

Memoria del proyecto

Plataforma para el seguimiento y la realización de ejercicios físicos.

Trabajo de Fin de Grado

INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
D SALAMANCA**

Julio de 2020

Autor:

Carlos Valdunciel Gonzalo

Tutores:

Héctor Sánchez San Blas

Juan Francisco de Paz Santana

Gabriel Villarrubia González

Memoria del proyecto

Memoria del proyecto

CERTIFICADO DE LOS TUTORES

D. Héctor Sánchez San Blas, D. Gabriel Villarrubia González y D. Juan Francisco De Paz Santana, profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

HACEN CONSTAR:

Que el trabajo titulado “Plataforma para el seguimiento y realización de ejercicios físicos” ha sido realizada por D. Carlos Valdunciel Gonzalo, con el número de documento 70925424W y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado de Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 7 de julio de 2022

D. Héctor Sánchez San
Blas

D. Gabriel Villarrubia
González

D. Juan Francisco De Paz
Santana

Memoria del proyecto

Memoria del proyecto

Resumen

El deporte siempre ha sido muy importante para mantener una buena salud, ya sea yendo al gimnasio, practicando deportes de equipo, individuales o incluso dar un simple paseo. Sin embargo, hay que destacar que no todo el mundo posee los conocimientos para realizarlo correctamente, no encuentra el momento de empezar o alguien que le guíe en este proceso que, en un inicio, puede ser complicado.

Con el objetivo de dar solución a estos problemas nace el presente proyecto. Una aplicación web que permita aprender ejercicios y rutinas sencillas mediante un proceso monitorizado. Es ideal para personas con poca experiencia en el mundo del deporte y de cara a la rehabilitación de personas mayores que, normalmente, tienen dificultades para acceder o entender este tipo de servicios. Mediante una interfaz sencilla se facilita esta tarea al usuario y se analizan los movimientos para detectar malas posturas o fallos que se comenten en realización del ejercicio.

Teniendo en cuenta que a veces es difícil mantener la constancia, la aplicación presenta un sistema de ranking, para que los usuarios compitan por estar en lo más alto, siendo esto una simple excusa para incentivar la realización de ejercicio físico con la finalidad de crear un hábito saludable en las personas ayudándolas a introducirse en el mundo del deporte. Esto último está enfocado a los usuarios que no usan la plataforma como rehabilitación por lesiones o por enfermedades derivadas de la edad, como puede ser la artrosis.

Palabras clave: deporte, ejercicio, monitorizado, rutina, rehabilitación.

Memoria del proyecto

Memoria del proyecto

Summary

Sport has always been very important for maintaining good health, whether it is going to the gym, practicing team sports, individual sports or even just going for a walk. However, it should be noted that not everyone has the knowledge to do it correctly, they do not find the time to start or someone to guide them in this process, which can be complicated at first.

With the aim of providing a solution these problems, the present project was born. A web application that allows you to learn simple exercises and routines through a monitored process. It is ideal for people with little experience in the world of sports and for the rehabilitation of elderly people who normally have difficulties in accessing or understanding this type of services. By means of a simple interface, this task is made easier for the user, and the movements are also analysed to detect bad postures or mistakes during the performance of a specific exercise.

Bearing in mind that sometimes it is difficult to maintain constancy, the application presents a ranking system, so that users compete to be at the top, this being a simple excuse to encourage the performance of physical exercise in order to create a healthy habit in people helping them to enter the world of sports. The latter is focused on users who do not use the platform for rehabilitation purposes due to injuries or age-related illnesses, such as osteoarthritis.

Keywords: sport, exercise, monitored, routine, rehabilitation.

Memoria del proyecto

Memoria del proyecto

Tabla de Contenido

1.	INTRODUCCIÓN.....	14
2.	ESTADO DEL ARTE	15
3.	OBJETIVOS.....	19
3.1.	OBJETIVOS DEL SISTEMA.....	19
3.2.	OBJETIVOS PERSONALES	19
4.	CONCEPTOS TEÓRICOS.....	20
4.1.	HERRAMIENTAS CASE.....	20
4.2.	VALOR MET	21
4.3.	REST.....	21
5.	TÉCNICAS Y HERRAMIENTAS	22
5.1.	HTML	22
5.2.	CSS.....	22
5.3.	JAVASCRIPT	22
5.4.	ML5.....	23
5.5.	P5.....	23
5.6.	PHP.....	23
5.7.	PHP MAILER.....	24
5.8.	APACHE.....	24
5.9.	MARIADB.....	24
5.10.	PHPMYADMIN	25
5.11.	VISUAL STUDIO CODE.....	25
5.12.	REMOTE - SSH.....	26
5.13.	HERRAMIENTAS CASE.....	26
5.13.1.	VISUAL PARADIGM	26
5.13.2.	EZESTIMATE.....	27
5.13.3.	MICROSOFT PROJECT	28
5.13.4.	NGROK.....	29
5.14.	HERRAMIENTAS DE GENERACIÓN DE DOCUMENTACIÓN	30
6.	ASPECTOS RELEVANTES DEL DESARROLLO	31
6.1.	MARCO DE TRABAJO	31
6.2.	ESTIMACIÓN DE LA DURACIÓN DEL PROYECTO.....	32
6.3.	PLANIFICACIÓN TEMPORAL	34
6.4.	ESPECIFICACIÓN DE REQUISITOS	37
6.4.1.	PARTICIPANTES.....	38
6.4.2.	OBJETIVOS DEL SISTEMA.....	38

Memoria del proyecto

6.4.3.	REQUISITOS DE INFORMACIÓN.....	39
6.4.4.	REQUISITOS FUNCIONALES.....	40
6.4.5.	REQUISITOS NO FUNCIONALES.....	43
6.5.	ANÁLISIS DE REQUISITOS.....	44
6.5.1.	MODELO DE DOMINIO.....	44
6.5.2.	REALIZACIÓN DE CASOS DE USO EN EL MODELO DE ANÁLISIS.....	45
6.5.3.	CLASES DE ANÁLISIS.....	46
6.5.4.	VISTA DE ARQUITECTURA DEL MODELO DE ANÁLISIS.....	47
6.6.	DISEÑO DEL SISTEMA.....	47
6.6.1.	PATRONES ARQUITECTÓNICOS.....	47
6.6.2.	SUBSISTEMAS DE DISEÑO.....	48
6.6.3.	CLASES DE DISEÑO.....	49
6.6.4.	VISTA ARQUITECTÓNICA.....	51
6.6.5.	REALIZACIÓN DE CASOS DE USO.....	52
6.6.6.	DISEÑO DE LA BASE DE DATOS.....	53
6.6.7.	MODELO DE DESPLIEGUE.....	53
6.7.	IMPLEMENTACIÓN.....	54
6.8.	PRUEBAS.....	54
6.9.	FUNCIONALIDAD DEL SISTEMA.....	55
6.9.1.	GESTIÓN DE USUARIOS.....	55
6.9.2.	GESTIÓN DE ESTADÍSTICAS.....	58
6.9.3.	GESTIÓN DE LA PLATAFORMA.....	60
6.9.4.	GESTIÓN DE EJERCICIOS.....	63
6.9.5.	GESTIÓN DE RUTINAS.....	64
7.	CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS.....	68
7.1.	CONCLUSIONES.....	68
7.2.	LÍNEAS DE TRABAJO FUTURAS.....	68
8.	REFERENCIAS.....	70

Índice de Figuras

Figura 1: Aplicación web Kentai	15
Figura 2: Aplicación móvil - Ejercicios en casa	16
Figura 3: Adidas training	17
Figura 4: Virtuagym fitness	18
Figura 5: Entrenamiento en 7 minutos	18
Figura 6: Interfaz phpMyAdmin	25
Figura 7: Interfaz Visual Studio Code	26
Figura 8: Interfaz Visual Paradigm.....	27
Figura 9: Interfaz EZEstimate	28
Figura 10: Interfaz Microsoft Project	29
Figura 11: Interfaz de ngrok	29
Figura 12: Fases e iteraciones del Proceso Unificado	32
Figura 13: Casos de uso EZEstimate	32
Figura 14: Factores de complejidad técnica EZEstimate.....	33
Figura 15: Factores de complejidad de entorno EZEstimate	33
Figura 16: Estimación obtenida EZEstimate.....	34
Figura 17: Calendario de trabajo.....	35
Figura 18: Identificación de tareas e hitos	35
Figura 19: Asignación de recursos.....	36
Figura 20: Diagrama de Gantt	36
Figura 21: Camino crítico.....	37
Figura 22: Diagrama de paquetes	40
Figura 23: Diagrama de actores	40
Figura 24: Paquete Gestión de usuarios	41
Figura 25: Diagrama de clases del Modelo de dominio	44
Figura 26: Diagrama de secuencia Entrar	45
Figura 27: Diagrama de comunicación paquete Gestión de rutinas.....	46
Figura 28: Vista de arquitectura del modelo de análisis.....	47
Figura 29: Patrón Model-View-ViewModel (MVVM).....	48
Figura 30: Diagrama de subsistemas.....	48
Figura 31: Capa View.....	49
Figura 32: Capa View Model	50
Figura 33: Capa Model	50
Figura 34: Vista arquitectónica	51
Figura 35: Diagrama de secuencia Entrar	52
Figura 36: Base de datos	53
Figura 37: Modelo de despliegue.....	54
Figura 38: Pantalla de login	55
Figura 39: Pantalla de registro	56
Figura 40: Pantalla de perfil	56
Figura 41: Pantalla de modificar perfil.....	57
Figura 42: Pantalla solicitud email	58
Figura 43: Pantalla cambio de contraseña	58
Figura 44: Pantalla de inicio / estadísticas	59
Figura 45: Pantalla de ranking.....	60
Figura 46: Pantalla de panel de control	61

Memoria del proyecto

Figura 47: Buscar usuario	61
Figura 48: Listar administradores.....	62
Figura 49: Listar entrenadores	62
Figura 50: Cambiar rol	62
Figura 51: Cambiar estado de cuenta	62
Figura 52: Eliminar cuenta.....	63
Figura 53: Desplegable ejercicio	63
Figura 54: Información ejercicio.....	63
Figura 55: Realización de ejercicio	64
Figura 56: Listado de rutinas	64
Figura 57: Detalles de rutina	65
Figura 58: Creación de rutina	65
Figura 59: Listado de rutinas propias	66
Figura 60: Detalles de rutina propia.....	66
Figura 61: Renombrar rutina	66
Figura 62: Añadir ejercicio a rutina	67
Figura 63: Eliminar ejercicio de rutina	67
Figura 64: Eliminar rutina	67

Memoria del proyecto

Índice de Tablas

Tabla 1: Participante Valdunciel Gonzalo, Carlos.....	38
Tabla 2: OBJ-02 Altas, bajas, modificaciones y visualización de usuarios.....	39
Tabla 3: IRQ-01 Información sobre usuario	40
Tabla 4: ACT-02 Usuario logueado	41
Tabla 5: UC-06 Modificar perfil	42
Tabla 6: NFR-06 Hacer uso de la webcam	43

Memoria del proyecto

1. INTRODUCCIÓN

Cada vez hay más personas que se incorporan al mundo del deporte, sin embargo, gran parte de estas carecen de la experiencia necesaria para realizarlo correctamente o les da vergüenza pedir ayuda por pensar que los más experimentados no les ayudarían o son personas en rehabilitación que necesitan una especial atención por su situación.

Por estas razones, muchas acaban siendo autodidactas, lo que provoca abandonos prematuros por falta de conocimientos, molestias derivadas de realizar incorrectamente los ejercicios o falta de motivación al no saber muy bien qué hacer.

Este proyecto tiene como objetivo proporcionar una aplicación web a estos usuarios que les guíe en el proceso de realización de un ejercicio predeterminado mediante una monitorización y análisis de sus movimientos o de una rutina añadida al sistema por alguno de los entrenadores de la plataforma.

En este documento se explicarán los aspectos más importantes en el desarrollo del proyecto. Su estructura será la siguiente:

- **Estado de arte:** se realiza un estudio de mercado con aplicaciones similares existentes pertenecientes al mismo ámbito.
- **Objetivos:** se especifican los objetivos que se quieren alcanzar con la realización del proyecto.
- **Conceptos teóricos:** se explican los conceptos necesarios para entender correctamente el proyecto.
- **Técnicas y herramientas:** documentación de las técnicas y herramientas empleadas a lo largo de la realización del proyecto.
- **Aspectos relevantes del desarrollo:** se recogen los aspectos más importantes del desarrollo de la aplicación
- **Limitaciones de la aplicación:** se explican cuáles son las limitaciones técnicas encontradas al final del proyecto.
- **Conclusiones y líneas de trabajo futuras:** se expondrán las conclusiones obtenidas tras la finalización del proyecto y que ampliaciones podría tener este en un futuro.
- **Bibliografía.**

Este documento se complementa con los siguientes anexos:

- **Anexo I – Planificación temporal:** documento en que se realiza una estimación de la duración del proyecto y una planificación temporal de las tareas a llevar a cabo.
- **Anexo II – Especificación de requisitos:** documento en que se especifican los requisitos del sistema: objetivos, requisitos de información, requisitos funcionales y requisitos no funcionales.
- **Anexo III – Análisis de requisitos:** documentación sobre la fase de análisis del sistema.
- **Anexo IV – Diseño del sistema software:** documentación sobre la fase de diseño del sistema.
- **Anexo V – Documentación técnica:** documento en que se explica el código desarrollado para facilitar su comprensión.
- **Anexo VI – Manual de usuario:** documento en que se explica la funcionalidad de las distintas partes del sistema de cara a la utilización de este por el usuario final.

2. ESTADO DEL ARTE

Antes de escoger este proyecto, se realizó un estudio de mercado para determinar que campo no se encontraba muy explotado. Si bien las aplicaciones móviles para la realización de ejercicio físico de forma autónoma se cuentan por decenas, casi no existen aplicaciones web de este estilo. Cabe destacar que la mayoría de las aplicaciones móviles se limitan a una serie de rutinas, algunas con vídeos explicativos sobre la realización de los distintos ejercicios que las componen, pero no se ha encontrado ninguna que capture la posición corporal e indique los fallos en tiempo real. Esto también se debe a una limitación física, ya que las pantallas de los dispositivos móviles suelen tener un tamaño demasiado reducido como para incorporar esta funcionalidad. Entre las aplicaciones analizadas se encuentran:

- **Kemtai** (Aplicación Kemtai, s.f.): se trata de una aplicación web que ofrece un servicio para la realización de ejercicio de manera virtual. Aprovecha una tecnología avanzada de visión artificial para analizar el movimiento humano y proporcionar información sobre los ejercicios en tiempo real para crear entrenamientos más seguros y eficaces. Entre sus funcionalidades principales se encuentran:
 - Detección y análisis preciso del movimiento y la posición corporal.
 - Algoritmos de redes neuronales que clonan el movimiento específico del entrenador para proporcionar una representación precisa de los movimientos y objetivos específicos.
 - Entrenamientos adaptativos que analizan el rendimiento en tiempo real para ajustar la dificultad del ejercicio y proporcionar entrenamientos óptimos.
 - Permite realizar rutinas sin ningún material adicional más que una cámara web.
 - En la *Figura 1* se puede observar la interfaz de la realización de un ejercicio de la aplicación.

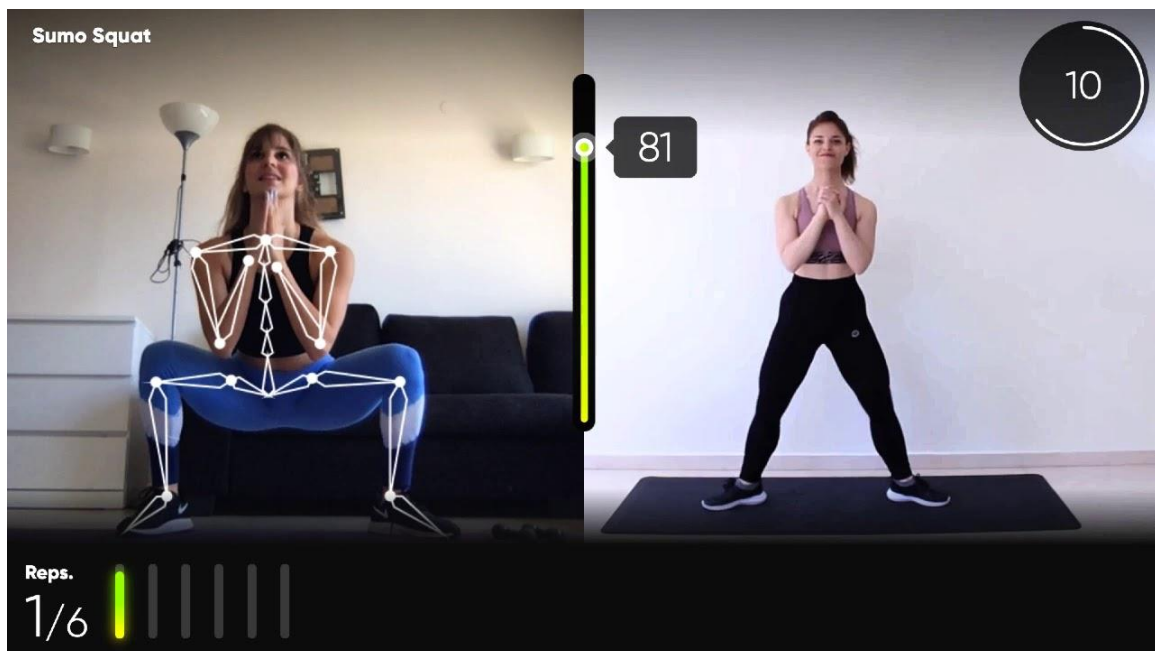


Figura 1: Aplicación web Kemtai

- **Ejercicios en casa** (Aplicación Ejercicios en casa: Sin equipo, s.f.): Sin equipo: se trata de una aplicación móvil que proporciona rutinas de ejercicios para todos los grupos musculares. No se necesita equipo ni entrenador, ya que todos los ejercicios presentes

Memoria del proyecto

se pueden hacer con el propio peso corporal. Garantiza que todos los ejercicios han sido diseñados por expertos. Entre las funcionalidades principales de la aplicación se encuentran:

- Rutinas de calentamientos y estiramientos.
- Registro del progreso del entrenamiento de forma automática.
- Gráfico para el seguimiento de las tendencias de peso del usuario.
- Personalización de recordatorios de ejercicios.
- Guías de vídeo y animación detalladas.
- En la *Figura 2* se puede observar un ejemplo de una rutina de abdominales para principiantes.

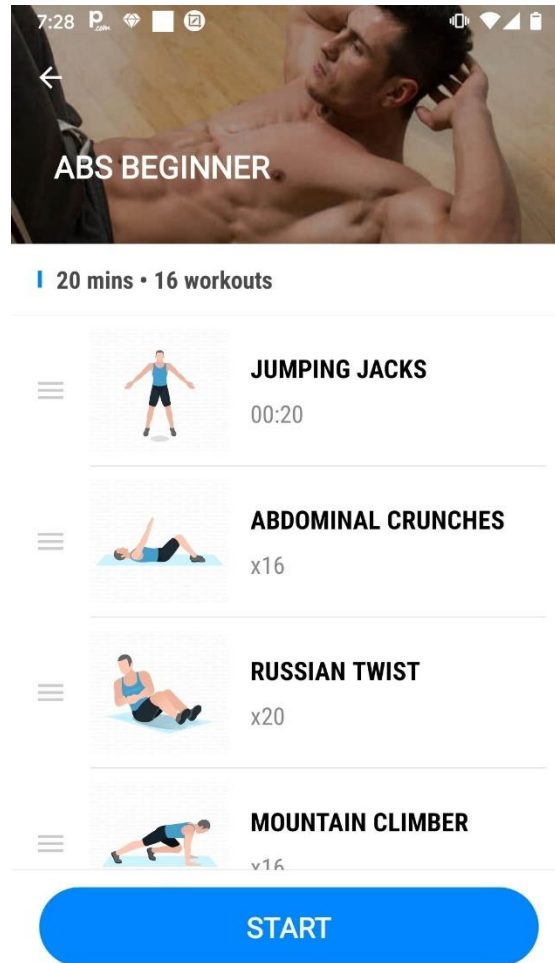


Figura 2: Aplicación móvil - Ejercicios en casa

- **Adidas training** (Aplicación adidas Training: Home Workout, s.f.): se trata de una aplicación móvil que tiene programas de entrenamiento y rutinas de ejercicios para hacer en casa, sin máquinas ni pesas. Además, ofrece la posibilidad de personalizar las sesiones de entrenamiento para ajustarla a los objetivos particulares, sea cual sea: adelgazar, tonificar, quemar grasa, ganar músculo, fortalecer el cuerpo, ponerse en forma o introducirse en los entrenamientos con el propio peso. Entre sus características destacables se encuentran:
 - Planes de entrenamiento creados en base a las necesidades particulares.
 - Entrenamientos enfocados a los distintos grupos musculares: glúteos, bíceps, tríceps, abdominales, muslos, entre otros.

Memoria del proyecto

- Vídeos de demostración de cada ejercicio.
- Colección de entrenamientos: una gran variedad de rutinas de diferentes tipos como los entrenamientos HIIT, por ejemplo, para quemar aún más calorías, ganar músculo, trabajar los glúteos, marcar abdominales, tonificar, perder peso, etc.
- Entrenamientos guiados: sesiones en vídeo para seguir a los entrenadores en tiempo real.
- En la *Figura 3* se puede observar un ejemplo de su interfaz.

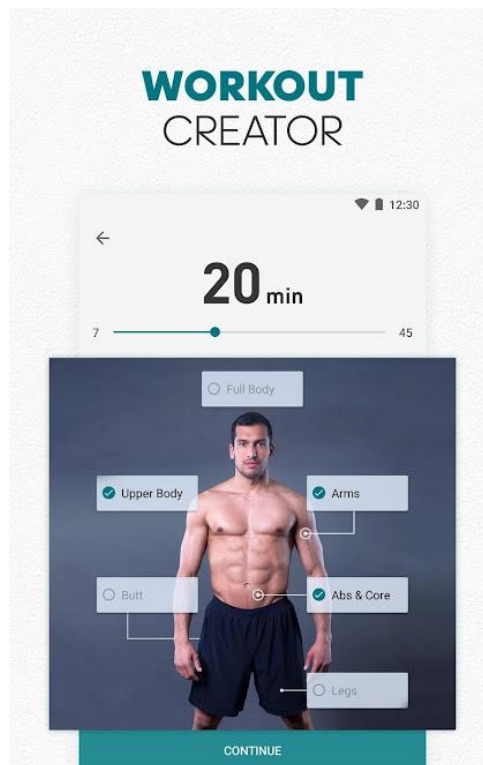


Figura 3: Adidas training

- **Virtuagym Fitness** (Aplicación Virtuagym Fitness - Home & Gym, s.f.): se trata de una aplicación móvil cuyo objetivo es que el usuario se ponga en forma. Dispone de un entrenador personal en 3D, que muestra cómo hacer cada ejercicio de forma correcta. Virtuagym incluye una cada vez más extensa base de datos de ejercicios, actualmente con más de 4000 ejercicios distintos y muchos entrenamientos predefinidos que se pueden realizar desde casa o el gimnasio. En la *Figura 4* se puede observar un ejemplo de su interfaz.

