre del producto
- Imagen del producto
- Precio del producto

Salidas: Mensaje de confirmación de adición de producto en la tienda

Precondición: El usuario tiene permisos de administrador y ha iniciado sesión en la aplicación.

Postcondición: El nuevo producto se añade correctamente en la ventana de tienda de la aplicación.

Manejo de Situaciones Anormales:

- Si el usuario intenta agregar un producto sin completar todos los campos requeridos, se notificará que todos los campos deben ser rellenados correctamente.

Criterios de Aceptación:

- Los usuarios con permisos de administrador pueden acceder a la ventana de adición de productos en la tienda.
- El usuario puede completar los campos requeridos para agregar un nuevo producto.
- Se muestra un mensaje de confirmación después de agregar el producto en la tienda.
- El producto recién agregado se muestra correctamente en la ventana de la tienda con su nombre, imagen y precio.

Destinatarios: Usuarios que se han registrado y poseen permisos de administrador, y desean añadir productos en la ventana de tienda de la aplicación.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Tabla 7. Modificación de productos

R007: Modificación de Productos por Usuarios Administradores en la Tienda

Tipo: Funcional

Estado: En Producción

Prioridad: Alta

Documentos:

Descripción: Permitir a los usuarios registrados que posean permisos de administrador modificar productos en la ventana de tienda de la aplicación. Estos usuarios deben tener permisos otorgados para acceder a esta funcionalidad. En la ventana de modificación de productos en la tienda, se proporcionarán campos para ingresar información del producto, incluyendo su nombre, una imagen representativa y el precio. Los administradores deberán completar estos campos para realizar la modificación del producto. Para ello, los usuarios deben acceder a una ventana previa con un listado de productos, desde la cual seleccionarán el producto que desean modificar. Una vez realizada la modificación, se mostrará un mensaje de confirmación y el usuario será redirigido a la ventana de tienda para verificar la actualización del producto con su nombre, imagen y precio correspondientes.

Entradas:

- Nombre del producto
- Imagen del producto
- Precio del producto

Salidas: Mensaje de confirmación de modificación de producto en la tienda

Precondición: El usuario tiene permisos de administrador y ha iniciado sesión en la aplicación.

Postcondición: El producto se modifica correctamente en la ventana de tienda de la aplicación.

Manejo de Situaciones Anormales:

- Si los usuarios no completan todos los campos requeridos para la modificación, se notificará que todos los campos deben ser rellenados correctamente.

Criterios de Aceptación:

- Los usuarios con permisos de administrador pueden acceder a la ventana de modificación de productos en la tienda.
- Los usuarios pueden seleccionar un producto de la lista para modificar.
- El usuario puede completar los campos requeridos para modificar el producto.
- Se muestra un mensaje de confirmación después de modificar el producto en la tienda.
- El producto modificado se muestra correctamente en la ventana de la tienda con su nombre, imagen y precio actualizados.

Destinatarios: Usuarios registrados que poseen permisos de administrador, y desean modificar productos en la ventana de tienda de la aplicación.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Tabla 8. Eliminación de Productos

R008: Eliminación de Productos por Usuarios Administradores en la Tienda

Tipo: Funcional

Estado: En Producción

Prioridad: Alta

Documentos:

Descripción: Permitir a los usuarios registrados que posean permisos de administrador eliminar productos en la ventana de tienda de la aplicación. Estos usuarios deben tener permisos otorgados para acceder a esta funcionalidad. En la ventana de eliminación de productos en la tienda, se mostrará un listado de productos por su nombre, y para cada producto se proporcionará un botón para eliminarlo. Los administradores podrán seleccionar el producto que desean eliminar y, al hacerlo, la aplicación llevará a cabo la acción de borrado. Después de la eliminación, el usuario será redirigido a la ventana de tienda para verificar que el producto se ha eliminado correctamente.

Entradas:

- Selección del producto a eliminar

Salidas: Redirección a la ventana de tienda con mensaje de confirmación de eliminación de producto

Precondición: El usuario tiene permisos de administrador y ha iniciado sesión en la aplicación.

Postcondición: El producto seleccionado se elimina correctamente de la ventana de tienda.

Manejo de Situaciones Anormales:

- Si no se selecciona ningún producto para eliminar, se notificará al usuario que debe elegir un producto para eliminar.

Criterios de Aceptación:

- Los usuarios con permisos de administrador pueden acceder a la ventana de eliminación de productos en la tienda.
- Se muestra un listado de productos en la ventana de eliminación, con botones individuales para cada producto.
- Los usuarios pueden seleccionar un producto de la lista para eliminarlo.
- Al eliminar un producto, se muestra un mensaje de confirmación y el usuario es redirigido a la ventana de tienda.
- El producto eliminado ya no se muestra en la ventana de tienda.

Destinatarios: Usuarios registrados que poseen permisos de administrador, y desean eliminar productos en la ventana de tienda de la aplicación.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Tabla 9. Notificaciones

R009: Notificaciones
Tipo: Funcional
Estado: En Producción
Prioridad: Alta
Documentos:
Descripción: Permitir el envío de notificaciones dirigidas al administrador para la gestión de acciones necesarias internas en la empresa, con el fin alertar al usuario administrador cuando existan culminaciones de suscripciones, considerando todos los campos ingresados con anterioridad en el registro de la suscripción.
Entradas:
No existen entradas
Salidas: Mensajes de notificación en una bandeja de entrada
Precondición: El suscriptor aproxima un cambio de estado en su suscripción
Postcondición: El mensaje permanece con alerta hasta que sea visualizado
Manejo de Situaciones Anormales:
- Si no se observan los mensajes de notificación permanecerá un atenuante rojo sobre la bandeja de entrada.
Criterios de Aceptación:
- Usuarios registrados pueden acceder a la función de notificaciones.
- Los usuarios ingresan a la bandeja de notificaciones para visualizar una lista de pendientes.
<i>Autor: Alvaro Enríquez</i>
<i>Fuente: Del sistema aplicado</i>

Atributos del sistema

Los atributos fundamentales del sistema están intrínsecamente interconectados para ofrecer un software de alta calidad y funcionamiento. En lo que respecta a la confiabilidad, la aplicación se ha diseñado con el propósito de asegurar un rendimiento coherente y constante, lo que infunde confianza en su disponibilidad ininterrumpida. La facilidad de mantenimiento y adaptación del código es un componente esencial, ya que la aplicación ha sido completamente programada y estructurada de manera modular, lo que permite realizar modificaciones y mejoras futuras de manera fluida y sin generar efectos secundarios no deseados.

La portabilidad se consigue mediante la compatibilidad de los lenguajes de programación utilizados y la base de datos con múltiples sistemas operativos, lo que otorga la flexibilidad necesaria para ejecutarse en diversos entornos con uniformidad y eficiencia.

La seguridad se ha abordado de manera integral: los usuarios registrados acceden a través de contraseñas seguras y ciertas funcionalidades críticas exigen la posesión de permisos de administrador para mitigar riesgos de accesos no autorizados. Las prácticas avanzadas de seguridad se aplican en el almacenamiento de contraseñas y datos sensibles, reduciendo al mínimo las posibles vulnerabilidades.

En síntesis, estos atributos están entrelazados para garantizar una aplicación confiable, adaptable, versátil y segura. Esto, a su vez, crea una experiencia positiva para los usuarios y facilita a los desarrolladores llevar a cabo el mantenimiento y la mejora continua del sistema de manera efectiva. (Maldonado, 2018)

Otros requerimientos

No existen otros requerimientos.

Plan de desarrollo

En esta sección del informe, se detallará la planificación que se ha implementado para la ejecución del proyecto utilizando la metodología ágil Extreme Programming (XP). Aunque XP es comúnmente utilizado en equipos de trabajo, en este caso se ha adaptado para ser aplicado por un único individuo, ya que el desarrollo del proyecto se ha llevado a cabo de manera individual.

Es importante señalar que la etapa de estudio de la tecnología a utilizar, así como las fases de diseño y análisis del proyecto, no forman parte de la planificación. Antes de iniciar el desarrollo, se dedicó tiempo al estudio de la tecnología y se llevaron a cabo conversaciones con el cliente. Durante estas interacciones, se realizó la fase de análisis y diseño. A excepción de situaciones específicas en las que se requirió un estudio adicional de algún aspecto tecnológico para una funcionalidad en particular, o cuando el cliente proporcionó nuevos requisitos o aclaraciones, que se incorporaron en las interacciones, el tiempo asignado a estos aspectos se consideró en los ciclos restantes.

La primera fase para lograr la implementación del proyecto en un período de tiempo específico consistió en la planificación de iteraciones. Esto implicó definir y desarrollar una lista de objetivos que se deberían alcanzar a medida que se completaran las iteraciones. Para garantizar el éxito de esta metodología, se evitaron crear iteraciones con demasiadas tareas y se procuró una estimación realista de la velocidad de trabajo. Se diseñaron 12 iteraciones, cada una con una duración de una semana, en las cuales se determinaron las tareas a realizar.

Dado que este proyecto marcaba la primera aplicación desarrollada utilizando esta tecnología, las primeras iteraciones tuvieron una carga de trabajo relativamente baja. Se centraron en el diseño inicial de la aplicación y en la creación de las primeras páginas, que no requerían códigos de gran complejidad. En contraste, las iteraciones posteriores se planificaron

con una carga de trabajo más significativa, considerando que se había adquirido experiencia con la tecnología. A pesar de ello, las iteraciones no estaban ajustadas al límite temporal, ya que se dejó un margen de horas por posibles complicaciones. Este margen permitía la incorporación de nuevas funcionalidades sugeridas por el usuario, ajustes requeridos o estudios adicionales sobre aspectos específicos de la tecnología para su aplicación en el proyecto. (Aliaga Melo, 2012)

Inicialmente, la planificación pudo parecer ajustada en tiempo, a pesar del margen de horas previamente mencionado para afrontar posibles complicaciones o cambios. Sin embargo, después de varias semanas cumpliendo exitosamente con los objetivos de las iteraciones, quedó claro que la planificación había sido precisa al asignar las horas y tareas correspondientes. De hecho, el proyecto se completó según el cronograma inicialmente estimado en la planificación, a pesar de algunas iteraciones más intensivas que otras. Esto demuestra que la decisión de elaborar una planificación propia resultó acertada, ya que permitió mantener un ritmo constante de trabajo durante el desarrollo del proyecto, similitud a la forma en que se gestionaría el desarrollo de una aplicación en un entorno empresarial, con plazos establecidos para la finalización del proyecto.

Metodología de desarrollo de software XP.

Fase 1: Planificación

- Compilación de Requerimientos.
- Plan de iteraciones para el desarrollo.

Fase 2: Diseño

- Creación de un diagrama de clases que abarque los requisitos funcionales
- Diseño de la interfaz de usuario

Fase 3: Codificación

- Utilización del Framework Django con Python

Fase 4: Pruebas

- Interacción del usuario administrador con la gestión del sistema.

Fase 5: Implementación

- Elaboración de un prototipo de la página web
- Despliegue inicial del servidor de forma local para una revisión final

Planificación

En esta sección, nos sumergiremos en el proceso detallado de desarrollo y análisis que ha dado vida a este proyecto. Nuestro enfoque se centrará en explorar los aspectos técnicos, las decisiones estratégicas tomadas y los desafíos que se enfrentaron durante la ejecución de esta iniciativa. El proyecto se ha llevado a cabo siguiendo una adaptación de una metodología ágil, focalizada en la individualidad y basada en Extreme Programming (XP). A lo largo de esta sección, analizaremos cómo se ha traducido esta metodología en acciones concretas que culminaron en la exitosa creación de la aplicación.

Antes de entrar en detalles, se realizó un análisis exhaustivo y un diseño preliminar del proyecto, junto con la selección de la tecnología más apropiada para su implementación. Estos cimientos sirvieron como base para la fase de planificación, en la que se esbozaron las iteraciones y se establecieron los hitos a alcanzar en cada una de ellas. Es importante señalar que, además de las tareas técnicas, se incluyeron momentos de interacción con el cliente, permitiendo ajustes y refinamientos en los objetivos del proyecto durante su desarrollo.

A medida que avancemos en esta sección, exploraremos los hitos clave del proyecto, desde la concepción de la arquitectura y la creación de las funcionalidades, hasta las etapas de

pruebas y optimización del código. También abordaremos los enfoques empleados para asegurar la confiabilidad, seguridad y eficiencia del sistema, así como las estrategias implementadas para abordar imprevistos y requisitos adicionales que surgieron durante el proceso de desarrollo.

Este segmento ofrecerá una visión en profundidad del proceso de creación de la aplicación, destacando las elecciones tomadas en cada fase y cómo se aplicaron los principios ágiles para adaptarse a las cambiantes necesidades y lograr una implementación exitosa. Mediante un análisis completo del desarrollo, esta sección proporcionará una comprensión integral de cómo se transformó la idea inicial en una aplicación funcional y sólida. (Rahmadi Islam, 2018)

Plan de Iteración para el Desarrollo de la Aplicación de Gimnasio

A lo largo de este plan, se enfatiza la importancia de las pruebas continuas y la retroalimentación constante. Cada iteración está diseñada para llevar a cabo el desarrollo incremental de funcionalidades clave, garantizando que cada componente esté probado y optimizado antes de avanzar al siguiente paso. Además, se presta especial atención a la interacción con los usuarios y la incorporación de sus comentarios para asegurar que la aplicación cumpla con los requisitos y ofrezca una experiencia excepcional.

