

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
CONCEPCIÓN - CHILE



**“GYM USM: SISTEMA DE GESTIÓN DE RESERVAS
PARA GIMNASIO UNIVERSIDAD TÉCNICA FEDERICO
SANTA MARÍA SEDE CONCEPCIÓN”**

VICTOR ALEXIS MARIÑAN SEPÚLVEDA

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERÍA EN INFORMÁTICA**

Profesor Guía: Leonardo Emilio Hernández Vera
Profesor Correferente: Cristian Antonio Lara Valenzuela

Marzo - 2024

DEDICATORIA

A mis padres, por su amor incondicional, apoyo constante y por ser mi fuente de inspiración. A cada profesor y mentores que han iluminado mi camino académico, gracias por su guía y sabiduría. A mis amigos y compañeros, por las risas compartidas y el aliento inquebrantable. Este logro es también de ustedes.

A los estudiantes, quienes son el motor de mi trabajo. Este proyecto está dedicado a todos ustedes, a su dedicación y pasión por el aprendizaje. Que este esfuerzo contribuya a hacer sus días en la universidad un poco más fáciles y sus metas un poco más alcanzables.

A mi alma mater, por proporcionarme un espacio para crecer, aprender y soñar. Que este trabajo sea un pequeño tributo a la comunidad que me ha acogido y enriquecido.

Con gratitud y dedicación,

AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a todas las personas que han contribuido de alguna manera en la realización de este trabajo. Sus valiosos aportes y apoyo han sido fundamentales para llevar a cabo este proyecto.

En primer lugar, agradezco a mis orientadores, Leonardo Emilio Hernández Vera y Cristian Antonio Lara Valenzuela, por su orientación experta y sus sugerencias significativas que guiaron cada paso de esta investigación. Su sabiduría y dedicación fueron fuentes invaluable de inspiración.

Agradezco profundamente a mis profesores, cuyos conocimientos y enseñanzas han sido el fundamento sobre el cual se ha construido este trabajo. Sus lecciones han iluminado mi camino académico y han enriquecido mi comprensión del tema.

Mi gratitud se extiende a mis amigos y compañeros de clase, quienes han compartido este viaje conmigo, brindando apoyo emocional y colaboración intelectual. Sus debates y sugerencias han sido esenciales para dar forma a mis ideas.

Agradezco a mi familia por su amor incondicional y su apoyo constante. Su aliento ha sido mi fuerza motriz, y este logro es también el suyo.

Finalmente, quiero agradecer a todas las personas que participaron en las entrevistas y encuestas, cuya información y perspectivas fueron cruciales para este estudio. Su generosidad al compartir sus conocimientos ha enriquecido significativamente este trabajo.

Con profundo agradecimiento.

RESUMEN

En este sistema web, adaptable a Smartphone y Tablet, se presenta un sistema intuitivo para la reserva de horarios en el gimnasio de la universidad. El autor desarrolló una plataforma fácil de usar que permite a los usuarios crear cuentas, seleccionar horarios disponibles y recibir confirmaciones instantáneas. La metodología (SCRUM) se centra en un diseño de interfaz amigable que simplifica el proceso de reserva, promoviendo la accesibilidad siendo eficaz. Los resultados principales incluyen una experiencia de usuario mejorada y una reducción significativa en el tiempo dedicado a la reserva de horarios. La relevancia de estos resultados radica en la optimización del uso del gimnasio, fomentando la participación y el compromiso de los estudiantes, profesores y personal en actividades físicas, beneficiando así la salud y el bienestar de la comunidad universitaria.

Reserva de Horarios; Gimnasio Universitario; Experiencia del Usuario; Accesibilidad; Salud y Bienestar.

ABSTRACT

This webpage presents an intuitive system for scheduling gym sessions at the university. The author developed a user-friendly platform allowing users to create accounts, select available time slots, and receive instant confirmations. The methodology focuses on a streamlined interface design that simplifies the scheduling process, promoting accessibility and efficiency. Key outcomes include an enhanced user experience and a significant reduction in time spent on scheduling. The relevance lies in optimizing gym facility utilization, encouraging active participation from students, faculty, and staff, thereby enhancing the overall health and well-being of the university community.

KEYWORDS

Scheduling; University Gym; User Experience; Accessibility; Health and Wellness.

GLOSARIO

1. **Sistema de Gestión de Horarios (SGH):** Plataforma tecnológica que permite a los usuarios reservar y gestionar horarios de manera eficiente.
2. **Interfaz de Usuario (UI):** Diseño visual y disposición de elementos en una aplicación o página web, centrada en la experiencia del usuario.
3. **Experiencia del Usuario (UX):** Interacción del usuario con una aplicación y cómo se siente durante esa interacción, que se centra en la usabilidad y satisfacción del usuario.
4. **Desarrollo Front-end:** Parte del desarrollo web que se enfoca en la interfaz y la experiencia del usuario, incluyendo tecnologías como HTML, CSS.
5. **Desarrollo Back-end:** Parte del desarrollo web que se ocupa de la lógica detrás de escena de una aplicación, incluyendo el manejo de bases de datos y el procesamiento de datos del usuario.
6. **Base de Datos Relacional:** Tipo de base de datos que utiliza tablas para organizar los datos y establecer relaciones entre ellos, utilizando un lenguaje estructurado de consulta (SQL).
7. **Seguridad en Aplicaciones Web:** Conjunto de prácticas y tecnologías utilizadas para proteger las aplicaciones web contra amenazas y ataques cibernéticos, como autenticación, autorización y cifrado de datos.
8. **Metodologías Ágiles:** Enfoques de desarrollo de software que se basan en la colaboración, la adaptabilidad y la entrega continua, como Scrum.
9. **Reserva Instantánea:** Funcionalidad que permite a los usuarios reservar un horario de forma inmediata y recibir confirmaciones automáticas.
10. **Bootstrap:** Framework de diseño web que facilita la creación de interfaces modernas y receptivas mediante el uso de HTML, CSS y JavaScript, con componentes preestablecidos y estilos.
11. **HTML (Lenguaje de Marcado de Hipertexto):** Lenguaje estándar utilizado para la creación y estructuración de contenido en páginas web, utilizando etiquetas para definir elementos y su presentación.
12. **Python:** Lenguaje de programación de alto nivel, versátil y fácil de aprender, utilizado en el desarrollo web para construir aplicaciones y servicios.

13. **CSS (Hojas de Estilo en Cascada):** Lenguaje utilizado para definir el estilo y la presentación de documentos HTML, permitiendo el diseño y la personalización visual de las páginas web.

14. **Flask:** Framework de desarrollo web en Python que simplifica la creación de aplicaciones web, proporcionando herramientas para enrutamiento, manejo de solicitudes y respuestas, y conectividad con bases de datos.

INDICE DE CONTENIDOS

DEDICATORIA.....	2
AGRADECIMIENTOS	3
RESUMEN.....	4
ABSTRACT	4
INDICE DE IMAGENES	9
INDICE DE TABLAS.....	10
INTRODUCCIÓN	11
CAPÍTULO 2: MARCO CONCEPTUAL.....	19
3.1.1 REQUERIMIENTOS	21
3.1.2 Requerimientos funcionales	21
3.1.3 Requerimientos no funcionales	24
3.1.4. Requerimientos Técnicos.	25
CAPÍTULO 4: CASOS DE USOS Y MODELAMIENTO DE DATOS	27
4.1. CASO DE USO.....	27
4.2 CASOS DE USOS EXTENDIDOS	29
4.3 DIAGRAMA DE CLASE	37
4.4 MODELO DE DATOS LÓGICO	38
4.5 MODELO DE DATOS FISICO	39
4.6 DICCIONARIO DE DATOS	40
CAPÍTULO 5: DISEÑO INTERFAZ.....	46
5.1. Inicio de sesión	46
5.2. Registro.....	47
5.3. Mostrar Horarios	48
5.4. Tomar horario	49
5.5 Confirmación de reservas.....	50

5.6 Tipos de entrenamiento	51
5.7 Notificaciones	52
5.8. Gestión de usuarios	53
5.9. Ingreso de usuarios manual.....	54
5.10. Crear eventos para notificar a los usuarios	55
6.1. PostgreSQL.....	56
6.2. Amazon Web Services	57
6.3. Python	59
6.3. Bootstrap	61
6.3. Flask	63
CAPÍTULO 7: PROPUESTA DE SOLUCION	65
CAPÍTULO 8: VALIDACIÓN DE LA SOLUCIÓN.....	67
CAPÍTULO 9: CONCLUSIONES	69
REFERENCIAS BIBLIOGRÁFICAS.....	73

INDICE DE IMAGENES

Sala de musculación 1.....	13
Sala de musculación 2.....	13
Sala de musculación 3.....	14
Imagen 1. UML.....	27
Imagen 2. Diagrama de clase	37
Imagen 3. Modelo de datos lógico	38
Imagen 4. Modelo de datos físico.....	39
Imagen 5. Inicio de sesión	46
Imagen 6. Registro de usuario	47
Imagen 7. Mostrar horarios	48
Imagen 8. Toma de horarios	49
Imagen 9. Reservas realizadas.....	50
Imagen 10. Entrenamientos	51
Imagen 11. Notificaciones	52
Imagen 12. Gestión de usuarios	53
Imagen 13. Ingreso de usuarios manual	54
Imagen 14. Ingresar evento a compartir	55
Imagen15.PostgreSQL.....	56
Imagen 16. Amazon.	58
Imagen 17. Python.....	59
Imagen 18. Bootstrap.....	61
Imagen 19. Flask.	63

INDICE DE TABLAS

Tabla 3.1.2.1 Requerimiento N°1.	21
Tabla 3.1.2.2 Requerimiento N°2.	21
Tabla 3.1.2.3 Requerimiento N°3.	22
Tabla 3.1.2.4 Requerimiento N°4.	22
Tabla 3.1.2.5. Requerimiento N°5.	22
Tabla 3.1.2.6. Requerimiento N°6.	23
Tabla 3.1.2.7 Requerimiento N°7.	23
Tabla 3.1.2.8 Requerimiento N°8.	23
Tabla 3.1.3.1 Requerimiento N°1.	24
Tabla 3.1.3.2 Requerimiento N°2.	24
Tabla 3.1.3.3. Requerimiento N°3.	25
Tabla 3.1.4.1 Requerimiento N°1.	25
Tabla 3.1.4.2 Requerimiento N°2.	26
4.6.1 Usuario.....	40
4.6.2. Reservas.....	41
4.6.3. Entrenamiento.....	41
4.6.4 Registros	42
4.6.5 Horario.....	43
4.6.6. Evento.....	44
4.6.6. Profesor	44
4.6.6. registro_profesor_turno.....	45

INTRODUCCIÓN

La presente memoria se propone desarrollar un Sistema de Gestión de Horarios (SGH) innovador y robusto en el contexto del gimnasio universitario, un área crucial para la promoción del bienestar físico y mental en la comunidad académica. En este contexto, surge la necesidad de abordar el desafío de simplificar y optimizar el proceso de reservas, permitiendo a los usuarios acceder de manera rápida y efectiva a las instalaciones del gimnasio.

Este sistema web, destinado a simplificar y mejorar la experiencia de los usuarios al reservar y gestionar horarios, se fundamentará en un enfoque tecnológico avanzado, haciendo uso de diversas herramientas y tecnologías líderes en la industria.

Arquitectura Tecnológica

El núcleo del desarrollo se basará en un enfoque de desarrollo web moderno, combinando tanto tecnologías frontend como backend. Para la interfaz de usuario (UI), se adoptará el framework Bootstrap, permitiendo un diseño receptivo y una experiencia de usuario atractiva.