Memoria del proyecto

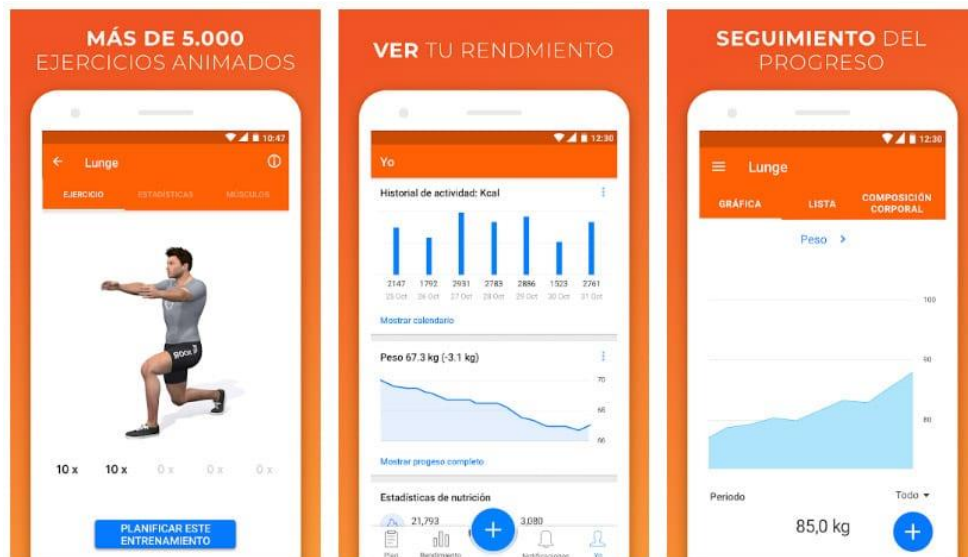


Figura 4: Virtugym fitness

- **Entrenamiento de 7 minutos** (Aplicación 7 Minute Workout, s.f.): se trata de una aplicación móvil de realización de rutinas de 7 minutos basada en el HICT (entrenamiento de circuito de alta intensidad) que se demostró como el método más "seguro, eficaz y eficiente" para mejorar la forma física muscular y aeróbica. Solo consta de 12 ejercicios que deben realizarse en 30 segundos, con descansos de 10 segundos entre ejercicio. Todo lo que se necesita es una silla y una pared. En la Figura 5 se puede observar un ejemplo de su interfaz.

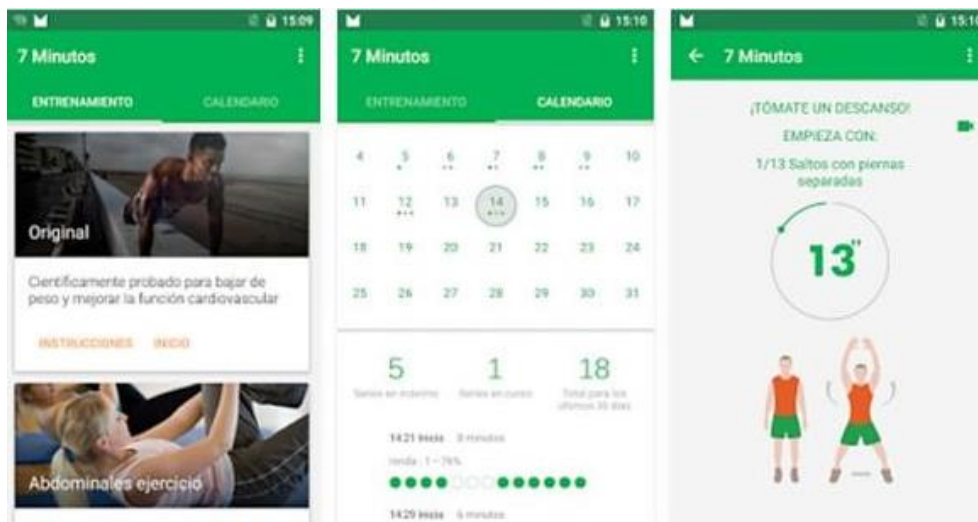


Figura 5: Entrenamiento en 7 minutos

3. OBJETIVOS

En este apartado se exponen los objetivos que tiene que cumplir el sistema para llevar a cabo el desarrollo de este Trabajo de Fin de Grado. Se detallan tanto los objetivos del sistema como los objetivos personales que se pretenden alcanzar con la realización del proyecto.

3.1.OBJETIVOS DEL SISTEMA

El objetivo principal del sistema es el desarrollo de una plataforma para el seguimiento y realización de ejercicio físico, que monitorice y analice los movimientos realizados por el usuario en tiempo real siendo capaz de detectar posturas incorrectas. En el siguiente listado se presentan los objetivos del sistema de forma más detallada:

- **Detección de los movimientos del usuario:** el sistema debe ser capaz de detectar la posición corporal mientras el usuario realiza movimientos.
- **Altas, bajas, modificaciones y visualización de usuarios:** el sistema debe permitir a los usuarios darse de alta, baja y modificar sus datos, así como visualizar sus perfiles.
- **Gestión de usuarios por los administradores:** el sistema debe permitir a los administradores buscar usuarios, cambiar su rol, deshabilitar y habilitar sus cuentas, eliminar cuentas, así como obtener listados de los administradores y entrenadores de la plataforma.
- **Altas, bajas y modificaciones de rutinas:** el sistema debe permitir a los entrenadores crear, modificar y eliminar rutinas.

3.2.OBJETIVOS PERSONALES

La finalidad de este apartado es exponer los objetivos personales que han llevado a la realización de este proyecto. El interés en este tema en parte es debido a que he sido una persona que siempre ha hecho deporte, pasando por natación, baloncesto, pádel y, finalmente, gimnasio combinado con ejercicios en casa. Este último es el más relacionado con el presente proyecto, ya que permite una gran variedad de ejercicios. En un inicio, como cualquier persona, haya hecho o no deporte, tenía poca idea de la técnica correcta, lo que me llevó en más de una ocasión a sufrir molestias. En cierto modo, el desarrollo de esta plataforma está pensado para ayudar a esas personas que, como yo, echaron en falta ser guiados en el inicio y evitar que comentan los mismos errores que yo.

Por otro lado, también está pensado para la rehabilitación, ya sea de personas mayores que por lo general sufren artrosis o para la recuperación de lesiones de gente de cualquier edad. Al igual que la primera motivación, también he sufrido lesiones a lo largo de mi vida y en ocasiones no sabía muy bien que ejercicios serían buenos para acelerar la recuperación y fortalecer la zona afectada.

Ya que la aplicación está pensada para que cualquier persona pueda hacer uso de ella, el principal objetivo es que esta sea amigable y fácil de usar independientemente de las condiciones de cada uno. A esto se le añade que la tecnología de detectar y la mayoría de los lenguajes de programación empleados eran nuevos para mí, lo que supuso un reto personal y acabar adquiriendo conocimientos y desarrollando habilidades que antes no tenía.

4. CONCEPTOS TEÓRICOS

En este apartado se exponen los conceptos teóricos que utilizados a lo largo del proyecto para facilitar su comprensión.

4.1.HERRAMIENTAS CASE

Las herramientas CASE (*Computer Aided Software Engineering*) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el costo de estas en términos de tiempo y de dinero (Definición herramientas CASE, s.f.).

Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida del desarrollo software, en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. En este tipo de herramientas se plantean los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas que, al ser realizadas con una herramienta, agilizan el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes software.
- Permitir un desarrollo y refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

Por otro lado, las herramientas CASE se componen de:

- Repositorio: diccionario donde se almacenan los elementos definidos o creados por la herramienta, y cuya gestión se realiza mediante el apoyo de un Sistema de Gestión de Base de Datos (SGBD) o de un sistema de gestión de ficheros.
- Metamodelo: constituye el marco para la definición de las técnicas y metodologías soportadas por la herramienta
- Carga o descarga de datos: son facilidades que permiten cargar el repertorio de la herramienta CASE con datos provenientes de otros sistemas, o bien generar a partir de la propia herramienta esquemas de base de datos, programas, etc. que pueden, a su vez, alimentar otros sistemas. Este elemento proporciona así un medio de comunicación con otras herramientas.
- Comprobación de errores: facilidades que permiten llevar a cabo un análisis de la exactitud, integridad y consistencia de los esquemas generados por la herramienta.
- Interfaz de usuario: consta de editores de texto y herramientas de diseño gráfico que permiten, mediante la utilización de un sistema de ventanas, iconos y menús , con la ayuda del ratón, definir los diagramas, matrices, etc. que incluyen las distintas metodologías

Memoria del proyecto

4.2. VALOR MET

Es la unidad de medida del índice metabólico y corresponde a 3,5 ml O₂/kg x min, que es el consumo mínimo de oxígeno que el organismo necesita para mantener sus constantes vitales (Compendium, s.f.).

Se utiliza el compendio de actividades físicas que se elaboró para su utilización en estudios epidemiológicos a fin de normalizar la asignación de intensidades MET para las actividades físicas.

El **índice metabólico** es la cantidad de energía que se libera o consume en una determinada unidad de tiempo. En un adulto, el índice metabólico basal (IMB) es la cantidad de las calorías diarias necesarias para mantener las funciones básicas de su cuerpo. El metabolismo basal se calcula en kilocalorías/día y depende del sexo, la altura y el peso, entre otros factores.

Los MET son la razón entre el metabolismo de una persona durante la realización de un trabajo y su metabolismo basal. Un MET se define como el costo energético de estar sentado tranquilamente y es equivalente a un consumo de 1 kcal/kg/h.

Un equivalente metabólico (1 MET) es la cantidad de energía (oxígeno) que el cuerpo utiliza cuando se está sentado tranquilamente, por ejemplo, leyendo un libro. La intensidad se puede describir como un múltiplo de este valor. Cuanto más trabaja el cuerpo durante una actividad física, más elevado es el nivel MET al que se está trabajando.

A cada actividad física se le asigna un MET en función de la intensidad de esta. Se hacen dos distinciones principales:

- **Actividad física moderada** (aproximadamente 3 – 6 MET): requiere un esfuerzo moderado, que acelera de forma perceptible el ritmo cardíaco.
- **Actividad física intensa** (aproximadamente > 6 MET): requiere una gran cantidad de esfuerzo, provocando una respiración rápida y un aumento sustancial de la frecuencia cardíaca.

4.3. REST

La transferencia de estado representacional (REST, REpresentational State Transfer) es un estilo de arquitectura software que define un conjunto de restricciones que se utilizan para crear servicios web (REST, s.f.).

Los servicios que siguen esta arquitectura, se les conoce como RESTful. Permiten a los sistemas solicitantes acceder y manipular representaciones textuales de los recursos de la Web mediante el uso de un conjunto uniforme y predefinido de operaciones sin estado.

Esta arquitectura hace uso del protocolo HTTP debido a que permite compartir información entre cliente y servidor.

5. TÉCNICAS Y HERRAMIENTAS

En este apartado se presentan las técnicas metodológicas, herramientas de desarrollo y bibliotecas software que se han utilizado en el proyecto.

5.1.HTML

HTML (HyperText Markup Language), es el lenguaje de marcas de hipertexto usado para estructurar una página web y su contenido. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones. El estándar está a cargo del World Wide Web Consortium (W3C), organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación (Documentación HTML, s.f.).

En su nombre aparecen dos términos “Hypertext” (hipertexto) y “Markup” (marcas):

- “Hypertext” hace referencia a los enlaces que conectan unas páginas con otras, ya sea dentro de un solo sitio web o entre varios sitios web. Los enlaces son un aspecto fundamental de la WEB. Al subir contenido a internet y vincularlo a páginas creadas por otras personas, se convierte en un participante activo en la Word Wide Web.
- “Markup”, HTML utiliza marcas para anotar texto, imágenes y otro contenido para su visualización en la web. El marcado HTML tiene elementos especiales como <head>, <body>, <footer>, <div>, <form>, <a>, entre otros.

5.2.CSS

CSS (Cascade Style Sheets), es un lenguaje de diseño gráfico para definir y crear la representación de un documento estructurado escrito en un lenguaje de marcado como HTML o XML. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas e interfaces de usuario para aplicaciones web.

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características de presentación, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

5.3.JAVASCRIPT

JavaScript es un lenguaje de programación interpretado basado en el estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Ofrece la funcionalidad a la página mientras que HTML ofrece la estructura y CSS el diseño (Documentación Javascript, s.f.).

Este lenguaje se ha usado principalmente en la programación de los ejercicios y captura de movimientos del usuario.

Memoria del proyecto

5.4.ML5

ML5.js es una librería que proporciona acceso a algoritmos y modelos de aprendizaje automático usando JavaScript y como única dependencia tensorflow.js, que es una librería de Machine Learning para JavaScript creada por Google (Documentación ml5.js, s.f.).

En este caso no se ha usado para crear un modelo de aprendizaje, sino que se ha empleado un modelo existente para la detección de la posición corporal llamado PoseNet. Se trata de un modelo de aprendizaje automático que permite la estimación de la postura humana en tiempo real. Se puede usar para estimar una o varias poses.

5.5.P5

P5.js es una librería de JavaScript de código abierto enfocada en la programación creativa, que permite crear dibujos y animaciones en el navegador (Documentación p5.js, s.f.).

Esta librería se ha usado en el proyecto para crear el canvas en el que se captura la imagen del usuario y dibujar el esqueleto, uniendo los puntos de las articulaciones, a partir de la monitorización de los movimientos del usuario, de manera que se coloree de verde si la postura es correcta y de rojo si no lo es.

5.6.PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web, ya que puede ser incluido en HTML, sin embargo, se trata de un lenguaje de propósito general (Documentación PHP, s.f.).

El código PHP suele ser procesado del lado del servidor web por un intérprete PHP implementado como un módulo, un Daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado -que puede ser cualquier tipo de datos, como el HTML generado o datos de imágenes binarias- formaría la totalidad o parte de una respuesta HTTP. Existen diversos sistemas de plantillas, sistemas de gestión de contenidos y frameworks que pueden emplearse para organizar o facilitar la generación de esa respuesta. Por otra parte, PHP puede utilizarse para muchas tareas de programación fuera del contexto de la web. Algunas de sus características principales son:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables, aunque estas se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

Memoria del proyecto

5.7.PHP MAILER

PHPMailer es una biblioteca php utilizada para enviar emails basada en el componente active server ASPMail (PHPMailer, s.f.). Permite realizar de forma sencilla tareas complejas como:

- Enviar correos electrónicos con ficheros adjuntos.
- Enviar correos electrónicos en formato HTML.

Con PHPMailer se pueden enviar emails vía sendmail, PHP mail() o con SMTP. Lo más recomendable es SMTP por dos razones:

- Con phpmailer se pueden usar varios servidores SMTP. Esto permite repartir la carga entre varias computadoras, con lo que se podrán enviar un mayor número de mensajes en un tiempo menor.
- Además, los servidores SMTP permiten mensajes con múltiples destinatarios .

5.8.APACHE

Apache es un servidor HTTP de código abierto para plataformas Unix, Microsoft Windows, Macintosh y otras (multiplataforma), que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616 (Documentación Apache, s.f.).

El servidor Apache es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la Apache Software Foundation dentro del proyecto HTTP Server (httpd).

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido.

Actualmente es el servidor web más utilizado, con mayor número de instalaciones a nivel mundial.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan solo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales maliciosos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

Este servidor se ha utilizado durante el desarrollo para tener disponible un servidor local facilitando las tareas de programación.

5.9.MARIADB

MariaDB es un sistema de gestión de bases de datos multiplataforma derivado de MySQL con licencia GPL (General Public License). Es desarrollado por Michael (Monty) Widenius (fundador de MySQL), la fundación MariaDB y la comunidad de desarrolladores de software libre. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, API y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente (Documentación MariaDB, s.f.).

Este SGBD surge a raíz de la compra de Sun Microsystems por parte de Oracle. MariaDB es una bifurcación directa de MySQL que asegura la existencia de una versión de este producto con licencia GPL.

Memoria del proyecto

MariaDB, como base de datos relacional, utiliza tablas para almacenar y organizar la información. Destaca por su integración con lenguajes de programación ampliamente usados como PHP, Perl y Java y su integración en varios sistemas operativos.

En el proyecto se ha usado para garantizar la persistencia de los datos en el servidor sobre usuarios, ejercicios, rutinas y estadísticas.

5.10. PHPMYADMIN

PhpMyAdmin es una herramienta de software libre escrita en PHP , destinada facilitar la administración de bases de datos MySQL a través de la web. Se usa para crear, eliminar o modificar bases de datos, tablas o cualquier campo de estas. A parte, incorpora una sección para ejecutar sentencias SQL directamente (Documentación phpMyAdmin, s.f.).

Esta herramienta ha sido usada a lo largo de todo el desarrollo ya que, al disponer de una interfaz gráfica muy intuitiva, su manejo se hace más sencillo. En la *Figura 6* se puede observar su interfaz.

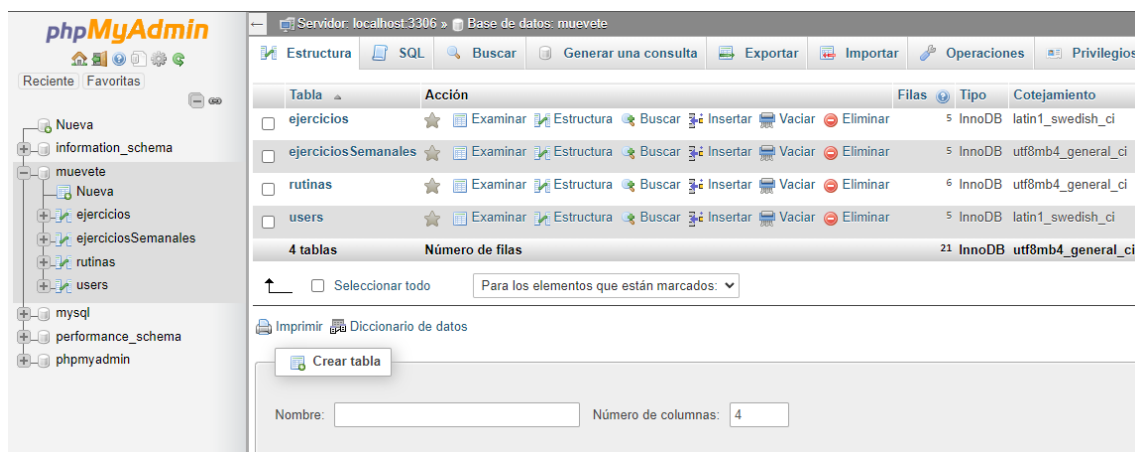
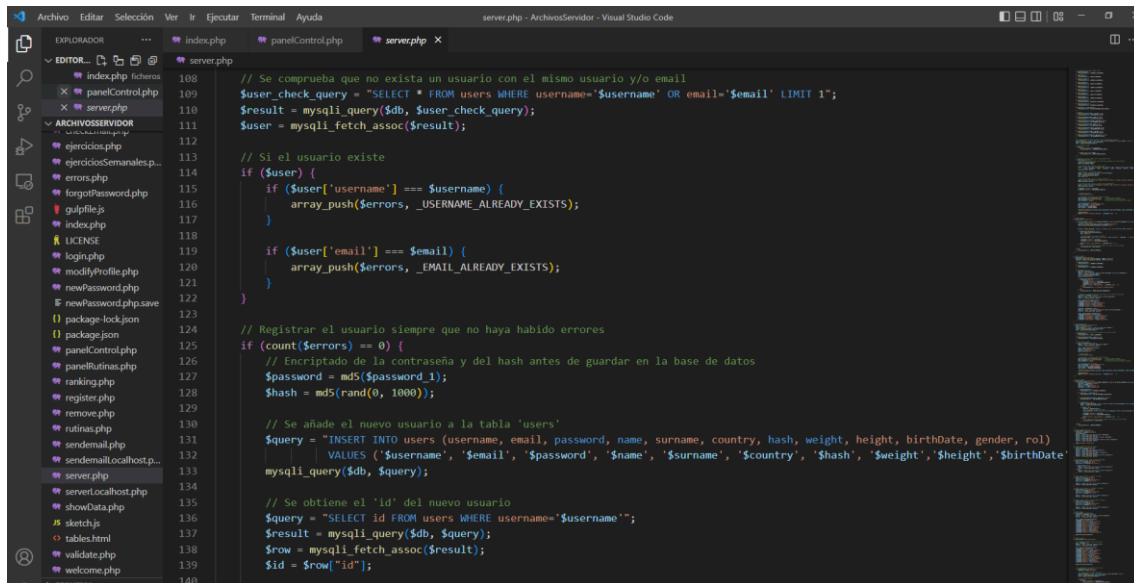


Figura 6: Interfaz phpMyAdmin

5.11. VISUAL STUDIO CODE

Se trata de un editor de código fuente que incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis y finalización inteligente de código entre otros. Es compatible con varios lenguajes de programación entre los que se encuentran PHP y JavaScript. Esta es una de las razones por las que se escogió este editor para el desarrollo del proyecto, a parte que ya se contaba con experiencia en el mismo (Documentación Visual Studio Code, s.f.). En la *Figura 7* se puede observar la interfaz del programa.