Tabla 10. Iteración 1

Iteración 1:

Duración: 2 semanas

Establecer el entorno de desarrollo y configurar el framework Django.

Desarrollar y probar la autenticación de usuarios administradores.

Implementar la base de datos para almacenar información de usuarios y actividades.

*Autor: Alvaro Enríquez**Fuente: Del sistema aplicado*

Tabla 11. Iteración 2

Iteración 2:

Duración: 2 semanas

Desarrollar la funcionalidad de registro de nuevos usuarios, incluyendo los campos necesarios y validaciones.

Implementar las vistas y plantillas para la consulta de actividades y productos disponibles en la tienda.

Realizar pruebas de integración de las funcionalidades implementadas.

*Autor: Alvaro Enríquez**Fuente: Del sistema aplicado*

Tabla 12. Iteración 3

Iteración 3:

Duración: 2 semanas

Desarrollar la funcionalidad de notificaciones dirigidas al administrador para gestión interna.

Implementar la capacidad de cierre de sesión para usuarios registrados.

Refinar la interfaz de usuario para mejorar la experiencia del usuario.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Tabla 13. Iteración 4

Iteración 4:

Duración: 3 semanas

Desarrollar las funcionalidades de gestión de productos para usuarios administradores: adición, modificación y eliminación.

Implementar la funcionalidad de visualización de productos en la tienda para usuarios regulares.

Realizar pruebas exhaustivas de todas las funcionalidades implementadas hasta ahora.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Tabla 14. Iteración 5

Iteración 5:

Duración: 2 semanas

Implementar la funcionalidad de consultas bidireccionales para usuarios y trabajadores del gimnasio.

Realizar pruebas de interacción y comunicación directa para asegurarse de que las consultas funcionan correctamente.

Realizar ajustes y mejoras basados en los resultados de las pruebas de usuario.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Tabla 15. Iteración 6

Iteración 6:

Duración: 1 semana

Refinar y optimizar el rendimiento de la aplicación.

Realizar pruebas exhaustivas de seguridad y corregir posibles vulnerabilidades.

Preparar la aplicación para el despliegue en un entorno de producción.

Autor: Alvaro Enríquez

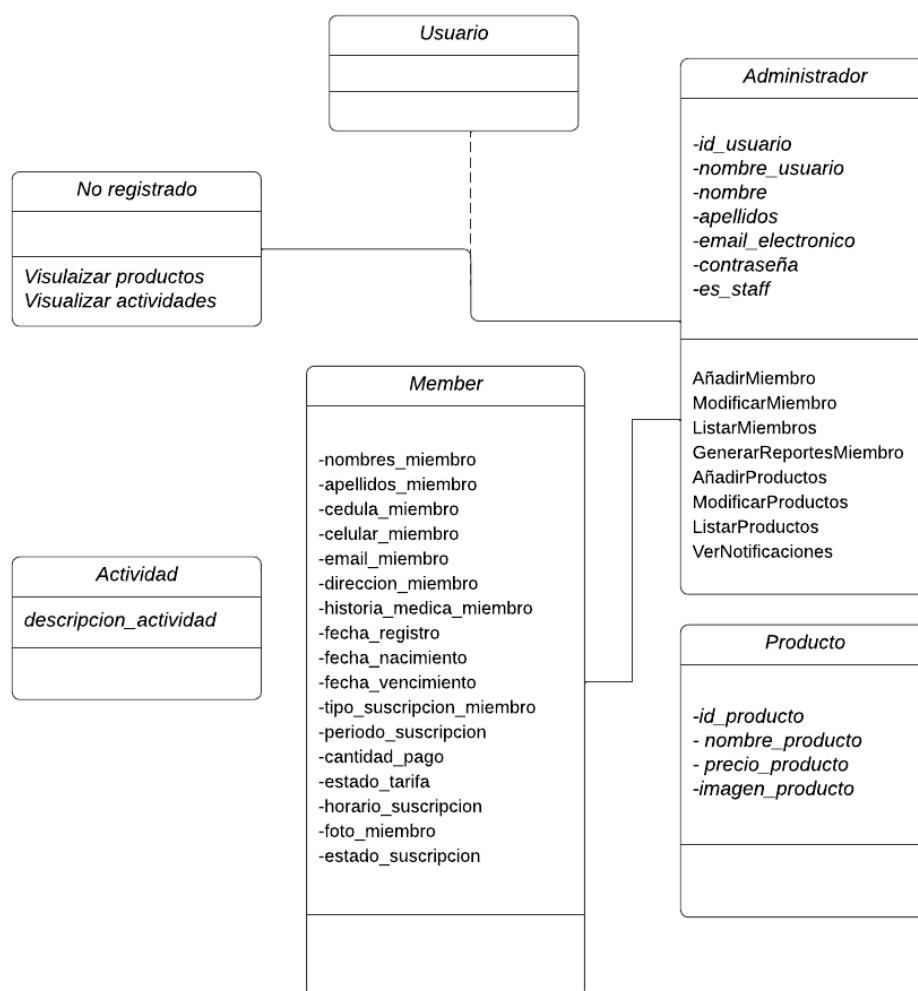
Fuente: Del sistema aplicado

Diseño

Diagrama de clases

En esta sección, vamos a presentar un diagrama de clases que proporciona una representación visual de los modelos de las tablas de usuario, actividad y producto en nuestro proyecto. A través de esta imagen, podremos apreciar la estructura y las interconexiones entre estos elementos esenciales.

Figura 1. Diagrama de Clases.



Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Es importante resaltar que, en el diagrama, se puede observar que las tablas de producto y actividad no presentan relaciones directas con las tablas de usuarios. Esta particularidad surge debido a que, aunque los administradores tienen la capacidad de crear objetos tanto en la tabla de producto como en la de actividad, no existe una vinculación inherente entre estas entidades y la de usuarios en términos de relaciones de base de datos.

Es relevante señalar que los administradores pueden realizar estas creaciones con tan solo contar con los permisos adecuados. Estos permisos están representados en el atributo "role", el cual, cuando tiene un valor "admin", habilita la función de creación de objetos en las mencionadas tablas. Sin embargo, la falta de una conexión directa entre estas entidades en el diagrama no implica que no exista una relación funcional en el sistema, ya que los administradores pueden interactuar con ellas mediante los permisos correspondientes.

En este contexto, la ausencia de conexiones visibles entre las tablas en el diagrama no significa que no exista una relación en el funcionamiento real de la aplicación. Por el contrario, demuestra que la interacción entre estas entidades se basa en los roles y permisos de los usuarios, lo cual se refleja en la arquitectura interna de la aplicación y en la lógica de sus funcionalidades.

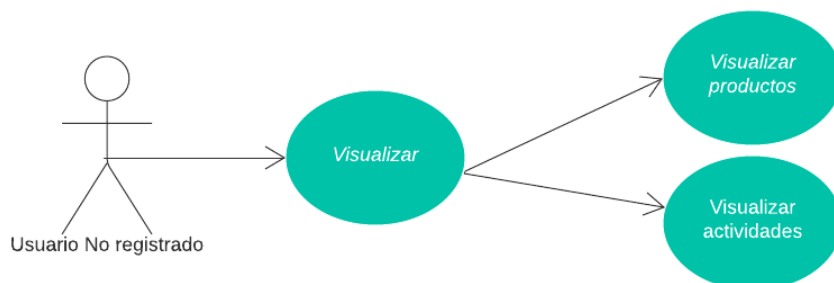
Diagrama de casos

En las próximas secciones, presentaremos los diagramas de casos de uso de nuestra aplicación. Estos diagramas tienen la finalidad de visualizar de manera detallada los diversos procesos que la aplicación es capaz de llevar a cabo, destacando la participación de los usuarios en cada acción y las operaciones específicas que pueden realizar. (Carlos Billy Reynoso, 2004)

Comenzaremos por mostrar el caso de uso fundamental, que sirve como punto de partida para nuestro análisis. En este caso, se presentan dos diferentes tipos de usuarios que pueden interactuar con la aplicación, cada uno con sus respectivas acciones y capacidades. Este

escenario base nos brinda una comprensión esencial de cómo se desarrollan las interacciones en la aplicación a través de sus distintos usuarios.

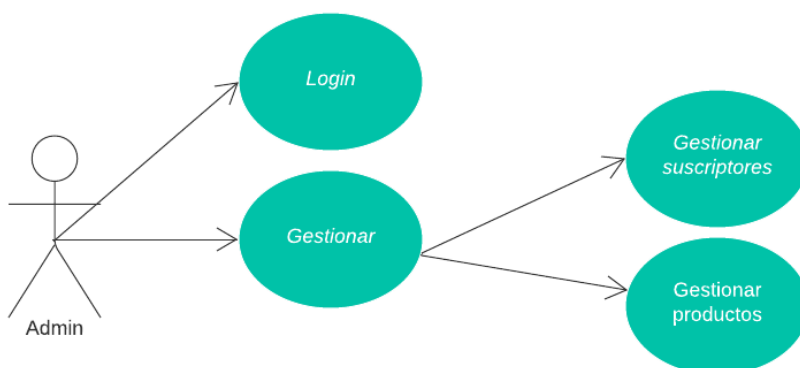
Figura 2. Diagrama de Casos.



Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Figura 3. Diagrama de Casos.



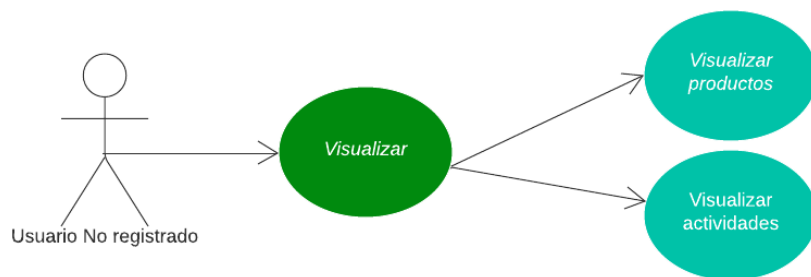
Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Casos de Uso

A continuación, presentaremos los escenarios de uso que exhiben todas las capacidades que la aplicación puede llevar a cabo para los diversos perfiles de usuarios.

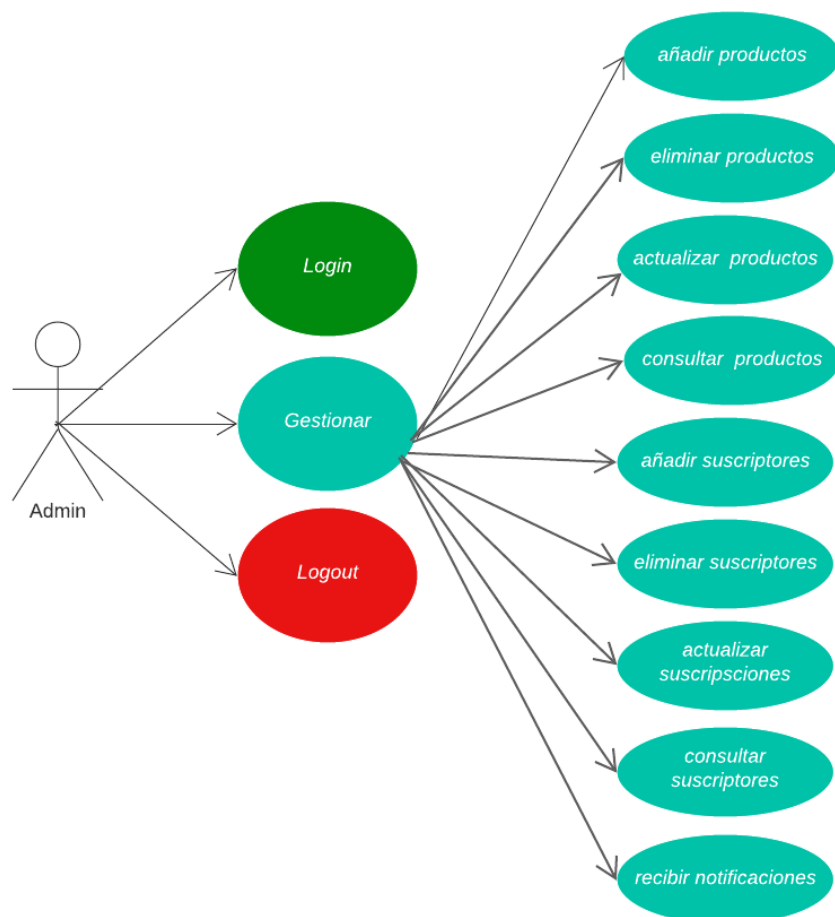
Figura 4. Caso de uso.



Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Figura 5. Caso de uso.



Autor: Alvaro Enríquez

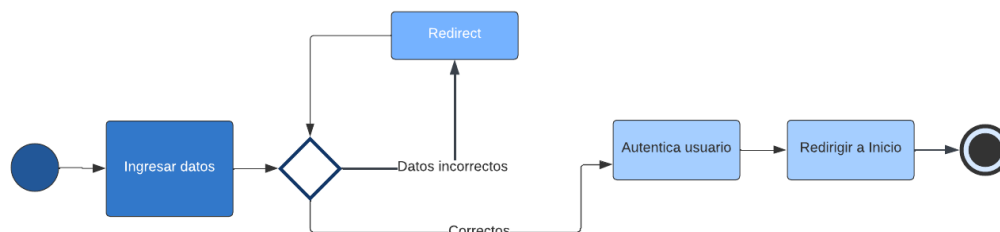
Fuente: Del sistema aplicado

Diagramas de Flujo

En esta sección, ilustraremos las funciones más elaboradas de la aplicación empleando diagramas de flujo de actividad.