En el backend, la elección recae en el uso del framework Flask, impulsado por el lenguaje de programación Python. Flask ofrecerá un entorno flexible y escalable para el desarrollo de la lógica de la aplicación, gestionando las solicitudes del usuario y garantizando una interacción eficiente con la base de datos.

Persistencia de Datos

La persistencia de datos se llevará a cabo mediante el sistema de gestión de bases de datos PostgreSQL. Esta elección se fundamenta en la robustez y la capacidad de manejar grandes volúmenes de datos de manera eficiente.

Despliegue en la Nube

Para garantizar un acceso ininterrumpido y una escalabilidad eficaz, la implementación del sistema se llevará a cabo en la infraestructura de Amazon Web Services (AWS). Los servicios de AWS, tales como EC2 para la implementación de la aplicación, RDS para la gestión de la base de datos PostgreSQL, y S3 para el almacenamiento de archivos estáticos, formarán parte integral de la arquitectura.

Metodología de Desarrollo y Seguridad

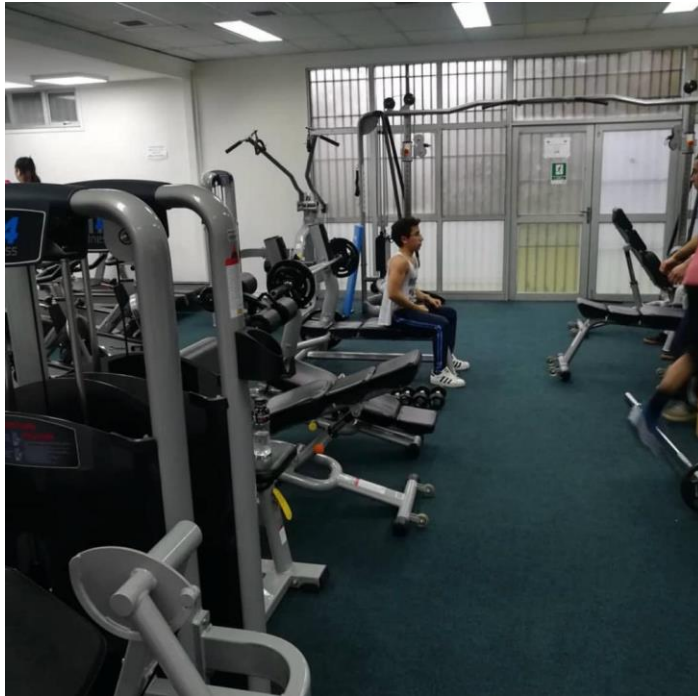
El desarrollo se regirá por metodologías ágiles, fomentando una entrega continua y adaptativa. Se implementarán rigurosas prácticas de seguridad en aplicaciones web, abarcando la autenticación de usuarios, la autorización de accesos y el cifrado de datos sensibles.

Este enfoque integral en la arquitectura y tecnologías seleccionadas busca no solo abordar las necesidades actuales de la gestión de horarios, sino también anticipar y preparar el sistema para futuras expansiones y mejoras.

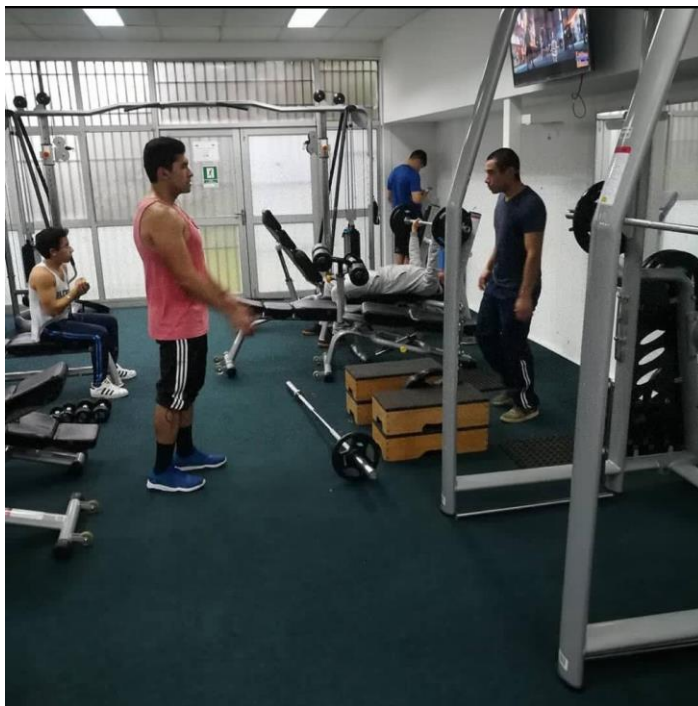
Este trabajo se enfoca en comprender la complejidad de los sistemas de reserva actuales y propone una solución innovadora y accesible. A lo largo de esta memoria, se detalla cómo se diseñó y desarrolló una plataforma intuitiva que facilita a los usuarios la creación de cuentas, la selección de horarios y la confirmación instantánea de sus reservas. Esta propuesta no solo aborda las limitaciones del sistema anterior que consistía en un Excel en el cual uno tiene que ir presencialmente a inscribirse para una reserva, sino que también busca mejorar significativamente la experiencia del usuario y promover una participación más activa en las actividades físicas.

El enfoque metodológico adoptado se basa en la creación de una interfaz de usuario amigable y eficiente, respaldada por tecnologías de vanguardia para garantizar la fiabilidad y la seguridad del sistema. Esta introducción proporciona un vistazo a la estructura del documento, que incluye una revisión detallada de la problemática existente, seguida de una descripción minuciosa de la metodología implementada. A medida que se avanza en estas páginas, se revelarán los desafíos encontrados, las soluciones desarrolladas y los resultados obtenidos.

En última instancia, esta investigación busca no solo mejorar la gestión de horarios en el gimnasio universitario, sino también sentar las bases para futuras innovaciones en la optimización de servicios similares en otros contextos. Al proporcionar una visión general de la problemática abordada y del enfoque metodológico adoptado, esta introducción sirve como un portal que invita al lector a explorar en detalle el proceso de desarrollo y los logros alcanzados en esta propuesta innovadora.



Sala de musculación 1



Sala de musculación 2



Sala de musculación 3

OBJETIVO GENERAL

- Desarrollar un sistema web con el fin de facilitar la inscripción en el gimnasio de la universidad y fijar hora para asistir a él.

OBJETIVOS ESPECÍFICOS

- Entregar las herramientas necesarias para generar la inscripción en el gimnasio
- Creación de una página intuitiva y responsiva para una buena interacción con el usuario
- Establece el tiempo que puede dedicar a actividades de la sala de musculación.

JUSTIFICACIÓN

El gimnasio de la universidad es un lugar que permite practicar o hacer ejercicio en un recinto cerrado con varias máquinas y artículos deportivos a disposición de quienes lo visiten y muchas veces hay que ir presencialmente a inscribirse, en el cual existe un Excel donde se registran todos los alumnos de la universidad lo cual es un trabajo de mucha manualidad y tedioso.

PROPÓSITO

Esto les facilitaría a todos los alumnos que quisieran inscribirse en el gimnasio sin la necesidad de tener que asistir presencialmente a rellenar ese Excel al recinto para ver su horario disponible para realizar su respectivo ejercicio.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

En el contexto actual del gimnasio universitario, la gestión de horarios se enfrenta a un desafío significativo: la falta de eficiencia y accesibilidad en el proceso de reserva. Actualmente, la toma de horarios se lleva a cabo mediante un sistema basado en hojas de cálculo de Excel. Este método, aunque ha sido utilizada durante un tiempo, se ha vuelto cada vez más tediosa y propensa a errores. Los usuarios, incluyendo estudiantes, profesores y personal, se ven obligados a navegar por un documento compartido, lo que no solo consume tiempo, sino que también dificulta la planificación y coordinación de sus actividades físicas.

ACTORES INVOLUCRADOS

Los actores principales involucrados son los usuarios del gimnasio universitario, incluyendo estudiantes de diversos programas académicos, profesores y personal administrativo. Estos individuos dependen del gimnasio como una fuente vital de actividad física y bienestar.

DIFICULTADES ACTUALES Y CANTIDAD DE USUARIOS AFECTADOS

Las dificultades actuales radican en la naturaleza poco práctica del sistema actual basado en Excel. Los usuarios enfrentan desafíos al tratar de encontrar horarios disponibles, lo que lleva a la frustración y, en muchos casos, a no reservar. Se estima que cientos de usuarios se ven afectados diariamente por este problema, lo que subraya la necesidad crítica de una solución más eficiente y amigable. El gimnasio se utiliza con regularidad un 70 %, la gente que llega y accede a él muchas veces ahí mismo se tiene que inscribir para entrar y a veces está copado y no pudo obtener su hora reservada ya que no tiene como enterarse de ello.

CONSECUENCIAS DE NO ABORDAR EL PROBLEMA

Si esta problemática no se resuelve a corto plazo, las consecuencias incluirán una disminución en la participación de los usuarios, lo que afectará negativamente la salud y el bienestar de la comunidad universitaria en general. Además, la falta de un sistema eficaz para la gestión de horarios podría llevar a la sobreutilización o subutilización de las instalaciones, lo que afectaría la disponibilidad para aquellos que deseen participar en actividades físicas.

COMPETENCIA EXISTENTE

Aunque existen algunas soluciones de gestión de horarios en el mercado, ninguna ha sido específicamente adaptada para satisfacer las necesidades y dinámicas únicas del gimnasio universitario. La mayoría de las soluciones existentes son complejas y costosas, lo que no las hace adecuadas para la comunidad universitaria al momento de tener que financiar un software.

ALCANCES DE LA SOLUCIÓN PROPUESTA

La solución propuesta debe proporcionar a los usuarios una experiencia de reserva de horarios fácil, rápida y precisa. Además, la plataforma deberá ser segura y confiable, garantizando la integridad de los datos y la privacidad de los usuarios. Al desarrollar esta solución, se busca mejorar significativamente la eficiencia de la gestión de horarios y, en última instancia, fomentar una mayor participación en actividades físicas, promoviendo así el bienestar en toda la comunidad universitaria.

PROCESOS DEL USUARIO

Para hacer un sistema web para la toma de horarios del gimnasio de la universidad, es importante la experiencia del usuario al ingresar a la página y cómo se podría facilitar el proceso para reservar un horario en el gimnasio. Estos serían los procedimientos por considerar:

1. **Página de inicio:** en la página principal debería incluir una breve descripción del gimnasio, horarios de apertura y cierre, así como los servicios que ofrece. También debería incluir una opción para iniciar sesión o registrarse en caso de que el usuario aún no tenga una cuenta.
2. **Creación de una cuenta de preferencia el correo institucional:** si el usuario no tiene una cuenta, debería haber una opción para registrarse en la página. Al crear una cuenta, el usuario debería ingresar su información personal, como nombre, correo electrónico institucional, por ejemplo: @usm.cl, @sansano.usm.cl, número de teléfono y contraseña.
3. **Selección de horario:** una vez que el usuario ha iniciado sesión, debería haber una sección donde pueda seleccionar el horario que desea reservar. Esto puede hacerse a través de un calendario en el que se muestren los horarios disponibles y el usuario pueda seleccionar el que desee.
4. **Confirmación de reserva:** después de seleccionar el horario, el usuario debería recibir una confirmación de que su reserva ha sido aceptada. Además, debería recibir una confirmación por correo electrónico con los detalles de la reserva.
5. **Cancelación de reserva:** en caso de que el usuario necesite cancelar su reserva, debería haber una opción para hacerlo en el sistema web. Esto debería ser fácil de encontrar y permitir al usuario cancelar su reserva con un solo clic.
6. **Ayuda y soporte:** si el usuario tiene alguna pregunta o necesita ayuda con el sistema web, debería haber una sección de ayuda y soporte donde pueda encontrar información útil. Esto puede incluir preguntas frecuentes, una sección de contacto

con el equipo de soporte o una sección de comentarios donde el usuario pueda dejar sus opiniones y sugerencias.