Memoria del proyecto



```
server.php - ArchivosServidor - Visual Studio Code
EXPLORADOR
  ficheros
  panelControl.php
  server.php
  ARCHIVOS SERVIDOR
  ejercicios.php
  ejerciciosSemanales.php
  errors.php
  forgotPassword.php
  gulpfile.js
  index.php
  LICENSE
  login.php
  modifyProfile.php
  newPassword.php
  newPassword.php.save
  package-lock.json
  package.json
  panelControl.php
  panelRutinas.php
  ranking.php
  register.php
  remove.php
  rutinas.php
  sendemail.php
  sendemailLocalhost.php
  server.php
  serverLocalhost.php
  showData.php
  sketch.js
  tablect.html
  validate.php
  welcome.php
  ...

server.php
108 // Se comprueba que no exista un usuario con el mismo usuario y/o email
109 $user_check_query = "SELECT * FROM users WHERE username='$username' OR email='$email' LIMIT 1";
110 $result = mysqli_query($db, $user_check_query);
111 $user = mysqli_fetch_assoc($result);
112
113 // Si el usuario existe
114 if ($user) {
115     if ($user['username'] === $username) {
116         array_push($errors, _USERNAME_ALREADY_EXISTS);
117     }
118
119     if ($user['email'] === $email) {
120         array_push($errors, _EMAIL_ALREADY_EXISTS);
121     }
122 }
123
124 // Registrar el usuario siempre que no haya habido errores
125 if (count($errors) == 0) {
126     // Encriptado de la contraseña y del hash antes de guardar en la base de datos
127     $password = md5($password_1);
128     $hash = md5(rand(0, 1000));
129
130     // Se añade el nuevo usuario a la tabla 'users'
131     $query = "INSERT INTO users (username, email, password, name, surname, country, hash, weight, height, birthDate, gender, rol)
132     VALUES ('$username', '$email', '$password', '$name', '$surname', '$country', '$hash', '$weight', '$height', '$birthDate', '$gender', '$rol')";
133     mysqli_query($db, $query);
134
135     // Se obtiene el 'id' del nuevo usuario
136     $query = "SELECT id FROM users WHERE username='$username'";
137     $result = mysqli_query($db, $query);
138     $row = mysqli_fetch_assoc($result);
139     $id = $row["id"];
140
```

Figura 7: Interfaz Visual Studio Code

5.12. REMOTE - SSH

Se trata de una extensión del editor Visual Studio Code que permite usar cualquier máquina remota con un servidor SSH como entorno de desarrollo. Esto simplifica en gran medida el desarrollo y la solución de problemas, ya que se puede trabajar directamente sobre los ficheros del servidor con un entorno de desarrollo agradable. Además, permite acceder a un entorno de desarrollo existente desde varias máquinas o ubicaciones (Documentación Visual Studio Code, s.f.).

5.13. HERRAMIENTAS CASE

Este apartado tiene como objetivo explicar cuáles son las herramientas CASE que se han utilizado. Estas herramientas son aquellos programas destinados a aumentar el balance de desarrollo software reduciendo el costo de estas en cuanto a tiempo se refiere.

5.13.1. VISUAL PARADIGM

Visual Paradigm for UML es una herramienta UML concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de distintos tipos de diagramas UML (Documentación Visual Paradigm, s.f.).

Esta herramienta ha sido usada en las fases de especificación de requisitos, análisis de requisitos y diseño del sistema para realizar los diagramas de casos de uso, de paquetes y de secuencia distribuidos en los distintos anexos de este proyecto. En la *Figura 8* se muestra la interfaz del programa.

Memoria del proyecto

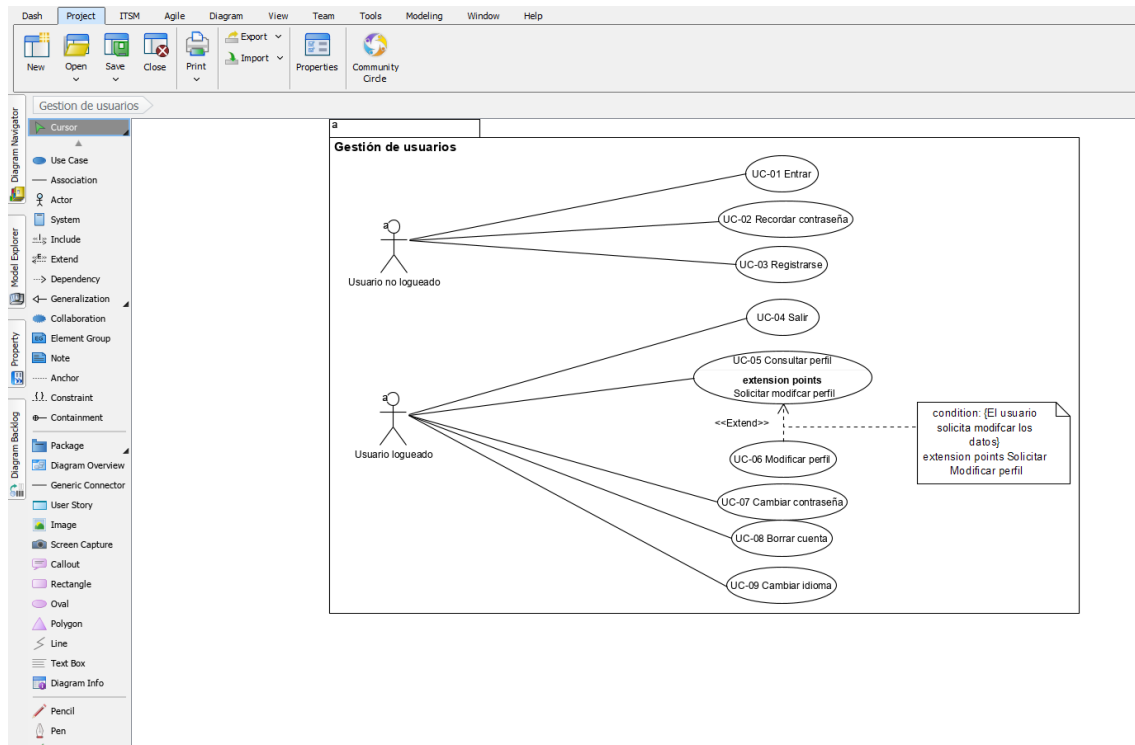


Figura 8: Interfaz Visual Paradigm

5.13.2. EZESTIMATE

EZEstimate es una herramienta utilizada para la estimación de la duración de un proyecto a partir de los puntos de casos de uso. Para ello hay que seguir un proceso que se describe a continuación:

- Crear un módulo por cada paquete del modelo de casos de uso del sistema
- Identificar los actores y asignarles su complejidad.
 - *Simple*: Si el actor es un sistema y la aplicación se comunica con él mediante una API.
 - *Medio*: Si el actor es un sistema y la aplicación se comunica con él mediante un protocolo (Internet).
 - *Complejo*: Persona con una interfaz gráfica.
- Identificar casos de uso y asignarles su complejidad en función del número de transacciones (acción del actor y respuesta del sistema).
 - *Simple*: 3 transacciones o menos.
 - *Medio*: de 4 a 7 transacciones.
 - *Complejo*: más de 7 transacciones.
- Determinar la influencia (complejidad percibida) de los factores técnicos y de los factores de entorno asignándole un valor entre 0 y 5.
- Asignar el número de horas de persona por punto de caso de uso (20 por defecto).

En la *Figura 9* se muestra su interfaz.

Memoria del proyecto

File Settings Help

Module
Gestion_usuarios
Add Module Delete

Summary
Total Modules 5 Excel Report Generate Report
Use cases Simple 32 Average 5 Complex 0
Actors Simple 0 Average 0 Complex 4

Add Actor / Use case
Actor / Use case Name Select Type Complexity Add

Tech / Env Factors
Set Tech Factor Set Env Factors

Estimation Summary
UAW 12
UUCW 210
UUPC = UAW + UUCW 222
TFactor 19
EFactor 12
TCF = 0.6 + (.01*TFactor) 0.79
EF = 1.4 + (-0.03*EFactor) 1.04
UCP = UUCP*TCT*EF 182,3952
Total Effort@ 8 Hrs/UCP 1459,1616

Use case / Actor List (Double click to delete)

Id	Module	Type	Name	complexity
31	Gestion_ejercici...	Usecase	Realizar ejercicio	Simple
32	Gestion_rutinas	Actor	Entrenador	Complex
33	Gestion_rutinas	Usecase	Crear rutina	Average
34	Gestion_rutinas	Usecase	Eliminar rutina	Average
35	Gestion_rutinas	Usecase	Listar rutinas	Simple
36	Gestion_rutinas	Usecase	Ver detalles rutina	Simple
37	Gestion_rutinas	Usecase	Listar rutinas pr...	Simple
38	Gestion_rutinas	Usecase	Añadir ejercicio ...	Simple
39	Gestion_rutinas	Usecase	Eliminar ejercici...	Simple
4	Gestion_usuarios	Usecase	Recordar contr...	Average
40	Gestion_rutinas	Usecase	Cambiar nombri...	Average
41	Gestion_rutinas	Usecase	Realizar rutina	Simple
5	Gestion_usuarios	Usecase	Registrarse	Average
6	Gestion_usuarios	Usecase	Salir	Simple
7	Gestion_usuarios	Usecase	Consultar perfil	Simple
8	Gestion_usuarios	Usecase	Modificar perfil	Simple
9	Gestion_usuarios	Usecase	Cambiar contra...	Simple

Figura 9: Interfaz EZEstimate

5.13.3. MICROSOFT PROJECT

Microsoft Project es una herramienta utilizada para la administración de proyectos que permite prestar asistencia en algunas tareas como:

- Elaborar un calendario de trabajo.
- Identificar tareas/subtareas tomando como referencia el Proceso Unificado (Fases-Iteraciones-disciplinas-tareas-subtareas).
- Identificar hitos.
- Asignar recursos a las diferentes tareas.
- Estimar la duración de las diferentes tareas.
- Establecer dependencias entre las tareas.
- Revisar y comentar los resultados obtenidos
 - Tareas críticas.
 - Distribución de recursos por tareas.
 - Carga de trabajo de cada recurso a lo largo del proyecto.

En la *Figura 10* se muestra la interfaz del programa (Documentación Microsoft Project, s.f.).

Memoria del proyecto

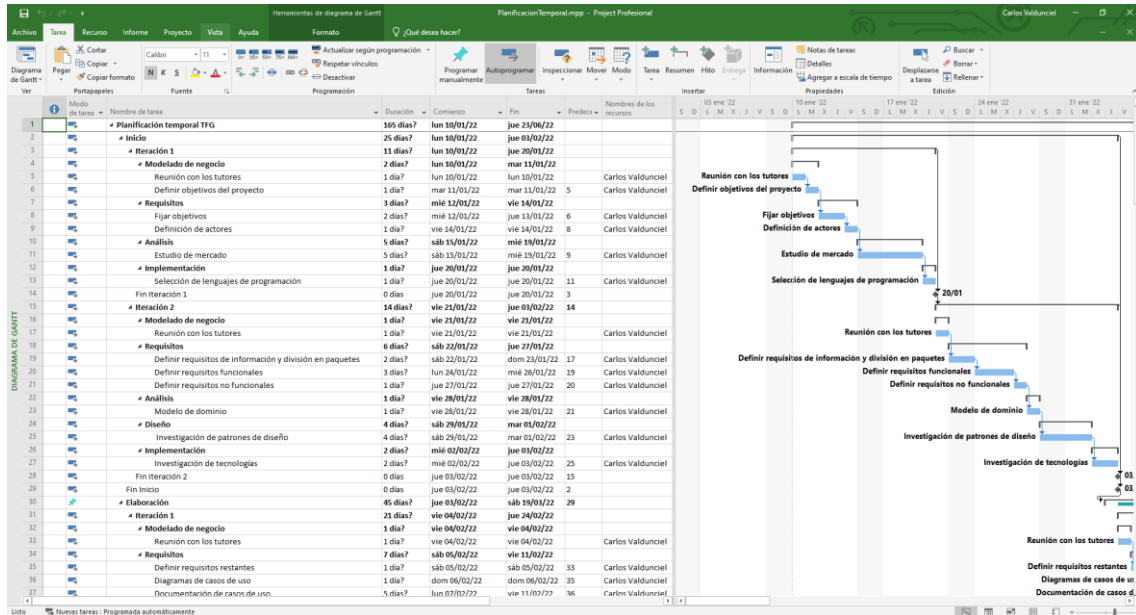


Figura 10: Interfaz Microsoft Project

5.13.4. NGROK

Ngrok es un programa que permite exponer a internet una URL generada dinámicamente, la cual apunta a un servicio web que se está ejecutando en la máquina local. Por ejemplo: si se tiene un servicio web arrancado en <http://localhost:8080>, ngrok genera una URL del tipo <http://xxxx.ngrok.io> (Documentación ngrok, s.f.).

Otra funcionalidad que brinda esta herramienta es que, aunque el servicio local que se está ejecutando lo haga usando http, proporciona https, evitando configuraciones adicionales en la máquina local o tener que expedir un certificado SSL.

Esta última funcionalidad es especialmente importante en el proyecto ya que para que el usuario pueda usar la cámara para realizar ejercicios, se requiere https, de lo contrario no se podría obtener imagen de la webcam. Esto es por razones de seguridad de la función `createCapture()` perteneciente a la biblioteca `p5.js` mencionada en el punto 5.5. En la Figura 11 se muestra su interfaz una vez abierto un túnel.

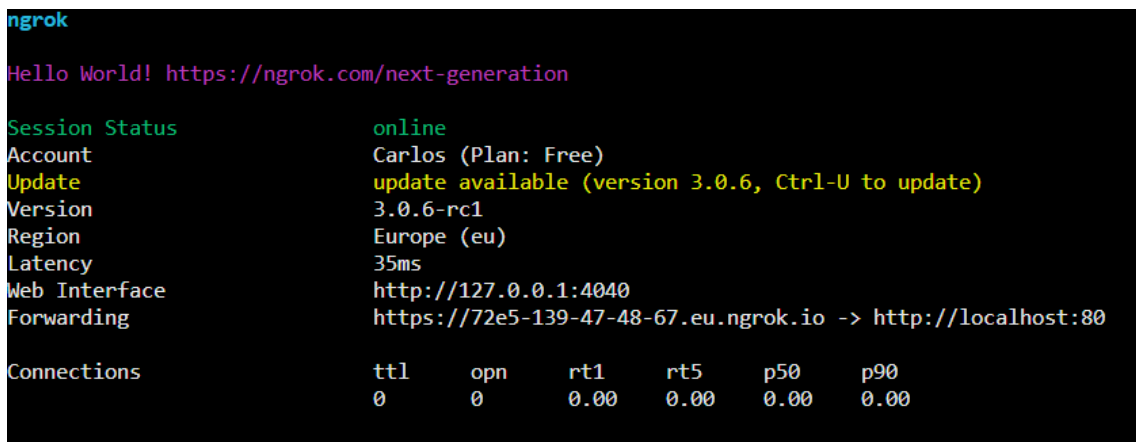


Figura 11: Interfaz de ngrok

Memoria del proyecto

5.14. HERRAMIENTAS DE GENERACIÓN DE DOCUMENTACIÓN

Doxygen es una herramienta de generación de documentación a partir del código fuente. Está disponible para lenguajes como C, C++, Java, Python y PHP entre otros. Se ha utilizado durante el desarrollo del *Anexo V -Documentación técnica*, para la creación automatizada de las páginas HTML de documentación del sistema (Documentación Doxygen, s.f.).

6. ASPECTOS RELEVANTES DEL DESARROLLO

En este apartado se especificarán las partes más importantes y representativas de las distintas fases de desarrollo del proyecto.

6.1. MARCO DE TRABAJO

Se ha empleado el Proceso Unificado como marco de trabajo durante el desarrollo del proyecto. Este se caracteriza por estar dirigido por casos de uso, definir las etapas de desarrollo por las que el proceso evoluciona de forma iterativa e incremental y estar centrado en la arquitectura.

- **Iterativo e incremental:** las fases del proceso, aunque están bien definidas, se solapan entre ellas, siendo cada una dependiente de su antecesora. Está compuesto por cuatro fases que se dividen en una serie de iteraciones cuyo resultado es un incremento del producto desarrollado:
 - **Inicio:** en esta primera fase se define el modelo de negocio, se realiza la elicitación de requisitos, definición de objetivos, estimación de costos y planificación temporal.
 - **Elaboración:** se obtiene una visión refinada del proyecto a realizar, aparecen nuevos requisitos y se ajustan las estimaciones.
 - **Construcción:** comprende el desarrollo del proyecto hasta su puesta en producción, es decir, hasta la obtención de un prototipo.
 - **Transición:** fase final del proyecto, el programa debe estar listo para ser probado, instalado y utilizado por el cliente sin ningún problema.
- **Dirigido por casos de uso:** en el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: modelado de negocio, requisitos, análisis, diseño, implementación y pruebas.
- **Centrado en la arquitectura:** el Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema.
- **Enfocados en riesgos:** el Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero. De esta manera se evitan sobrecostos y se agilizan las iteraciones.

En la *Figura 12* se pueden ver las fases e iteraciones del proceso unificado.

Memoria del proyecto

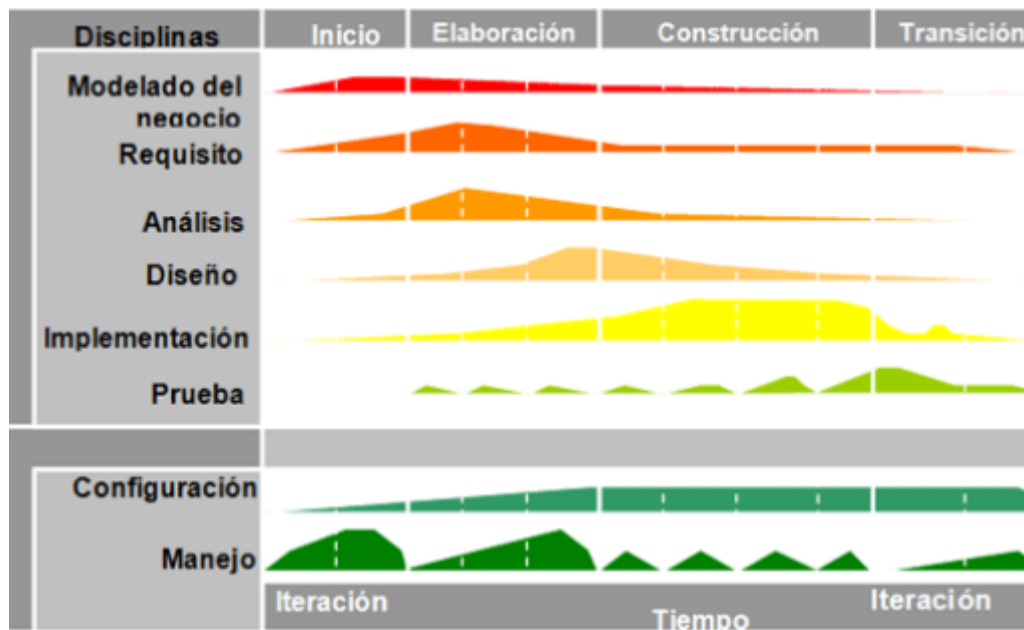


Figura 12: Fases e iteraciones del Proceso Unificado

6.2. ESTIMACIÓN DE LA DURACIÓN DEL PROYECTO

La estimación de la duración fue de las primeras tareas realizadas en el desarrollo del proyecto. Su objetivo era tener una estimación del tiempo de desarrollo calculado por la herramienta EZEstimate en función de diversos factores como: complejidad de los actores, complejidad de los casos de uso, factores de complejidad técnica y factores de complejidad de entorno.

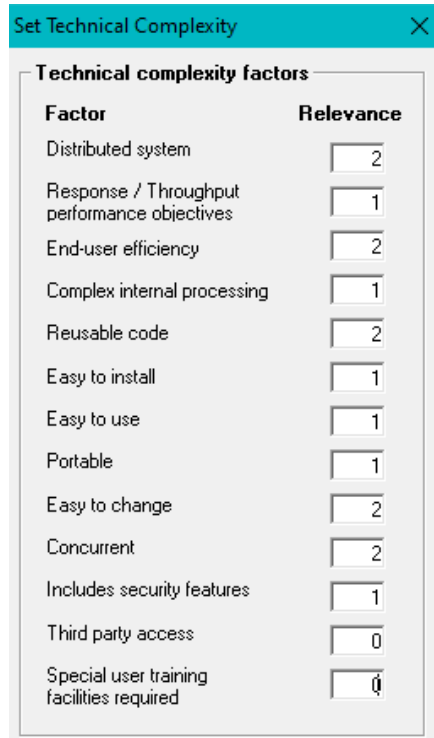
Para conocer la información detallada sobre esta estimación, se puede consultar en *Anexo I – Plan del proyecto*.

Los parámetros introducidos para obtener dicha estimación son los que se muestran en las figuras: *Figura 13: Casos de uso EZEstimate*, *Figura 14: Factores de complejidad técnica EZEstimate*, *Figura 15: Factores de complejidad de entorno EZEstimate* y *Figura 16: Estimación obtenida EZEstimate*.