Autenticarse

Figura 6. Diagrama de flujo 1.



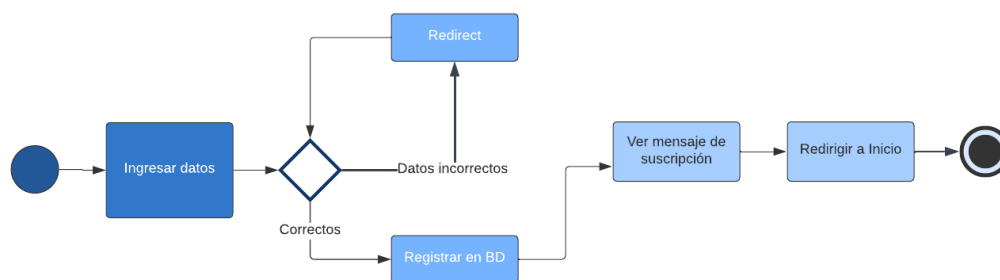
Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

El procedimiento de autenticación requiere que el usuario, previamente registrado en la aplicación, ingrese los datos correspondientes. En el caso de que alguno de los datos proporcionados sea incorrecto, el usuario será redirigido nuevamente a la página de inicio de sesión. Sin embargo, si los datos de autenticación son precisos y válidos, la aplicación procederá a autenticar al usuario, tras lo cual lo redirigirá de manera exitosa a la página principal de la aplicación para comenzar su sesión.

Registrar suscriptor

Figura 7. Diagrama de flujo 2.



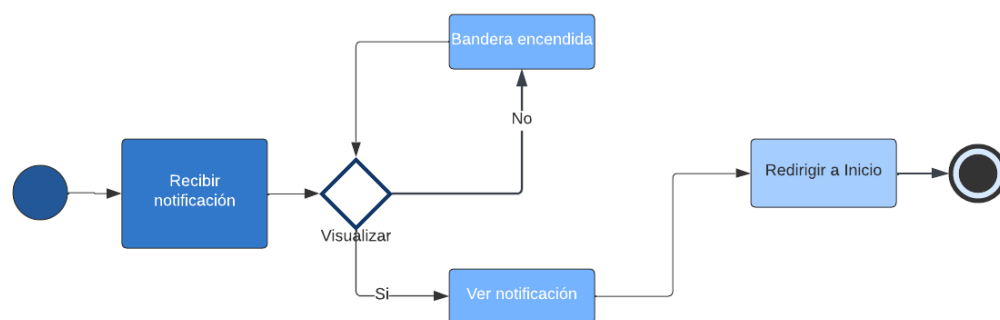
Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

El proceso de registro implica que el usuario complete los campos requeridos. En caso de que alguno de los datos proporcionados sea incorrecto, se le redireccionará nuevamente a la página de registro. Sin embargo, si todos los datos son precisos y válidos, la aplicación procederá a registrar al usuario en la base de datos. Una vez completado el registro, el usuario será automáticamente autenticado y dirigido a la página principal de la aplicación para comenzar su experiencia.

Notificaciones

Figura 8. Diagrama de flujo 3.



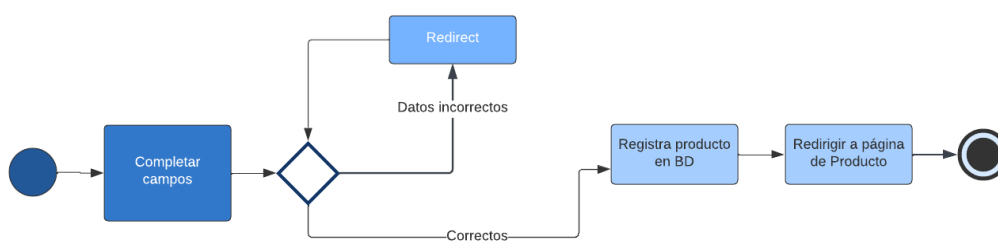
Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Los usuarios registrados que cuenten con permisos de administrador recibirán notificaciones en la página principal. En caso de que la notificación no ha sido abierta o visualizada se mantendrá una bandera encendida. No obstante, si el proceso se completa adecuadamente, el usuario con permisos de administrador será redirigido de vuelta a la página principal de la aplicación.

Añadir producto

Figura 9. Diagrama de flujo 4.



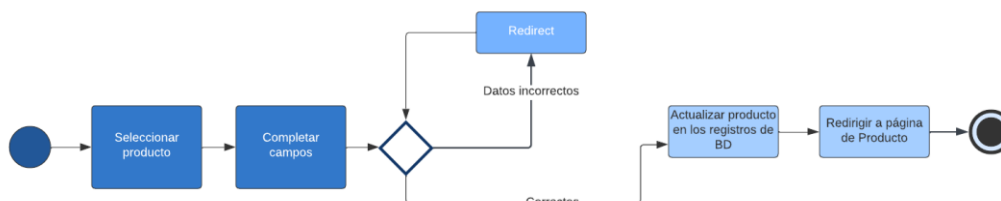
Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

El usuario que posea permisos de administrador y esté registrado deberá completar los campos asociados al producto. En caso de que algún campo quede sin completar, será redirigido automáticamente a la página de creación de productos. Por otro lado, si todos los campos están correctamente rellenos, se procederá a crear el nuevo producto en la base de datos y el usuario será dirigido a la ventana correspondiente de productos.

Actualizar producto

Figura 10. Diagrama de flujo 5.



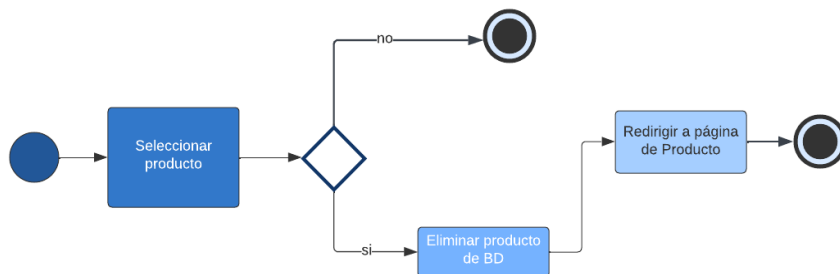
Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

El usuario registrado con privilegios de administrador deberá localizar el producto en la lista, realizar su selección y proceder a completar los campos que se deseen modificar. Si en algún caso algún campo queda en blanco, el sistema lo redirigirá automáticamente a la página que muestra el listado de productos. Por otro lado, si todos los campos son completados de manera adecuada, se efectuará la modificación del producto en la base de datos y el usuario será redirigido nuevamente a la ventana que presenta el listado de productos.

Eliminar producto

Figura 11. Diagrama de flujo 6.



Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

El usuario registrado con permisos de administrador deberá ubicar el producto que desea eliminar en el listado y realizar su selección. Posteriormente, se procederá a eliminar el producto correspondiente de la base de datos, lo que resultará en la redirección del usuario a la página que muestra el listado completo de productos.

Codificación

En el proceso de desarrollo, es esencial contar con una estrategia estructurada que guíe el equipo de desarrollo desde la concepción hasta la implementación exitosa del producto final. Uno de los enfoques más eficaces en este sentido es la metodología de Desarrollo Extremo (XP), que promueve la flexibilidad, la colaboración constante y la mejora continua a lo largo del ciclo de desarrollo. En este contexto, el plan de iteración se presenta a continuación como una hoja de ruta detallada para posterior diseño.

Estructura

En la construcción de esta aplicación, se ha implementado el patrón de diseño conocido como Modelo-Vista-Controlador (MVC). Este patrón arquitectónico se caracteriza por mantener una distinción clara y estructurada entre tres componentes fundamentales: la interfaz de usuario, la lógica de negocio y la gestión de datos.

El patrón MVC se traduce en una organización efectiva del código, donde cada uno de los componentes desempeña un papel específico. La "Vista" se encarga de mostrar la información al usuario y recibir sus interacciones, la "Lógica de Negocio" o "Controlador" gestiona las acciones y procesos de la aplicación, mientras que el "Modelo" se ocupa de gestionar y acceder a los datos subyacentes.

Esta separación de responsabilidades en módulos independientes proporciona varios beneficios. En primer lugar, simplifica el desarrollo al permitir que los equipos se centren en áreas específicas sin interferencias innecesarias. Además, facilita la comprensión y el mantenimiento del código, ya que cada componente es coherente y autónomo. Además, la modularidad del patrón MVC permite realizar cambios en un área sin afectar otras partes de la aplicación.

En resumen, la elección del patrón MVC para esta aplicación se basa en la búsqueda de una arquitectura clara y organizada que simplifique el desarrollo, el mantenimiento y la escalabilidad de la aplicación a lo largo del tiempo.(De Ingeniería et al., 2019)

Modelo

En el contexto del patrón de diseño MVC, la capa del Modelo desempeña un papel fundamental al controlar y gestionar tanto los datos como la lógica de negocio de la aplicación. Su responsabilidad radica en definir la estructura de los datos necesarios para el funcionamiento de la aplicación y en manejar cualquier cambio que ocurra en esos datos. En muchas ocasiones, cuando se produce una modificación en los datos, es el Modelo el encargado de notificar a la capa de Vista o al Controlador, según corresponda.

En el marco de nuestra aplicación, la administración de esta capa recae en el Sistema de Gestión de Bases de Datos (SGBD). Como se mencionó previamente, Django, por defecto, utiliza un gestor de bases de datos (aunque es posible emplear otro), que se encarga de realizar las operaciones de modificación en la base de datos. Esto se logra mediante el uso de modelos que se definen mediante código en la propia aplicación.

A continuación, procederemos a definir las clases que se utilizan en nuestra aplicación, las cuales son esenciales para representar los distintos elementos y entidades del sistema:

Usuario: Esta clase contiene los datos de los usuarios que se registran en la aplicación. Incluye información relevante para gestionar a los usuarios en el sistema.

Actividad: En este campo se guardan los datos relacionados con las actividades que son ofrecidas por el centro. Esta campo representa una actividad específica y contiene información detallada sobre la misma.

Producto: La clase Producto se encarga de almacenar los datos asociados a los productos que están disponibles para la venta en la aplicación. Cada instancia de esta clase representa un producto y contiene detalles relevantes como su nombre, descripción y precio.

Vista

Dentro del marco del patrón de diseño MVC, la capa de Vista se encarga de definir la presentación de los datos en la interfaz de usuario de la aplicación. En términos más específicos, esta capa abarca la parte visual de la aplicación que los usuarios pueden ver, incluyendo componentes como botones, cuadros de texto, etiquetas, entre otros. En el contexto de Django, es común utilizar plantillas HTML para establecer el diseño de las aplicaciones web. Además, se pueden aprovechar bibliotecas como Bootstrap y archivos CSS para realzar el aspecto visual de los componentes definidos en los documentos HTML.

Un aspecto crucial de la capa de Vista en cualquier aplicación web es la necesidad de que su diseño sea adaptable, es decir, que la aplicación pueda ajustarse y reorganizar sus componentes para adaptarse a las dimensiones del dispositivo en el que se esté utilizando. Esto garantiza una experiencia más cómoda y satisfactoria para el usuario, independientemente de si está accediendo desde una pantalla de ordenador, una tablet o un smartphone.

Controlador

Dentro del patrón MVC, la capa de Controlador desempeña un papel fundamental en la lógica de la aplicación. Su principal responsabilidad es actualizar tanto la capa de Modelo como la capa de Vista (o ambas simultáneamente) en respuesta a las interacciones que los usuarios llevan a cabo en la aplicación. En esencia, el Controlador gestiona el flujo de datos y transformaciones necesarias para adaptar la información según las necesidades de las otras dos capas.

En el entorno de Django, la capa de Controlador se manifiesta a través de los documentos de rutas y manejo de las entradas de los usuarios. Esto se refleja en la presencia de uno o varios archivos "urls.py" y uno o varios archivos "views.py" en cualquier aplicación desarrollada en Django.

El archivo "urls.py" asume el papel de gestionar las rutas de la aplicación, es decir, el enrutamiento. Si la aplicación consta de múltiples aplicaciones, cada una con su propio archivo "urls.py", estos archivos deben estar integrados en el archivo "urls.py" de la aplicación principal.

Contrariamente a lo que su nombre podría sugerir, los archivos "views.py" no están encargados de la capa de Vista, sino que se responsabilizan de gestionar las solicitudes (requests) realizadas por los usuarios y generar respuestas conforme a lo definido por los desarrolladores en dichos archivos. Desde los archivos "views.py", los desarrolladores pueden interactuar tanto con elementos de la base de datos como con componentes de la interfaz de usuario, estableciendo así una conexión entre las capas de Vista y Modelo. Esto posibilita la administración y facilita la interacción entre los usuarios y la aplicación, generando respuestas adecuadas ante diversas acciones realizadas por los usuarios.

La combinación de los archivos "views.py" y "urls.py" es la que da forma a la capa de Controlador en una aplicación desarrollada en Django. Esta combinación permite, además, manejar situaciones en las que es necesario redirigir a los usuarios a diferentes páginas. En estos casos, el archivo "urls.py", encargado de administrar las rutas de la aplicación, resulta esencial. De esta manera, estos dos archivos asumen un papel integral en la gestión de las interacciones entre los usuarios y la aplicación, otorgándoles un control absoluto sobre este proceso.