CAPÍTULO 2: MARCO CONCEPTUAL

El marco conceptual de esta memoria se fundamenta en una serie de conceptos y tecnologías clave que son esenciales para la comprensión y la implementación efectiva de la solución propuesta.

SISTEMAS DE GESTIÓN DE HORARIOS (SGH)

Los sistemas de gestión de horarios son plataformas tecnológicas que permiten a los usuarios reservar y gestionar horarios de manera eficiente. Estos sistemas han evolucionado para abordar diversas necesidades, desde la planificación académica hasta la gestión de recursos en instituciones como gimnasios y centros deportivos. Comprender las mejores prácticas en SGH es fundamental para diseñar una solución eficaz y centrada en el usuario.

INTERFAZ DE USUARIO (UI) Y EXPERIENCIA DEL USUARIO (UX)

La interfaz de usuario se refiere al diseño visual y la disposición de los elementos en una aplicación o página web. La experiencia del usuario, por otro lado, se centra en cómo los usuarios interactúan con una aplicación y cómo se sienten durante esa interacción. La combinación efectiva de UI y UX es esencial para crear una plataforma fácil de usar y atractiva para los usuarios finales.

DESARROLLO WEB Y TECNOLOGÍAS FRONT-END

El desarrollo web moderno implica el uso de tecnologías front-end como HTML, CSS para crear interfaces interactivas y receptivas. Frameworks como Python han revolucionado el desarrollo web, proporcionando herramientas poderosas para crear aplicaciones web dinámicas y eficientes.

BASES DE DATOS Y GESTIÓN DE DATOS

Las bases de datos son fundamentales para almacenar y gestionar datos de manera estructurada y eficiente. La tecnología de base de dato relacional a utilizar seria PostgreSQL que es la opción más recomendable por su escalabilidad en el desarrollo del sistema web.

SEGURIDAD EN APLICACIONES WEB

La seguridad en aplicaciones web es crucial para proteger los datos del usuario y garantizar la integridad del sistema. Conceptos como autenticación, autorización, cifrado de datos y protección contra ataques comunes, serán necesarios para el desarrollo seguro de la aplicación web.

METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE

Las metodologías ágiles, como Scrum, proporcionan enfoques flexibles y colaborativos para el desarrollo de software. Estas metodologías se centran en la adaptabilidad, la colaboración y la entrega continua, lo que es especialmente relevante en el proyecto de desarrollo web.

CAPÍTULO 3: ANÁLISIS DE REQUERIMIENTOS DE LA PAGINA GYMUSM.

Los siguientes requerimientos fueron formulados por el administrador de la sala de musculación en base a la experiencia que él tiene como usuario en el uso de Excel.

3.1.1 REQUERIMIENTOS

3.1.2 Requerimientos funcionales

Tabla 3.1.2.1 Requerimiento N°1.

N° Requerimiento 1

Nombre	Registro de usuario
Tipo	Acceso
Descripción	El usuario debe poder registrarse en caso de no haber ningún registro con el mismo nombre.

Tabla 3.1.2.2 Requerimiento N°2.

N° Requerimiento 2

Nombre	Inicio de sesión de usuario
Tipo	Acceso
Descripción	El usuario debe poder iniciar sesión para acceder al menú principal, en caso de no haber registro del usuario debe registrarse.

Tabla 3.1.2.3 Requerimiento N°3.

N° Requerimiento 3

Nombre	Ver Horarios
Tipo	Acceso
Descripción	El usuario puede observar los horarios al iniciar sesión en el menú principal.

Tabla 3.1.2.4 Requerimiento N°4.

N° Requerimiento 4

Nombre	Reservar horario
Tipo	Selección
Descripción	El usuario en el menú principal tendrá la opción de tomar el horario que se encuentre disponible.

Tabla 3.1.2.5. Requerimiento N°5.

N° Requerimiento 5

Nombre	Confirmación de reserva
Tipo	acceso
Descripción	El usuario en el menú principal estará la opción de confirmación de reservas.

Tabla 3.1.2.6. Requerimiento N°6.

N° Requerimiento 6

Nombre	Tipo de ejercicio a realizar
Tipo	Acceso
Descripción	El usuario en el menú principal estará la opción que le permitirá ingresar el tipo de ejercicio que desee realizar.

Tabla 3.1.2.7 Requerimiento N°7.

N° Requerimiento 7

Nombre	Cancelación de reservas
Tipo	Acceso
Descripción	El usuario en el menú principal estará la opción que le permitirá ver las reservas realizadas y en ellas podrá también cancelar las reservas.

Tabla 3.1.2.8 Requerimiento N°8.

N° Requerimiento 8

Nombre	Notificaciones
Tipo	Acceso
Descripción	El usuario en el menú principal mostrara las notificaciones en caso de cualquier evento que suceda o vía correo electrónico se le informara.

3.1.3 Requerimientos no funcionales

Tabla 3.1.3.1 Requerimiento N°1.

N° Requerimiento 1

Nombre	Creación y diseño sencillo e intuitivo
Tipo	Diseño
Descripción	La página web debe ser responsiva e intuitiva.

Tabla 3.1.3.2 Requerimiento N°2.

N° Requerimiento 2

Nombre	Compatibilidad
Tipo	Presentación
Descripción	Que no exija muchos requerimientos tanto en hardware y que sea compatible con el dispositivo que lo esté ocupando como computadora, smartphone o Tablet.

Tabla 3.1.3.3. Requerimiento N°3.

N° Requerimiento 3

Nombre	Creación y diseño de una presentación del sistema
Tipo	Presentación
Descripción	Presentación del sistema para el usuario inicial.

3.1.4. Requerimientos Técnicos.**Tabla 3.1.4.1 Requerimiento N°1.**

N° Requerimiento 1

Nombre	Error de DNS
Tipo	Conectividad
Descripción	No se puede acceder al sistema por el NIC. En caso de que el dispositivo este fuera del servidor que se conecta mediante el backend con la base.

Tabla 3.1.4.2 Requerimiento N°2.

N° Requerimiento 2

Nombre	AWS
Tipo	Tecnológico
Descripción	Una amplia gama de servicios y recursos informáticos que permiten a las organizaciones construir, implementar y escalar aplicaciones y servicios de manera eficiente.

CAPÍTULO 4: CASOS DE USOS Y MODELAMIENTO DE DATOS

4.1. CASO DE USO

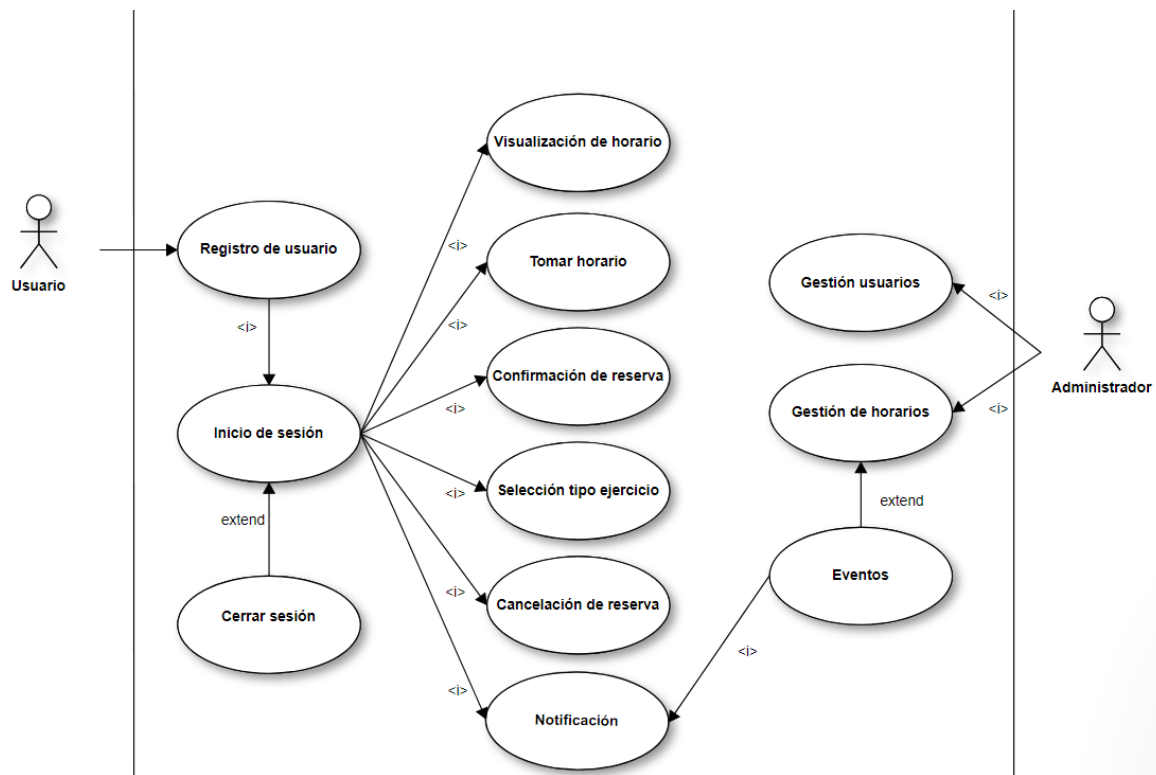


Imagen 1. UML

Los casos de uso son una técnica utilizada en el desarrollo de software para capturar y definir las interacciones entre un sistema y sus actores externos (usuarios o sistemas externos). Proporcionan una descripción detallada de cómo los usuarios interactúan con el sistema para lograr un objetivo específico. Como se puede ver aquí en la imagen los elementos claves relacionados con el sistema:

Actor:

Descripción: Un actor es cualquier entidad externa que interactúa con el sistema. Puede ser un usuario de la universidad.

Ejemplo: Un "Estudiante" que desea reservar un horario en el gimnasio.

Caso de Uso:

Descripción: Un caso de uso representa una interacción específica entre un actor y el sistema para lograr un objetivo particular. Describe lo que hace el sistema en respuesta a una entrada del usuario.

Ejemplo: "Realizar Reserva de Horario".

Escenario:

Descripción: Un escenario es una instancia particular de un caso de uso. Es una secuencia específica de acciones que ocurren cuando un actor interactúa con el sistema.

Ejemplo: Un escenario para "Realizar Reserva de Horario" podría incluir pasos como iniciar sesión, registrarse, seleccionar un horario y confirmar la reserva.

Relaciones:

Las relaciones entre casos de uso y actores representan cómo interactúan los actores con diferentes partes del sistema.

Ejemplo: Un actor "Administrador" puede tener un caso de uso separado para "Gestionar Usuarios", "Gestionar Horarios", "Eventos" que está relacionado con ello, pero diferente de "Realizar Reserva de Horario".

Diagrama de Casos de Uso:

Un diagrama que visualiza las relaciones entre actores y casos de uso en un sistema. Es una herramienta útil para entender la estructura general de las interacciones del sistema en UML.

4.2 CASOS DE USOS EXTENDIDOS

Descripción: Un sistema web para la toma de horarios del gimnasio USM que permite a los usuarios registrarse, seleccionar horarios disponibles, confirmar sus reservas, cancelar reservas, consultar horarios y recibir notificaciones.