Use case / Actor List (Double click to delete)				
Id	Module	Type	Name	complexity ^
29	Gestion_ejercici...	Usecase	Ver video de ej...	Simple
3	Gestion_usuarios	Usecase	Entrar	Simple
30	Gestion_ejercici...	Usecase	Listar dificultad...	Simple
31	Gestion_ejercici...	Usecase	Realizar ejercicio	Simple
32	Gestion_rutinas	Actor	Entrenador	Complex
33	Gestion_rutinas	Usecase	Crear rutina	Average
34	Gestion_rutinas	Usecase	Eliminar rutina	Average
35	Gestion_rutinas	Usecase	Listar rutinas	Simple
36	Gestion_rutinas	Usecase	Ver detalles rutina	Simple
37	Gestion_rutinas	Usecase	Listar rutinas pr...	Simple
38	Gestion_rutinas	Usecase	Añadir ejercicio ...	Simple
39	Gestion_rutinas	Usecase	Eliminar ejercici...	Simple
4	Gestion_usuarios	Usecase	Recordar contr...	Average
40	Gestion_rutinas	Usecase	Cambiar nombr...	Average
41	Gestion_rutinas	Usecase	Realizar rutina	Simple
5	Gestion_usuarios	Usecase	Registrarse	Average
6	Gestion_usuarios	Usecase	Salir	Simple
7	Gestinn_usuarios	Usecase	Consultar perfil	Simple

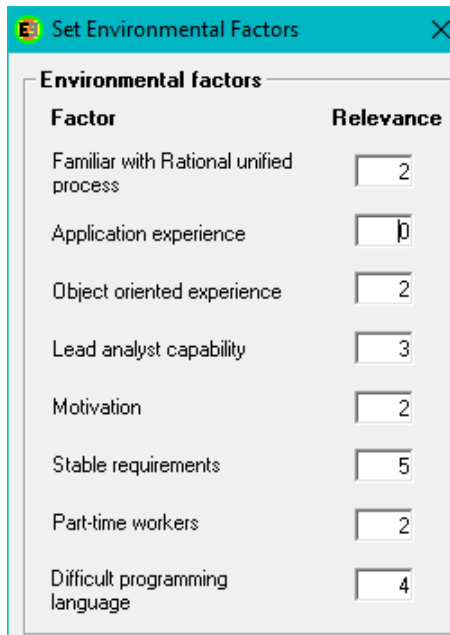
Figura 13: Casos de uso EZEstimate

Memoria del proyecto



Factor	Relevance
Distributed system	2
Response / Throughput performance objectives	1
End-user efficiency	2
Complex internal processing	1
Reusable code	2
Easy to install	1
Easy to use	1
Portable	1
Easy to change	2
Concurrent	2
Includes security features	1
Third party access	0
Special user training facilities required	0

Figura 14: Factores de complejidad técnica EZEstimate



Factor	Relevance
Familiar with Rational unified process	2
Application experience	0
Object oriented experience	2
Lead analyst capability	3
Motivation	2
Stable requirements	5
Part-time workers	2
Difficult programming language	4

Figura 15: Factores de complejidad de entorno EZEstimate

Memoria del proyecto

Estimation Summary	
UAW	12
UUCW	210
UUPC = UAW + UUCW	222
TFactor	19
EFactor	12
TCF = 0.6 + (.01*TFactor)	0,79
EF = 1.4 + (-0.03*EFactor)	1,04
UCP = UUCP*TCT*EF	182,3952
Total Effort@ 8	Hrs/UCP 1459,1616

Figura 16: Estimación obtenida EZEstimate

6.3.PLANIFICACIÓN TEMPORAL

Esta fase consiste en la división del proyecto en tareas, así como la identificación de dependencias entre ellas, asignándoles una duración estimada a cada una. Para su realización se ha utilizado la herramienta Microsoft Project. Las etapas en que se divide esta fase son las siguientes:

- **Elaboración del calendario de trabajo:** sirve para establecer la jornada laboral que se seguirá durante el desarrollo del proyecto, así como los días festivos y la fecha de comienzo. Ver *Figura 17*.
- **Identificación de tareas y subtareas:** en esta etapa se divide el proyecto en tareas, repartidas en las fases de inicio, elaboración, construcción y transición. Cada una de ellas se divide a su vez en las disciplinas de modelado de negocio, requisitos, análisis, diseño, implementación y pruebas de las cuales habrá más o menos iteraciones dependiendo de la fase de desarrollo en que se encuentre el proyecto. Ver *Figura 18*.
- **Identificación de hitos:** se trata del establecimiento de tareas simbólicas (sin duración) que coinciden con el final de una fase (hitos principales) o de una iteración (hitos secundarios). Ver *Figura 18*.
- **Asignación de recursos a las tareas.** Ver *Figura 19*.
- **Diagrama de Gantt:** se trata de un gráfico que muestra una vista general de las tareas programadas, así como las relaciones entre ellas y su duración. Este se genera de forma automática una vez se han realizado las etapas anteriores. Ver *Figura 20*.
- **Camino crítico:** se trata del mismo gráfico que el diagrama de Gantt, solo que en este se muestran las tareas críticas (aquellas que no pueden sufrir retrasos) dibujadas de color rojo. Ver *Figura 21*.

Memoria del proyecto

Cambiar calendario laboral [X]

Para calendario: Calendario TFG (Calendario del proyecto) Crear calendario...

El calendario 'Calendario TFG' es un calendario base.

Leyenda:

- Laborable
- No laborable
- 31** Horas laborables modificadas

En este calendario:

- 31** Día de excepción
- 31** Semana laboral no predeterminada

Haga clic en un día para ver sus períodos laborables:

Períodos laborables del 24 enero 2022:

- 7:00 a 15:00

Basado en:
Semana laboral predeterminada del calendario 'Calendario TFG'.

enero 2022

L	M	M	J	V	S	D
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Excepciones **Semanas laborales**

Nombre	Comienzo	Fin

Ayuda Opciones... Aceptar Cancelar

Figura 17: Calendario de trabajo

Planificación temporal TFG	139 días?	lun 10/01/22	sáb 28/05/22		
Inicio	23 días?	lun 10/01/22	mar 01/02/22		
Iteración 1	9 días?	lun 10/01/22	mar 18/01/22		
Modelado de negocio	2 días?	lun 10/01/22	mar 11/01/22		
Reunión con los tutores	1 día?	lun 10/01/22	lun 10/01/22		Carlos Valdunciel
Definir objetivos del proyecto	1 día?	mar 11/01/22	mar 11/01/22	5	Carlos Valdunciel
Requisitos	3 días?	mié 12/01/22	vie 14/01/22		
Fijar objetivos	2 días?	mié 12/01/22	jue 13/01/22	6	Carlos Valdunciel
Definición de actores	1 día?	vie 14/01/22	vie 14/01/22	8	Carlos Valdunciel
Análisis	3 días?	sáb 15/01/22	lun 17/01/22		
Estudio de mercado	3 días?	sáb 15/01/22	lun 17/01/22	9	Carlos Valdunciel
Implementación	1 día?	mar 18/01/22	mar 18/01/22		
Selección de lenguajes de programación	1 día?	mar 18/01/22	mar 18/01/22	11	Carlos Valdunciel
Fin Iteración 1	0 días	mar 18/01/22	mar 18/01/22	3	

Figura 18: Identificación de tareas e hitos

Memoria del proyecto

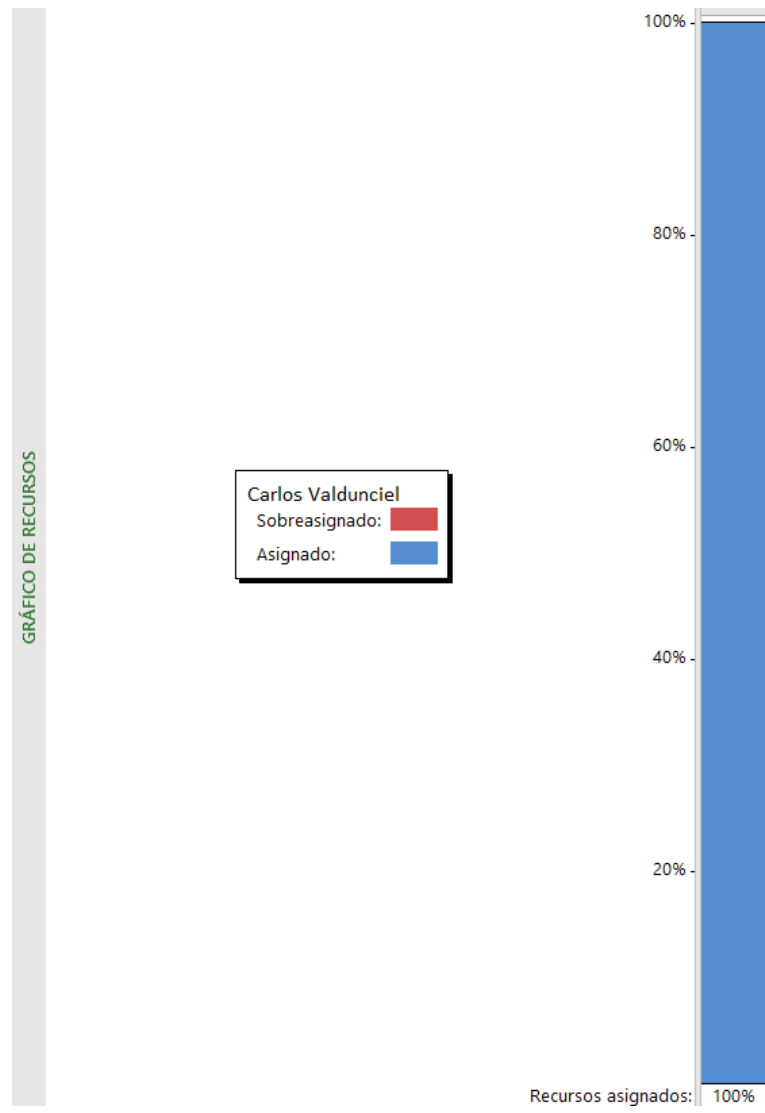


Figura 19: Asignación de recursos

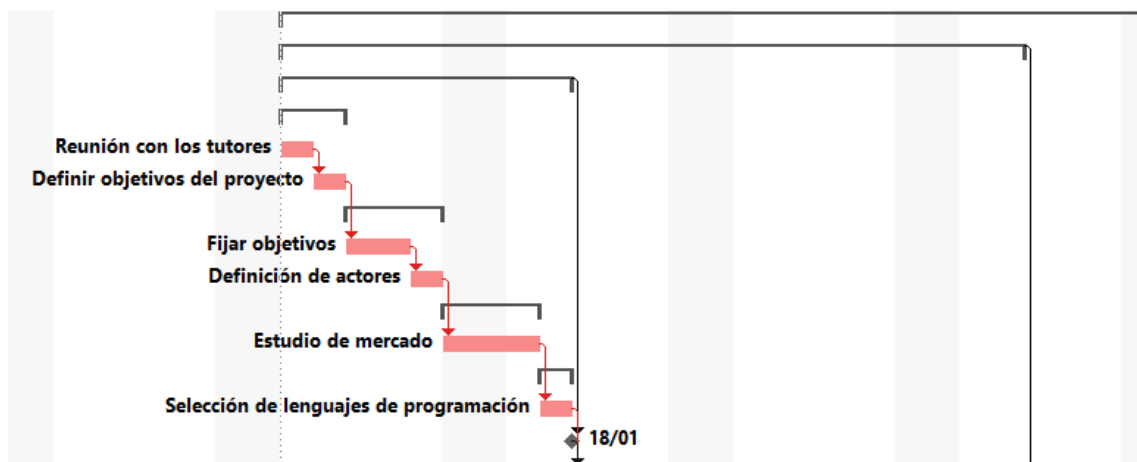


Figura 20: Diagrama de Gantt

Memoria del proyecto

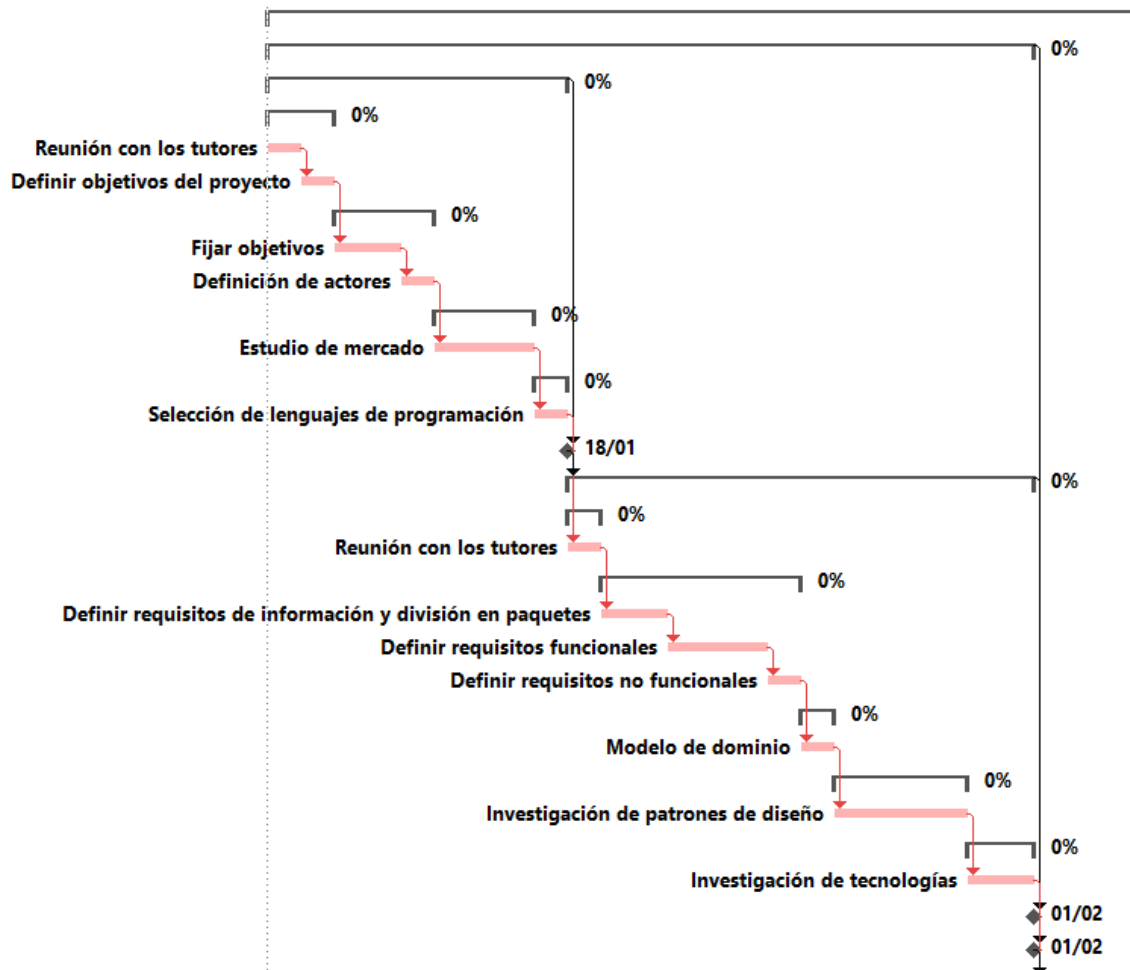


Figura 21: Camino crítico

Para entrar en mayor detalle sobre la planificación temporal del proyecto se puede consultar el documento *Anexo I – Plan de proyecto* que acompaña a esta memoria.

6.4.ESPECIFICACIÓN DE REQUISITOS

En esta fase se han recogido las especificaciones de requisitos software del sistema a desarrollar, entre los que se encuentran los requisitos de información, requisitos funcionales, objetivos, actores, casos de uso y requisitos no funcionales. Se ha llevado a cabo siguiendo la metodología de Durán y Bernárdez.

Para entrar en mayor detalle sobre la especificación de requisitos del sistema se puede consultar el documento *Anexo II – Especificación de requisitos del sistema* que acompaña a esta memoria.

En los siguientes puntos de este apartado se muestran ejemplos de cada etapa de la fase de especificación de requisitos.

Memoria del proyecto

6.4.1. PARTICIPANTES

El proyecto cuenta con los siguientes participantes:

- Valdunciel Gonzalo, Carlos.
- Sánchez San Blas, Héctor.
- De Paz Santana, Juan Francisco.
- Villarrubia González, Gabriel.

En la *Tabla 1* se muestra un ejemplo de especificación de participantes.

Participante	Valdunciel Gonzalo, Carlos
Organización	Universidad de Salamanca
Rol	Desarrollador
Es desarrollador	Sí
Es cliente	No
Es usuario	No
Comentarios	Ninguno

Tabla 1: Participante Valdunciel Gonzalo, Carlos

6.4.2. OBJETIVOS DEL SISTEMA

El proyecto consta de los siguientes objetivos:

- Detección de los movimientos del usuario.
- Altas, bajas, modificaciones y visualización de usuarios.
- Gestión de usuarios por administradores.
- Altas, bajas, modificación y visualización de rutinas.
- Visualizar ejercicios.
- Almacenamiento y visualización de estadísticas.

En la *Tabla 2* se muestra un ejemplo de especificación de objetivos.

OBJ-02	Altas, bajas, modificaciones y visualización de usuarios
Versión	1.0
Autores	Valdunciel Gonzalo, Carlos
Fuentes	Sánchez San Blas, Héctor De Paz Santana, Juan Francisco Villarrubia González, Gabriel
Descripción	<i>El sistema deberá permitir a los usuarios darse de alta, de baja y modificar sus datos, así como visualizar sus perfiles</i>
Subobjetivos	Ninguno

Memoria del proyecto

Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 2: OBJ-02 Altas, bajas, modificaciones y visualización de usuarios

6.4.3. REQUISITOS DE INFORMACIÓN

Los requisitos de información indican los datos que el sistema debe almacenar. El proyecto consta de los siguientes objetivos:

- Información sobre usuario.
- Información sobre ejercicios.
- Información sobre objetivos semanales.
- Información sobre rutinas.

En la *Tabla 3* se muestra un ejemplo de especificación de requisitos de información.

IRQ-01	Información sobre usuario
Versión	1.0
Autores	Valdunciel Gonzalo, Carlos
Fuentes	Sánchez San Blas, Héctor De Paz Santana, Juan Francisco Villarrubia González, Gabriel
Dependencias	Ninguno
Descripción	<i>El sistema deberá almacenar la información relativa a los usuarios registrados en el sistema. En concreto:</i>
Datos específicos	<ul style="list-style-type: none">• Identificador• Nombre de usuario• Email• Contraseña• Nombre• Apellido• País• Peso• Altura• Género• Fecha de nacimiento• Rol
Importancia	Vital

Memoria del proyecto

Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 3: IRQ-01 Información sobre usuario

6.4.4. REQUISITOS FUNCIONALES

Los requisitos funcionales definen cómo se comporta el sistema en situaciones particulares. Antes de realizar esta especificación, se ha construido un diagrama de paquetes que proporciona una visión general del sistema. Este diagrama se muestra en la *Figura 22*.

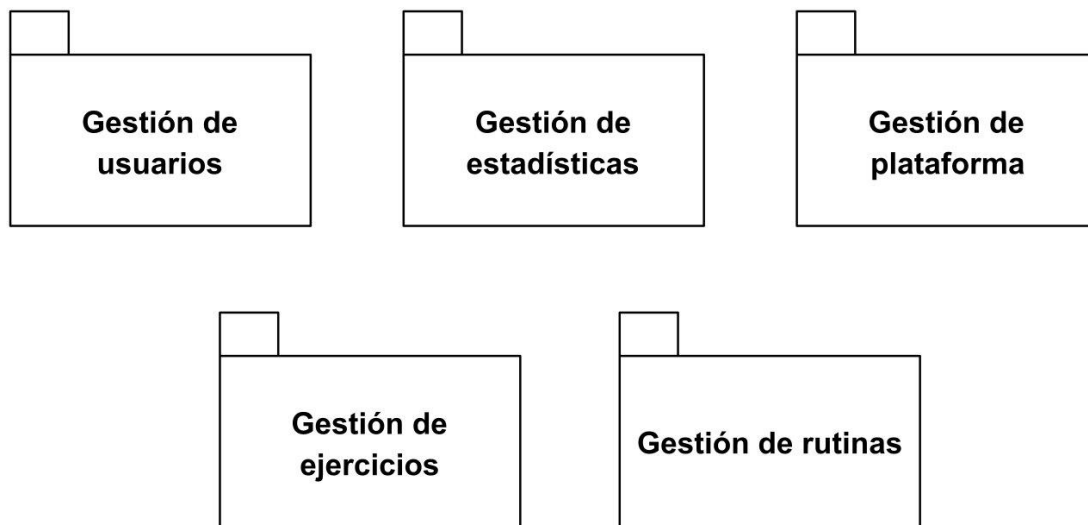


Figura 22: Diagrama de paquetes

Posteriormente, se definen los actores que representan a los usuarios que interactúan con el sistema, como se puede observar en la *Figura 23*.

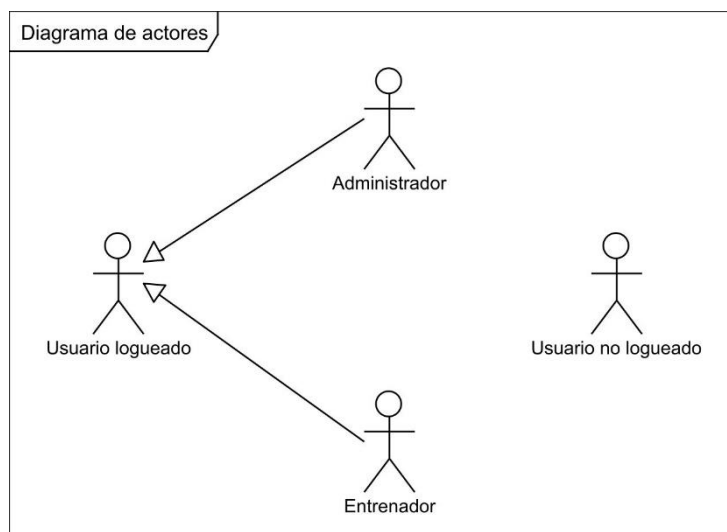


Figura 23: Diagrama de actores

Memoria del proyecto

En la *Tabla 4* se muestra un ejemplo de especificación de actor.

ACT-02	Usuario logueado
Versión	1.0
Autores	Valdunciel Gonzalo, Carlos
Fuentes	Sánchez San Blas, Héctor De Paz Santana, Juan Francisco Villarrubia González, Gabriel
Descripción	<i>Este actor representa al usuario que accede a la plataforma y que ha iniciado sesión con éxito</i>
Comentarios	Ninguno

Tabla 4: ACT-02 Usuario logueado

Por último, se definen los casos de uso del sistema, agrupados en los paquetes vistos anteriormente. Se realiza un diagrama de casos de uso por paquete y posteriormente se especifica cada uno individualmente. En la *Figura 24* se muestra un ejemplo de diagrama de casos de uso.