Entorno de desarrollo

Para iniciar este proyecto, el primer paso consistió en tomar la decisión de cambiar el sistema operativo de la laptop en la que llevaría a cabo el desarrollo de la aplicación. Esta elección surgió debido a la necesidad de utilizar instrucciones de línea de comandos para ejecutar y crear programas en Django. Dado que en la universidad ya contábamos con experiencia en el uso de la línea de comandos en entornos Linux, teníamos una base de conocimientos en este aspecto. En contraste, en Windows no habíamos desarrollado esa habilidad en la misma medida debido a que no se requería tanto, ya que Windows ofrece una experiencia más amigable sin la necesidad extensiva de la línea de comandos. Además, es importante señalar que la potencia de la línea de comandos en Linux supera a la de Windows en muchos aspectos.

Por lo tanto, el primer paso para dar inicio al proyecto fue crear un USB booteable. Un USB booteable permite que la computadora arranque desde el USB mismo, lo que facilita la ejecución del sistema operativo cuya imagen de arranque se encuentra en dicho dispositivo. También facilita la instalación del sistema operativo en la máquina.

Para este proyecto en particular, opté por utilizar el programa Rufus para crear el USB booteable. Rufus se destaca por ser uno de los programas más completos y fáciles de usar disponibles para la creación de este tipo de dispositivos. El proceso simplemente implicó contar con un USB con suficiente capacidad de almacenamiento y la imagen del sistema operativo deseado. En este caso, mi elección fue Windows, y más específicamente, la versión 10, junto con las últimas actualizaciones de seguridad.(Culhuacan et al., 2020)

Además, fue esencial instalar el lenguaje de programación Python para habilitar el entorno de desarrollo necesario. La instalación del lenguaje de programación requería ejecutar las siguientes instrucciones:

Python 3.10:

Para asegurar una base sólida, se optó por la versión más reciente de Python, la 3.10. Esta decisión se tomó en base a su compatibilidad y actualizaciones recientes. El proceso de instalación se llevó a cabo siguiendo estos pasos:

Se accedió al sitio oficial de Python en <https://www.python.org/downloads/release/> para descargar el instalador correspondiente a Windows.

Una vez descargado el instalador, se procedió a ejecutarlo.

Durante la instalación, se marcó la opción "Add Python 3.10 to PATH" para facilitar la accesibilidad a Python desde la línea de comandos.

Tras completar la instalación, se abrió el símbolo del sistema (Command Prompt) y se verificó la correcta instalación de Python ejecutando el comando `python --version`.

PIP (Gestor de paquetes de Python):

La herramienta pip es fundamental para administrar los paquetes y dependencias del proyecto. El proceso de instalación se documentó de la siguiente manera:

Se verificó si pip ya estaba instalado en el sistema ejecutando `pip --version` en el símbolo del sistema.

En caso de que pip no estuviera instalado, se descargó el script `get-pip.py` desde <https://bootstrap.pypa.io/get-pip.py>.

Se abrió una terminal y se navegó hasta la ubicación donde se descargó `get-pip.py` utilizando el comando `cd ruta\hasta\archivo`.

A continuación, se ejecutó el siguiente comando para instalar pip:

```
python get-pip.py
```

Django:

La instalación de Django, el framework elegido para el desarrollo del proyecto, fue una parte esencial del proceso de configuración del entorno:

Con pip instalado, se procedió a instalar Django utilizando el comando:

```
pip install django
```

La instalación de Django fue confirmada ejecutando `django-admin --version` en la terminal.

Proyecto de Django

Preparación del Ambiente:

Previo a la instauración del proyecto, se verificó que el sistema Windows contara con Python 3.10, el sistema de gestión de paquetes pip y el framework Django correctamente instalados, tal como se especifica en la sección "Configuración Inicial del Entorno de Desarrollo" en la documentación del proyecto.

Creación del Proyecto:

Para dar inicio al proyecto, se siguió el proceso estándar de creación proporcionado por Django:

Se procedió a abrir una terminal en el directorio seleccionado para el proyecto.

A continuación, se ejecutó el siguiente comando con el fin de instaurar un nuevo proyecto Django con el nombre "PGG":

```
django-admin startproject GYM
```

Exploración de la Estructura:

Una vez que el proyecto fue generado, se exploró a fondo la estructura resultante, compuesta por directorios y archivos creados por Django. Esta estructura proporciona una sólida base organizativa para el proyecto y sus diversas aplicaciones.

Configuración Inicial:

Posteriormente, se procedió a llevar a cabo ciertas configuraciones iniciales dentro del archivo settings.py del proyecto. Entre estas configuraciones se incluyó la definición de la zona horaria y la configuración pertinente de la base de datos.

Generación de la Aplicación:

En congruencia con la metodología de desarrollo basada en aplicaciones, se procedió a instaurar la primera aplicación dentro del marco del proyecto:

Se ingresó a la carpeta del proyecto utilizando la interfaz de línea de comandos.

Utilizando el comando `python manage.py startapp Gym`, se generó una nueva aplicación denominada "Gym".

Ajustes en la Configuración de la Aplicación:

Se llevó a cabo la exploración y configuración de los archivos generados automáticamente para la aplicación recién creada, tales como models.py, views.py y urls.py.

Ejecución de Migraciones y Creación de la Base de Datos:

Se efectuó la generación de las migraciones iniciales para la base de datos, haciendo uso de los siguientes comandos:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

Puesta en Marcha del Servidor de Desarrollo:

Finalmente, se procedió a arrancar el servidor de desarrollo proporcionado por Django para poner a prueba el proyecto:

Se llevó a cabo la ejecución del siguiente comando:

```
python manage.py runserver
```

Dicho comando permitió el acceso a la aplicación a través de un navegador web local, con la dirección `http://127.0.0.1:8000/`.

Con esta descripción minuciosa de la creación de un proyecto Django en el entorno Windows, se ha establecido una base sólida que facilita el desarrollo continuo de la aplicación. Además, esta documentación asegura que futuros colaboradores puedan comprender y replicar el proceso de manera efectiva y eficiente.

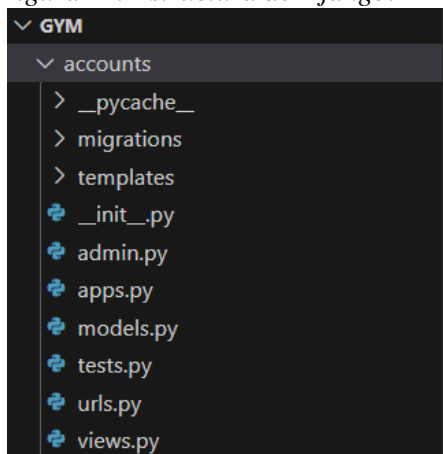
Estructura del proyecto

En el contexto del proyecto PGG, se aborda en detalle la estructura esencial que define una aplicación web desarrollada en Django en el entorno de Windows. A continuación, se sintetiza el panorama de la estructura básica del proyecto, así como las particularidades que se presentan al crear y organizar componentes clave en el marco de esta plataforma de desarrollo.

Estructura Básica del Proyecto:

En la siguiente representación visual se presenta la estructura fundamental de un proyecto Django, donde se resaltan cinco archivos de relevancia en el directorio principal "proyecto":

Figura 12. Estructura de Django.



Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Cada uno de estos archivos desempeña un papel crucial en la configuración, funcionamiento y administración del proyecto.

Archivo `__init__.py`:

Este archivo sirve como indicador para Python, señalando que la carpeta contenedora es un directorio de paquetes, lo que facilita la importación y organización de módulos.

Archivo `settings.py`:

En este archivo de configuración se definen diversos aspectos del proyecto, incluyendo la activación del depurador, las aplicaciones ejecutadas, la base de datos, el idioma y las rutas de archivos estáticos y multimedia.

Archivo `urls.py`:

Este archivo se utiliza para registrar las URLs que la aplicación va a utilizar, enlazando las vistas correspondientes y definiendo los patrones de URL.

Archivo `wsgi.py`:

El archivo Web Server Gateway Interface (WSGI) especifica cómo se comunica un servidor web con una aplicación web, gestionando el flujo de peticiones y respuestas.

Archivo manage.py:

Este archivo facilita la administración del sitio web, permitiendo arrancar un servidor web local sin necesidad de instalaciones adicionales.

Desarrollo de Componentes Esenciales:

En la construcción de una aplicación Django en Windows, diversos componentes clave son cruciales para la funcionalidad y la interacción del usuario:

Modelos de Base de Datos (models.py):

En la subcarpeta "proyecto", se encuentra el archivo "models.py", donde se definen los modelos de base de datos utilizando campos, nombres y tipos de atributos específicos. Tras definir el modelo, la ejecución de los comandos "makemigrations" y "migrate" permite registrar el modelo en la base de datos.

Figura 13. Creación de modelo en Django.

```
models.py 3 ●
accounts > models.py > ...
1  from django.db import models
2  from django.forms import.ModelForm
3
4  # Create your models here.
5  class Wallpaper(models.Model):
6      photo = models.FileField(verbose_name="Foto", upload_to='wallpaper/')
7
8  class WallpaperForm(ModelForm):
9      class Meta:
10         model = Wallpaper
11         fields = '__all__'
12
13 class Producto(models.Model):
14     nombre = models.CharField(max_length=100)
15     precio = models.DecimalField(max_digits=10, decimal_places=2)
16     imagen = models.ImageField(upload_to='productos/', null=True, blank=True)
17
18     def __str__(self):
19         return self.nombre
20
```

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Formularios (forms.py):

La creación de formularios para la aplicación se realiza a través del archivo "forms.py". Aquí se puede especificar el tipo de formulario, los campos que contendrá, las características de cada campo y su validación, contribuyendo a una interacción efectiva con los usuarios.

Figura 14. Creación de formulario de Django.

```
forms.py 2 x
members > forms.py > ...
1  from django import forms
2
3  class AddMemberForm(forms.Form):
4      first_name = forms.CharField(label='Nombres', max_length=100)
5      last_name = forms.CharField(label='Apellidos', max_length=100)
6      mobile_number = forms.IntegerField(label='Celular')
7      #email = forms.EmailField()
8
```

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Plantillas (templates):

Las plantillas utilizadas en la capa de Vista se almacenan generalmente en la carpeta "templates". Aunque es convencional, no obligatorio, este enfoque permite organizar y gestionar los archivos de interfaz de usuario de manera eficaz.

Figura 15. Plantillas Bootstrap.

```

update_member.html x
members > templates > update_member.html > ...
1  {% load tags %}
2  <div class="container" style="margin-top: 3%">
3      {% if deleted %}
4          <div class="alert alert-success">
5              {{ deleted }}
6          </div>
7      {% endif %}
8      <form method="POST">
9          {% csrf_token %}
10         <div class="row">
11             <div class="form-group col-lg-6 col-md-6 col-sm-12 col-xs-12 offset-md-3 text-center">
12                 <label for="search">{{ search_form.search.label }}</label>
13                 {{ search_form.search | add_css:"form-control" }}
14             </div>
15         </div>
16         <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 text-center">
17             <button type="submit" id="submit" class="btn btn-primary">Buscar</button>
18         </div>
19     </form>
20     <div id="ajax_result" style="margin-top: 2%;"></div>
21 </div>
22

```

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Codificación de los modelos

En el contexto del desarrollo de la aplicación de gestión de un gimnasio, la fase de codificación de los modelos es crucial para establecer la estructura de datos y la lógica de negocio subyacente. Los modelos representan las entidades centrales con las que interactúa la aplicación, y en este caso, nos centraremos en tres modelos clave: Usuario, Producto y Miembro.

Modelo member

El modelo de Miembro se enfoca en representar a los suscriptores o miembros del gimnasio. Los atributos como nombres, apellidos, fecha de inicio de la suscripción, fecha de vencimiento, estado de la suscripción y detalles de contacto son esenciales en este modelo. La codificación del modelo de Miembro garantiza un seguimiento preciso de los miembros, incluida la generación de notificaciones en caso de vencimiento de la suscripción.

Figura 16. Modelo miembro.

```

class Member(models.Model):
    member_id = models.AutoField(primary_key=True)
    first_name = models.CharField(('Nombres'), max_length=50)
    last_name = models.CharField(('Apellidos'), max_length=50)
    mobile_number = models.CharField(('Celular'), max_length=10, unique=True)
    email = models.EmailField(null=True, blank=True)
    address = models.CharField(('Dirección'), max_length=300, blank=True)
    medical_history = models.CharField(('Historia Medica'), max_length=300, blank=True, default='Ninguna')
    admitted_on = models.DateField(auto_now_add=True)
    registration_date = models.DateField(('Fecha de Registro'), default='dd/mm/yyyy')
    registration_upto = models.DateField()
    dob = models.DateField(default='dd/mm/yyyy')
    subscription_type = models.CharField(
        ('Tipo de suscripción'),
        max_length=30,
        choices=SUBSCRIPTION_TYPE_CHOICES,
        default=SUBSCRIPTION_TYPE_CHOICES[0][0]
    )
    subscription_period = models.CharField(
        ('Periodo de suscripción'),
        max_length=30,
        choices=SUBSCRIPTION_PERIOD_CHOICES,
        default=SUBSCRIPTION_PERIOD_CHOICES[0][0]
    )
    amount = models.CharField(('Cantidad'), max_length=30)
    fee_status = models.CharField(
        ('Estado de la tarifa'),
        max_length=30,
        choices=FEE_STATUS,
        default=FEE_STATUS[0][0]
    )
    batch = models.CharField(('Lote'),
        max_length=30,
        choices=BATCH,
        default=BATCH[0][0]
    )
    photo = models.FileField(upload_to='photos/', blank=True)
    notification = models.IntegerField(default=2, blank=True)
    stop = models.IntegerField(('Status'), choices=STATUS, default=STATUS[0][0], blank=True)

    def __str__(self):
        return self.first_name + ' ' + self.last_name

```

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Modelo producto

El modelo de Producto define la estructura de los productos ofrecidos en la tienda del gimnasio. Esto puede incluir atributos como nombre, descripción, precio y cantidad en stock. La codificación del modelo de Producto asegura que los datos relacionados con los productos se almacenen y recuperen de manera eficiente, facilitando la gestión y visualización de los artículos disponibles para los usuarios.