Definición de actores

Usuario: estudiantes de la universidad o funcionarios de la universidad

CDU Extendidos

- Nombre:** Registro
- Actores:** Usuarios USM
- Descripción:** El usuario selecciona la opción de registro.
- Pre condición:** El usuario accede a la página web del gimnasio
- Post condición:** Ingresar y Registrarse en forma satisfactoria.
- Responsable:** Víctor Mariñan

Curso Normal

Usuario	Sistema
1. Ingresar al sistema web del gimnasio.	2. El sistema solicita la información del Usuario: a. Correo. b. Contraseña. c. Registrarse
3. Seleccionar la opción de Registrar.	4. Ofrece las opciones de logearse o salir.
5. Ingresa la información solicitada por el sistema.	6. Ofrece las opciones de ingresar a registrarse.
7. Selecciona avanzar.	8. Valida que todos los datos estén ingresados correctamente e inicia el login del cliente, mostrando en pantalla el perfil del usuario guardado junto a la frase “bienvenido a GymUsm SR. (a) logueado”.

Excepciones

Administrador	Sistema
1. El Usuario deja un campo en blanco y selecciona registrar.	2. Informa al usuario que quedaron datos en blanco y solicita que lo ingrese (mostrando detalle).
3. Selecciona Salir	4. pregunta si está seguro si desea salir de la App: a. si es así, cierro la App b. si no es así, volvemos al CDU actual.

Nombre: Login

Actores: Usuarios USM

Descripción: Permite al Usuario Logearse en el sistema.

Pre condición: Estar en el sistema y contar con el registro.

Post condición: Logearse en forma satisfactoria.

Responsable: Víctor Mariñan

Curso Normal

Usuario	Sistema
1. Seleccionar la opción de Login.	2. El sistema solicita la información del Usuario: a. Correo. b. Contraseña.
3. Ingresa la información solicitada por el sistema.	4. Valida que todos los datos estén ingresados correctamente y guarda el nuevo registro, mostrando en pantalla el perfil del usuario guardado junto a la frase "El registro se ha realizado exitosamente".

Excepciones

Administrador	Sistema
1. El Usuario deja un campo en blanco y selecciona login.	2. Informa al usuario que quedaron datos en blanco y solicita que lo ingrese (mostrando detalle).
3. Selecciona Cancelar o cerrar.	4. pregunta si está seguro si desea salir el login: a. si es así, cierro la ventana. b. si no es así, volvemos al CDU actual.

- Nombre:** Selección de horarios
- Actores:** Usuarios USM
- Descripción:** El usuario navega por las opciones de horarios disponibles, filtrando por tipo de clase, instructor, etc. El usuario selecciona un horario deseado y lo añade a su reserva.
- Pre condición:** El sistema verifica las credenciales del usuario y lo redirige a la página de selección de horarios.
- Post condición:** El usuario puede agregar más horarios o proceder a la confirmación de reserva.
- Responsable:** Víctor Mariñan

Curso Normal

Usuario	Sistema
1. Seleccionar página de selección de horarios.	2. El sistema muestra horarios disponibles, filtrando por tipo de clase, instructor, etc.
3. El usuario selecciona un horario deseado y lo añade a su carrito de reserva.	4. Valida las selecciones sobre horarios disponibles, filtrando por tipo de clase, instructor. Mostrando en pantalla "reserva se ha agrega exitosamente".

Excepciones

Administrador	Sistema
1. El Usuario deja un campo sin seleccionar.	2. Informa al usuario que no se han seleccionado campos solicita que lo seleccione (mostrando detalle).
3. Selecciona Cancelar o cerrar.	4. pregunta si está seguro si desea salir de la selección: a. si es así, cierro la ventana. b. si no es así, volvemos al CDU actual.

- Nombre:** Confirmación de reserva
- Actores:** Usuarios USM
- Descripción:** El sistema muestra una confirmación de reserva en la página web y/o envía un correo electrónico al usuario con los detalles de la reserva..
- Pre condición:** El usuario revisa los horarios seleccionados en su carrito de reserva.
- Post condición:** El usuario confirma su reserva.
- Responsable:** Víctor Mariñan

Curso Normal

Usuario	Sistema
1. Seleccionar confirma la selección de horarios.	2. El sistema muestra si está seguro de confirmar. a. Aceptar. b. Cancelar.
3. El usuario selecciona Aceptar o Cancelar.	4. Valida las selecciones. Mostrando en pantalla “reserva confirmada exitosamente” en el caso de aceptar. Al cancelar muestra en pantalla “reserva cancelada”.

Excepciones

Administrador	Sistema
1. El Usuario olvida confirmar.	2. Informa al usuario que no confirmado su selección. (mostrando detalle).
3. Selecciona Cancelar o cerrar.	4. pregunta si está seguro si desea salir de la selección: a. si es así, cierro la ventana. b. si no es así, volvemos al CDU actual.

- Nombre:** Cancelación de reserva
- Actores:** Usuarios USM
- Descripción:** El usuario accede a su perfil o sección de reservas para cancelar reservaciones hechas.
- Pre condición:** El usuario inicia sesión en la página web.
- Post condición:** El usuario confirma la cancelación de la reserva.
- Responsable:** Víctor Mariñan

Curso Normal

Usuario	Sistema
1. Seleccionar cancelar el horario confirmado.	2. El sistema muestra si está seguro de cancelar. a. Aceptar. b. Cancelar.
3. El usuario selecciona Aceptar o Cancelar.	4. Valida las selecciones. Mostrando en pantalla "reserva cancelada" en el caso de aceptar. Al cancelar el horario confirmado sigue en curso.

Excepciones

Administrador	Sistema
1. El Usuario olvida cancelar.	2. Informa al usuario que no a cancelado su selección. (mostrando detalle).
3. Selecciona Cancelar o cerrar.	4. pregunta si está seguro si desea salir de la selección: a. si es así, cierro la ventana. b. si no es así, volvemos al CDU actual.

Nombre:	Consulta de horarios
Actores:	Usuarios USM
Descripción:	El usuario navega a la sección de horarios disponibles y el sistema muestra una lista de horarios disponibles con información relevante, como tipo de clase, instructor, capacidad y número de reservas.
Pre condición:	El usuario inicia sesión en la página web.
Post condición:	El usuario puede filtrar los horarios por criterios específicos, como fecha, tipo de clase, etc.
Responsable:	Víctor Mariñan

Curso Normal

Usuario	Sistema
1. Navega en la selección de horarios disponibles.	2. El sistema muestra una lista de horarios disponibles.
3. El usuario filtra los horarios.	4. El sistema muestra el tipo de filtro por criterios específicos, como fecha, tipo de clase, etc.

Excepciones

Administrador	Sistema
1. Caída de internet.	2. Informa al usuario que no hay conexión (mostrando detalle).

Nombre: Notificaciones

Actores: Usuarios USM

Descripción: El sistema envía notificaciones a los usuarios, ya sea por correo electrónico o mensajes de texto, para recordarles sus horarios reservados.

Pre condición: El usuario debe tener una hora confirmada para el gimnasio.

Post condición: Las notificaciones se envían antes de la hora programada para la clase

Responsable: Víctor Mariñan

Curso Normal

Administrador	Sistema
1. Programa la entrega de notificaciones.	2. Las notificaciones se envían antes de la hora programada para la clase, recordando al usuario su reserva y proporcionando información adicional, como el nombre del instructor, ubicación del gimnasio, etc.

Excepciones

Administrador	Sistema
1. Caída de internet.	2. Informa al usuario que no hay conexión (mostrando detalle).