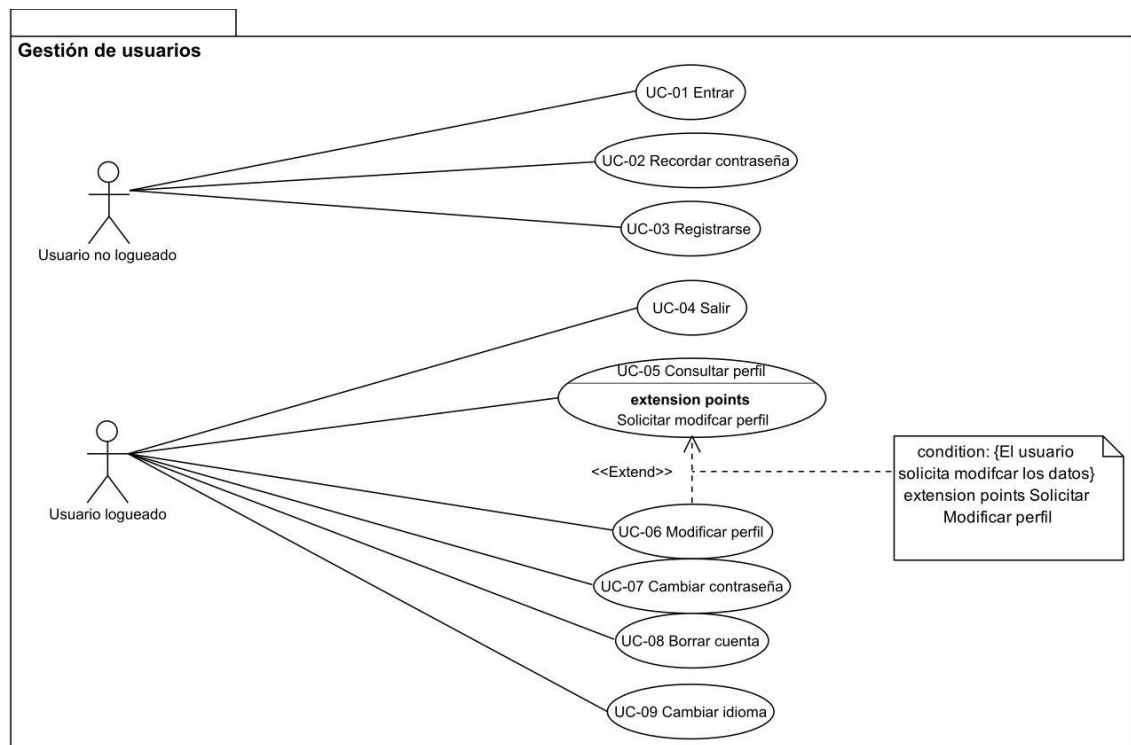


Figura 24: Paquete Gestión de usuarios

Memoria del proyecto

En la *Tabla 5* se muestra un ejemplo de especificación de requisitos funcionales.

UC-06	Modificar perfil	
Versión	1.0	
Autores	Valdunciel Gonzalo, Carlos	
Fuentes	Sánchez San Blas, Héctor De Paz Santana, Juan Francisco Villarrubia González, Gabriel	
Dependencias	Ninguna	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario solicite modificar los datos de su perfil o durante la realización de los siguientes casos de uso: [UC-08] Consultar perfil	
Precondición	Se ha mostrado al usuario un formulario con los datos de su cuenta	
Secuencia normal	Paso	Acción
	1	El actor Usuario logueado (ACT-02) solicita modificar sus datos
	2	El sistema solicita los cambios
	3	El actor Usuario logueado (ACT-02) introduce los cambios
	4	El sistema comprueba los datos
	5	El sistema informa al usuario del éxito de la acción
Postcondición	Ninguna	
Excepciones	Paso	Acción
	4	Si los datos introducidos son incorrectos, el sistema informa de ello al usuario, a continuación, el caso de uso queda sin efecto
	4	Si ya existe un usuario con el mismo nombre de usuario o email, el sistema informa de ello al usuario, a continuación, el caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	-	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 5: UC-06 Modificar perfil

Memoria del proyecto

6.4.5. REQUISITOS NO FUNCIONALES

Los requisitos no funcionales imponen restricciones en el diseño e implementación del sistema, además de estándares de calidad. El proyecto consta de los siguientes requisitos no funcionales:

- Protección de datos.
- Concurrencia.
- Interfaz.
- Compatibilidad.
- Manejabilidad.
- Hacer uso de la webcam.
- Implementación.

En la *Tabla 6* se muestra un ejemplo de especificación de requisitos funcionales.

NFR-06	Hacer uso de la webcam
Versión	1.0
Autores	Valdunciel Gonzalo, Carlos
Fuentes	Sánchez San Blas, Héctor De Paz Santana, Juan Francisco Villarrubia González, Gabriel
Dependencias	Ninguno
Descripción	<i>El sistema deberá usar la webcam disponible en el equipo en su implementación</i>
Importancia	Importante
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 6: NFR-06 Hacer uso de la webcam

6.5. ANÁLISIS DE REQUISITOS

En esta fase se ha llevado a cabo el análisis de los requisitos identificados y descritos en la elicitación de requisitos. El resultado de este análisis permite una comprensión más precisa del sistema que ayudará a estructurarlo. Para entrar en mayor detalle sobre el análisis de requisitos del sistema se puede consultar el documento *Anexo III – Análisis de requisitos* que acompaña a esta memoria.

6.5.1. MODELO DE DOMINIO

El modelo de dominio se trata de una representación de clases conceptuales del mundo real, no de componentes software. Este recoge las necesidades de almacenamiento y gestión de la información del sistema. Hay que determinar aquellas entidades del modelo de negocio que sean necesarias gestionar, sus atributos relevantes y las relaciones entre ellas. Para representarlo, se ha realizado el diagrama de clases que se puede observar en la *Figura 25*, en el que no se han incluido atributos para aportar mayor claridad.

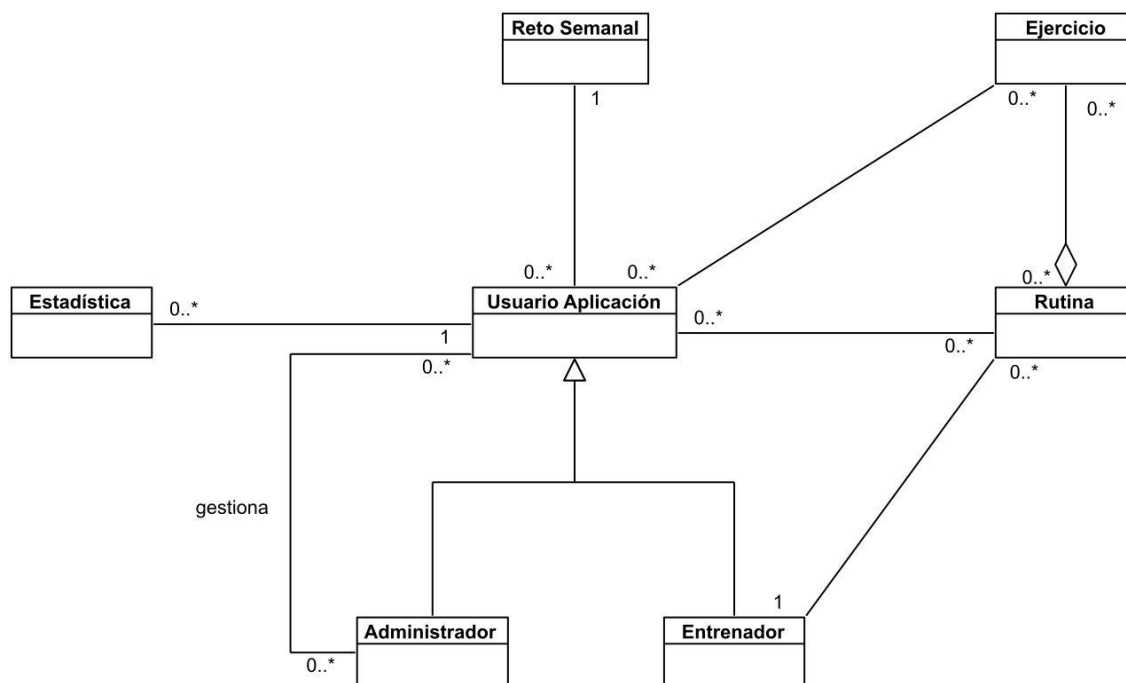


Figura 25: Diagrama de clases del Modelo de dominio

Memoria del proyecto

6.5.2. REALIZACIÓN DE CASOS DE USO EN EL MODELO DE ANÁLISIS

En este apartado se incluirán los diagramas de secuencia de casos de uso que muestran el intercambio de mensajes entre los distintos componentes del sistema. En la *Figura 26* se muestra un ejemplo de diagrama de secuencia correspondiente al caso de uso *Entrar*.

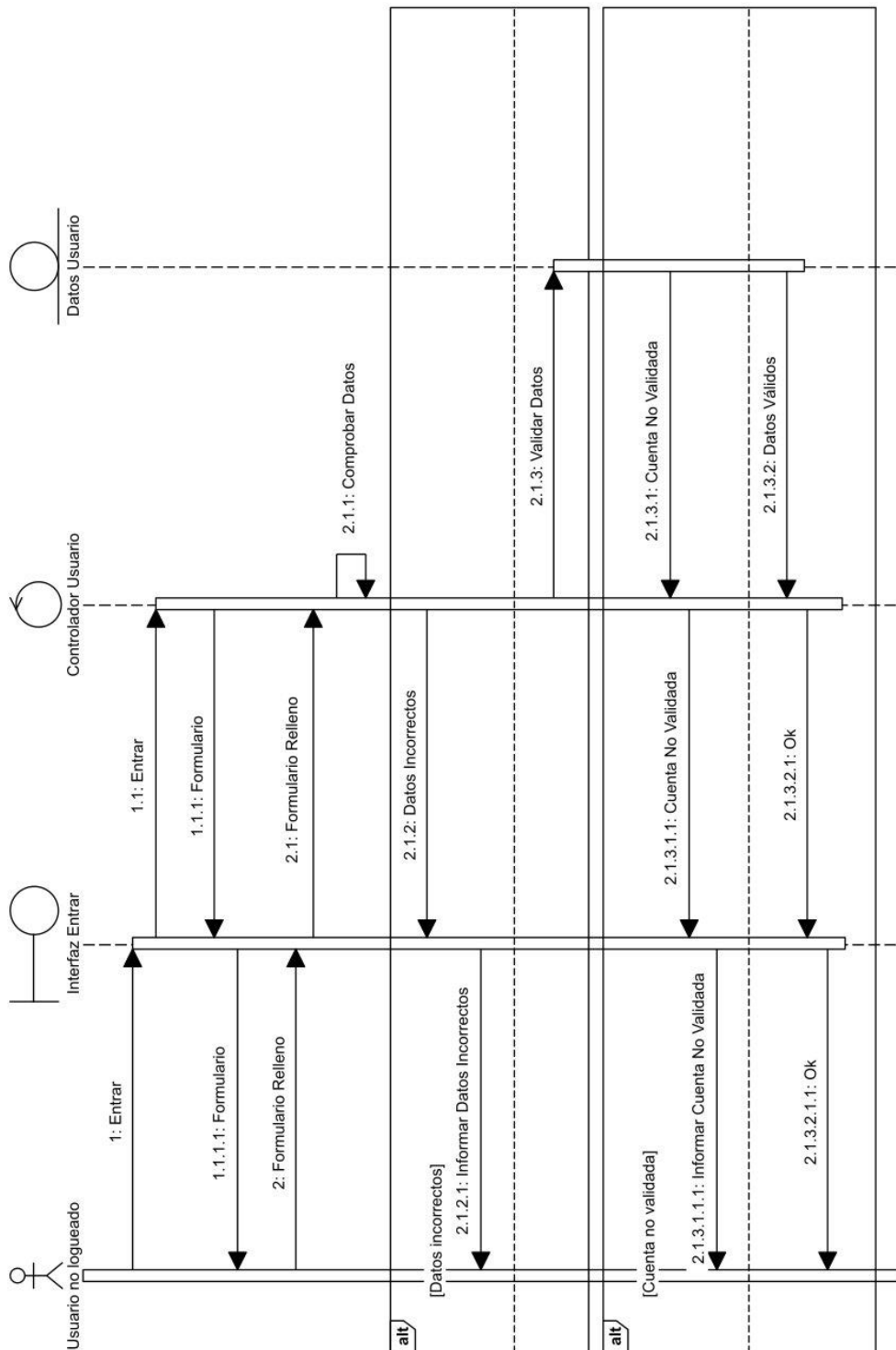


Figura 26: Diagrama de secuencia Entrar

Memoria del proyecto

6.5.3. CLASES DE ANÁLISIS

En este apartado se van a mostrar los diagramas de comunicación con el objetivo de determinar cómo se comunican y distribuyen las clases de análisis de los diagramas del punto anterior. Se tendrá un diagrama por cada paquete del sistema. En la *Figura 27* se muestra un ejemplo de clase de análisis correspondiente al paquete de Gestión de rutinas.

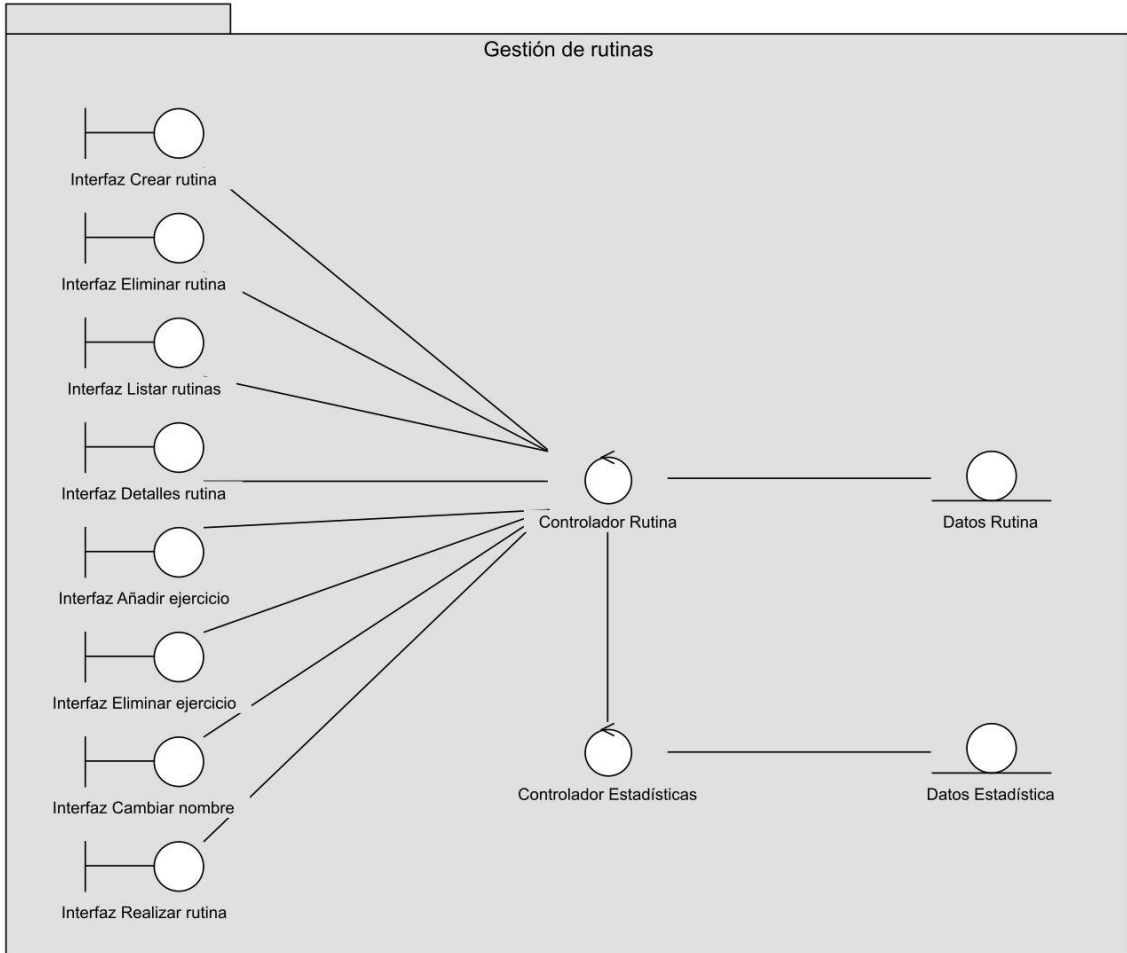


Figura 27: Diagrama de comunicación paquete Gestión de rutinas

Memoria del proyecto

6.5.4. VISTA DE ARQUITECTURA DEL MODELO DE ANÁLISIS

En este apartado se presenta un diagrama en la *Figura 28* de la vista de arquitectura del sistema, que será refinada en el siguiente anexo dedicado a la fase de diseño (*Anexo IV – Diseño del sistema*).

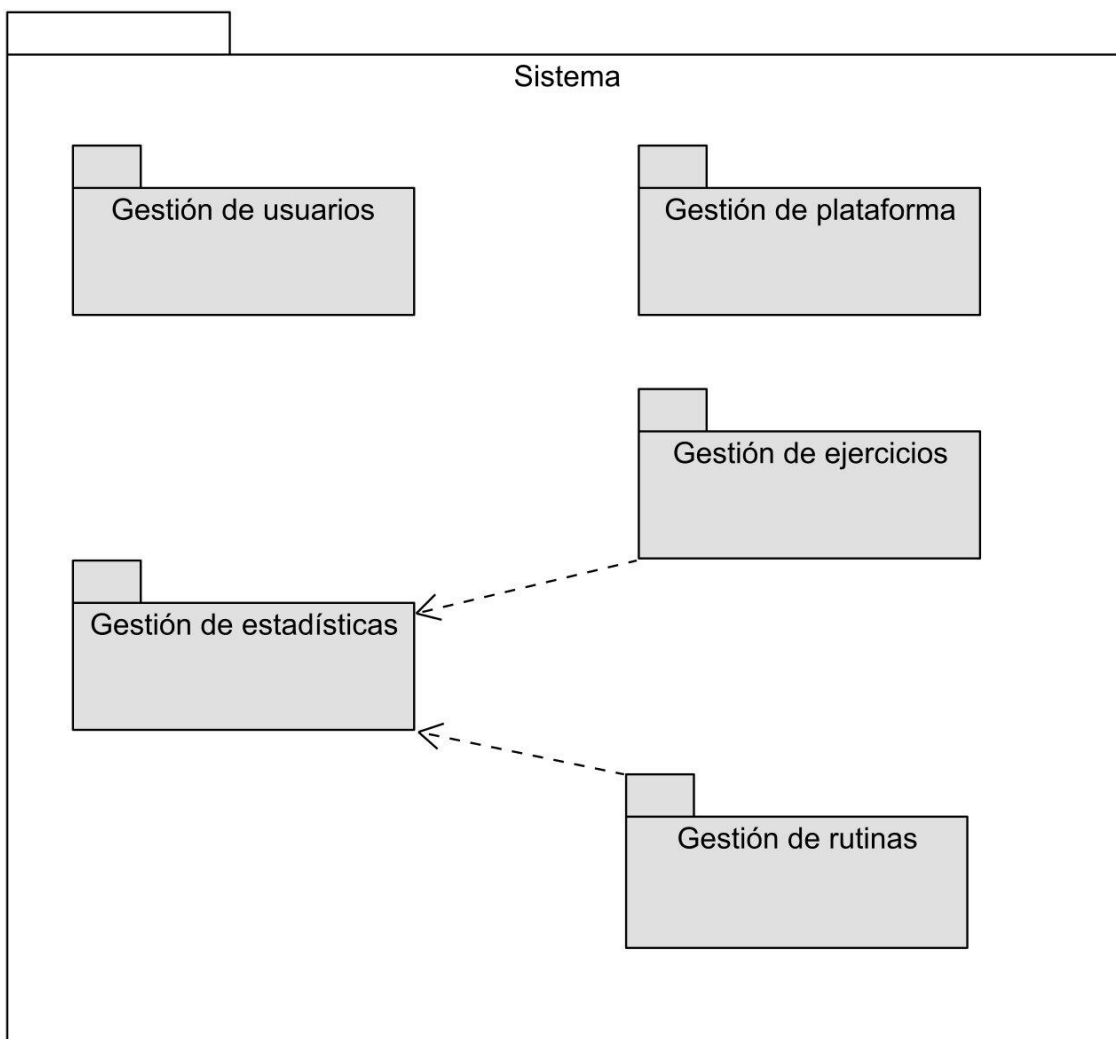


Figura 28: Vista de arquitectura del modelo de análisis

6.6. DISEÑO DEL SISTEMA

El objetivo de esta fase es documentar el modelo de diseño del sistema, que se centra en el dominio de la solución. Se trata de una aproximación a la implementación en cuanto a nombres de clases, funciones y atributos. Para entrar en mayor detalle sobre la fase de diseño del sistema se puede consultar el documento *Anexo IV – Diseño del sistema* que acompaña a esta memoria.

6.6.1. PATRONES ARQUITECTÓNICOS

Para la realización de este proyecto se seguirá la estructura del patrón *Model-View-ViewModel* (MVVM), que ayuda a separar la lógica de negocio de la interfaz de usuario, facilitando las pruebas, mantenimiento y escalabilidad de los proyectos. Para ello, existen tres componentes que nos ayudan a lograrlo

Memoria del proyecto

- **Modelo:** contiene una representación de los datos del sistema, su lógica de negocio y mecanismos de persistencia.
- **Vista:** se encarga de la representación de la información a través de elementos visuales, sin embargo, no obtiene los datos directamente del modelo.
- **Vista del modelo:** actúa como intermediario entre el modelo y la vista, encargándose de capturar las acciones sobre la vista, como puede ser la pulsación de un botón, interpretarlas y actuar en función de ellas. También realiza tareas de transformación de datos para hacer que los componentes de la vista y el modelo se entiendan.

En la *Figura 29* se muestra una imagen que detalla el intercambio de información entre estos componentes.

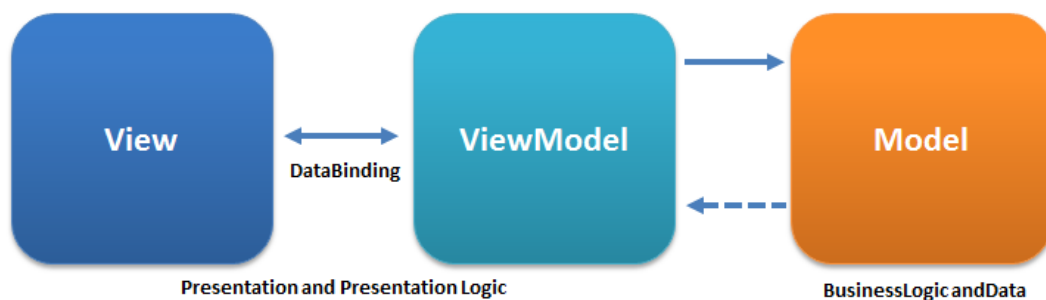


Figura 29: Patrón Model-View-ViewModel (MVVM)

6.6.2. SUBSISTEMAS DE DISEÑO

En este apartado se realiza una división de las distintas partes del sistema siguiendo el patrón MVVM mencionado en el punto anterior, así como la relación entre estas. Atendiendo a un criterio personal se ha decidido aplicar dicho patrón a cada paquete de gestión del sistema de modelado para una mejor comprensión. Por tanto, cada paquete de gestión consta a su vez de tres paquetes que se corresponden con la vista, la vista del modelo y el modelo. El diagrama se detalla en la *Figura 30*.