Figura 17. Modelo producto.

```
class Producto(models.Model):
    nombre = models.CharField(max_length=100)
    precio = models.DecimalField(max_digits=10, decimal_places=2)
    imagen = models.ImageField(upload_to='productos/', null=True, blank=True)

    def __str__(self):
        return self.nombre
```

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Controlador de miembros:

Este controlador gestionaría las acciones relacionadas con la administración de usuarios. Podría incluir la creación, edición y eliminación de usuarios, así como la asignación de roles y permisos. También podría manejar las solicitudes de restablecimiento de contraseña y otras funciones relacionadas con la gestión de cuentas de usuario.

Controlador de Notificaciones

Este controlador sería responsable de enviar notificaciones a los usuarios. Podría manejar las lógicas para generar y enviar mensajes de recordatorio sobre vencimientos de suscripción, actualizaciones de productos y otros eventos importantes para los usuarios administradores.

Controlador de Consultas

El usuario administrador puede realizar consultas sobre las suscripciones tomando en cuenta sus campos principales.

Panel de Administración de Django:

En el marco de esta documentación sobre PGG, se resalta la relevancia del Panel de Administración de Django como una herramienta esencial para la administración y gestión de la aplicación web. Este apartado se centra en la importancia de este panel y su función fundamental dentro del entorno de desarrollo Django.

Entrada a la Administración:

El Panel de Administración de Django emerge como una característica valiosa para los desarrolladores, habilitándoles el acceso a una interfaz de gestión después de haber creado un superusuario mediante el comando "createsuperuser". Una vez se ingresa en este panel, los desarrolladores tienen la capacidad de gestionar los objetos generados en los modelos de bases de datos, así como también la posibilidad de crear nuevos usuarios.

Figura 18. Administración de Django.

The screenshot displays the Django Admin interface. At the top, a dark blue header contains the text 'Administración de Django' on the left and 'BIENVENIDOS, ADMIN. VER EL SITIO / CAMBIAR CONTRASEÑA /' on the right. Below the header, the main content area is titled 'Sitio administrativo'. It features three main sections: 'ACCOUNTS' with a 'Productos' item and '+ Añadir' and 'Modificar' buttons; 'AUTENTICACIÓN Y AUTORIZACIÓN' with 'Grupos' and 'Usuarios' items, each having '+ Añadir' and 'Modificar' buttons; and 'MEMBERS' with a 'Members' item and '+ Añadir' and 'Modificar' buttons. On the right side, there is a sidebar titled 'Acciones recientes' containing a list of 'Mis acciones' with green plus icons and product names: 'PRENSA OLÍMPICA 45°', 'JACA OLÍMPICA', 'PS-HYDROBCAA (90 SERV) PASSION FRUIT', 'MUTANT MASS VAINILLA 12LB', and 'B-NOX WATERMELON 35 SERVIDAS', each followed by the word 'Producto'.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Transición Potencial:

Este recurso resulta especialmente útil en escenarios donde la aplicación no brinda de manera nativa la capacidad de gestionar los objetos de la base de datos, ya sea por designación o por la naturaleza misma de los objetos. En tales situaciones, el Panel de Administración emerge como un aliado que permite la creación, modificación y eliminación de estos objetos, otorgando a los desarrolladores el control necesario para adaptar la aplicación a las necesidades específicas del cliente.

Figura 19. Gestionar a través de Administración de Django.

Añadir producto

Nombre:	<input type="text"/>
Precio:	<input type="text"/>
Imagen:	<input type="button" value="Seleccionar archivo"/> Ninguno archivo selec.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Interfaz Versátil:

El Panel de Administración de Django exhibe una interfaz amigable y versátil, permitiendo la manipulación de diversos elementos directamente desde su ventana principal. Además, ofrece la capacidad de añadir y modificar usuarios, grupos (en caso de aplicar), así como supervisar las últimas acciones ejecutadas en calidad de administrador.

Visualización Ejemplar:

La siguiente imagen ilustra un ejemplo representativo del Panel de Administración, resaltando su funcionalidad y las opciones disponibles para los administradores:

Figura 20. Interfaz de suscripciones.

The screenshot shows the Django Admin interface for managing subscriptions. At the top, there is a navigation bar with links: 'Ingresar', 'Suscripción', 'Ver todos los miembros', 'Informes', 'Notificaciones' (with a notification icon), 'Cambiar la contraseña', 'Fondo de pantalla', and 'Cerrar sesión'. Below the navigation bar, the form is divided into two main sections. The first section is for user information, including a photo upload area with a 'Subir foto:' label and a 'Seleccionar archivo' button, followed by input fields for 'Nombres *', 'Apellidos *', 'Celular *', 'Email', 'Fecha de Nacimiento: *' (with a date picker), 'Dirección', and 'Historia Médica' (with a dropdown menu set to 'Ninguna'). A dark 'Agregar' button is located below these fields. The second section is for subscription details, featuring dropdown menus for 'Tipo de suscripción *' (set to 'Gym'), 'Período de suscripción *' (set to '1 Mes'), 'Estado de la tarifa *' (set to 'Pagado'), and 'Fecha de Registro *' (with a date picker). There is also a 'Cantidad *' field with a currency symbol '\$' and a 'Lote *' dropdown menu set to 'Mañana'.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

En esta ilustración, se aprecia cómo el panel posibilita no solo la gestión de nuevos suscriptores, sino también la manipulación de los modelos específicos creados para la aplicación. Esta visión panorámica y la capacidad de realizar acciones fundamentales desde un punto central facilita la administración y agiliza las tareas de los desarrolladores y administradores.

En conclusión, el Panel de Administración de Django es un recurso esencial en el arsenal del desarrollador, brindando un enfoque eficaz y eficiente para gestionar y adaptar la aplicación según las necesidades del cliente. La versatilidad y accesibilidad de esta herramienta resaltan su importancia como una pieza fundamental en la construcción y administración de aplicaciones web sólidas y versátiles.

Estilos

Haciendo referencia nuevamente a conceptos previamente abordados, es fundamental resaltar la faceta del diseño dentro del contexto de las aplicaciones web desarrolladas a través de Django. Para esta etapa, Django emplea una herramienta esencial denominada "templates" o plantillas. Estas plantillas son ficheros de texto que se utilizan con el propósito de definir tanto la estructura como el diseño de las aplicaciones Django, funcionando de manera análoga a las páginas codificadas en HTML.

Un aspecto distintivo de los "templates" en Django radica en su capacidad de heredar de otras plantillas, comúnmente referidas como "base". Esta práctica se utiliza con el objetivo de otorgar cohesión visual y diseño uniforme a todas las páginas de una aplicación web, brindando a los usuarios una experiencia visualmente atractiva. Para llevar a cabo esta herencia de plantillas en Django, se emplean etiquetas de plantillas.(Gudiño León. et al., 2021)

Dentro de esta representación, es posible identificar cuatro etiquetas esenciales:

- `{ % if % }`, indica a Django que muestre u oculte el contenido que se encuentra entre esta etiqueta y su contraparte `{ % endif % }`, basado en si se cumple la condición especificada en la etiqueta del "if".
- `{ % endif % }`, marca el final de la sección condicional, anulando la condición aplicada después de esta etiqueta.
- `{ % block content % }`, señala el comienzo de la plantilla que heredará de la plantilla base, garantizando que todo contenido previo a esta etiqueta sea compartido por todas las plantillas derivadas.
- `{ % endblock % }`, establece el cierre de la plantilla que hereda de la base, permitiendo que el contenido de la plantilla base resurja en las secciones subsiguientes.

Además, Django permite la creación de hojas de estilo CSS para manejar la apariencia de los elementos HTML presentes en las plantillas. Para hacer uso de los estilos definidos en las hojas de estilo CSS, únicamente se aplica la etiqueta "class" al elemento deseado, haciendo referencia al identificador asignado al estilo en la hoja CSS correspondiente.

De igual manera, es posible incorporar la biblioteca Bootstrap a Django mediante una sencilla instrucción en la línea de comandos, seguida por la inclusión de 'bootstrap4' en la lista de INSTALLED APPS en el archivo settings.py de la aplicación.

Figura 21. Aplicaciones instaladas de proyecto Django.

```
35 # Application definition
36
37 INSTALLED_APPS = [
38     'django.contrib.admin',
39     'django.contrib.auth',
40     'django.contrib.contenttypes',
41     'django.contrib.sessions',
42     'django.contrib.messages',
43     'django.contrib.staticfiles',
44     'accounts',
45     'members',
46     'notifications',
47     'reports',
48     'payments',
49 ]
```

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

La capacidad de fusionar plantillas, estilos CSS y herramientas como Bootstrap provee a los desarrolladores con un conjunto de recursos valiosos para dar forma a la apariencia visual de las aplicaciones Django, asegurando una experiencia estética y funcional satisfactoria para los usuarios finales.

Pruebas de desarrollo

A continuación, se presentan las pruebas de desarrollador que se realizaron con su respectivo código.

Figura 22. Prueba de Autenticación de usuarios.

```
def test_login_valid_user(self):
    # Crear un usuario de prueba
    User.objects.create_user(username='usuario_prueba', password='contrasena123')

    # Enviar una solicitud POST al formulario de inicio de sesión
    response = self.client.post('/login/', {'username': 'usuario_prueba', 'password': 'contrasena123'})

    # Verificar que la página de inicio se cargue correctamente después del inicio de sesión
    self.assertEqual(response.status_code, 200)
    self.assertContains(response, 'Bienvenido, usuario_prueba')

def test_login_invalid_user(self):
    # Intentar iniciar sesión con credenciales incorrectas
    response = self.client.post('/login/', {'username': 'usuario_inexistente', 'password': 'contrasena_erronea'})

    # Verificar que el usuario sea redirigido al formulario de inicio de sesión nuevamente
    self.assertEqual(response.status_code, 302)
    self.assertRedirects(response, '/login/?error=True')

def test_logout(self):
    # Iniciar sesión como un usuario
    self.client.login(username='usuario_prueba', password='contrasena123')

    # Enviar una solicitud POST para cerrar sesión
    response = self.client.post('/logout/')

    # Verificar que el usuario sea redirigido a la página de inicio
    self.assertEqual(response.status_code, 302)
    self.assertRedirects(response, '/')
```

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Figura 23. Prueba de creación de productos.

```
def test_create_product_admin(self):
    # Iniciar sesión como un usuario administrador
    self.client.login(username='admin', password='contrasena_admin')

    # Enviar una solicitud POST para crear un producto
    response = self.client.post('/create_product/', {'name': 'Producto de Prueba', 'price': 10.0})

    # Verificar que el producto se haya creado correctamente
    self.assertEqual(response.status_code, 302)

    # Acceder a la página de la tienda para verificar la creación del producto
    response = self.client.get('/store/')
    self.assertContains(response, 'Producto de Prueba')
```

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Pruebas de despliegue

Después de completar la fase de desarrollo de la aplicación, se han llevado a cabo una serie de evaluaciones exhaustivas para garantizar que tanto el rendimiento como la usabilidad de nuestra aplicación web sean adecuados.

Mediante estas evaluaciones, hemos verificado que la aplicación web funciona de manera correcta en todas sus funcionalidades y que los tiempos de carga se ajustan a las expectativas previstas. A continuación, se proporcionará una descripción detallada de algunas de las evaluaciones realizadas en el contexto de la aplicación.

Autenticación

La evaluación de autenticación para los usuarios con rol de administrador ha cumplido con los criterios y requisitos establecidos. El proceso de inicio de sesión y autenticación ha demostrado ser efectivo y seguro, permitiendo que solo los usuarios con privilegios de administrador accedan a las funciones de gestión de la aplicación. Las pruebas realizadas han confirmado que el sistema de autenticación funciona de manera confiable y proporciona el nivel adecuado de seguridad para proteger las áreas sensibles de la aplicación.

Figura 24. Prueba de despliegue.



Ingresar

Nombre de usuario

Contraseña

Ingresar

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Ingreso de suscriptores

La evaluación de la prueba de ingreso de suscriptores a la base de datos ha sido satisfactoria, ya que se verificó que el proceso de registro de suscriptores se realiza de manera exitosa. Los usuarios interesados en inscribirse en actividades y recibir notificaciones pudieron completar el proceso sin dificultades. Las pruebas demostraron que los datos ingresados por los suscriptores se almacenan correctamente en la base de datos y están disponibles para su posterior utilización. Esta funcionalidad ha sido validada y cumple con los objetivos de brindar a los usuarios una forma efectiva de registrarse y participar en las actividades ofrecidas por el centro deportivo.

Figura 25. Prueba de Despliegue.

The screenshot shows a registration form with the following fields and values:

- Nombres ***: Juan
- Apellidos ***: Perez
- Celular ***: 0987654321
- Email**: juanperez@mail.com
- Fecha de Nacimiento: ***: 01/01/2000
- Dirección**: Loja, calles ...
- Tipo de suscripción ***: Cross Fit
- Estado de la tarifa ***: Pagado
- Cantidad ***: \$ 25
- Lote ***: Mañana

A calendar widget is open, showing the date 10/08/2023.

