



Universidad de Valladolid

# **ESCUELA DE INGENIERÍA INFORMÁTICA DE SEGOVIA**

**Grado en Ingeniería Informática  
de Servicios y Aplicaciones**

---

**Efficient Time-Management (ETM-App): Aplicación  
responsive para la gestión efectiva del tiempo**

---

**Julián Peromingo Jiménez**  
**Tutor: José Vicente Álvarez Bravo**



# Índice general

|   |    |
|---|----|
| Índice de figuras .....                     | 5  |
| Índice de cuadros .....                     | 7  |
| Agradecimientos .....                       | 10 |
| Resumen .....                               | 11 |
| Abstract.....                               | 13 |
| Parte I Introducción del proyecto .....     | 15 |
| Capítulo 1 Descripción del proyecto .....   | 18 |
| 1.1. Introducción .....                     | 18 |
| 1.2. Objetivos del trabajo .....            | 19 |
| 1.3. Entorno de la aplicación .....         | 19 |
| 1.3.1. TomatoTimers.....                    | 20 |
| 1.3.2. Forest .....                         | 21 |
| 1.3.3. Comparativa .....                    | 22 |
| 1.4. Entorno y tecnologías utilizadas ..... | 24 |
| 1.4.1. PHP .....                            | 24 |
| 1.4.2. MySQL .....                          | 25 |
| 1.4.3. JavaScript .....                     | 26 |
| 1.4.4. Gulp .....                           | 26 |
| 1.4.5. Composer.....                        | 28 |
| 1.4.6. HTML.....                            | 28 |
| 1.4.7. Apache .....                         | 29 |
| 1.4.8. SAAS .....                           | 30 |
| 1.5. Estructura de la memoria .....         | 31 |
| Capítulo 2 Planificación .....              | 33 |
| 2.1. Metodología de trabajo .....           | 33 |
| 2.2. Planificación del proyecto .....       | 34 |
| 2.2.1. Planificación inicial .....          | 34 |
| 2.2.2. Análisis de riesgos .....            | 36 |
| 2.2.3. Planificación Final.....             | 39 |
| 2.2.4. Presupuesto económico: .....         | 40 |
| Parte II Documentación técnica.....         | 43 |
| Capítulo 3 Análisis .....                   | 45 |
| 3.1. Requisitos.....                        | 45 |

|                |   |     |
|----------------|---|-----|
| 3.1.1.         | Requisitos funcionales.....                   | 46  |
| 3.1.2.         | Requisitos no funcionales:.....               | 51  |
| 3.1.3.         | Requisitos de información.....                | 52  |
| 3.2.           | Diagrama de casos de uso.....                 | 53  |
| 3.3.           | Historias de usuario.....                     | 54  |
| Capítulo 4     | Diseño.....                                   | 58  |
| 4.1.           | Arquitectura Lógica.....                      | 58  |
| 4.2.           | Arquitectura Física.....                      | 61  |
| 4.3.           | Diagramas de secuencia.....                   | 62  |
| 4.3.1.         | Diagrama de secuencia crear actividad.....    | 62  |
| 4.3.2.         | Diagrama de secuencia almacenar registro..... | 63  |
| 4.3.3.         | Diagrama de secuencia invitar usuario.....    | 64  |
| 4.4.           | Modelado de los datos.....                    | 64  |
| 4.4.1.         | Modelo Entidad Relación.....                  | 65  |
| 4.4.2.         | Diccionario de datos.....                     | 66  |
| 4.5.           | Diseño de la interfaz de usuario.....         | 69  |
| 4.5.1.         | Disposición de los elementos.....             | 69  |
| 4.5.2.         | Esquemas de colores.....                      | 76  |
| Capítulo 5     | Implementación.....                           | 78  |
| 5.1.           | Estructura del proyecto.....                  | 78  |
| 5.2.           | Framework creado.....                         | 83  |
| 5.2.1.         | Patrones de diseño.....                       | 83  |
| 5.3.           | Funcionamiento del Framework.....             | 85  |
| 5.3.1.         | Funcionamiento en el desarrollo.....          | 85  |
| 5.3.2.         | Funcionamiento general.....                   | 86  |
| 5.4.           | Dificultades en el desarrollo.....            | 88  |
| Capítulo 6     | Pruebas.....                                  | 91  |
| 6.1.           | Pruebas de caja blanca.....                   | 91  |
| 6.2.           | Pruebas de caja negra.....                    | 92  |
| Capítulo 7     | Conclusiones.....                             | 98  |
| 7.1.           | Líneas de trabajo futuras.....                | 98  |
| Parte III      | Manuales de usuario.....                      | 101 |
| Capítulo 8     | Manual de Usuario.....                        | 103 |
| Webgrafía..... |   | 113 |