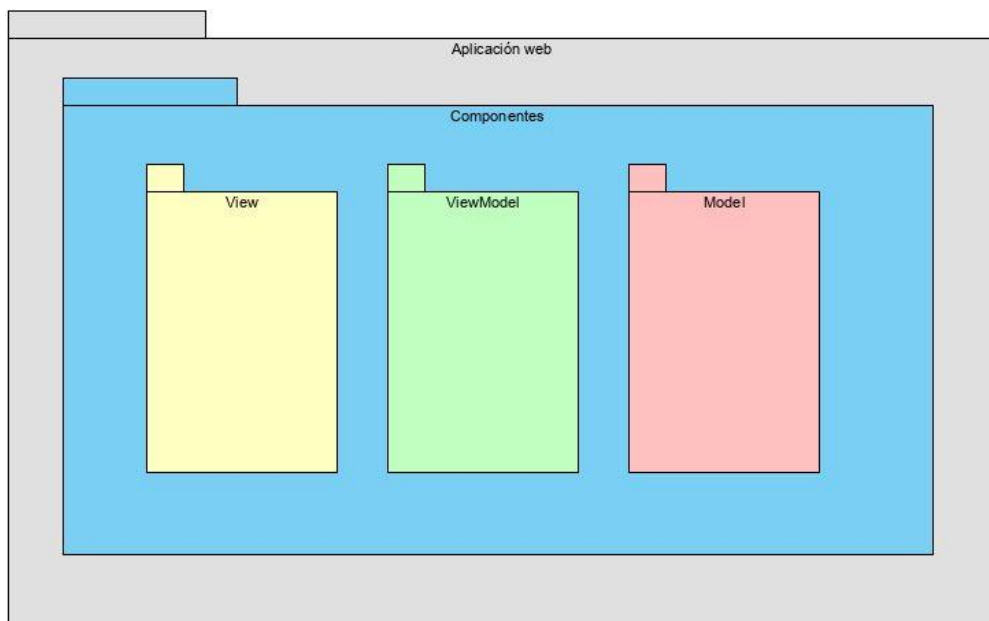


Figura 30: Diagrama de subsistemas

Memoria del proyecto

6.6.3. CLASES DE DISEÑO

Este apartado tiene como propósito especificar los contenidos en cada capa del patrón MVVM vistas en el punto anterior, así como las clases de cada paquete y los métodos de dichas clases. Se tienen tres capas: vista, modelo de la vista y modelo, que se muestran en las figuras *Figura 31*, *Figura 32* y *Figura 33* respectivamente.

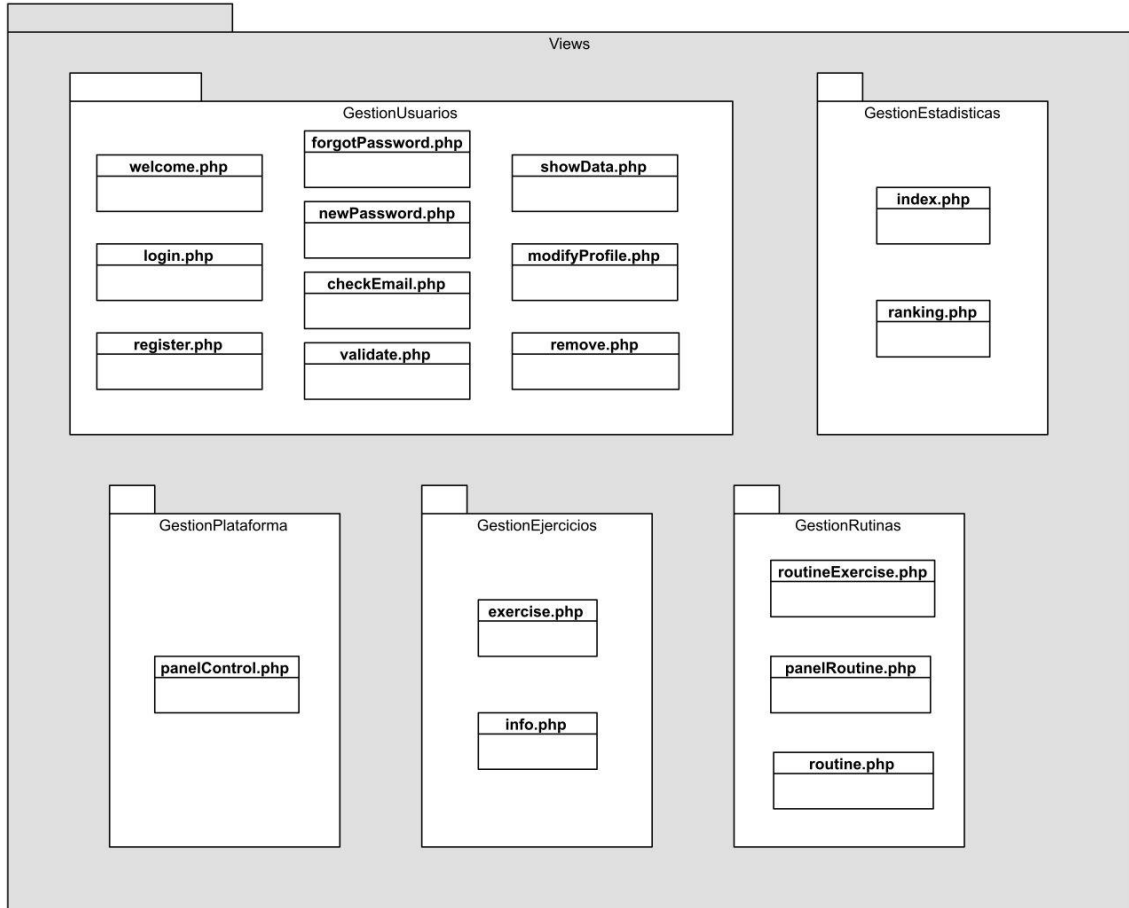


Figura 31: Capa View

Memoria del proyecto

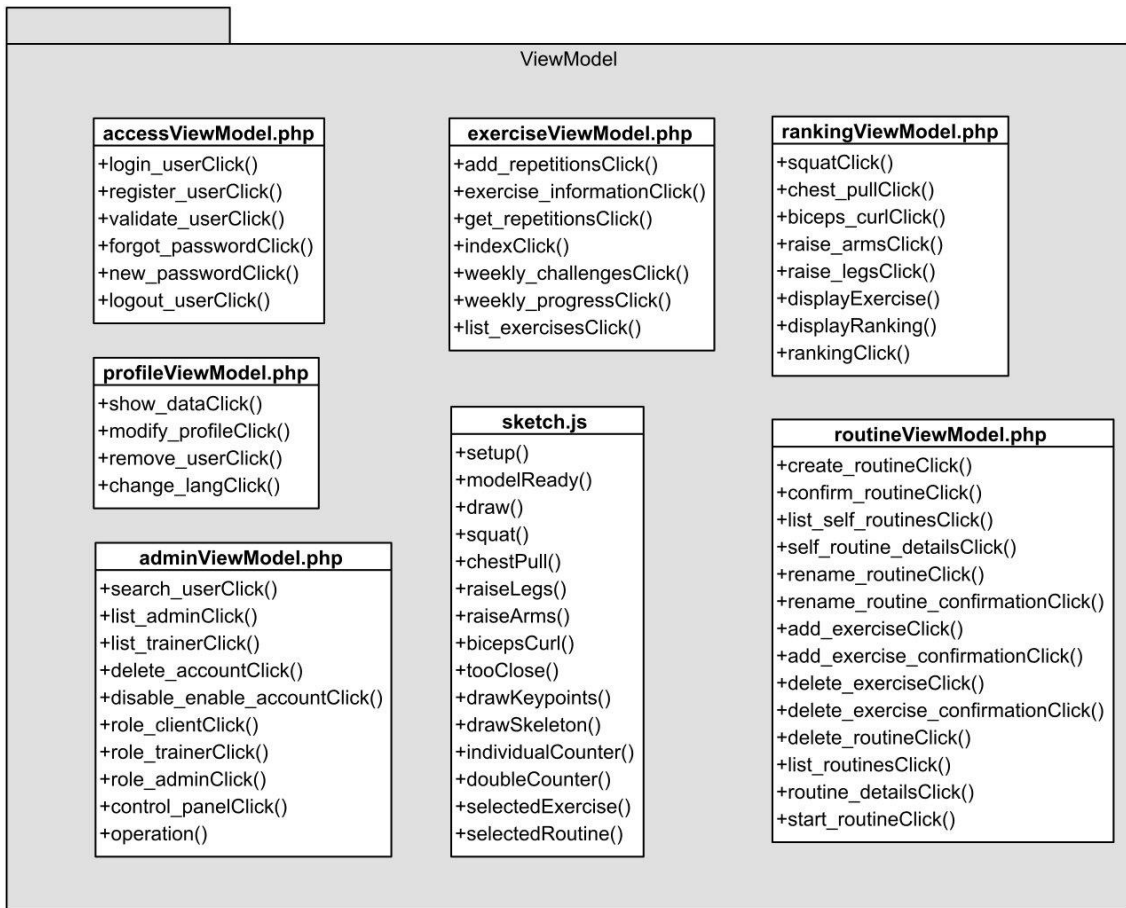


Figura 32: Capa View Model

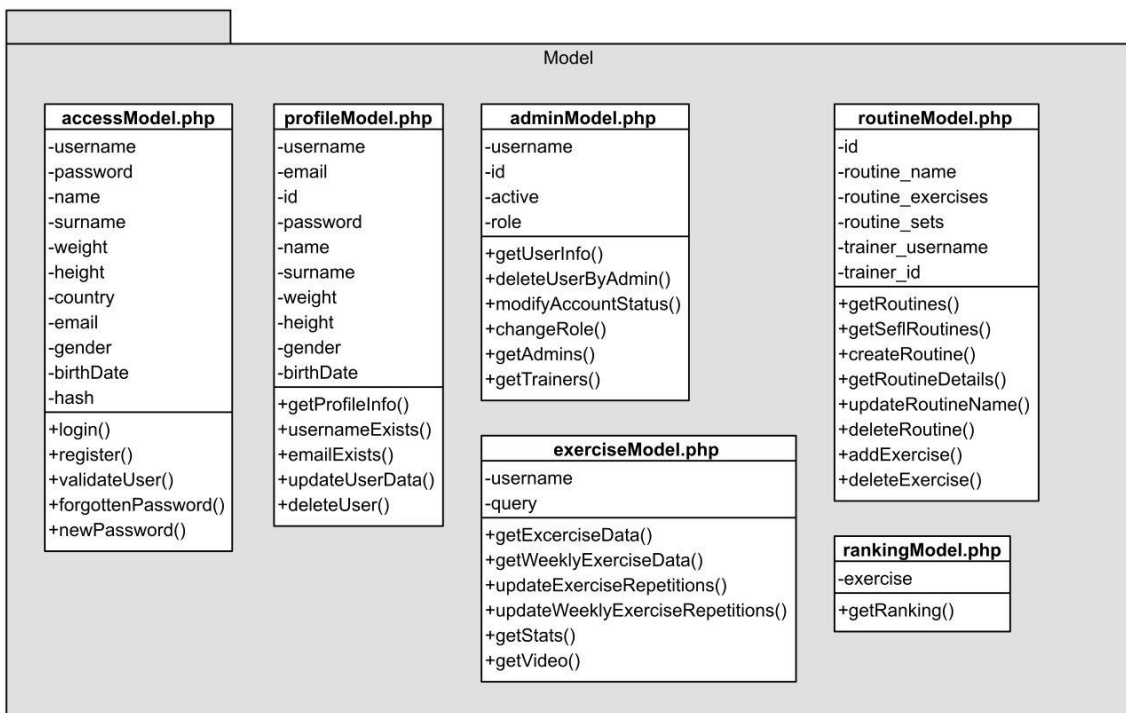


Figura 33: Capa Model

Memoria del proyecto

6.6.4. VISTA ARQUITECTÓNICA

En este apartado se presenta una visión general de la arquitectura del sistema. Para que el diagrama sea entendible se han omitido las clases pertenecientes a cada paquete, las cuales se pueden consultar en el apartado anterior. El diagrama se muestra en la *Figura 34*.

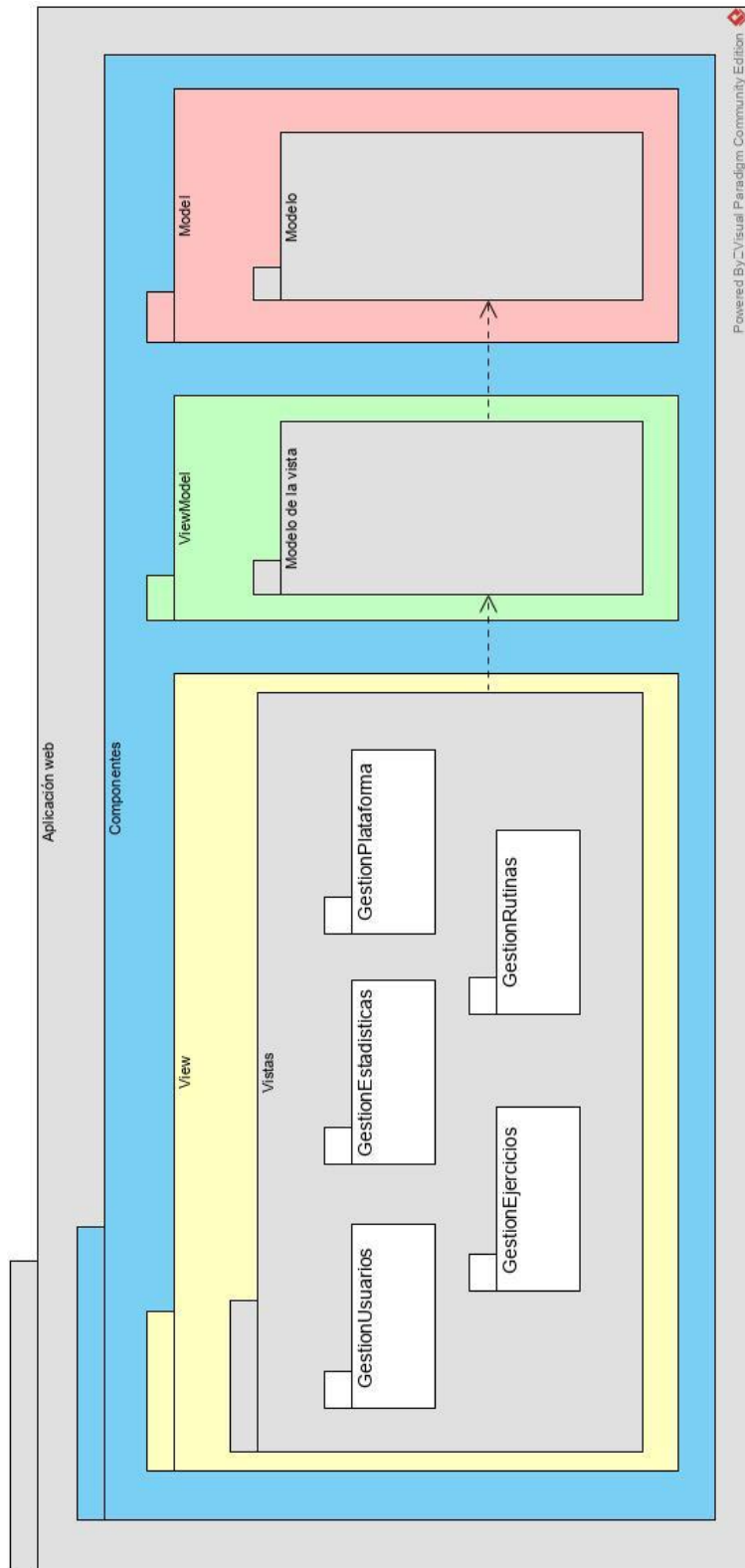


Figura 34: Vista arquitectónica

Memoria del proyecto

6.6.5. REALIZACIÓN DE CASOS DE USO

Este apartado tiene como objetivo detallar el intercambio de mensajes entre objetos mediante diagramas de secuencia de diseño, que muestran pasos más detallados de los casos de uso del sistema. En la *Figura 35* se puede observar un diagrama de secuencia a modo de ejemplo del caso de uso *Entrar*.

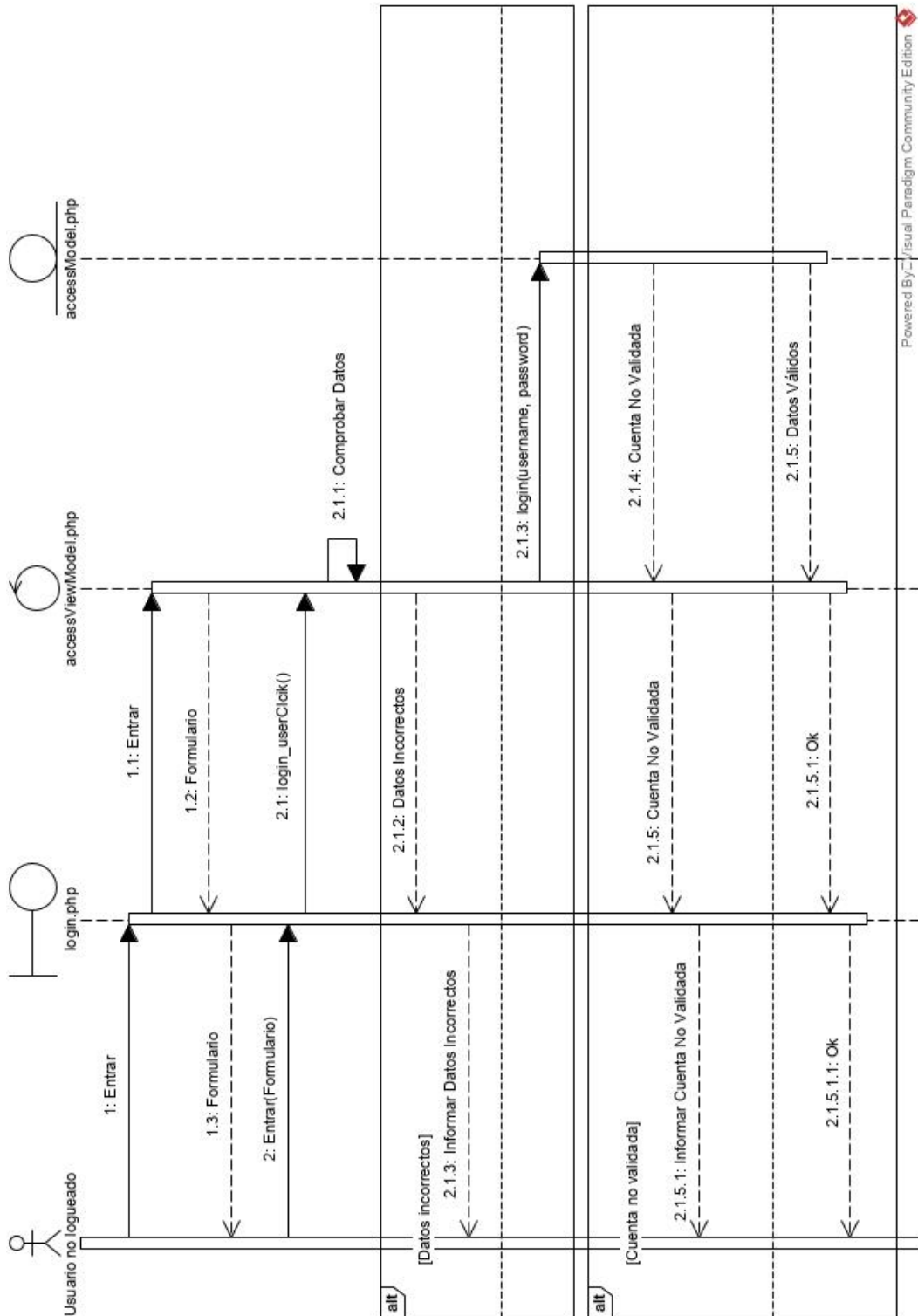


Figura 35: Diagrama de secuencia Entrar

Memoria del proyecto

6.6.6. DISEÑO DE LA BASE DE DATOS

Para almacenar y gestionar la información que el sistema ha de manejar, se usará una base de datos MySQL (Mariadb) con el esquema que se muestra en la *Figura 36*.

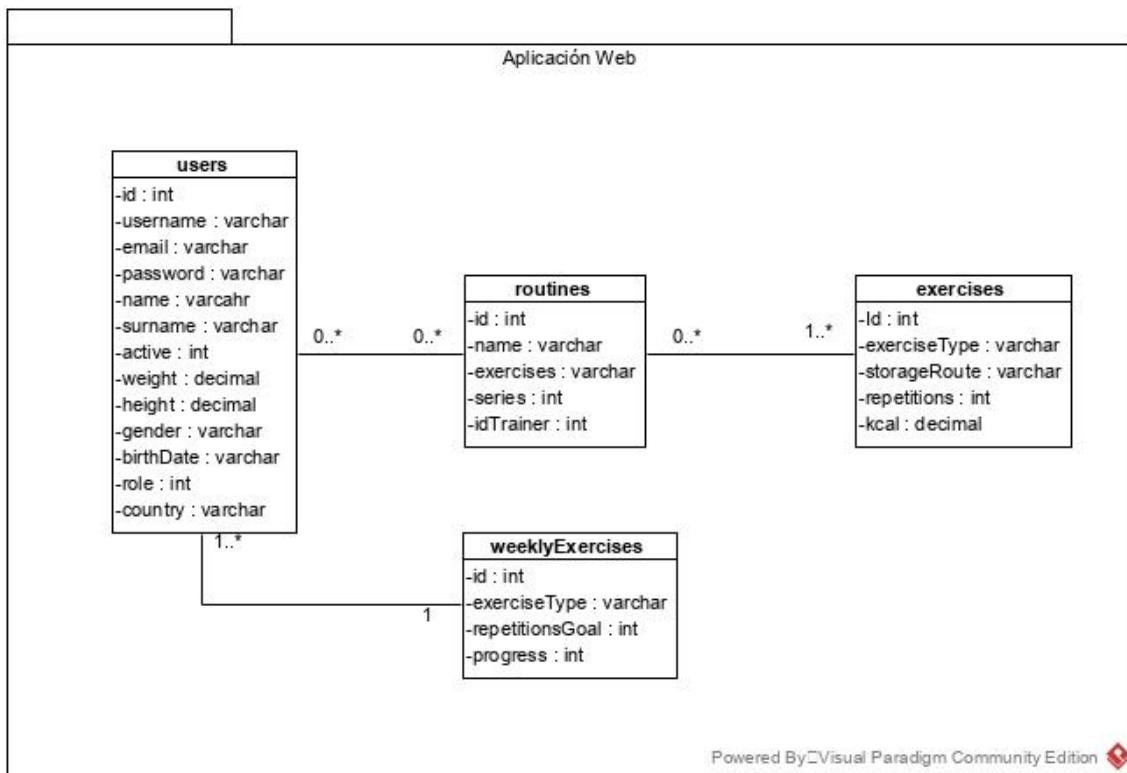


Figura 36: Base de datos

6.6.7. MODELO DE DESPLIEGUE

En el modelo de despliegue se tienen dos nodos principales:

- **Ciente Web:** se trata de la página web en el artefacto navegador web donde accederán los usuarios, independientemente de su rol. De este nodo se podrán tener varias instancias, ya que podrán acceder varios usuarios de manera simultánea
- **Servidor aplicación Web:** servidor en que están almacenados los datos que la aplicación requiere para su correcto funcionamiento. En este caso, al contrario que en el Cliente Web, solo se tendrá una instancia, ya que se dispone de un único servidor.