Autor: Alvaro Enríquez

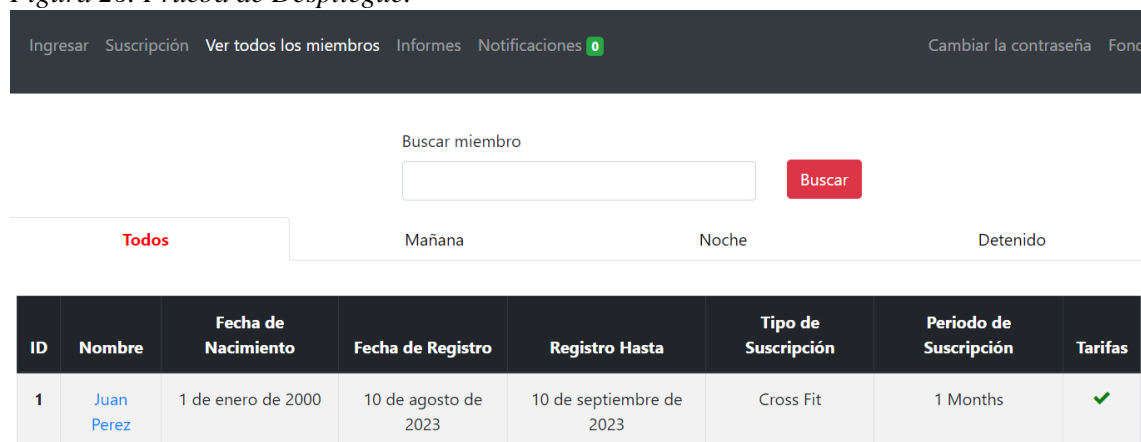
Fuente: Del sistema aplicado

Listar suscriptores

La evaluación de la prueba de listar suscriptores arrojó resultados positivos, demostrando que la funcionalidad de mostrar la lista de suscriptores se ejecuta de manera eficaz. La aplicación logra recuperar y presentar la información de los suscriptores almacenados en la base de datos de manera ordenada y legible. Los usuarios con el rol de administrador pueden acceder a esta lista y visualizar los detalles de los suscriptores inscritos en las diferentes

actividades. Esta prueba confirma que la aplicación brinda una visión clara y concisa de la información de los suscriptores, cumpliendo con el propósito de facilitar la gestión de la base de datos y proporcionar un panorama general de los usuarios registrados en el centro deportivo.

Figura 26. Prueba de Despliegue.



The screenshot shows a navigation bar with links: Ingresar, Suscripción, Ver todos los miembros, Informes, Notificaciones 0, Cambiar la contraseña, and Fondos. Below the navigation bar is a search section with the text 'Buscar miembro', an input field, and a red 'Buscar' button. Underneath are filter tabs: 'Todos' (selected), 'Mañana', 'Noche', and 'Detenido'. The main content is a table with the following data:

ID	Nombre	Fecha de Nacimiento	Fecha de Registro	Registro Hasta	Tipo de Suscripción	Periodo de Suscripción	Tarifas
1	Juan Perez	1 de enero de 2000	10 de agosto de 2023	10 de septiembre de 2023	Cross Fit	1 Months	✓

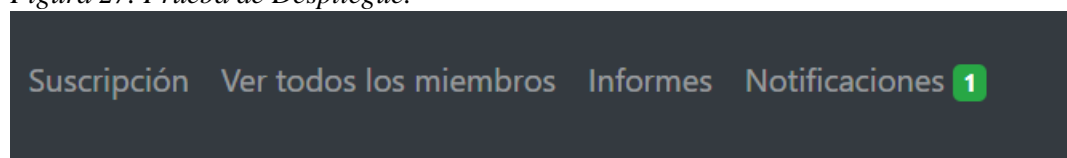
Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Notificaciones

La prueba destinada a evaluar las notificaciones por día y por cuotas vencidas arrojó resultados satisfactorios. Durante el proceso de evaluación, se confirmó que la aplicación es capaz de identificar de manera precisa los suscriptores cuyas cuotas han vencido y aquellos que deben ser notificados por eventos diarios.

Figura 27. Prueba de Despliegue.



Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Esta prueba refleja que la aplicación es efectiva en el proceso de seguimiento y notificación de suscriptores con cuotas vencidas y en la difusión de información diaria. Esto

garantiza una interacción proactiva con los suscriptores, manteniéndolos informados sobre sus obligaciones y las novedades del centro deportivo, contribuyendo así a una gestión más efectiva y a una experiencia positiva para los usuarios.

Figura 28. Prueba de Despliegue.

Nombre	Nacimiento	Fecha de Registro	Fin de registro	Tipo de suscripción	Editar	Eliminar notificación
Usuario Cupo vencido	19 de febrero de 1998	19 de julio de 2023	19 de agosto de 2023	Gym	Editar	Eliminar

Asadas

Nombre	Nacimiento	Fecha de Registro	Fin de Registro	Tipo de Suscripción	Mes Pendiente	Editar	Eliminar Notificación
Usuario Cupo vencido	19 de febrero de 1998	19 de julio de 2023	19 de agosto de 2023	Gym	1	Editar	Eliminar

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Crear productos

La evaluación de la funcionalidad de crear productos en la aplicación arrojó resultados exitosos. Durante el proceso de prueba, se confirmó que los administradores pueden crear nuevos productos en la tienda del gimnasio de manera efectiva y eficiente. La interfaz proporciona una experiencia intuitiva y simplificada, permitiendo a los administradores ingresar la información necesaria, como descripción, precio y la imagen.

Figura 29. Prueba de Despliegue.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

Además, se constató que la inclusión de imágenes de los productos es igualmente fluida, mejorando la presentación visual de los artículos en la tienda. Esta funcionalidad garantiza la actualización y diversidad de productos disponibles para los usuarios.

Visualización de Productos y Actividades

La evaluación de la funcionalidad de visualización de productos reveló un cumplimiento satisfactorio de los objetivos establecidos. Durante las pruebas, se comprobó que los usuarios, al acceder a la sección de productos en la aplicación, pueden ver de manera clara y ordenada la lista de artículos disponibles en la tienda del gimnasio.

Figura 30. Prueba de Despliegue.



Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

La interfaz de visualización proporciona información esencial de cada producto, como su descripción y precio. Además, la inclusión de imágenes permite a los usuarios tener una idea precisa de cómo luce cada artículo. La estructura de la página y la disposición de los elementos contribuyen a una experiencia de usuario intuitiva y agradable.

DISCUSIÓN

En el contexto del desarrollo de una aplicación web dirigida a un centro deportivo, este segmento se enfoca en la evaluación de los descubrimientos obtenidos a lo largo de la investigación. El propósito esencial de esta evaluación es analizar cómo los objetivos iniciales del proyecto han sido tratados y resueltos a través de la implementación de soluciones concretas. Estos objetivos se centran en la creación de una plataforma de administración eficiente, la simplificación de la divulgación de información sobre actividades y productos, la incorporación de un sistema de comunicación bidireccional y la instauración de un mecanismo de notificación.

Cada uno de estos objetivos abordó desafíos particulares que afectaban la interacción entre el centro deportivo y sus usuarios. La ausencia de una plataforma integral para la administración, las dificultades en la comunicación con los usuarios, la carencia de un método efectivo para compartir información y la carencia de un sistema de notificación eran obstáculos que necesitaban soluciones prácticas y sólidas.

En esta evaluación, se analizarán minuciosamente los resultados obtenidos en relación con cada objetivo y se contextualizarán en las circunstancias problemáticas previas a su implementación. Igualmente, se examinará cómo se ha afrontado cada desafío, destacando las estrategias, enfoques y herramientas empleadas para alcanzar soluciones eficaces. Mediante este análisis, se podrá comprender la influencia y la efectividad de las soluciones aplicadas en el entorno de la aplicación del centro deportivo.

En última instancia, esta evaluación no solo proporcionará una visión global de cómo se han logrado los objetivos, sino también cómo estas soluciones han transformado la experiencia tanto del propietario como de los usuarios de la aplicación. El segmento explorará en profundidad cómo se ha logrado una gestión administrativa más eficiente, una comunicación

mejorada, una distribución de información eficaz y un sistema de notificación oportuno. De esta manera, se resalta cómo la aplicación ha evolucionado desde su etapa inicial hasta convertirse en una herramienta valiosa y funcional para el centro deportivo y sus usuarios.

Hallazgo 1: Crear una interfaz de administración eficiente:

Antes de obtener los resultados, el propietario del centro deportivo enfrentaba dificultades en la administración de actividades y productos debido a la falta de una plataforma eficiente. No tenían una herramienta centralizada para agregar, modificar o eliminar entradas, lo que limitaba su capacidad para adaptarse rápidamente a los cambios y actualizaciones.

Los resultados de la investigación destacan que se ha logrado implementar una interfaz de administración sólida y eficiente. Se ha empleado el patrón de diseño MVC para estructurar la aplicación, lo que permite una separación clara entre la interfaz del usuario, la lógica de negocio y los datos. Se identificó que el panel de administración de Django desempeña un papel fundamental en esta área, permitiendo al propietario gestionar actividades y productos de manera efectiva. Se han presentado ejemplos de la creación, modificación y eliminación de entradas, lo que cumple con la necesidad de una administración flexible.

Hallazgo 2: Facilitar la difusión de información:

Antes de los resultados obtenidos, el centro deportivo carecía de una forma efectiva de comunicar las ofertas actuales de actividades y productos a los usuarios. La presentación de la información carecía de cohesión visual y atractivo, lo que dificultaba la comprensión y el interés por parte de los usuarios.

La investigación revela que la aplicación desarrollada cumple con éxito el objetivo de difundir información sobre actividades y productos del centro deportivo. Los "templates" de Django permiten definir la estructura y el diseño de las páginas, lo que brinda una presentación atractiva de la información. La herencia de plantillas, en conjunto con la integración de hojas de estilo

CSS y la posibilidad de utilizar Bootstrap, proporcionan una experiencia visualmente agradable para los usuarios. Esto asegura que las ofertas actuales del centro deportivo sean comunicadas de manera efectiva y accesible.

Hallazgo 3: Implementación de un Sistema de Control Integral:

Previo a la implementación del sistema de control, la gestión de suscriptores y productos en el centro deportivo carecía de una estructura eficaz. No existía un mecanismo adecuado para monitorear y supervisar las inscripciones a las actividades, la duración de las membresías y los pagos correspondientes.

La investigación destaca que se ha logrado implementar con éxito un sistema de control integral. Mediante el uso de privilegios de administrador en la aplicación, se ha habilitado a los responsables del centro deportivo para gestionar y supervisar detalladamente las inscripciones de los usuarios a las actividades, así como definir la duración de las membresías y tarifas asociadas. El enfoque basado en roles proporcionado por la estructura MVC ha permitido establecer una jerarquía de acceso, otorgando a los administradores las herramientas necesarias para una gestión efectiva.

La implementación de este sistema de control integral asegura una administración más precisa y organizada de las actividades y membresías, mejorando la eficiencia y la calidad del servicio ofrecido por el centro deportivo.

Hallazgo 4: Establecer un mecanismo de notificaciones:

Antes de obtener los resultados, los usuarios registrados carecían de una forma efectiva de recibir notificaciones sobre novedades y actualizaciones relevantes del centro deportivo. La falta de un sistema de notificaciones dificultaba mantener a los usuarios informados de manera oportuna.

La investigación señala que se ha logrado establecer un mecanismo de notificaciones efectivo. Se ha identificado que Django ofrece la posibilidad de enviar notificaciones a los usuarios registrados sobre novedades, actualizaciones y eventos relevantes del centro deportivo. Estas notificaciones contribuyen a mantener a los usuarios informados de manera oportuna y eficiente. La incorporación de la biblioteca Bootstrap en la aplicación permite una presentación visual atractiva de las notificaciones, lo que asegura que los usuarios estén al tanto de la información relevante.

CONCLUSIONES

Se permite con la aplicación alcanzar los propósitos estipulados para el desarrollo de una plataforma web orientada a un centro deportivo. A través de un enfoque metódico y una ejecución precisa, se han obtenido resultados notables que generan un impacto positivo en la gestión, la comunicación y la interacción entre el centro y sus usuarios.

Se creó un entorno de administración que solventa de manera eficiente la gestión de usuarios y productos del centro deportivo. Se analizó minuciosamente la situación actual de la empresa para dar un trato adecuado a los requerimientos, diseñando e implementando las soluciones a todas las limitaciones en la gestión de procesos observados.

La implementación de un sistema de control integral en el centro deportivo ha demostrado ser un paso crucial hacia la eficaz gestión y administración de las actividades y productos ofrecidos. La introducción de privilegios de administrador y la asignación de roles en la aplicación ha permitido una supervisión detallada de las inscripciones de usuarios a las actividades, así como el seguimiento de la duración de las membresías y los pagos correspondientes. Esta estructura jerárquica de acceso, respaldada por la arquitectura MVC, ha proporcionado a los administradores las herramientas necesarias para tomar decisiones informadas y llevar a cabo una gestión efectiva.

Finalmente, la instauración de un mecanismo de notificaciones ha contribuido a mantener a los usuarios debidamente informados. Antes de la implementación, la carencia de comunicación acerca de novedades y actualizaciones generaba desconexión. Ahora, los usuarios registrados son notificados en tiempo real, lo que fortalece su sentido de pertenencia y su compromiso con el centro deportivo.