4.3 DIAGRAMA DE CLASE

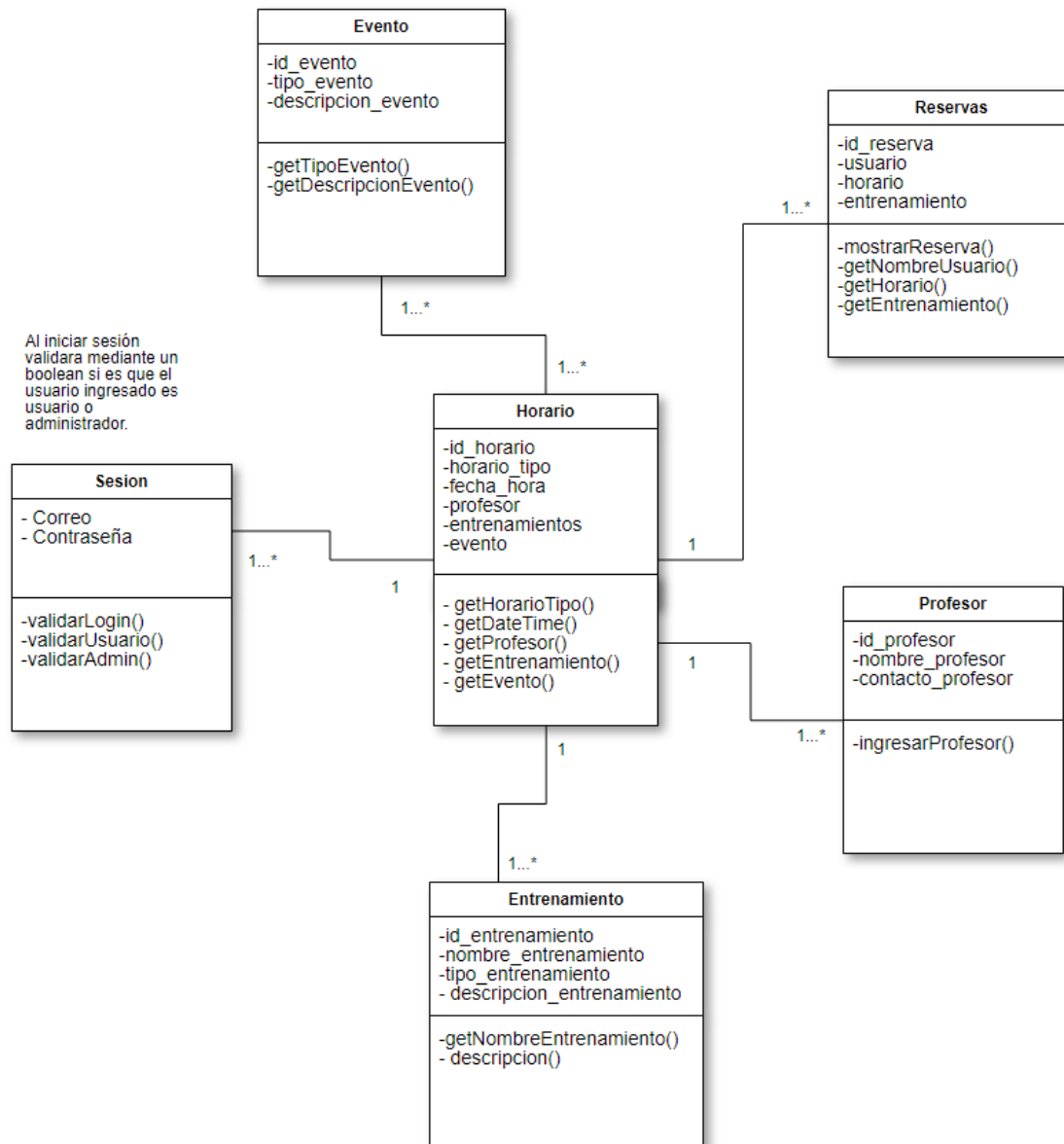


Imagen 2. Diagrama de clase

4.4 MODELO DE DATOS LÓGICO

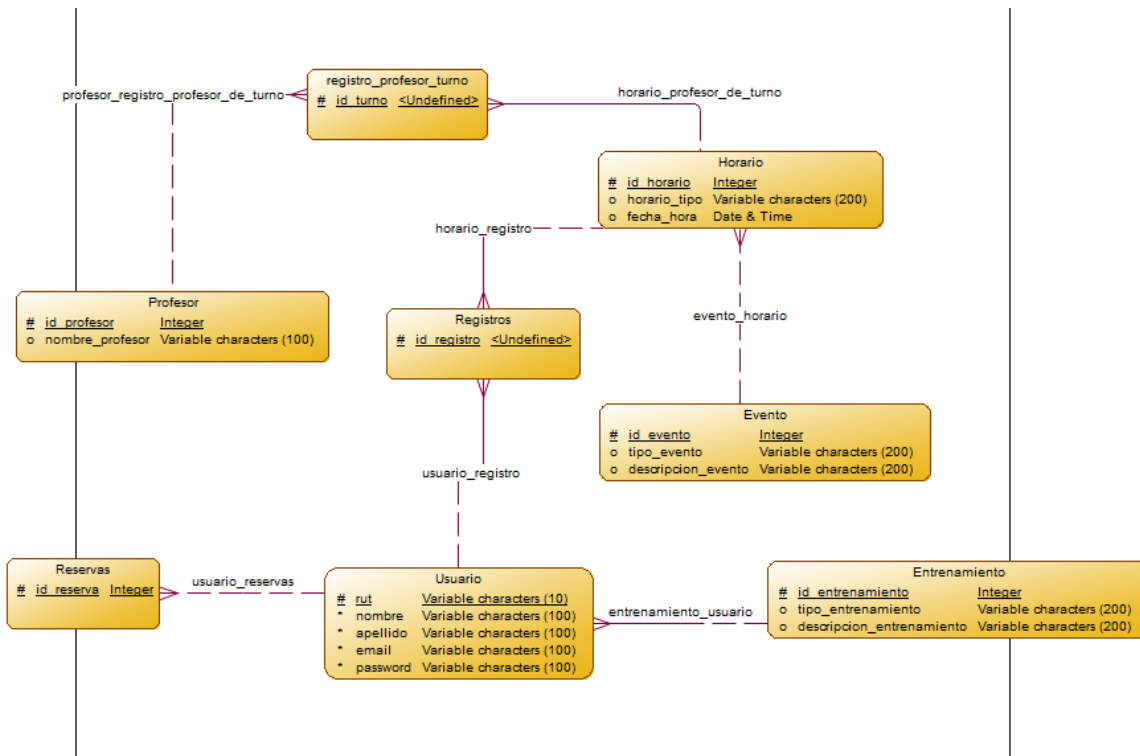


Imagen 3. Modelo de datos lógico

4.5 MODELO DE DATOS FISICO

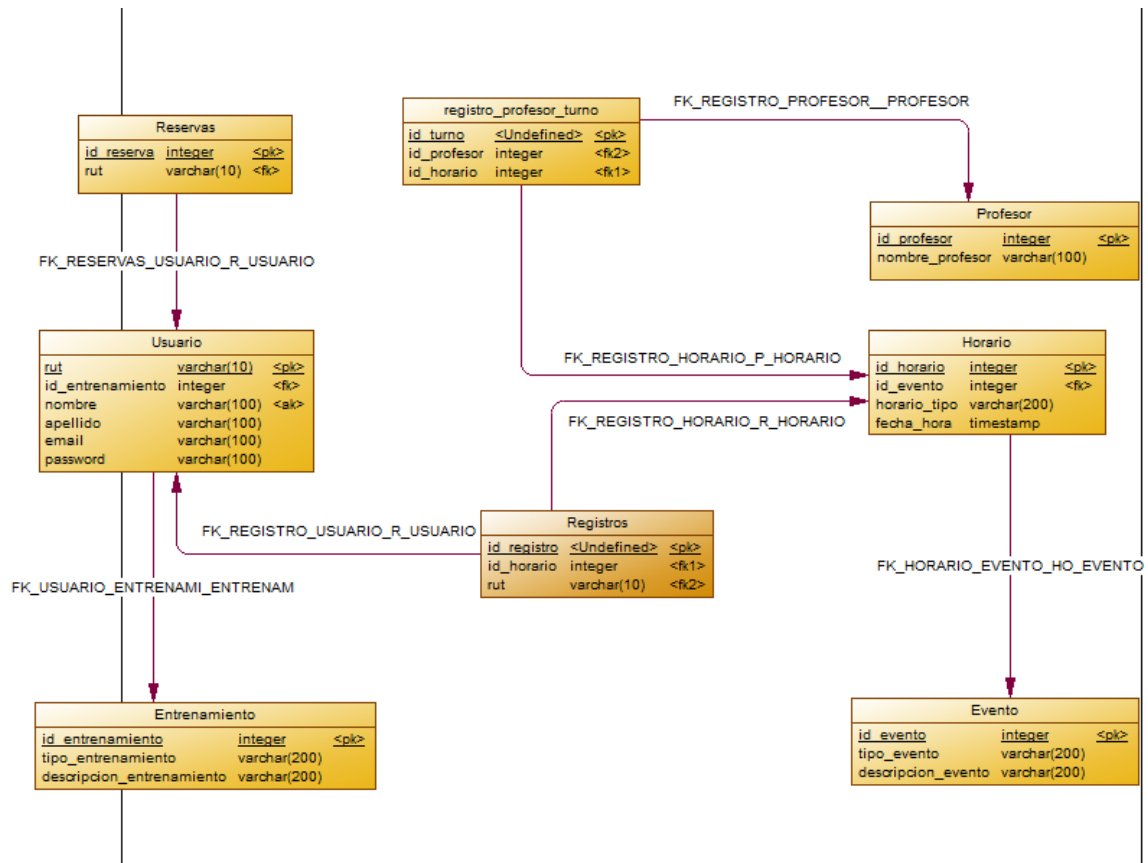


Imagen 4. Modelo de datos físico

4.6 DICCIONARIO DE DATOS

El diccionario de datos nos otorga una descripción de cada una de las tablas que se encuentra en la base de datos de la sistema web.

4.6.1 Usuario

En esta tabla se almacena todo acerca del usuario para que una vez registrado e iniciado sesión pueda acceder al menú principal.

Nombre del campo	Tipo De Dato	Descripción
rut	Varchar(10)	Será el Identificador del usuario.
nombre	Varchar (100)	nombre que el usuario digitará cuando se registre.
email	Varchar (100)	Correo que utilizara el usuario para poder registrarse.
Password	Varchar (100)	Contraseña que asignara el usuario cuando se registre.

4.6.2. Reservas

En esta tabla se almacena los datos sobre las reservas realizadas en la toma de horarios del gym.

Nombre del campo	Tipo De Dato	Descripción
id_reserva	integer	Identificador único de la reserva.

4.6.3. Entrenamiento

En esta tabla se almacenan todos los entrenamientos.

Nombre del campo	Tipo De Dato	Descripción
id_entrenamiento	integer	Identificador único del entrenamiento.
tipo_entrenamiento	Varchar (200)	Nombre del tipo de entrenamiento a realizar en el horario solicitado.
Descripción_entrenamiento	Varchar (200)	Se describe el tipo de entrenamiento a realizar.

4.6.4 Registros

En esta tabla se almacenan los registros de todos los horarios tomados por los usuarios manteniendo un tipo de auditoría que refleje la cantidad de usuarios que asisten frecuentemente al gym.

Nombre del campo	Tipo De Dato	Descripción
id_registro	Integer	Identificador único de los registros.
rut	Varchar(10)	Será el Identificador del usuario.
id_horario	Integer	Identificador único de los horarios.

4.6.5 Horario

En esta tabla se almacena el horario del gym con sus horas disponibles.

Nombre del campo	Tipo De Dato	Descripción
id_horario	integer	Identificador único de los horarios.
id_profesor	integer	Identificador único del profesor.
id_evento	integer	Identificador único del evento.
horario_tipo	Varchar(200)	Tipo de horario que puede ser mañana o tarde.
fecha_hora	Data&time	Fecha y hora sobre los horarios disponibles para seleccionar.

4.6.6. Evento

En esta tabla se asignarán los eventos que podrían suceder tanto como de imprevisto o planeado en el gym.

Nombre del campo	Tipo De Dato	Descripción
id_evento	integer	Identificador único del evento.
tipo_evento	Varchar(200)	Se indica el tipo de evento que puede suceder de imprevisto o planeado.
descripción_evento	Varchar(200)	Se describe el tipo de evento que ocurrió.

4.6.6. Profesor

En esta tabla se almacena la información acerca donde se jugará hora cancha.

Nombre del campo	Tipo De Dato	Descripción
id_profesor	integer	Identificador único de los profesores.

4.6.6. registro_profesor_turno

En esta tabla se almacena la información acerca de los profesores que registran sus turnos.

Nombre del campo	Tipo De Dato	Descripción
id_turno	Integer	Identificador único de los turnos.
id_profesor	integer	Identificador único de los profesores.
id_horario	Integer	Identificador único de los horarios.

CAPÍTULO 5: DISEÑO INTERFAZ

5.1. Inicio de sesión



Iniciar Sesión en
GymUsm

Email address

Password

[Regístrame](#)

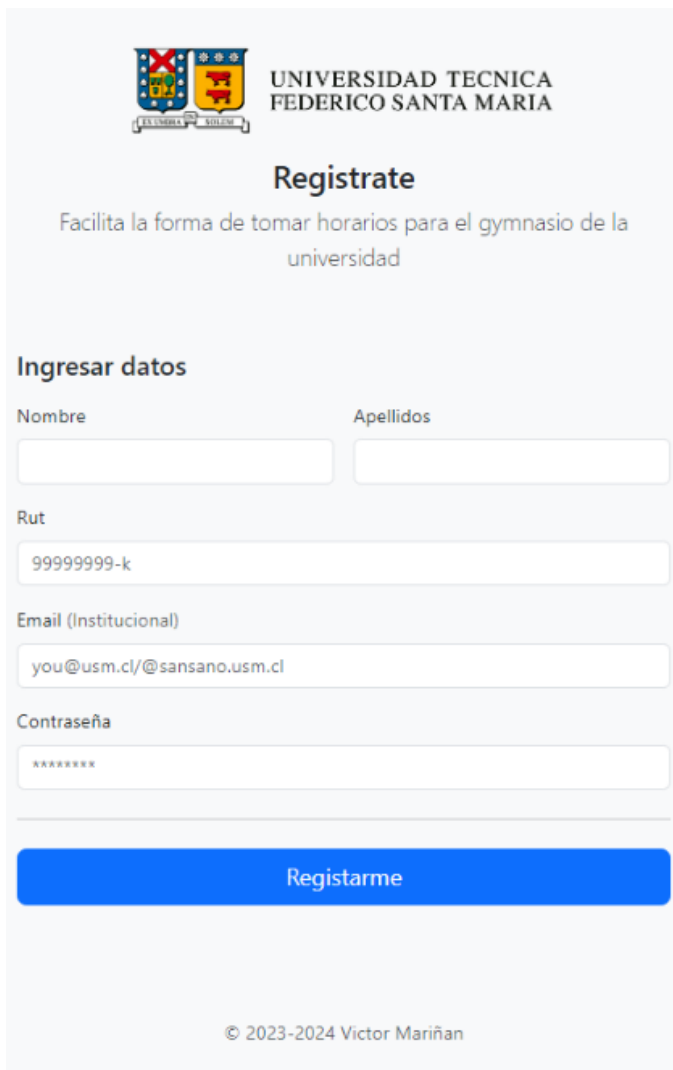
Sign in

©2023

Esta interfaz representa el inicio de sesión, el usuario deberá ingresar sus datos para poder acceder al menú principal de la página web, que le permitirá realizar distintas funciones estando una vez logueado. Al no tener cuenta esta le permitirá registrarse llevándolo a otra interfaz de registro.