Las comunicaciones entre estos dos nodos, como es lógico, se realizarán a través de internet usando el protocolo HTTP. El diagrama asociado a este modelo se muestra en la *Figura 37*.

Memoria del proyecto

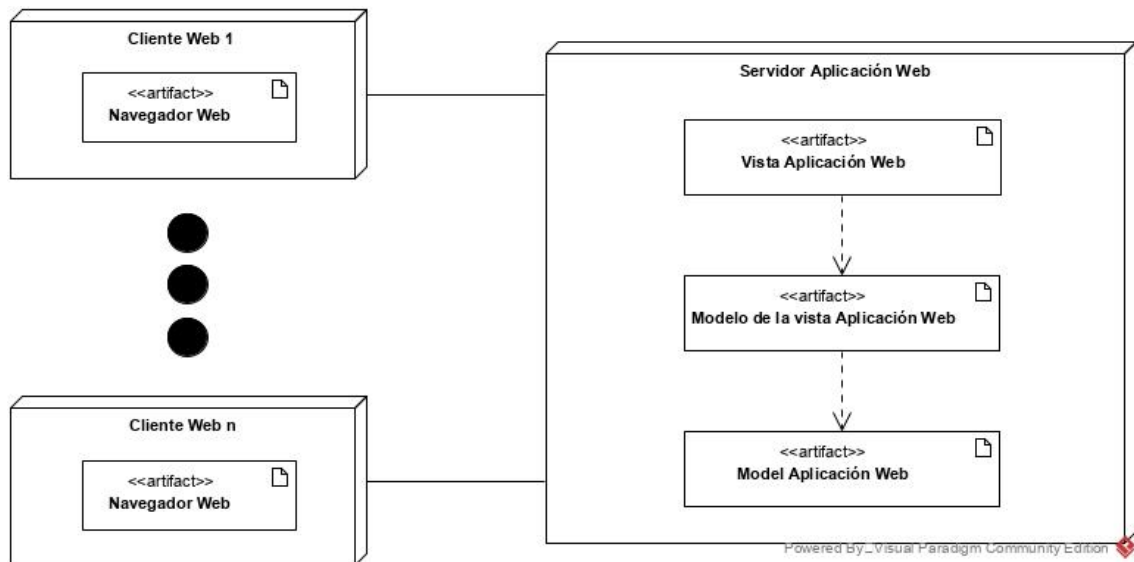


Figura 37: Modelo de despliegue

6.7. IMPLEMENTACIÓN

En esta fase se ha llevado a cabo la codificación del sistema, siendo una de las etapas más largas, en la cual se han empleado las técnicas y herramientas explicadas en el apartado *TÉCNICAS Y HERRAMIENTAS* de este documento. Para su desarrollo se ha seguido el esquema de la fase de diseño. Esto ha facilitado el proceso, obteniendo como resultado un proyecto bien estructurado.

Durante este proceso se ha llevado a cabo la codificación de las páginas web acompañadas de sus respectivos ficheros para darles funcionalidad y que todas las acciones del usuario se satisfagan correctamente. Para entrar en mayor detalle sobre la estructura del sistema y programación realizada, se puede consultar el *Anexo V – Documentación técnica*, que acompaña a esta memoria.

6.8. PRUEBAS

La realización de pruebas es esencial en el desarrollo de un proyecto, ya que permite comprobar el correcto funcionamiento del sistema y descubrir errores inesperados, enfocándolas desde distintos escenarios.

Se han realizado pruebas unitarias de cada componente mientras este se desarrollaba y a su finalización. Esto permite detectar pequeños errores en una etapa temprana y que no se propaguen a lo largo del desarrollo, lo que haría que fuese mucho más difícil identificarlos.

Una vez finalizado el desarrollo del proyecto, se han realizado pruebas completas de las funcionalidades del sistema para comprobar su correcto funcionamiento antes de realizar la entrega. Además, estas pruebas permiten comprobar si todos los componentes están bien integrados y su comportamiento trabajando juntos es el esperado.

Memoria del proyecto

6.9.FUNCIONALIDAD DEL SISTEMA

En este apartado se explicará de forma resumida el funcionamiento del sistema, así como sus funcionalidades de cara al usuario final. Para abordarlo de esta forma, se explicará teniendo en cuenta los paquetes utilizados en la fase de elicitación de requisitos.

Para entrar en mayor detalle sobre la funcionalidad del sistema, se puede consultar el *Anexo VI – Manual de usuario*, que acompaña a esta memoria.

6.9.1. GESTIÓN DE USUARIOS

La aplicación permite a los usuarios crear una cuenta de usuario, registrarse y modificar su contraseña en caso de olvidarla. Por otro lado, el usuario puede consultar datos de su perfil y modificarlos. En las figuras *Figura 38* y *Figura 39* se pueden observar las pantallas de login y registro respectivamente.

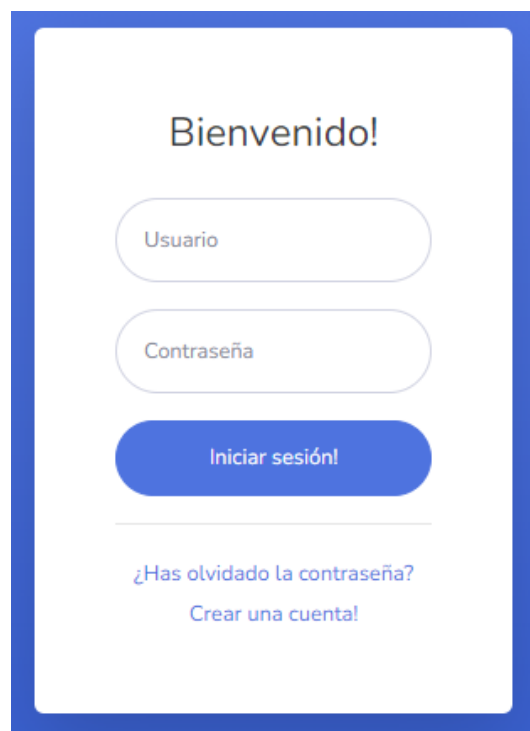


Figura 38: Pantalla de login

Memoria del proyecto

Crear una cuenta!

Nombre Apellidos

Correo

Usuario Hombre

País dd/mm/aaaa

Peso(Kg) Altura(cm)

Contraseña Confirmar

Registrarse

[¿Has olvidado la contraseña?](#)
[¿Ya tienes cuenta? Iniciar sesión!](#)

Figura 39: Pantalla de registro

En las imágenes de las figuras *Figura 40* y *Figura 41* se pueden observar las pantallas de perfil y modificación de perfil respectivamente.

Mostrar perfil



Tus datos

Nombre	Carlos
Apellidos	Valdunciel
Usuario	Ejemplo
Correo electrónico	valduncielcarlos@gmail.com
Cumpleaños	1999-08-03
Peso(Kg)	75.00
Altura(cm)	180.00
Género	Hombre

Modificar perfil

Figura 40: Pantalla de perfil

Modificar perfil

Nombre	<input type="text" value="Carlos"/>
Apellidos	<input type="text" value="Valdunciel"/>
Usuario	<input type="text" value="Ejemplo"/>
Correo electrónico	<input type="text" value="valduncielcarlos@gmail.com"/>
Cumpleaños	<input type="text" value="dd/mm/aaaa"/> 
Peso(Kg)	<input type="text" value="75.00"/>
Altura(cm)	<input type="text" value="180.00"/>
Género	<input type="text" value="Hombre"/> 

Contraseña	<input type="password"/>
Confirmar contraseña	<input type="password"/>

[Actualizar Perfil](#)

[Eliminar Cuenta](#)

Figura 41: Pantalla de modificar perfil

Memoria del proyecto

Por último, se muestran las pantallas que permiten al usuario solicitar recuperar su contraseña introduciendo su email y cambiar su contraseña olvidada en las figuras *Figura 42* y *Figura 43* respectivamente.



¿Has olvidado tu contraseña?

Lo entendemos, estas cosas pasan. Simplemente ingrese su dirección de correo electrónico a continuación y le enviaremos un enlace para restablecer su contraseña.

Correo

Cambiar contraseña

[Crear una cuenta!](#)

[¿Ya tienes cuenta? Iniciar sesión!](#)

Figura 42: Pantalla solicitud email



Recuperar contraseña

Ingrese su nueva contraseña, asegúrese de no compartirla con nadie.

Clave

Confirmar

Recuperar contraseña

Figura 43: Pantalla cambio de contraseña

6.9.2. GESTIÓN DE ESTADÍSTICAS

La aplicación permite a los usuarios visualizar sus estadísticas y acceder a un ranking de aquellos con mayor número de repeticiones mediante distintos tipos de gráficos. En la figura *Figura 44* se puede observar el gráfico de progreso que indica el número de repeticiones de cada ejercicio predeterminado de la plataforma, mientras que en la figura *Figura 45* se muestra el progreso de los objetivos semanales, que tratan de completar 100 repeticiones de cada ejercicio.

Memoria del proyecto

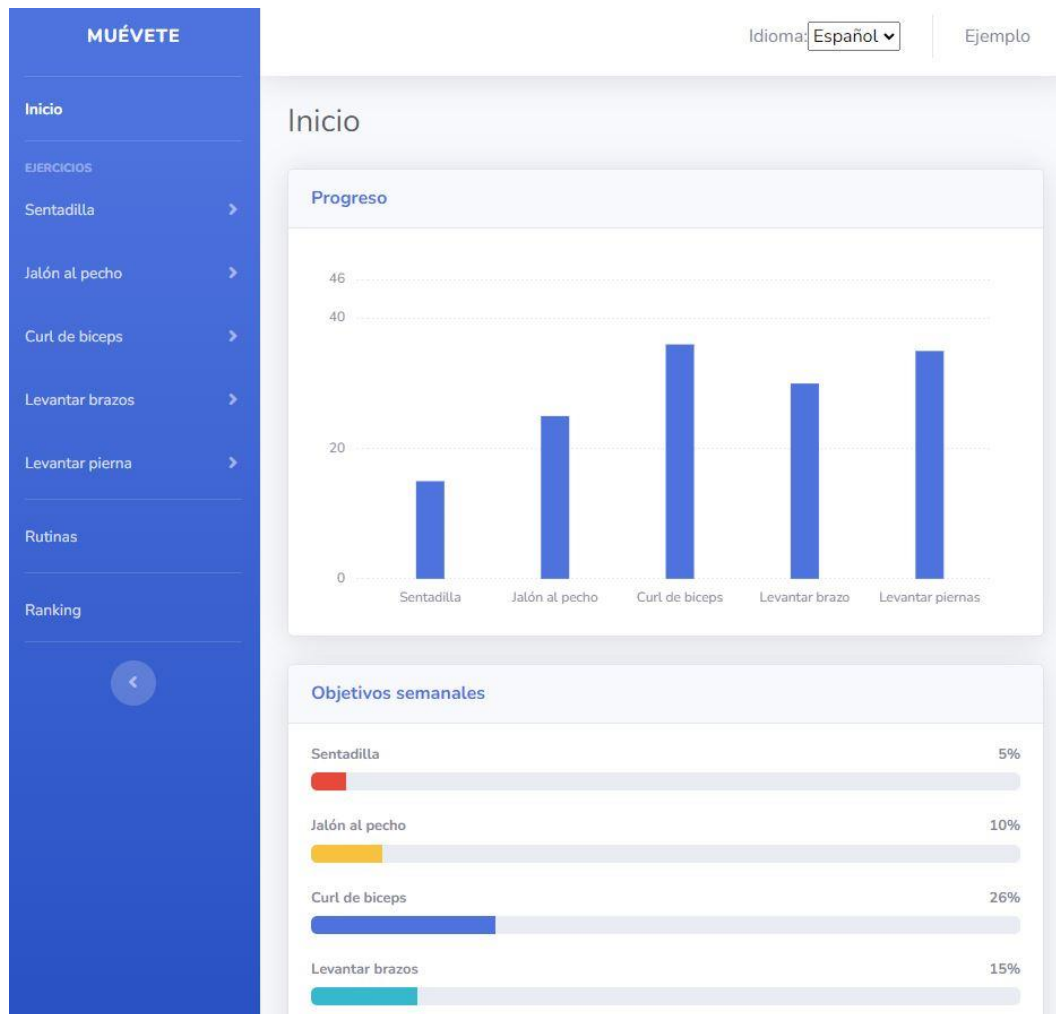


Figura 44: Pantalla de inicio / estadísticas

Memoria del proyecto

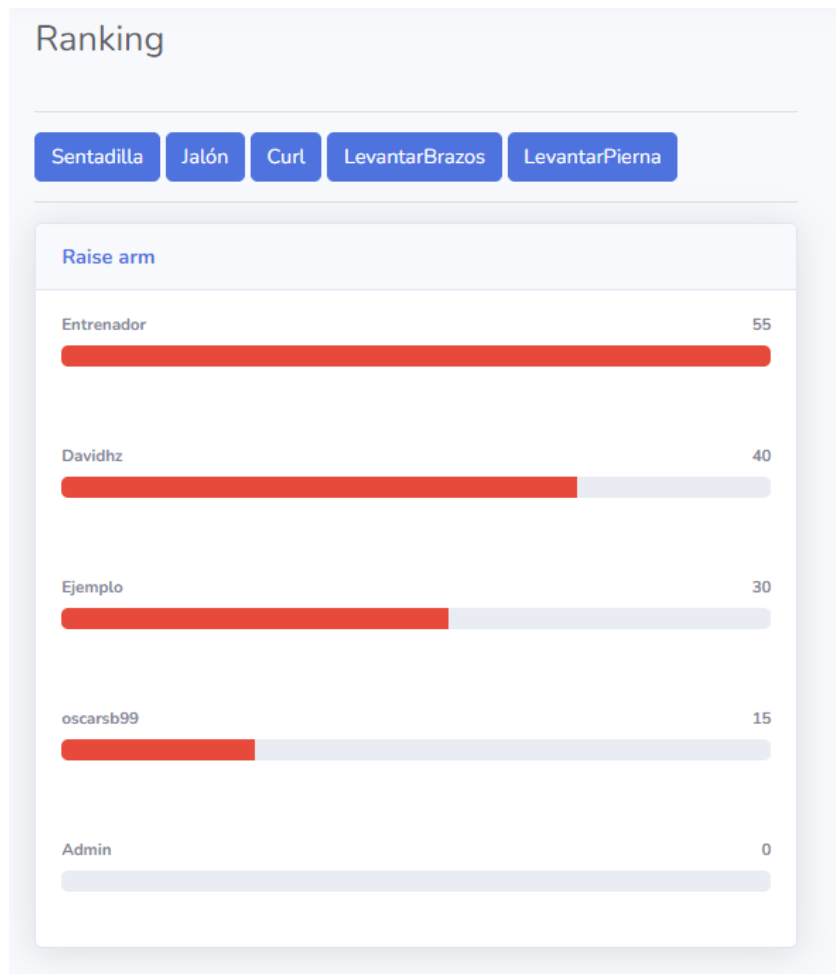


Figura 45: Pantalla de ranking

6.9.3. GESTIÓN DE LA PLATAFORMA

La aplicación permite a los administradores de la plataforma gestionar las cuentas de otros usuarios siempre y cuando el usuario en cuestión no sea otro administrador. En la figura *Figura 46* se muestra la interfaz del panel de control.

Memoria del proyecto

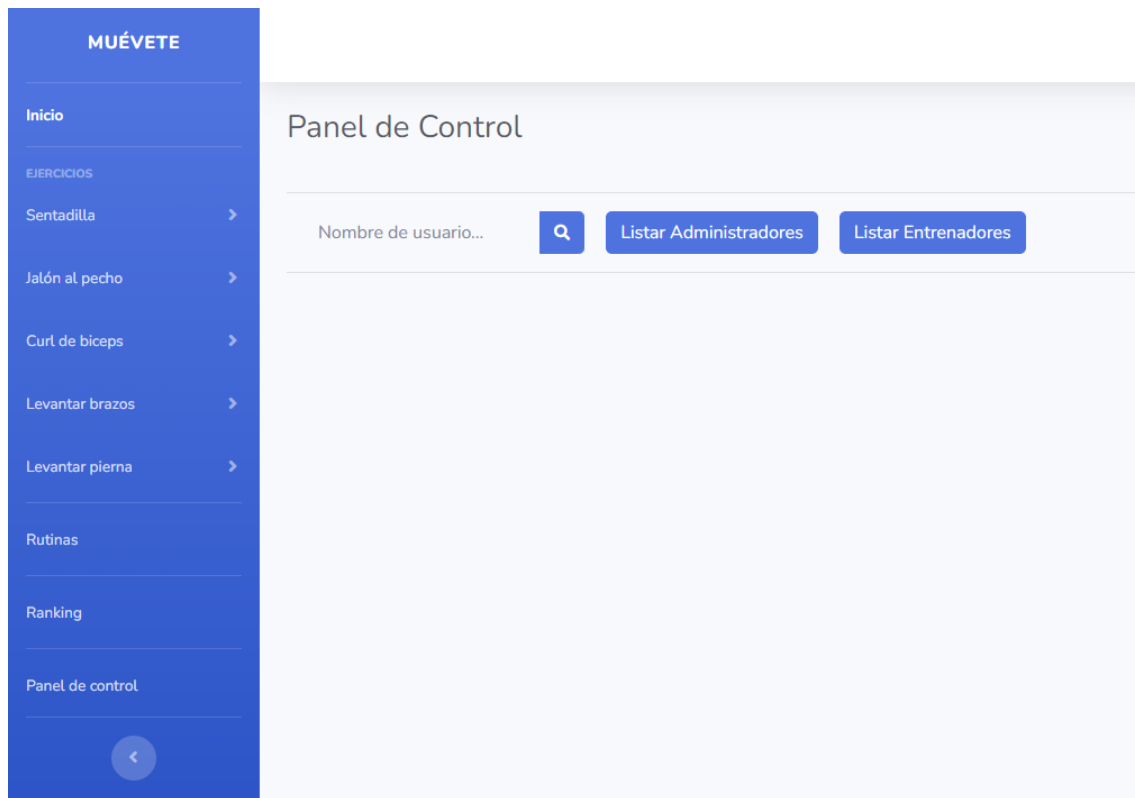


Figura 46: Pantalla de panel de control

Desde este panel de control el administrador puede buscar un usuario para ver los detalles de su cuenta y gestionarla, listar los administradores y listar los entrenadores, tal y como se puede observar en las figuras *Figura 47*, *Figura 48* y *Figura 49* respectivamente.

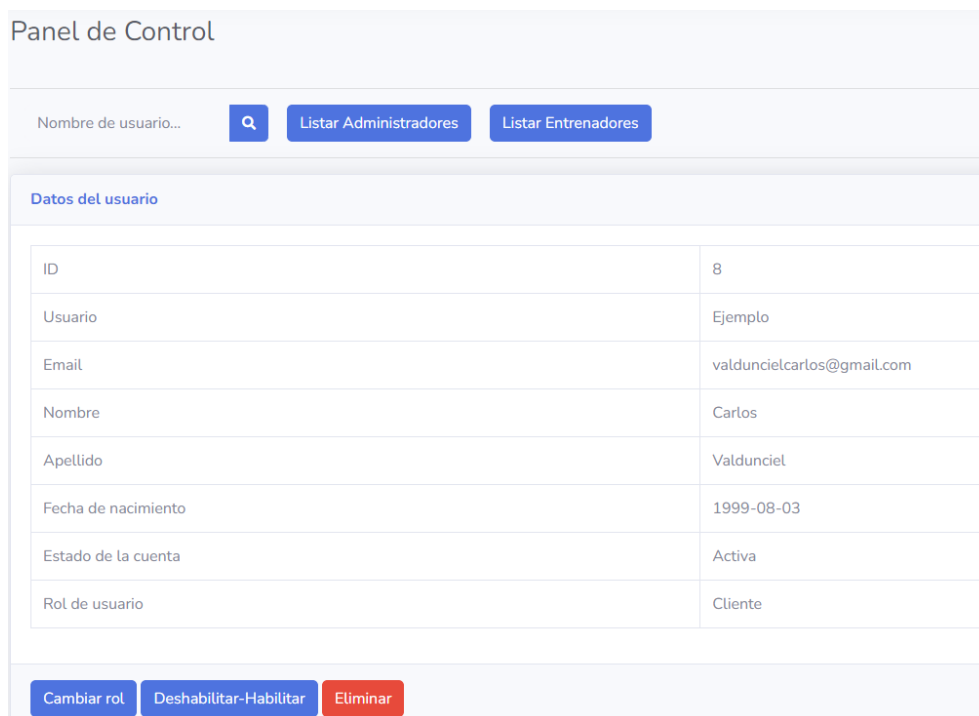


Figura 47: Buscar usuario

Memoria del proyecto

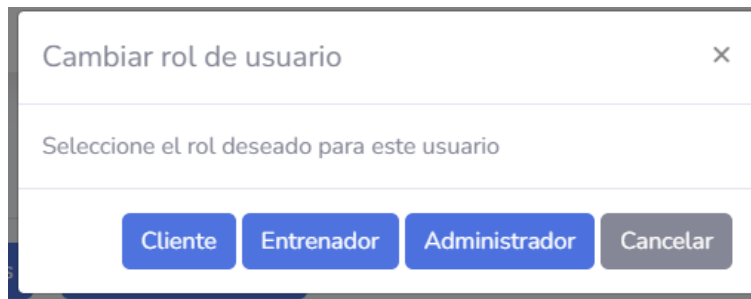
Listado de administradores				
ID	Nombre de usuario	Nombre	Apellidos	Email
8	Ejemplo	Carlos	Valdunciel	valduncielcarlos@gmail.com
9	Admin	Carlos	Valdunciel Gonzalo	carlosvg@usa.es

Figura 48: Listar administradores

Listado de entrenadores				
ID	Nombre de usuario	Nombre	Apellidos	Email
10	Entrenador	Alberto	García	valduncielcarlos99@gmail.com

Figura 49: Listar entrenadores

Como se muestra en la figura *Figura 47*, cuando el usuario buscado no es un administrador, se pueden realizar tres acciones sobre su cuenta: cambiar su rol, cambiar el estado de esta y eliminarla, tal y como se puede observar en las figuras *Figura 50*, *Figura 51* y *Figura 52* respectivamente.