En conjunto, estos logros reflejan el impacto positivo y la transformación que la aplicación ha generado en el centro deportivo y su comunidad de usuarios. La atención meticulosa a los objetivos originales y la implementación efectiva de soluciones han culminado en una herramienta sumamente valiosa que cumple de manera sobresaliente su función de optimizar la administración, la comunicación y la satisfacción en el ámbito deportivo.

RECOMENDACIONES

Estas sugerencias son esenciales para preservar la excelencia y el constante progreso exitoso de la aplicación, asegurando una vivencia favorable tanto para los usuarios como para los administradores involucrados.

- **Mantenimiento y Actualización Constante:** Asegurarse de mantener la plataforma actualizada con las últimas versiones de Django y componentes relacionados. Esto garantizará un rendimiento óptimo, correcciones de seguridad y la adopción de nuevas características.
- **Capacitación Integral para Administradores:** Proporcionar una formación completa tanto a los usuarios finales como a los administradores encargados de la gestión de la plataforma. Un conocimiento profundo de las funcionalidades mejorará su experiencia y eficiencia.
- **Monitoreo y Respaldos Periódicos:** Establecer un sistema de monitoreo regular para identificar y abordar problemas potenciales a tiempo, asegurando un funcionamiento sin interrupciones. Además, implementar un protocolo de respaldo de datos para prevenir la pérdida de información crítica.
- **Enfoque en la Seguridad y Protección de Datos:** Priorizar la seguridad implementando medidas robustas de protección de datos y privacidad de usuarios. Permanecer actualizado en las mejores prácticas de ciberseguridad y realizar auditorías regulares para detectar vulnerabilidades.

REFERENCIAS BIBLIOGRÁFICAS

- Aliaga Melo, G. D. (2012). Implementación y Metodología para la elaboración de modelos BIM para su aplicación en proyectos industriales multidisciplinarios. *Universidad De Chile*, 77. <http://repositorio.uchile.cl/handle/2250/112356>
- Carlos Billy Reynoso. (2004). Introducción a la Arquitectura de Software. *Universidad de Buenos Aires*, 33. https://www.fceia.unr.edu.ar/~mcristia/Introduccion_a_la_Arquitectura_de_Software.pdf
- Crespo, M. L. (2014). *EN INSTRUMENTACIÓN* (Issue December). <https://doi.org/10.18845/mct.v19i1.2014>
- Cristina, I., & Buriticá, R. (2015). *Sistema de Información como apoyo al proceso de evaluación Docente en la Universidad del Valle sede Tuluá Sistema de Información como apoyo al proceso de evaluación Docente en la Universidad del Valle sede Tuluá.*
- Culhuacan, U., Alberto, E. J., Iovanna, D., & Rodríguez, A. (2020). *Instituto politécnico nacional.*
- De Ingeniería, E., Electrónica, E. N., Redes, Y., De, I., Prototipo, U. N., Gafas, D. E., & Bastón, Y. (2019). *Escuela Superior Politécnica De Chimborazo Facultad De Informática Y Electrónica.* <http://dspace.esPOCH.edu.ec/handle/123456789/11018>
- Estrategias para el desarrollo de videojuegos libres para la educación formal de Colombia.* (n.d.). <http://www.freesound.org/people/Ryding/sounds/125968/>
- Gudiño León., A. R., Acuña López., R. J., & Terán Torres., V. G. (2021). *No 10 Title.* 6.
- León Soberón, J. J. (2020). Análisis comparativo de sistemas gestores de bases de datos postgresql y mysql en procesos crud. *Repositorio Institucional - USS*, 128. <http://repositorio.uss.edu.pe/handle/20.500.12802/7012>

- López Herrera, P. (2016). Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL. *Universidad Tecnológica Del Sur Del Estado de México*, 72.
<http://ri.uaemex.mx/handle/20.500.11799/62548><http://hdl.handle.net/20.500.11799/62548><http://ri.uaemex.mx/bitstream/handle/20.500.11799/62548/TesisPatriciaLopezHerrera.pdf?sequence=3>
- Mahapsari. (2013). No Title66, עלון הנוטע: תמונת מצב (1997), 39–37.
- Maldonado, J. (2018). *FUNDAMENTOS DE CALIDAD TOTAL José Ángel Maldonado*. 421.
- Merangin, D. I. D., Pattiselanno, F., Mentansan, G., Nijman, V., Nekaris, K. A. I., Pratiwi, A. I. N., Studi, P., Nutrisi, I., Makanan, D. A. N., Peternakan, F., Penulisan, P., Ilmiah, K., Berbagai, P., Cahaya, I., Lapangan, D. I., Eropa, A., Geometry, R., Analysis, G., Nasution, R. D., ... Bismark, M. (2018). No Title. 2(2), 2016.
<https://doi.org/10.1016/j.gecco.2019.e00539><https://doi.org/10.1016/j.foreco.2018.06.029>[http://www.cpsg.org/sites/cbsg.org/files/documents/Sunda Pangolin National Conservation Strategy and Action Plan%28LoRes%29.pdf](http://www.cpsg.org/sites/cbsg.org/files/documents/Sunda_Pangolin_National_Conservation_Strategy_and_Action_Plan%28LoRes%29.pdf)<https://doi.org/10.1016/j.forec>
- Pizarro Castro, D. (2014). *Daniela pizarro castro*.
- Rahmadi Islam. (2018). No Title. (3), 1–13. <http://dx.doi.org/10.1186/s13662-017-1121-6><https://doi.org/10.1007/s41980-018-0101-6><https://doi.org/10.1016/j.cnsns.2018.04.019><https://doi.org/10.1016/j.cam.2017.10.014><http://dx.doi.org/10.1016/j.apm.2011.07.041><http://arxiv.org/abs/1502.020>
- Severo Malaspina, M., & Tonarelli Alvarez, S. P. (2021). *Tecnologías de software para interfaz de usuario adaptativa y multiplataforma*. <http://hdl.handle.net/20.500.11968/4497>

Unzurrunzaga, C. (2013). *Propuesta de aplicación del modelo FRAD para el control de nombres de personas y de entidades corporativas en el contexto de la Facultad de Humanidades*. <http://www.memoria.fahce.unlp.edu.ar/library?a=d&c=tesis&d=Jte881>

Vásquez Rudas, J. F. (2014). Diseño de un sistema basado en tecnología web para el control y gestión de venta de unidades móviles. *Alicia Concytec*, 1–120. http://repositorio.uncp.edu.pe/bitstream/handle/UNCP/1153/DISEÑO_DE_UN_SISTEMA_BASADO_EN_TECNOLOGÍA_WEB.pdf?sequence=1

Django Software Foundation. (2021). Django - The Web framework for perfectionists with deadlines. <https://www.djangoproject.com/>

Leff, A., & Rayfield, J. T. (2001). Web-application development using the model/view/controller design pattern. *ACM SIGPLAN Notices*, 36(7), 16-21.

Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley Professional.

Martin, R. C., & Odell, J. J. (1998). *Object-oriented methods: A foundation, UML edition*. Prentice Hall.

Python Software Foundation. (2021). Welcome to Python.org. <https://www.python.org/>

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional.

Django Software Foundation. (2021). The Django admin site. <https://docs.djangoproject.com/en/3.2/ref/contrib/admin/>

Django Software Foundation. (2021). Django templates. <https://docs.djangoproject.com/en/3.2/topics/templates/>

Bootstrap. (2021). Get started. <https://getbootstrap.com/docs/5.1/getting-started/introduction/>

Django Software Foundation. (2021). Channels — Channels 3.0.3 documentation.
<https://channels.readthedocs.io/en/stable/>

Django Software Foundation. (2021). Optimization.
<https://docs.djangoproject.com/en/3.2/topics/db/optimization/>

Django Software Foundation. (2021). Django security.
<https://docs.djangoproject.com/en/3.2/topics/security/>

Django REST framework. (2021). Django REST framework. <https://www.django-rest-framework.org/>

Django Software Foundation. (2021). Testing.
<https://docs.djangoproject.com/en/3.2/topics/testing/>

Django Software Foundation. (2021). Scalability and Performance.
<https://docs.djangoproject.com/en/3.2/topics/cache/>

Nielsen, J. (1993). Usability Engineering. Academic Press.

Sommerville, I. (2011). Software Engineering (9th ed.). Pearson.

Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice (3rd ed.). Addison-Wesley Professional.

ANEXOS

ANEXO N° 1

Figura 31. Anexo.

The screenshot displays the Gym App interface. At the top, the logo 'Gym App' is visible, along with navigation links: 'Ingresar', 'Suscripción', 'Ver todos los miembros', 'Informes', 'Notificaciones 1', 'Cambiar la contraseña', 'Fondo de pantalla', and 'Cerrar sesión'. The main banner features a dramatic image of a barbell with a quote in Polish: 'Cena sukcesu to ciężka praca, poświęcenie i determinacja, by niezależnie od tego czy w danej chwili wygrywamy czy przegrywamy dawać z siebie wszystko.' attributed to '(Vince Lombardi)'. Below the banner, there are four small images illustrating gym activities. The 'Productos disponibles' section lists four items:

Producto	Precio
BPI- ISO HD VAINILLA COOKIES 5.2 LBS	\$120,00
B-NOX WATERMELON 35 SERVIDAS	\$60,00
MUTANT MASS VAINILLA 12LB	\$95,45
PS-HYDROBCAA (90 SERV) PASSION FRUIT	\$78,65

Below the product list, there are four images of gym equipment: a bench press, a leg press, a stationary bike, and a pair of sunglasses.

Autor: Alvaro Enríquez

Fuente: Del sistema aplicado

ANEXO N° 2

Ficha informativa de la Empresa

DomainGym Pindal - Tu Casa del Fitness en Pindal, Loja, Ecuador
Descripción General:
DomainGym Pindal es más que un gimnasio; es tu casa para alcanzar tus metas de acondicionamiento físico y bienestar en Pindal, Loja, Ecuador. Nuestro compromiso es brindarte una experiencia de entrenamiento excepcional en un ambiente acogedor y amigable.
Horario de Atención:
Días de la Semana: Lunes a Viernes
Horario: 6:00 AM - 10:30 PM
Ubicación:
Dirección: Barrio La Pampa, junto a la gasolinera, Pindal, Loja, Ecuador.
Instalaciones y Servicios:
Amplia variedad de equipos de entrenamiento de última generación.
Entrenadores personales altamente calificados.
Clases de grupo, incluyendo bailo terapia, spinning y más.
Área de pesas y entrenamiento de fuerza.
Vestuarios y duchas limpias y modernas.
Espacio de estacionamiento conveniente.
Nuestra Filosofía:
En DomainGym Pindal, creemos en el poder de la actividad física para transformar vidas. Nuestra misión es proporcionarte las herramientas y el apoyo necesarios para lograr tus objetivos de salud y bienestar. Ya sea que estés buscando perder peso, aumentar la fuerza, mejorar la flexibilidad o simplemente mantenerte activo, estamos aquí para ayudarte a alcanzar tu mejor versión.
Compromiso con la Comunidad:
Somos una parte activa de la comunidad de Pindal y nos enorgullece apoyar iniciativas locales de bienestar y salud. Colaboramos con organizaciones locales para promover estilos de vida activos y saludables.
Contacto:
Teléfono: 0992074487
Correo Electrónico: alejoperez32@hotmail.com
Nombre del propietario: GUSTAVO PEREZ GONZALES

ANEXO N° 3

Entrevista de involucrados

1. ¿Cuáles son los principales desafíos que enfrenta actualmente el centro deportivo en términos de gestión de actividades y comunicación con los usuarios?
2. ¿Qué funcionalidades considera esenciales para una aplicación que aborde estos desafíos y mejore la interacción con los usuarios?
3. ¿Qué tipo de interacción espera tener con la aplicación como administrador y cómo cree que esta interacción podría ser más eficiente?
4. ¿Qué información considera vital para registrar sobre los suscriptores? ¿Cómo imagina que debería ser el proceso de registro de nuevos usuarios?
5. En términos de seguridad y acceso, ¿cómo visualiza la autenticación de usuarios y la jerarquía de permisos para administradores y usuarios regulares?
6. ¿Existe algún proceso interno en el centro deportivo que requiera notificaciones y alertas? ¿Qué tipo de notificaciones serían útiles para usted como administrador?
7. ¿Cómo espera que los usuarios puedan consultar las actividades y los productos disponibles en el centro deportivo? ¿Tiene alguna preferencia sobre cómo deberían acceder a esta información?
8. En relación con la gestión de productos en la tienda, ¿qué acciones considera que deberían ser posibles para los administradores? ¿Y para los usuarios regulares?
9. ¿Qué características considera importantes para facilitar la administración y el mantenimiento de la aplicación a largo plazo?
10. ¿Hay alguna otra funcionalidad o requisito que considere crucial para el éxito de esta aplicación, pero que no hayamos discutido hasta ahora?