# Índice de figuras

|      |   |    |
|------|---|----|
| 1.1  | Página de tomatotimers .....  | 20 |
| 1.2  | Temporizador de tomatotimers .....  | 21 |
| 1.3  | Forest web .....  | 22 |
| 1.4  | PHP .....   | 24 |
| 1.5  | MySQL .....   | 25 |
| 1.6  | JavaScript .....  | 26 |
| 1.7  | Gulp .....  | 26 |
| 1.8  | Composer.....   | 28 |
| 1.9  | HTML.....   | 28 |
| 1.10 | Apache .....  | 29 |
| 1.11 | SAAS .....  | 30 |
| 2.1  | Gráfico modelo incremental .....  | 33 |
| 2.2  | Escala de probabilidad.....   | 37 |
| 2.3  | Escala de impacto .....   | 37 |
| 3.1  | Diagrama de casos de uso.....   | 53 |
| 4.1  | Arquitectura lógica de alto nivel.....                                      | 59 |
| 4.2  | Arquitectura lógica detallada .....   | 60 |
| 4.3  | Arquitectura física.....  | 61 |
| 4.4  | Diagrama de secuencia "crear actividad" .....                               | 62 |
| 4.5  | Diagrama de secuencia "almacenar registro" .....                            | 63 |
| 4.6  | Diagrama de secuencia "invitar usuario" .....                               | 64 |
| 4.7  | Modelo Entidad-Relación.....  | 65 |
| 4.8  | Diseño de interfaz vistas de Login .....                                    | 69 |
| 4.9  | Diseño de interfaz vistas principales .....                                 | 70 |
| 4.10 | Diseño de interfaz de una actividad.....                                    | 71 |
| 4.11 | Diseño de interfaz vista de invitaciones .....                              | 71 |
| 4.12 | Diseño de interfaz vista de los temporizadores de una tarea .....           | 72 |
| 4.13 | Diseño de interfaz vista de estadísticas de una tarea .....                 | 73 |
| 4.14 | Diseño de interfaz vista del histórico de una actividad .....               | 74 |
| 4.15 | Diseño de interfaz barra lateral.....                                       | 74 |
| 4.16 | Diseño interfaz vista móvil de la vista de temporizadores de una tarea..... | 75 |
| 4.17 | Color de eliminación de elementos .....                                     | 76 |
| 4.18 | Color de creación de elementos.....   | 76 |
| 4.19 | Muestra de colores utilizados en los tiempos .....                          | 76 |
| 5.1  | Carpetas .....  | 78 |
| 5.2  | Classes .....   | 79 |
| 5.3  | Controladores .....   | 79 |
| 5.4  | Carpeta includes .....  | 80 |
| 5.5  | Modelos .....   | 80 |
| 5.6  | Módulos de Node .....   | 80 |
| 5.7  | Carpeta public.....   | 81 |
| 5.8  | Carpeta src .....   | 81 |
| 5.9  | Vendor composer .....   | 81 |
| 5.10 | Vistas .....  | 82 |

|  |     |
|--|-----|
| 5.11 Archivos Composer .....   | 82  |
| 5.12 Archivos Gulp.....  | 82  |
| 5.13 Endpoints de la aplicación.....   | 87  |
| 8.1 Formulario de Login de Efficient .....                                     | 103 |
| 8.2 Formulario de creación de cuenta Efficient.....                            | 103 |
| 8.3 Email de confirmación de cuenta de Efficient.....                          | 104 |
| 8.4 Vista de confirmación de cuenta Efficient.....                             | 104 |
| 8.5 Vista principal de Efficient .....   | 105 |
| 8.6 Formulario de creación de una nueva actividad .....                        | 105 |
| 8.7 Actividad de Efficient .....   | 105 |
| 8.8 Actividad de Efficient con botón de los temporizadores resaltado.....      | 106 |
| 8.9 Temporizadores de una tarea .....  | 106 |
| 8.10 Muestra de un descanso en el apartado de temporizadores .....             | 107 |
| 8.11 Actividad de Efficient con botón de estadísticas resaltado .....          | 107 |
| 8.12 Vista de estadísticas de una tarea.....                                   | 108 |
| 8.13 Muestra de porcentajes del diagrama .....                                 | 108 |
| 8.14 Formulario para añadir más tiempo a una tarea .....                       | 109 |
| 8.15 Actividad con botón para mostrar el histórico resaltado .....             | 109 |
| 8.16 Histórico de una actividad .....  | 109 |
| 8.17 Barra lateral contraída .....   | 110 |
| 8.18 Barra lateral expandida.....  | 110 |
| 8.19 Botón cierre de sesión.....   | 110 |
| 8.20 Actividad grupal con botón para compartir actividad grupal resaltado..... | 111 |
| 8.21 Ejemplo de invitación.....  | 111 |
| 8.22Ejemplo de histórico actividad grupal .....                                | 112 |
| 8.23 Muestra vista estadísticas tarea grupal .....                             | 112 |