Imagen 5. Inicio de sesión

5.2. Registro



The image shows a registration form for the Universidad Técnica Federico Santa María Gym. At the top, there is the university's logo and name. Below this, the title 'Registrate' is displayed, followed by a subtitle 'Facilita la forma de tomar horarios para el gymnasio de la universidad'. The form is titled 'Ingresar datos' and contains several input fields: 'Nombre' and 'Apellidos' (Name and Surname), 'Rut' (with a placeholder '99999999-k'), 'Email (Institucional)' (with a placeholder 'you@usm.cl/@sansano.usm.cl'), and 'Contraseña' (with a placeholder '*****'). A blue button labeled 'Registarme' is positioned below the fields. At the bottom, there is a copyright notice: '© 2023-2024 Victor Mariñan'.

**UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA**

Registrate

Facilita la forma de tomar horarios para el gymnasio de la universidad

Ingresar datos

Nombre

Apellidos

Rut

Email (Institucional)

Contraseña

Registarme

© 2023-2024 Victor Mariñan

Imagen 6. Registro de usuario

Esta interfaz representa el registro, el usuario deberá ingresar sus datos para poder crear su perfil registrándose en la página web.

5.3. Mostrar Horarios



Horarios sala de musculación

Lunes

☐

8:30-9:30pm

☐

9:30-10:30pm

☐

10:30-11:30pm

☐

11:30-12:30pm

☐

12:30-13:30pm

☐

Almuerzo

Cerrado

☐

14:30-15:30pm

☐

15:30-16:30pm

☐

16:30-17:30pm

☐

17:30-18:30pm

☐

18:30-19:30pm

Martes

☐

8:30-9:30pm

☐

9:30-10:30pm

☐

10:30-11:30pm

☐

11:30-12:30pm

☐

12:30-13:30pm

☐

Almuerzo

Cerrado

☐

14:30-15:30pm

☐

15:30-16:30pm

☐

16:30-17:30pm

☐

17:30-18:30pm

☐

18:30-19:30pm

Miercoles

☐

8:30-9:30pm

☐

9:30-10:30pm

☐

10:30-11:30pm

☐

11:30-12:30pm

☐

12:30-13:30pm

☐

Almuerzo

Cerrado

☐

14:30-15:30pm

☐

15:30-16:30pm

☐

16:30-17:30pm

☐

17:30-18:30pm

☐

18:30-19:30pm

Jueves

☐

8:30-9:30pm

☐

9:30-10:30pm

☐

10:30-11:30pm

☐

11:30-12:30pm

☐

12:30-13:30pm

☐

Almuerzo

Cerrado

☐

14:30-15:30pm

☐

15:30-16:30pm

☐

16:30-17:30pm

☐

17:30-18:30pm

☐

18:30-19:30pm

Viernes

☐

8:30-9:30pm

☐

9:30-10:30pm

☐

10:30-11:30pm

☐

11:30-12:30pm

☐

12:30-13:30pm

☐

Almuerzo

Cerrado

☐

14:30-15:30pm

☐

15:30-16:30pm

☐

16:30-17:30pm

☐

17:30-18:30pm

☐

18:30-19:30pm

Imagen 7. Mostrar horarios

En esta interfaz mostrara todos los horarios que el gym está operativo para el usuario vea las opciones que más le acomoden en cuanto a su disponibilidad y horario académico.

5.4. Tomar horario



Horarios sala de musculación

Lunes	Martes	Miercoles	Jueves	Viernes
☐ 8:30-9:30pm	☐ 8:30-9:30pm	☐ 8:30-9:30pm	☐ 8:30-9:30pm	☒ 8:30-9:30pm
☐ 9:30-10:30pm	☐ 9:30-10:30pm	☒ 9:30-10:30pm	☐ 9:30-10:30pm	☐ 9:30-10:30pm
☒ 10:30-11:30pm	☐ 10:30-11:30pm	☐ 10:30-11:30pm	☐ 10:30-11:30pm	☐ 10:30-11:30pm
☐ 11:30-12:30pm	☒ 11:30-12:30pm	☐ 11:30-12:30pm	☐ 11:30-12:30pm	☐ 11:30-12:30pm
☐ 12:30-13:30pm	☐ 12:30-13:30pm	☐ 12:30-13:30pm	☐ 12:30-13:30pm	☐ 12:30-13:30pm
☐ Almuerzo	☐ Almuerzo	☐ Almuerzo	☐ Almuerzo	☐ Almuerzo
☐ Cerrado	☐ Cerrado	☐ Cerrado	☐ Cerrado	☐ Cerrado
☐ 14:30-15:30pm	☐ 14:30-15:30pm	☐ 14:30-15:30pm	☐ 14:30-15:30pm	☐ 14:30-15:30pm
☐ 15:30-16:30pm	☐ 15:30-16:30pm	☐ 15:30-16:30pm	☐ 15:30-16:30pm	☐ 15:30-16:30pm
☐ 16:30-17:30pm	☐ 16:30-17:30pm	☐ 16:30-17:30pm	☐ 16:30-17:30pm	☐ 16:30-17:30pm
☐ 17:30-18:30pm	☐ 17:30-18:30pm	☐ 17:30-18:30pm	☐ 17:30-18:30pm	☐ 17:30-18:30pm
☐ 18:30-19:30pm	☐ 18:30-19:30pm	☐ 18:30-19:30pm	☐ 18:30-19:30pm	☐ 18:30-19:30pm

Tomar horarios seleccionados

Imagen 8. Toma de horarios

Esta interfaz muestra la toma de horarios que el usuario podrá realizar una vez ingresado en el sistema de la página web.

5.5 Confirmación de reservas



Reservas realizadas

Numero de reserva	Nombre	Apellidos	Horario		
1	Victor	Mariñan Sepulveda	01-12-2023 08:30		
1	Victor	Mariñan Sepulveda	02-12-2023 09:30		

Deseas confirmar reserva estas seguro? Si, confirmar No gracias	Deseas cancelar reserva estas seguro? Si, confirmar No gracias
--	---

Imagen 9. Reservas realizadas

En esta interfaz esta opción a poder confirmar y/o cancelar los horarios de las reservas que el usuario ha agendado para asistir al gym de la universidad.

5.6 Tipos de entrenamiento



Esta interfaz representa el tipo de ejercicio que el usuario desee realizar al asistir al gym de la universidad, con el fin de que el profesor encargado realice la rutina optima en cuanto a ejercicio seleccionado en la página web.

Imagen 10. Entrenamientos

5.7 Notificaciones

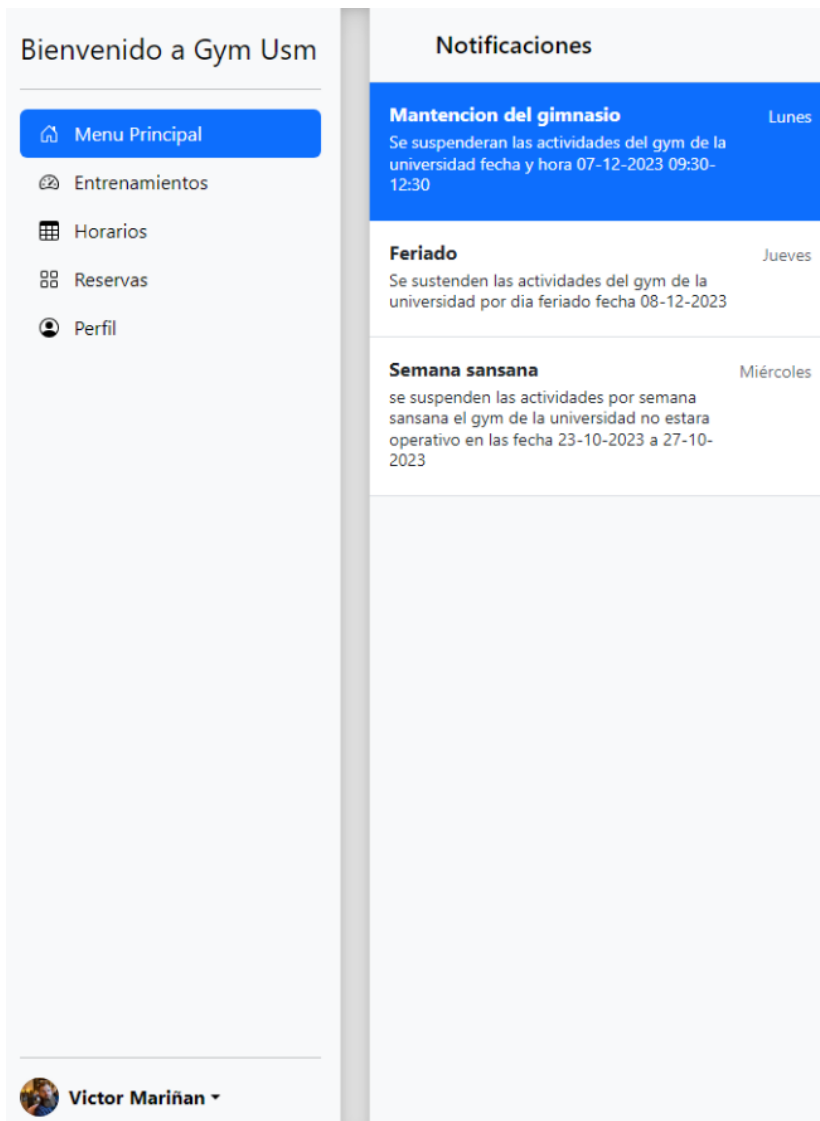


Imagen 11. Notificaciones

En esta interfaz se puede visualizar las notificaciones el cual dará avisos importantes de acuerdo con los acontecimientos y/o eventos que pueden ocurrir en la universidad informando sobre la operabilidad del gym de la universidad.

5.8. Gestión de usuarios



Gestión de Usuarios

Numero de reserva	Nombre	Apellidos	Horario		
1	Víctor	Mariñan Sepulveda	01-12-2023 08:30	✗	✓
2	Eitan	Candia Seguel	02-12-2023 09:30	✗	✓
3	Alexis	Mendoza Aravena	02-12-2023 09:30	✗	✓

Imagen 12. Gestión de usuarios

Esta interfaz está dedicada para el administrador que en este caso sería el profesor a cargo de la sala de musculación con el fin de obtener un registro de los usuarios que están ingresando en determinada fecha y hora también puede eliminar registros en caso de que no asista para otorgar cupo a otro usuario que desee ingresar.

5.9. Ingreso de usuarios manual



Ingresar usuario Manualmente

Conviertete en la mejor version de ti

Rellenar datos

Nombre Apellidos

Email (Institucional)

Fecha

Hora

Esta interfaz está dedicada para el administrador que en este caso sería el profesor a cargo de la sala de musculación con el fin de ingresar manualmente usuarios que no tengan el conocimiento de la pagina web y que deseen realizar sus ejercicios.

Imagen 13. Ingreso de usuarios manual

5.10. Crear eventos para notificar a los usuarios



Ingresar evento a compartir

dia	
Titulo del evento	
Descripción del evento	
Compartir evento	

Imagen 14. Ingresar evento a compartir

Esta interfaz está dedicada para el administrador que en este caso sería el profesor a cargo de la sala de musculación, con el fin de crear eventos que informen a los usuarios sobre las fechas y horas en que la sala de musculación no estará disponible por múltiples motivos.