ANEXO N° 4

Análisis de entrevista

- La autenticación es un aspecto crítico para nosotros. Queremos asegurarnos de que los usuarios puedan acceder a la aplicación de manera segura y que solo aquellos con los permisos adecuados puedan realizar acciones administrativas. Por eso, es necesario que los usuarios puedan autenticarse con su nombre de usuario y contraseña.
- Queremos que los usuarios registrados puedan iniciar sesión con su nombre de usuario y contraseña, lo que les dará acceso a sus perfiles y a las funciones relacionadas con su suscripción. Además, los usuarios administradores también deberán autenticarse para acceder a las funciones de administración, como la adición, modificación y eliminación de productos y suscriptores.
- Para nosotros es esencial tener un registro centralizado de los suscriptores y sus detalles. Los administradores deberían poder agregar nuevos usuarios, completando los campos de información requeridos. Esto incluye nombres, apellidos, fecha de nacimiento, información de contacto, período de suscripción, detalles de pago y otros datos relevantes. Además, los administradores deben poder modificar y eliminar registros según sea necesario.
- Las consultas son clave para nuestros usuarios. Queremos que puedan ver fácilmente las actividades que ofrecemos y los productos disponibles en nuestra tienda. Esto les permitirá tomar decisiones informadas y obtener información actualizada sobre nuestras ofertas. La consulta de actividades debe ser accesible para todos los usuarios, mientras que la consulta de productos será importante para usuarios interesados en la tienda.
- Las notificaciones son esenciales para mantenernos al tanto de las acciones y cambios importantes. Queremos que los administradores reciban alertas cuando haya culminaciones de suscripciones, para que puedan tomar medidas apropiadas. Esto nos

ayudará a mantener un flujo de trabajo eficiente y a asegurarnos de que estamos brindando un servicio de alta calidad a nuestros suscriptores.

ANEXO N° 5

Imágenes

Figura 32. Anexo.



Autor: Alvaro Enríquez

Fuente: Anexo

Figura 33. Anexo.



Autor: Alvaro Enríquez

Fuente: Anexo

ANEXO N° 6

Manual de Administración

Bienvenido al Manual de Usuario del Sistema de Gestión de Gimnasio. Este manual proporcionará instrucciones detalladas sobre cómo utilizar las diversas funcionalidades de la aplicación para gestionar las actividades, los productos y los miembros de su gimnasio de manera eficiente. A continuación, se presentarán las principales características y cómo interactuar con ellas.

Índice:

1. Inicio de Sesión
2. Consulta de Actividades
3. Consulta de Productos en la Tienda
4. Gestión de Usuarios
 - a. Registro de Nuevos Usuarios (Administradores)
 - b. Modificación de Usuarios (Administradores)
 - c. Eliminación de Usuarios (Administradores)
5. Gestión de Productos
 - a. Agregar Nuevos Productos (Administradores)
 - b. Modificar Productos (Administradores)
 - c. Eliminar Productos (Administradores)
6. Gestión de Miembros
 - a. Crear Nuevas Suscripciones (Administradores)
 - b. Modificar Suscripciones (Administradores)
 - c. Cancelar Suscripciones (Administradores)
7. Notificaciones
 - a. Visualización de Notificaciones (Administradores)
8. Cerrar Sesión

1. Inicio de Sesión:

Para acceder al sistema, inicie sesión con su nombre de usuario y contraseña proporcionados por el administrador. Haga clic en el botón "Ingresar" para acceder a la aplicación.

[Ingresar](#) [Suscripción](#) [Ver todos los miembros](#) [Informes](#) [Notificaciones](#)

Ingresar

Nombre de usuario

Contraseña

Ingresar

2. Consulta de Actividades:

En la sección "Actividades", podrá consultar las actividades actuales ofrecidas por el gimnasio. Simplemente accediendo a la página principal y se mostrará un contenedor de información acompañada de imágenes de las actividades disponibles.



3. Consulta de Productos en la Tienda:

Diríjase a la sección "Tienda" para ver los productos disponibles. Visualice el apartado de "Productos disponibles" y se mostrará una lista con los productos, incluyendo su nombre y precio, para posteriormente ser adquiridos en el centro deportivo

4. Gestión de Usuarios:

Los usuarios administradores pueden realizar las siguientes acciones ingresando a

http://nombre_de_dominio/admin/auth/user/

Registrar Nuevos Usuarios: Agregue nuevos usuarios al sistema, incluyendo empleados y miembros del gimnasio.

AUTENTICACIÓN Y AUTORIZACIÓN	
Grupos	+ Añadir ✎ Modificar
Usuarios	+ Añadir ✎ Modificar

Modificar Usuarios: Actualice la información de los usuarios, como sus datos de contacto o roles.

Seleccione usuario a modificar

Q

Acción: Ir seleccionados 0 de 1

<input type="checkbox"/>	NOMBRE DE USUARIO	DIRECCIÓN DE CORREO ELECTRÓNICO	NOMBRE	APELLIDOS	ES STAFF
<input type="checkbox"/>	admin	admin@dmin.com			<input checked="" type="checkbox"/>

1 usuario

Eliminar Usuarios: Elimine usuarios que ya no sean necesarios en el sistema.

Acción: 1 de 1 seleccionado

5. Gestión de Productos:

Los usuarios administradores pueden:

http://nombre_de_dominio/admin/acounts/producto/

Agregar Nuevos Productos: Añada nuevos productos a la tienda, incluyendo su nombre, imagen y precio.

Productos

[+ Añadir](#) [✎ Modificar](#)

Modificar Productos: Edite la información de los productos existentes, como su nombre o precio.

Seleccione producto a modificar

Acción: seleccionados 0 de 8

- PRODUCTO
- CINTURON LEV. PESAS
- BIKE AIR
- PRENSA OLÍMPICA 45°

Eliminar Productos: Elimine productos que ya no estén disponibles en la tienda.

Seleccione producto a modificar

Acción: 1 de 8 seleccionado

- PRODUCTO
- CINTURON LEV. PESAS
- BIKE AIR
- PRENSA OLÍMPICA 45°

6. Gestión de Miembros:

Los administradores pueden:

[Ingresar](#) [Suscripción](#) [Ver todos los miembros](#) [Informes](#)

Crear Suscripción: Agregue nuevos miembros y configure sus suscripciones, incluyendo duración y costo.

Subir foto:
Elegir archivo No se ...rchivo

Fecha de Nacimiento: *
dd/mm/aaaa

Agregar

Nombres *
Apellidos *

Celular *
Email

Dirección
Historia Medica
Ninguna

Tipo de suscripción *
Gym

Periodo de suscripción *
1 Mes


Cantidad *
\$

Estado de la tarifa *
Pagado

Fecha de Registro *
dd/mm/aaaa

Lote *
Mañana

Modificar Suscripciones: Edite las suscripciones existentes, ajustando su duración o costo.


Juan Perez
Miembro desde:
19 de agosto de 2023
fecha de nacimiento: 1 de enero de 2000
Celular: 0987654321
Loja , calles ...

INFORMACIÓN DE MEMBRESÍA: [Atrás](#)

Tipo de suscripción: Cross Fit
Periodo de suscripción: 1 Mes
Fecha de Registro: dd/mm/aaaa
Inscripción hasta: dd/mm/aaaa
Cantidad: 25
Tarifa: Pagado
Lote: Mañana
Status: Comenzar

PERSONAL INFORMATION:

Nombres: Juan
Apellidos: Perez
Nacimiento: dd/mm/aaaa
Cargar Foto: Elegir archivo N...o

Actualizar **Borrar cuenta**
Actualizar Inf **Exportar Datos**

Detalles del pago

Fecha de pago	Pagado	Cantidad pagada
10 de agosto de 2023	1 Meses	25

Cancelar Suscripciones: Termine las suscripciones de los miembros.

2	Usuario Cupo vencido	19 de febrero de 1998	19 de julio de 2023	19 de agosto de 2023	Gym	1 Months	✖
---	--------------------------------------	-----------------------	---------------------	----------------------	-----	----------	---

7. Notificaciones:

Notificaciones 1

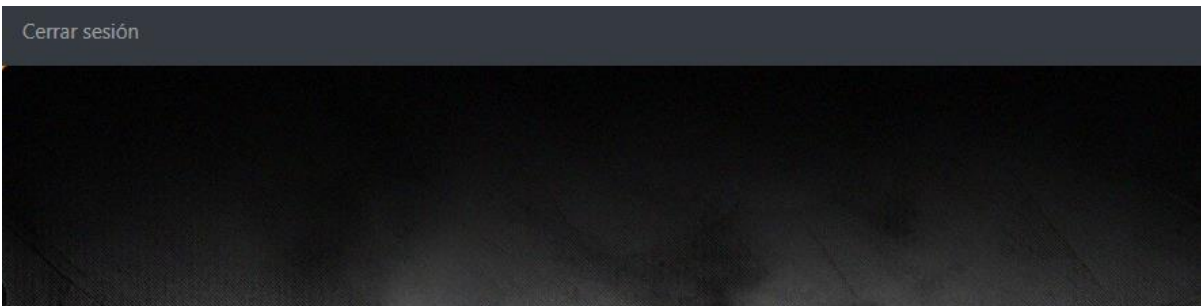
Los administradores pueden recibir notificaciones sobre eventos importantes, como vencimientos de suscripciones. Estas notificaciones se mostrarán en la bandeja de entrada.

Cuotas pasadas

Foto de Perfil	Nombre	Nacimiento	Fecha de Registro	Fin de Registro	Tipo de Suscripción	Mes Pendiente	Editar	Eliminar Notificación
	Usuario Cupo vencido	19 de febrero de 1998	19 de julio de 2023	19 de agosto de 2023	Gym	1	Editar	Eliminar

8. Cerrar Sesión:

Cuando haya terminado de utilizar el sistema, asegúrese de cerrar sesión haciendo clic en "Cerrar Sesión".



Manual del Desarrollador - Sistema de Gestión de Gimnasio

Este manual del desarrollador proporciona una guía completa para comprender la arquitectura, estructura y desarrollo del Sistema de Gestión de Gimnasio. Aquí encontrarás información sobre la configuración, las tecnologías utilizadas, la estructura del código y cómo contribuir al proyecto.

Contenido

1. Requisitos de Sistema.....	6
2. Configuración del Entorno de Desarrollo.....	6
3. Estructura del Proyecto.....	7
4. Tecnologías Utilizadas.....	8
5. Modelos de Datos.....	8
6. Vistas y Controladores.....	8
7. Plantillas y Diseño Frontend.....	8
8. Depuración y Pruebas.....	8
10. Mantenimiento y Escalabilidad.....	9
11. Recursos Adicionales.....	9

1. Requisitos de Sistema

Antes de comenzar a trabajar en el proyecto, asegúrate de tener instaladas las siguientes herramientas y tecnologías:

- Python 3.9 o superior
- Django 3.2 o superior
- Un entorno virtual (recomendado)
- Git
- Un editor de código (Ejemplo: Visual Studio Code)

2. Configuración del Entorno de Desarrollo

Para configurar tu entorno de desarrollo local, sigue estos pasos:

Clona el repositorio desde GitHub:

```
git clone https://github.com/nombre_de_usuario/gimnasio-app.git
```

Crea un entorno virtual (opcional pero recomendado):

```
python3 -m venv venv
```

Activa el entorno virtual:

En Windows:

```
venv\Scripts\activate
```

En macOS y Linux:

```
source venv/bin/activate
```

Instala las dependencias del proyecto:

```
pip install -r requirements.txt
```

Aplica las migraciones de la base de datos:

```
python manage.py migrate
```

Crea un superusuario para acceder al panel de administración:

```
python manage.py createsuperuser
```

Inicia el servidor de desarrollo:

```
python manage.py runserver
```

3. Estructura del Proyecto

El proyecto sigue una estructura típica de Django, con algunos directorios y archivos clave:

- Gym/: El directorio principal del proyecto.
- usuarios/: Aplicación para gestionar usuarios y autenticación.
- productos/: Aplicación para gestionar productos y actividades del gimnasio.
- templates/: Contiene las plantillas HTML utilizadas en el proyecto.
- static/: Almacena archivos estáticos como CSS, JavaScript e imágenes.
- manage.py: El script de administración de Django.

4. Tecnologías Utilizadas

El proyecto utiliza las siguientes tecnologías:

- Django: El framework web de Python.
- HTML, CSS, JavaScript: Para la creación de interfaces de usuario.
- Bootstrap: Para estilos y componentes frontend.
- SQLite (en desarrollo): Base de datos relacional para el almacenamiento de datos.
- Git y GitHub: Para el control de versiones.

5. Modelos de Datos

Los modelos de datos representan las tablas de la base de datos y se definen en el archivo `models.py` de cada aplicación. Algunos modelos principales incluyen:

- Usuario: Representa a los usuarios del gimnasio.
- Producto: Representa productos y actividades disponibles en el gimnasio.

6. Vistas y Controladores

Las vistas se definen en los archivos `views.py` y controlan la lógica de presentación. Los controladores, en este caso, son las funciones de vista.

7. Plantillas y Diseño Frontend

Las plantillas HTML se almacenan en el directorio `templates/`. El proyecto utiliza Bootstrap para el diseño frontend, con estilos personalizados definidos en archivos CSS en el directorio `static/`.

8. Depuración y Pruebas

Puedes realizar pruebas unitarias y de integración utilizando el módulo `unittest` de Python y el framework de pruebas de Django. Ejecuta las pruebas con el siguiente comando:

```
python manage.py test
```

Para depurar el código, puedes utilizar el depurador integrado de Django o herramientas externas como pdb. También se recomienda utilizar print y registros (logging) para rastrear el flujo de ejecución y depurar problemas.

10. Mantenimiento y Escalabilidad

Para mantener el proyecto, es importante aplicar actualizaciones de seguridad y mantener las dependencias actualizadas. Puedes escalar la aplicación utilizando servidores de base de datos más robustos, servidores de caché y equilibrio de carga si es necesario.

11. Recursos Adicionales

Aquí hay algunos recursos adicionales útiles para el desarrollo:

- Documentación de Django: La documentación oficial de Django es una excelente fuente de referencia.
- Bootstrap Docs: Documentación de Bootstrap para personalizar estilos y componentes.
- GitHub Guides: Tutoriales de GitHub para colaborar en proyectos.