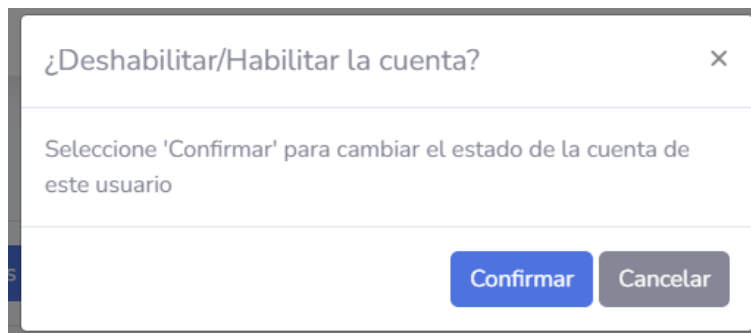


Cambiar rol de usuario

Seleccione el rol deseado para este usuario

Cliente Entrenador Administrador Cancelar

Figura 50: Cambiar rol



¿Deshabilitar/Habilitar la cuenta?

Seleccione 'Confirmar' para cambiar el estado de la cuenta de este usuario

Confirmar Cancelar

Figura 51: Cambiar estado de cuenta

Memoria del proyecto

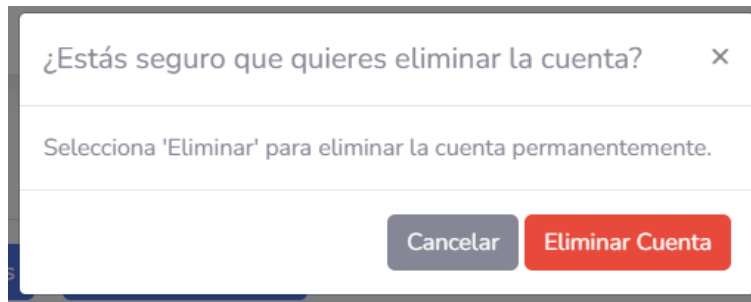


Figura 52: Eliminar cuenta

6.9.4. GESTIÓN DE EJERCICIOS

Esta sección permite al usuario listar los distintos ejercicios predeterminados que incorpora la plataforma con sus tres dificultades, así como acceder a una pequeña descripción del mismo tal y como se muestra en las figuras *Figura 53* y *Figura 54* respectivamente. Estas funcionalidades son accedidas desde el desplegable que incorpora cada uno de los ejercicios.

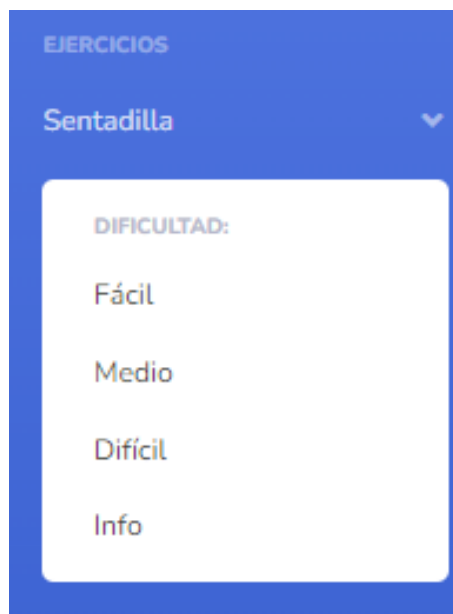


Figura 53: Desplegable ejercicio

La Sentadilla es un ejercicio físico que se lleva a cabo para desarrollar los músculos y fortalecer los tendones y los ligamentos de las piernas. También permite tonificar los glúteos y aporta beneficios a la cadera.

El ejercicio consiste en flexionar las rodillas y bajar el cuerpo manteniendo la verticalidad, para luego regresar a una posición erguida.

Figura 54: Información ejercicio

Memoria del proyecto

Si se desea realizar un ejercicio basta con presionar en una de las dificultades mostradas en la figura *Figura 53* y se dirigirá al usuario a la pantalla de realización de ejercicios, como se puede observar en la figura *Figura 55*

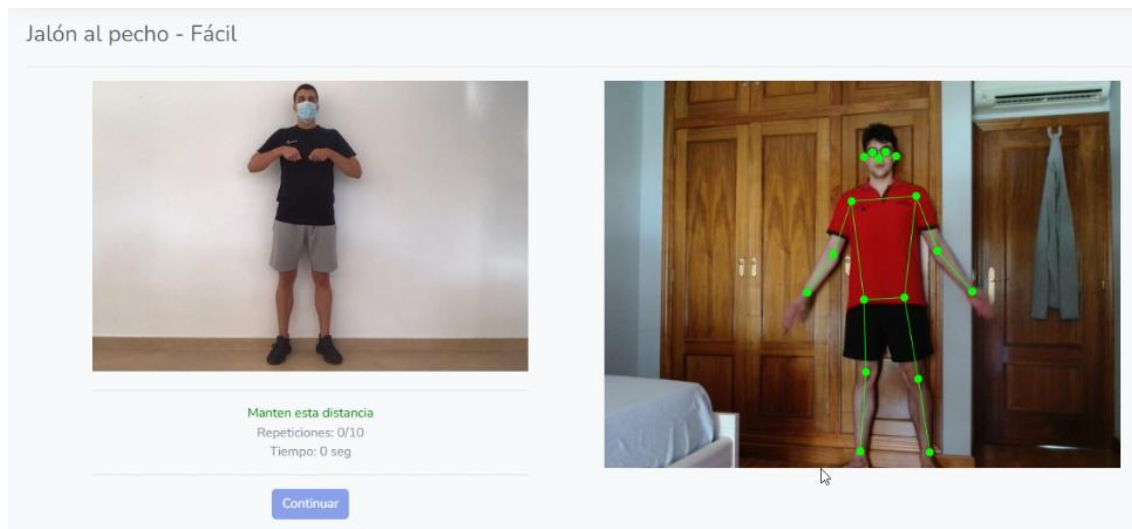


Figura 55: Realización de ejercicio

6.9.5. GESTIÓN DE RUTINAS

Esta sección permite al usuario realizar rutinas de ejercicio y a los entrenadores crear, listar y modificar sus rutinas. Para realizar una rutina el usuario se debe dirigir a la sección "Rutinas" en la barra de navegación lateral. Una vez en esta pantalla, se le permitirá ver un listado de las rutinas disponibles creadas por los entrenadores y ver los detalles de una rutina, tal y como se puede observar en las figuras *Figura 56* y *Figura 57* respectivamente.



Figura 56: Listado de rutinas

Memoria del proyecto

The screenshot shows a web interface for managing routines. At the top, there is a header 'Rutinas' and a button 'Listar rutinas'. Below this is a section titled 'Detalles de rutina' containing a table with the following data:

Nombre de rutina	Ejercicios	Número de series	Creador
Principiante	curlBiceps-levantarBrazo-levantarPierna	2	Ejemplo

At the bottom of the details section, there is a button 'A entrenar!'.

Figura 57: Detalles de rutina

Si el usuario desea empezar la rutina se le redirigirá a una pantalla igual a la de la figura *Figura 55*.

El panel de rutinas disponible para los entrenadores crear rutinas, listar sus propias rutinas, no las de los otros entrenadores y ver los detalles de estas, tal y como se puede observar en las figuras *Figura 58*, *Figura 59* y *Figura 60* respectivamente.

The screenshot shows a form titled 'Panel de Rutinas'. It contains two buttons at the top: 'Crear Rutina' and 'Listar Rutinas'. Below these are two input fields: 'Nombre de la rutina...' and 'Número de series...'. Underneath the input fields is a list of exercises with checkboxes:

- Sentadilla
- Jalón al pecho
- Curl de bíceps
- Levantar brazos
- Levantar piernas

At the bottom of the form is a 'Confirmar' button.

Figura 58: Creación de rutina

Memoria del proyecto

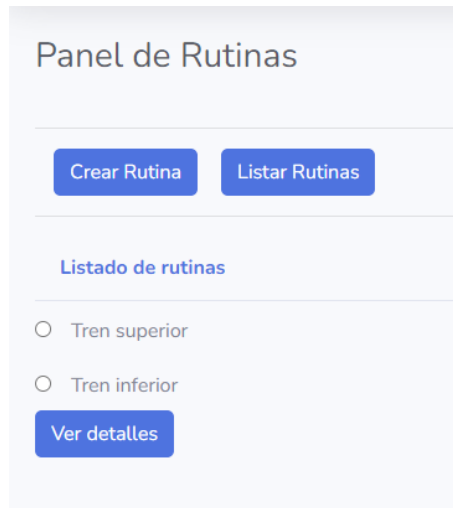


Figura 59: Listado de rutinas propias

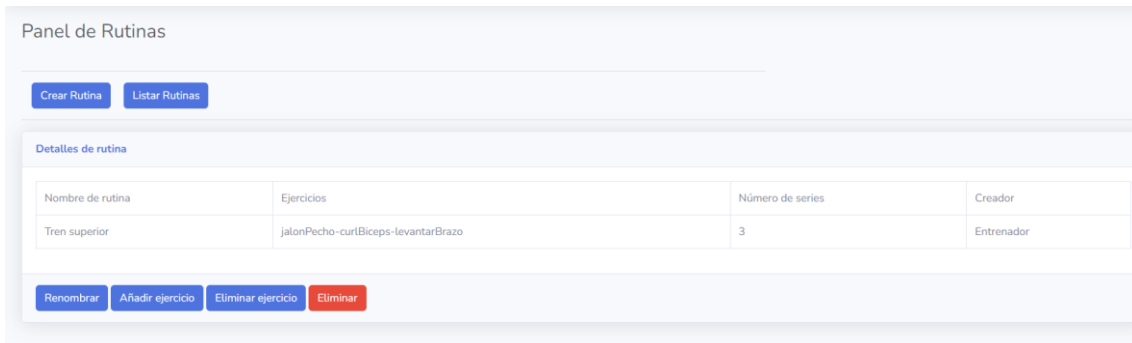


Figura 60: Detalles de rutina propia

Como se muestra en la figura *Figura 60*, se pueden realizar cuatro acciones sobre la rutina: cambiar su nombre, añadirle un ejercicio, eliminarle un ejercicio y eliminarla, tal y como se puede observar en las figuras respectivamente.

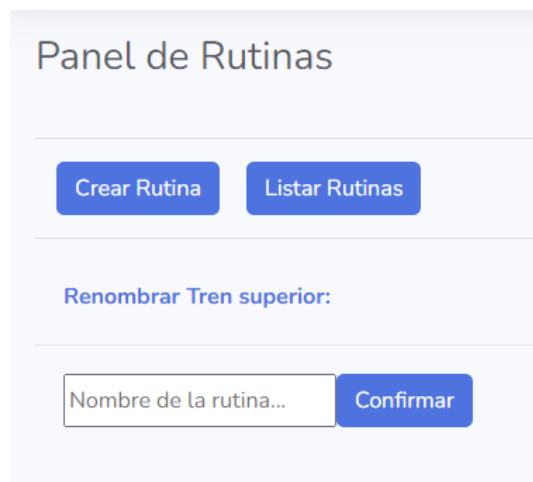
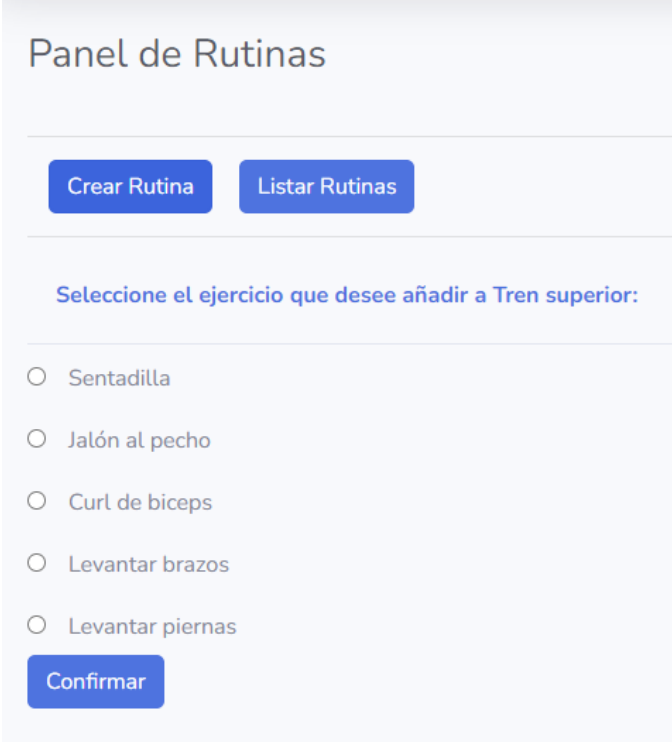


Figura 61: Renombrar rutina

Memoria del proyecto



Panel de Rutinas

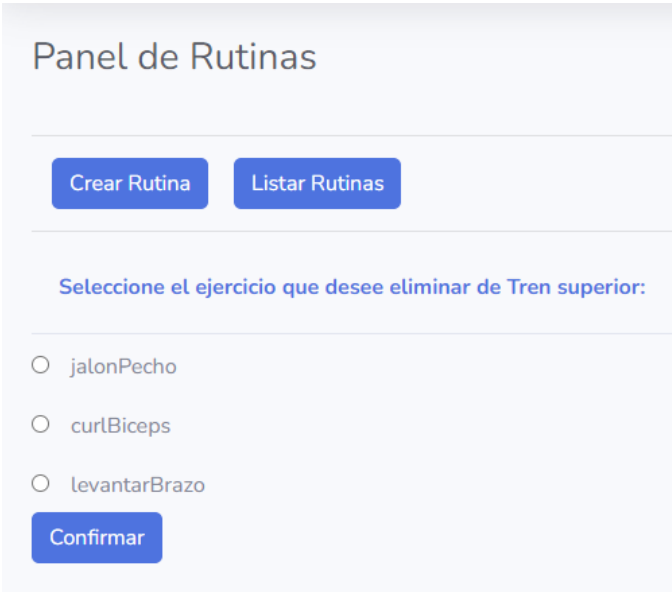
Crear Rutina Listar Rutinas

Seleccione el ejercicio que desee añadir a Tren superior:

- Sentadilla
- Jalón al pecho
- Curl de biceps
- Levantar brazos
- Levantar piernas

Confirmar

Figura 62: Añadir ejercicio a rutina



Panel de Rutinas

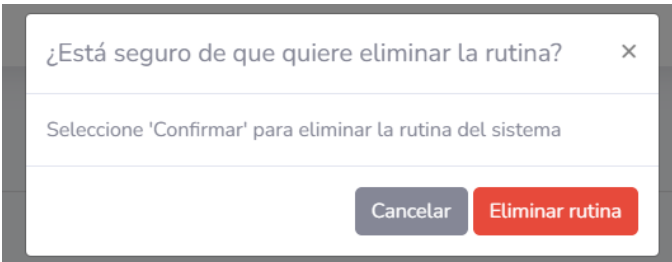
Crear Rutina Listar Rutinas

Seleccione el ejercicio que desee eliminar de Tren superior:

- jalonPecho
- curlBiceps
- levantarBrazo

Confirmar

Figura 63: Eliminar ejercicio de rutina



¿Está seguro de que quiere eliminar la rutina? ×

Seleccione 'Confirmar' para eliminar la rutina del sistema

Cancelar Eliminar rutina

Figura 64: Eliminar rutina

7. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

En este apartado se recogen las conclusiones a las que se han llegado una vez finalizadas todas las fases del proyecto *“Plataforma para el seguimiento y la realización de ejercicios físicos”*. Además, se incluye un apartado de líneas de trabajo futuras con ideas para mejorar la aplicación.

7.1. CONCLUSIONES

Finalizado el proyecto, se puede afirmar que se ha cumplido con todos los requisitos y objetivos que se habían planteado al inicio del proyecto y otros que se incluyeron más tarde. Entre ellos destacan:

- Desarrollo de un sistema capaz de detectar y monitorizar los movimientos realizados por el usuario de una forma bastante fluida dentro de la aplicación.
- La aplicación permite a los usuarios entrenadores crear sus propias rutinas y al resto de usuarios de la plataforma llevarlas a cabo.
- La aplicación permite a los usuarios administradores llevar un control eficiente de las cuentas de usuarios pudiendo realizar acciones propias de cualquier sistema de gestión.
- Se han adquirido conocimientos en lenguajes de programación nuevos no estudiados a lo largo de la carrera como PHP y Javascript, siendo estos dos de los más usados en el mundo laboral.
- Se han adquirido conocimientos básicos de machine learning y como son entrenados los modelos, como en el caso de PoseNet, aplicados a visión artificial abriendo un gran abanico de posibilidades para la aplicación de esta tecnología.
- Por otro lado, se han puesto en práctica muchos otros conocimientos adquiridos a lo largo de la carrera, especialmente en lo que a la ingeniería del software y gestión de proyectos se refiere.
- Se ha comprobado de primera mano el esfuerzo y dedicación que supone la realización de un proyecto individual relativamente grande, al menos comparados con los que se habían llevado a cabo con anterioridad. Además, el seguir un proceso definido como es el Proceso Unificado, ha sido revelador en el aspecto de cómo facilita la estructuración y desarrollo del software, aunque requiera más tiempo en las fases iniciales. Quizás, esto último, en proyectos más pequeños realizados a lo largo de estos años, no parecía tan importante ni útil a modo personal.

7.2. LÍNEAS DE TRABAJO FUTURAS

Algunas funcionalidades que han quedado en el tintero, o que se podrían incluir en el futuro son las siguientes:

- Permitir a los entrenadores, aparte de crear rutinas, crear nuevos ejercicios de una forma sencilla a través del reconocimiento de la posición corporal con uno de los modelos ya usados. Esto se podría desarrollar obteniendo la posición de los puntos clave en intervalos pequeños durante cierto periodo de tiempo.
- Llevar un control más exhaustivo durante la monitorización de los movimientos del usuario en la realización de ejercicios, pudiendo generar informes detallados sobre los errores cometidos.
- Permitir al usuario llevar un control de las calorías quemadas durante la realización de ejercicios. Esto es una funcionalidad que se planteó en un inicio, pero que finalmente

Memoria del proyecto

no se llegó a implementar. De hecho, en el apartado *VALOR MET* de los conceptos teóricos ya queda planteada la forma en que esto se llevaría a cabo.

- Habilitar un foro o algún mecanismo de feedback para que los usuarios puedan compartir ideas sobre qué les gustaría que incluyese la aplicación o mejoras sobre funcionalidades ya existentes.

Memoria del proyecto

8. REFERENCIAS

- Aplicación 7 Minute Workout.* (s.f.). Obtenido de <https://cutt.ly/KLr62oq>
- Aplicación adidas Training: Home Workout.* (s.f.). Obtenido de <https://cutt.ly/xLr7Wcl>
- Aplicación Ejercicios en casa: Sin equipo.* (s.f.). Obtenido de <https://cutt.ly/6Lr1c0P>
- Aplicación Kemtai.* (s.f.). Obtenido de <https://kentai.com/>
- Aplicación Virtuagym Fitness - Home & Gym.* (s.f.). Obtenido de <https://cutt.ly/NLr7a6P>
- Compendium.* (s.f.). Obtenido de <https://cutt.ly/ELr1OXm>
- Definición herramientas CASE.* (s.f.). Obtenido de <https://cutt.ly/ELr1OXm>
- Documentación Apache.* (s.f.). Obtenido de <https://httpd.apache.org/docs/2.4/>
- Documentación Doxygen.* (s.f.). Obtenido de <https://www.doxygen.nl/>
- Documentación HTML.* (s.f.). Obtenido de <https://developer.mozilla.org/es/docs/Web/html>
- Documentación Javascript.* (s.f.). Obtenido de <https://cutt.ly/GLr0nnz>
- Documentación MariaDB.* (s.f.). Obtenido de <https://mariadb.org/documentation/>
- Documentación Microsoft Project.* (s.f.). Obtenido de <https://cutt.ly/0Lr0TWo>
- Documentación ml5.js.* (s.f.). Obtenido de <https://learn.ml5js.org/#/>
- Documentación ngrok.* (s.f.). Obtenido de <https://ngrok.com/docs>
- Documentación p5.js.* (s.f.). Obtenido de <https://p5js.org/>
- Documentación PHP.* (s.f.). Obtenido de <https://www.php.net/manual/en/>
- Documentación phpMyAdmin.* (s.f.). Obtenido de <https://docs.phpmyadmin.net/en/latest/>
- Documentación sb-admin-2.* (s.f.). Obtenido de <https://startbootstrap.com/theme/sb-admin-2>
- Documentación Visual Paradigm.* (s.f.). Obtenido de <https://cutt.ly/8Lr077i>
- Documentación Visual Studio Code.* (s.f.). Obtenido de <https://code.visualstudio.com/docs>
- Ejemplos PHP.* (s.f.). Obtenido de <https://www.w3schools.com/php/>
- EZEstimate.* (s.f.). Obtenido de Manual de EZEstimate Asignatura de Gestión de Proyectos
- García Peñalvo, F. J. & García Holgado, A.* (s.f.). Obtenido de Apuntes de Ingeniería del Software I
- Moreno García, M. N.* (s.f.). Obtenido de Apuntes de la asignatura Ingeniería del Software II
- PHPMailer.* (s.f.). Obtenido de <https://github.com/PHPMailer/PHPMailer>
- Pressman, R. S.* (s.f.). Obtenido de Ingeniería del Software: Un Enfoque Práctico. 7a Edición. McGraw-Hill.
- REST.* (s.f.). Obtenido de https://en.wikipedia.org/wiki/Representational_state_transfer