CAPÍTULO 6: DESARROLLO DE LA SISTEMA WEB “GYM USM”

6.1. PostgreSQL



Imagen15.PostgreSQL.

PostgreSQL, también conocido como Postgres, es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL, similar a la BSD o MIT.

PostgreSQL es una base de datos muy flexible y escalable, que se puede utilizar para una amplia gama de aplicaciones, incluyendo:

- Aplicaciones web
- Aplicaciones empresariales
- Aplicaciones científicas

PostgreSQL es conocido por su rendimiento, seguridad y confiabilidad. Es una opción popular para aplicaciones críticas que requieren un alto nivel de disponibilidad.

Tiene una gran comunidad de usuarios y desarrolladores, lo que significa que hay una gran cantidad de recursos disponibles.

Ejemplos de empresas que utilizan PostgreSQL:

- Netflix utiliza PostgreSQL para almacenar datos de sus películas y programas de televisión.
- Airbnb utiliza PostgreSQL para almacenar datos de sus propiedades y reservas.
- Uber utiliza PostgreSQL para almacenar datos de sus conductores y pasajeros.

6.2. Amazon Web Services

AWS, o Amazon Web Services, es una plataforma de computación en la nube que ofrece una amplia gama de servicios, incluyendo:

- Computación
- Almacenamiento
- Bases de datos
- Red
- Análisis
- Inteligencia artificial
- Machine learning
- Internet de las cosas
- Desarrollo
- Pruebas
- Entrega

AWS permite a las empresas de todos los tamaños crear y ejecutar aplicaciones y servicios en la nube. La plataforma ofrece una gran flexibilidad y escalabilidad, lo que permite a las empresas adaptarse a sus necesidades cambiantes.

Algunos de los beneficios de usar AWS incluyen:

- Ahorro de costes: AWS puede ayudar a las empresas a ahorrar dinero en infraestructura y costes de mantenimiento.
- Flexibilidad: AWS ofrece una amplia gama de servicios que pueden adaptarse a las necesidades de cualquier empresa.
- Escalabilidad: AWS permite a las empresas escalar sus recursos según sea necesario.

AWS es una plataforma poderosa que puede ayudar a las empresas a mejorar su eficiencia y reducir sus costes.

Ejemplos de empresas que utiliza AWS:

Netflix utiliza AWS para alojar su servicio de streaming de vídeo.

Airbnb utiliza AWS para alojar su plataforma de alquiler de alojamientos.

Uber utiliza AWS para gestionar su flota de vehículos.



Imagen 16. Amazon.

6.3. Python



Imagen 17. Python.

Python es un lenguaje de programación de alto nivel, de propósito general, interpretado, de tipado dinámico, multiparadigma. Es un lenguaje de programación relativamente nuevo, pero ha ganado una popularidad sin precedentes en los últimos años.

Python es un lenguaje de programación muy versátil, que se puede utilizar para una amplia gama de aplicaciones, incluyendo:

- Desarrollo web
- Desarrollo de aplicaciones móviles
- Ciencia de datos
- Aprendizaje automático
- Inteligencia artificial

También es un lenguaje fácil de aprender, lo que lo convierte en una buena opción para principiantes. También es un lenguaje de programación muy eficiente, lo que lo hace adecuado para aplicaciones de alto rendimiento.

ventajas de usar Python incluyen:

- Es un lenguaje de programación fácil de aprender.
- Es un lenguaje de programación muy eficiente.
- Es un lenguaje de programación versátil, que se puede utilizar para una amplia gama de aplicaciones.
- Tiene una gran comunidad de usuarios y desarrolladores, lo que significa que hay una gran cantidad de recursos disponibles.

Ejemplos de empresas que utiliza Python:

- YouTube utiliza Python para desarrollar su plataforma de video.
- Instagram utiliza Python para desarrollar su aplicación de redes sociales.
- Spotify utiliza Python para desarrollar su plataforma de transmisión de música.
- Netflix utiliza Python para desarrollar su plataforma de transmisión de películas y programas de televisión.
- Airbnb utiliza Python para desarrollar su plataforma de alojamiento.

Características claves de Python:

- Interpretado: Python se interpreta, lo que significa que no necesita ser compilado antes de ejecutarse. Esto hace que Python sea más rápido de desarrollar que los lenguajes compilados.
- De tipado dinámico: Python tiene un sistema de tipado dinámico, lo que significa que no es necesario declarar el tipo de datos de las variables. Esto hace que Python sea más fácil de aprender que los lenguajes de tipado estático.
- Multiparadigma: Python es un lenguaje de programación multiparadigma, lo que significa que se puede utilizar para desarrollar aplicaciones utilizando diferentes paradigmas de programación, como la programación orientada a objetos, la programación funcional y la programación imperativa.

6.3. Bootstrap

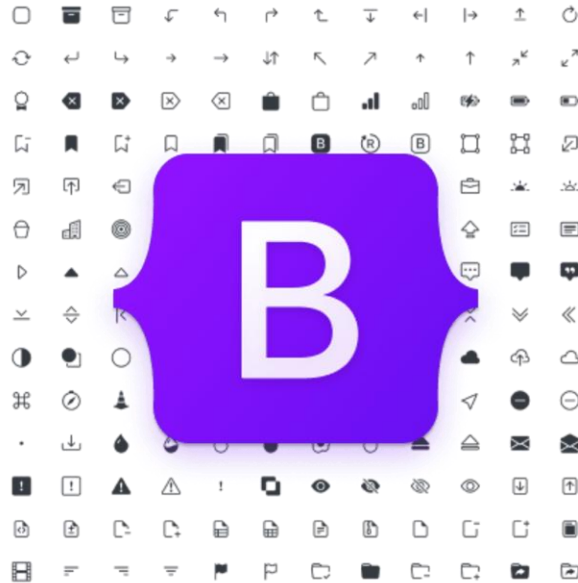


Imagen 18. Bootstrap.

Bootstrap es un framework de desarrollo web gratuito y de código abierto basado en HTML, CSS y JavaScript. Se utiliza para crear sitios web y aplicaciones web responsivas, es decir, que se adaptan automáticamente a diferentes tamaños de pantalla.

Bootstrap proporciona una amplia gama de componentes y herramientas para crear sitios web y aplicaciones web, incluyendo:

- CSS: Bootstrap proporciona una biblioteca de clases CSS para diseñar el aspecto de un sitio web o aplicación web.
- JavaScript: Bootstrap proporciona una biblioteca de JavaScript para agregar funcionalidad a un sitio web o aplicación web.
- HTML: Bootstrap proporciona un conjunto de componentes HTML para crear elementos de interfaz de usuario comunes.

Es muy popular para el desarrollo web, utilizada por empresas de todos los tamaños. Es una buena opción para desarrolladores web que buscan crear sitios web y aplicaciones web responsive de forma rápida y sencilla.

Características claves de Bootstrap:

- **Responsive:** Bootstrap está diseñado para ser responsive, lo que significa que se adapta automáticamente a diferentes tamaños de pantalla.
- **Personalizable:** Bootstrap es muy personalizable, lo que permite a los desarrolladores crear sitios web y aplicaciones web que se adapten a sus necesidades específicas.
- **Documentado:** Bootstrap está bien documentado, lo que facilita a los desarrolladores aprender a usarlo.
- **Comunidad:** Bootstrap tiene una gran comunidad de usuarios y desarrolladores, lo que significa que hay una gran cantidad de recursos disponibles.

Ejemplos de empresas que utilizan Bootstrap:

- **Twitter:** Twitter utiliza Bootstrap para diseñar su sitio web y su aplicación móvil.
- **LinkedIn:** LinkedIn utiliza Bootstrap para diseñar su sitio web y su aplicación móvil.
- **Medium:** Medium utiliza Bootstrap para diseñar su sitio web y su aplicación móvil.
- **Shopify:** Shopify utiliza Bootstrap para diseñar su plataforma de comercio electrónico.

6.3. Flask



Imagen 19. Flask.

Flask es un framework web micro escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.

Flask es una buena opción para desarrolladores web que buscan crear aplicaciones web sencillas y de código limpio. Es un framework ligero y flexible, que permite a los desarrolladores centrarse en la lógica de la aplicación.

Características claves de Flask:

- **Microframework:** Flask es un microframework, lo que significa que proporciona un conjunto básico de funciones para crear aplicaciones web. Esto permite a los desarrolladores centrarse en la lógica de la aplicación, sin tener que preocuparse por las características que no necesitan.
- **Código limpio:** Flask está diseñado para escribir código limpio y legible. Esto hace que sea más fácil mantener y depurar las aplicaciones.

- Flexible: Flask es muy flexible, lo que permite a los desarrolladores adaptarlo a sus necesidades específicas.
- Documentado: Flask está bien documentado, lo que facilita a los desarrolladores aprender a usarlo.
- Comunidad: Flask tiene una gran comunidad de usuarios y desarrolladores, lo que significa que hay una gran cantidad de recursos disponibles.

Aquí hay algunos ejemplos de empresas que utilizan Flask:

- Blogs: Flask se utiliza para crear blogs simples y de código limpio.
- Aplicaciones web de una sola página: Flask se utiliza para crear aplicaciones web de una sola página, que se cargan en una sola consulta HTTP.
- APIs: Flask se utiliza para crear APIs RESTful, que permiten a las aplicaciones comunicarse entre sí.

CAPÍTULO 7: PROPUESTA DE SOLUCION

La solución propuesta para mejorar la gestión de horarios en el gimnasio universitario se basa en el desarrollo de una plataforma web innovadora y fácil de usar, diseñada específicamente para abordar las limitaciones del sistema actual. La metodología empleada se basa en un enfoque ágil de desarrollo de software, permitiendo una adaptabilidad continua a medida que evolucionan los requisitos y las necesidades del usuario.

Diseño de la Interfaz de Usuario (UI): Se creará una interfaz de usuario intuitiva y atractiva, centrada en la experiencia del usuario. Los usuarios podrán navegar fácilmente a través de los horarios disponibles y realizar reservas con unos pocos clics. Se implementarán principios de diseño responsivo para garantizar una experiencia coherente en dispositivos móviles y computadoras de escritorio.

Desarrollo Front-end y Back-end: Se utilizarán tecnologías modernas de desarrollo web, como Python. Esto permitirá la creación de una aplicación web dinámica y eficiente, con funcionalidades interactivas que responden en tiempo real a las acciones del usuario.

Base de Datos Segura y Eficiente: Se implementará una base de datos relacional segura para almacenar la información de los usuarios y los horarios de reserva. Se aplicarán técnicas de cifrado y medidas de seguridad robustas para proteger la integridad de los datos del usuario, y la herramienta a utilizar sería PostgreSQL.

Funcionalidades Clave

Registro de Usuarios: Los usuarios podrán crear cuentas personalizadas, proporcionando información básica y credenciales seguras para el acceso.

Exploración de Horarios: Los usuarios podrán ver horarios disponibles por día y hora, con filtros para actividades específicas y entrenadores disponibles.

Reservas Instantáneas: Las reservas se podrán hacer de forma instantánea con confirmaciones automáticas enviadas por correo electrónico y notificaciones a través de la plataforma.

Gestión de Reservas: Los usuarios tendrán la opción de ver, modificar o cancelar sus reservas existentes, lo que garantizará flexibilidad en caso de cambios de planes.

Pruebas y Optimización: Se llevarán a cabo pruebas exhaustivas para asegurar la funcionalidad y la seguridad de la plataforma. Se realizarán pruebas de carga para evaluar el rendimiento del sistema bajo condiciones de uso intensivo. Las iteraciones se llevarán a cabo para abordar cualquier problema identificado durante las pruebas.