# Índice de cuadros

|   |    |
|---|----|
| Tabla 1 Resumen comparativa entre aplicaciones .....  | 23 |
| Tabla 2 Cálculos de probabilidad e impacto .....  | 37 |
| Tabla 3 Exposición de riesgos y planes de contingencia .....                                | 38 |
| Tabla 4 Presupuesto total .....   | 41 |
| Tabla 5 Requisitos funcionales sistema de Login .....                                       | 46 |
| Tabla 6 Requisitos funcionales vista principal .....  | 46 |
| Tabla 7 Requisitos funcionales de la vista de estadísticas de una tarea .....               | 47 |
| Tabla 8 Requisitos funcionales del sistema de temporizadores .....                          | 47 |
| Tabla 9 Requisitos funcionales histórico de una actividad .....                             | 48 |
| Tabla 10 Requisitos funcionales de la vista principal de las actividades grupales .....     | 48 |
| Tabla 11 Requisitos funcionales de la vista de estadísticas de una actividad grupal .....   | 49 |
| Tabla 12 Requisitos funcionales de la vista de temporizadores de una actividad grupal ..... | 49 |
| Tabla 13 Requisitos funcionales de la barra lateral de navegación .....                     | 50 |
| Tabla 14 Requisitos funcionales del histórico de una actividad grupal .....                 | 50 |
| Tabla 15 Requisitos funcionales del sistema de invitaciones .....                           | 51 |
| Tabla 16 Requisitos no funcionales de la aplicación .....                                   | 51 |
| Tabla 17 Requisitos de información de la aplicación .....                                   | 52 |
| Tabla 18 Historia de usuario del caso de uso "iniciar sesión" .....                         | 54 |
| Tabla 19 Historia de usuario del caso de uso "Crear cuenta" .....                           | 54 |
| Tabla 20 Historia de usuario del caso de uso "Recuperar contraseña" .....                   | 54 |
| Tabla 21 Historia de usuario del caso de uso "Gestionar actividades" .....                  | 54 |
| Tabla 22 Historia de usuario del caso de uso "Gestionar tareas" .....                       | 55 |
| Tabla 23 Historia de usuario del caso de uso "Visualizar histórico actividad" .....         | 55 |
| Tabla 24 Historia de usuario del caso de uso "Exportar histórico actividad" .....           | 55 |
| Tabla 25 Historia de usuario del caso de uso "Añadir tiempo tareas" .....                   | 55 |
| Tabla 26 Historia de usuario del caso de uso "Visualizar tiempo tareas" .....               | 56 |
| Tabla 27 Historia de usuario del caso de uso "Enviar invitaciones" .....                    | 56 |
| Tabla 28 Historia de usuario del caso de uso "Enviar invitaciones" .....                    | 56 |
| Tabla 29 Historia de usuario del caso de uso "Aceptar invitaciones" .....                   | 56 |
| Tabla 30 Diccionario de datos de la entidad "actividades" .....                             | 66 |
| Tabla 31 Diccionario de datos de la entidad "histórico" .....                               | 66 |
| Tabla 32 Diccionario de datos de la entidad "invitar" .....                                 | 67 |
| Tabla 33 Diccionario de datos de la entidad "participar" .....                              | 67 |
| Tabla 34 Diccionario de datos de la entidad "tareas" .....                                  | 68 |
| Tabla 35 Diccionario de datos de la entidad "invitar" .....                                 | 68 |
| Tabla 36 Prueba de caja negra "registro de un usuario" .....                                | 92 |
| Tabla 37 Prueba de caja negra "Iniciar sesión" .....  | 93 |
| Tabla 38 Prueba de caja negra "Recuperar contraseña" .....                                  | 93 |
| Tabla 39 Prueba de caja negra "Crear actividad" .....                                       | 93 |
| Tabla 40 Prueba de caja negra "Eliminar actividad" .....                                    | 94 |
| Tabla 41 Prueba de caja negra "Crear tarea" .....   | 94 |
| Tabla 42 Prueba de caja negra "Eliminación de tarea" .....                                  | 94 |
| Tabla 43 Prueba de caja negra "Añadir tiempo a una tarea" .....                             | 95 |
| Tabla 44 Prueba de caja negra "visualizar grafico tarea" .....                              | 95 |
| Tabla 45 Prueba de caja negra "visualizar histórico actividad" .....                        | 95 |
| Tabla 46 Prueba de caja negra "exportar histórico a PDF" .....                              | 96 |

|  |    |
|--|----|
| Tabla 47 Prueba de caja negra "invitar usuario" .....    | 96 |
| Tabla 48 Prueba de caja negra "aceptar invitación" ..... | 96 |





# Agradecimientos

Quiero agradecer a todas las personas que siempre han confiado en mí y han estado para apoyarme tanto en los momentos buenos como malos.