Documentación Detallada: Se proporcionará una documentación completa que incluirá manuales de usuario, descripciones técnicas y guías de mantenimiento para garantizar una comprensión clara de la plataforma tanto para los usuarios finales como para los administradores del sistema.

CAPÍTULO 8: VALIDACIÓN DE LA SOLUCIÓN

1. Definir Criterios de Validación:

Establece criterios claros y específicos para evaluar el éxito de la solución. Esto podría incluir la facilidad de uso, la velocidad de reserva, la tasa de adopción por parte de los usuarios, la precisión en las reservas, entre otros.

2. Desarrollar Casos de Prueba:

Crea una serie de casos de prueba que cubran diferentes escenarios de uso de la plataforma. Esto puede incluir casos de usuarios nuevos registrándose con un límite a tal punto que estresen el sistema y el modelo subyacente verificando que, si tolera con éxito o fallan las modificaciones y cancelaciones de reservas, entre otros.

3. Implementar Pruebas Piloto:

Lanza una versión piloto de la plataforma en un entorno controlado. Invita a un grupo selecto de usuarios representativos para probar la plataforma y proporcionar retroalimentación. Observa su interacción y registra cualquier problema que encuentren.

4. Recopilar Retroalimentación de los Usuarios:

Realiza encuestas y entrevistas con los usuarios que participaron en las pruebas piloto. Pregunta sobre su experiencia, cualquier problema que hayan enfrentado y sus sugerencias para mejoras. La retroalimentación cualitativa y cuantitativa es crucial para entender la percepción del usuario.

5. Realizar Pruebas de Estrés y Rendimiento:

Somete la plataforma a pruebas de carga para evaluar su rendimiento bajo condiciones de uso intensivo. Esto te ayudará a identificar posibles cuellos de botella y a garantizar que el sistema pueda manejar múltiples usuarios simultáneamente sin problemas de rendimiento.

6. Comparar con la Situación Anterior:

Realiza una comparación entre la situación anterior, donde se utilizaba el sistema basado en Excel, y la situación actual con la nueva plataforma implementada. Evalúa los indicadores cuantitativos, como el tiempo dedicado a la reserva, la precisión y la satisfacción del usuario.

7. Iterar y Mejorar:

Basándote en la retroalimentación y los resultados obtenidos, realiza ajustes y mejoras en la plataforma. Las soluciones tecnológicas a menudo requieren iteraciones para alcanzar su máximo potencial. Escucha atentamente a los usuarios y ajusta la plataforma según sus necesidades y expectativas.

8. Documentar los Resultados:

Documenta detalladamente los resultados de las pruebas, las mejoras realizadas y la retroalimentación de los usuarios. Esta documentación servirá como evidencia de la validación de la solución en tu memoria.

CAPÍTULO 9: CONCLUSIONES

En el transcurso de este proyecto, se ha abordado de manera integral el desafío de mejorar la gestión de horarios en el gimnasio universitario. A través de un análisis detenido del sistema anterior basado en hojas de cálculo y una cuidadosa planificación, se diseñó y se desarrollara una solución digital innovadora y efectiva. Esta solución, basada en un sistema de gestión de horarios intuitivo y fácil de usar, representa un paso significativo hacia la optimización de la experiencia de los usuarios y la promoción de un estilo de vida activo y saludable en nuestra comunidad universitaria.

La implementación de la plataforma pretende demostrar su valía durante las pruebas piloto y las pruebas de estrés. Esperar que los usuarios respondan positivamente, destacando la facilidad de reserva, la precisión y la rapidez del sistema. Además, se observaron mejoras significativas en la eficiencia del proceso de reserva y en la satisfacción general de los usuarios, en comparación con el sistema anterior que solo era un Excel modificado.

Es importante subrayar que este proyecto no solo representa una mejora tecnológica, sino también un cambio cultural en la forma en que nuestra comunidad universitaria interactúa con los servicios del gimnasio. La implementación de esta solución ha allanado el camino para una mayor participación en actividades físicas, fomentando un ambiente más saludable y activo para todos los usuarios.

Como resultado de este proyecto, se ha logrado un hito significativo en la mejora de los servicios ofrecidos por el gimnasio universitario. Sin embargo, esta implementación marca el comienzo de un viaje continuo. Se deben seguir recopilando datos y comentarios de los usuarios para realizar ajustes y mejoras periódicas, asegurando que la plataforma siga siendo relevante y efectiva en el futuro.

En última instancia, este proyecto representa un testimonio del poder de la innovación tecnológica y la dedicación a la mejora continua. Al proporcionar una solución práctica y eficiente para un problema cotidiano, este proyecto no solo ha mejorado la gestión de horarios, sino que también ha contribuido a fortalecer el sentido de comunidad y bienestar en nuestra universidad. Este trabajo es un paso adelante hacia un campus más saludable, activo y conectado.

La implementación del sistema de gestión de horarios en la Universidad Técnica Federico Santa María utilizando Python con Flask y Bootstrap representa un avance significativo hacia la modernización y eficiencia de los procesos en el gimnasio universitario. La combinación de Flask como marco de desarrollo web en el lado del servidor y Bootstrap para el diseño del lado del cliente proporciona una plataforma robusta y atractiva que atiende las necesidades de los usuarios de manera efectiva.

Funcionamiento del Sistema:

La arquitectura basada en Flask permite la creación de rutas y controladores que gestionan las solicitudes de los usuarios, facilitando la manipulación de datos y la interacción con la base de datos de PostgreSQL.

El uso de Bootstrap proporciona una interfaz de usuario receptiva y amigable, lo que facilita la navegación y la realización de acciones como la reserva de horarios.

El proyecto busca como finalidad

- Optimización de Procesos:

La automatización de la gestión de horarios elimina la dependencia de métodos manuales, como hojas de cálculo, reduciendo el riesgo de errores y mejorando la eficiencia del personal.

- Mejora de la Experiencia del Usuario:

La interfaz intuitiva y atractiva creada con Bootstrap mejora la experiencia del usuario, facilitando la navegación y la realización de reservas de horarios de manera rápida y sencilla.

- Accesibilidad y Disponibilidad:

El sistema, al ser basado en la web, brinda a los usuarios la flexibilidad de acceder y realizar reservas desde cualquier dispositivo con conexión a Internet, mejorando la accesibilidad y la disponibilidad del servicio.

- Experiencia del Usuario:

El diseño con Bootstrap garantiza una interfaz atractiva y moderna, lo que contribuye a una experiencia visualmente agradable para los usuarios.

- Proceso Simplificado:

La navegación fluida y las funcionalidades intuitivas simplifican el proceso de reserva, proporcionando a los usuarios una experiencia sin complicaciones.

Notificaciones Instantáneas:

La implementación de notificaciones instantáneas en tiempo real, posiblemente a través de WebSocket, mantendrá a los usuarios informados sobre cambios en los horarios y confirmaciones de reservas.

En conclusión, el desarrollo de este sistema utilizando tecnologías como Python, Flask y Bootstrap no solo representa un avance tecnológico, sino que también tiene un impacto significativo en la eficiencia operativa y la satisfacción del usuario. La combinación de un back-end sólido con un front-end atractivo crea un entorno que facilita la participación de los usuarios y mejora la calidad de los servicios ofrecidos por el gimnasio universitario. Este proyecto sienta las bases para un futuro más digital, accesible y centrado en el usuario en la gestión de horarios.

También la elección de PostgreSQL como motor de bases de datos para el proyecto de gestión de horarios ha demostrado ser una decisión acertada y estratégica. La robustez, versatilidad y escalabilidad de PostgreSQL han contribuido significativamente a la eficacia y fiabilidad del sistema, proporcionando una base de datos sólida para respaldar las operaciones críticas del gimnasio universitario y para realizar mejoras continuamente.

También la elección de PostgreSQL como motor de bases de datos para el proyecto de gestión de horarios de la universidad ha demostrado ser una decisión acertada y altamente beneficiosa. La arquitectura optimizada y las capacidades de optimización de consultas garantizan un rendimiento eficiente, la implementación de integridad referencial y restricciones asegura la coherencia y precisión de los datos, y el soporte para transacciones complejas es esencial en entornos concurrentes. La escalabilidad, flexibilidad y la facilidad de integración con herramientas como Flask contribuyen a una experiencia del desarrollador fluida. Las herramientas de administración robustas facilitan la gestión continua, y las consideraciones futuras, como la resiliencia mediante copias de seguridad y la mejora del rendimiento, aseguran la sostenibilidad y evolución del sistema. En conjunto, PostgreSQL ha establecido una base sólida para un sistema eficiente, confiable y preparado para el crecimiento futuro.

Con AWS, se logra una gestión eficiente de recursos, permitiendo escalabilidad según las necesidades del sistema. La facilidad de integración con servicios como Amazon RDS simplifica la administración de la base de datos, garantizando resiliencia mediante copias de seguridad automatizadas y proporcionando herramientas de monitoreo avanzadas. En resumen, la combinación de PostgreSQL y AWS no solo fortalece la infraestructura tecnológica del proyecto, sino que también asegura un despliegue eficiente y sostenible en la nube, respaldando una gestión de horarios moderna y efectiva.

Con servicios como Amazon EC2 y Amazon RDS, se logra una implementación eficiente y segura, respaldada por una arquitectura de red robusta. La flexibilidad y capacidad de escalabilidad de AWS permiten adaptarse dinámicamente a las demandas del usuario, asegurando una experiencia continua. La gestión simplificada de recursos y la posibilidad de integrar servicios adicionales de AWS para mejorar la funcionalidad refuerzan la solidez del despliegue. En última instancia, la elección de AWS no solo garantiza la fiabilidad operativa de la plataforma, sino que también sienta las bases para futuras expansiones y mejoras, respaldando así la visión a largo plazo del sistema de gestión de horarios.

Por último, un profundo reconocimiento mis profesores que han sido parte fundamental de mi carrera universitaria y desarrollo profesional. A cada uno de ellos que han desempeñado un papel crucial, guiándome, inspirándome, apoyándome y desafiándome a alcanzar mis metas académicas y profesionales. Su dedicación a la enseñanza, mentoría y orientación ha sido un faro que ha iluminado mi camino educativo, brindándome conocimientos y perspectivas valiosas. A través de su apoyo constante, he adquirido no solo conocimientos técnicos, sino también habilidades de resolución de problemas, ética laboral y una comprensión más profunda de mi campo de estudio. Este reconocimiento no solo se dirige a los profesores que impartieron clases, sino también a aquellos que colaboraron en proyectos, brindaron asesoramiento académico y compartieron su experiencia profesional.

Su influencia ha trascendido las aulas, impactando positivamente mi desarrollo personal y profesional. En cada desafío y éxito, puedo trazar un hilo que conecta directamente a la orientación sabia de mis profesores. Agradezco sinceramente su dedicación y contribución a mi formación, reconociendo que sus enseñanzas perdurarán a lo largo de mi carrera y más allá.

REFERENCIAS BIBLIOGRÁFICAS

Magno, A. (2015). Bootstrap in Practice. Manning Publications.

S. Jones, "The Future of Artificial Intelligence," presentado en International Conference on Technology Trends, New York, NY, 2018.

E. Matthes, "Python Crash Course ". No Starch Press, 2015.

Gkamas, Theodosios, Vasileios Karaiskos y Sotirios Kontogiannis. "Performance Evaluation of Distributed Database Strategies Using Docker as a Service for Industrial IoT Data: Application to Industry 4.0". Information 13, n.º 4 , 9 de abril de 2022.