Gracias a mi familia, amigos y seres queridos por acompañarme en este camino y haber ayudado a conseguir la mejor versión de mí mismo.

# Resumen

Cada día es más frecuente ver cómo las nuevas tecnologías, lejos de ayudar, suponen un impedimento a la hora de concentrarse para realizar distintas tareas, lo que provoca que la gente no sea consciente del tiempo que invierte en las mismas y que su eficiencia sea mucho menor.

Este Trabajo de Fin de Grado (TFG) tiene como objetivo desarrollar una aplicación web que fomente de manera sencilla la concentración y consecución de las tareas que se proponga el usuario. Para lograrlo, se facilita al usuario información del tiempo invertido en la actividad y tiempo de descanso a través de registros que le permitan monitorizar su progreso y mejora. De forma adicional, a través de esta aplicación, podrá apoyarse y colaborar con otros usuarios para obtener una motivación extra y facilitar el trabajo en equipo.

**Palabras claves:** Actividad, tarea, tiempo, gestión del tiempo, eficiencia.



# Abstract

In the modern world, it is becoming more and more common to see how new technologies, rather than helping, are an impediment when it comes to focusing on different tasks. This lack of concentration means that people are not aware of the time they invest in these technologies and, subsequently, their own lack of efficiency in day-to-day life.

This TFG has as main goal creating a web application that helps the user to easily concentrate on, and achieve the different tasks they propose. To accomplish this, the application makes the user aware of how much time they invest on a task, and how much time they rest. To achieve this, the application provides records that allow them to see whether they are improving or not. Moreover, if necessary, the platform provides the opportunity to rely on other users for collaboration and extra motivation.

**Key words:** Activity, task, time, time management, efficiency.



# Parte I

## Introducción del proyecto







# Capítulo 1

## Descripción del proyecto

### 1.1. Introducción

La pérdida de tiempo es algo que está a la orden del día. Para un estudiante resulta difícil determinar cuánto tiempo ha invertido en prepararse para un examen, pero aún más realizar predicciones sobre cuánto tiempo le conllevará preparárselos.

Las distracciones continuas como pueden ser el teléfono móvil o las redes sociales tienen un impacto directo en la gestión del tiempo. Acciones como contestar un WhatsApp se acaban convirtiendo en media hora de uso del teléfono y la consiguiente desmotivación y agobio por la propia pérdida de tiempo.

A lo largo de la vida de un estudiante este es un problema recurrente, y siempre se escuchan frases típicas como: “le dediqué dos horas, pero si me hubiese puesto en serio habría sido media hora”.

Y de este tipo de respuestas se acaban obteniendo dos conclusiones:

1. No se puede estimar el tiempo que conlleva realmente realizar las actividades.
2. Es muy difícil ser eficiente.

En este contexto surge EFFICIENT, una aplicación web que permitirá obtener tiempos realistas de las actividades que se realizan (siendo un aliciente para que se disminuya el tiempo “perdido”) como obtener la capacidad de realizar predicciones realistas sobre los tiempos invertidos.

Para lograr estos objetivos, mientras se realizan las actividades la aplicación permitirá observar el tiempo que se está invirtiendo en ellas. Además, se podrán visualizar históricos de las actividades que contienen la proyección y avance total de las mismas.

Como añadido, la aplicación contará con una gran versatilidad, ya que se podrá utilizar para cualquier actividad, como puede ser saber el número de horas que conlleva la realización de un TFG o incluso para medir los descansos y la progresión de una persona cuando sale a correr.

Y para un mayor aliciente a la hora de lograr objetivos, la aplicación podrá utilizarse con compañeros, tanto para ver el tiempo que invierte cada uno en las actividades grupales, como para ayudar de motivación extra ya que podrán visualizar el tiempo que tú inviertes y pierdes y viceversa.

## 1.2. Objetivos del trabajo

Con el desarrollo de este trabajo se busca la implementación de una aplicación que monitorice y controle el tiempo invertido, tiempo efectivo y los descansos en las distintas actividades elegidas por el usuario, mostrándolo de una manera visual fácil y sencilla para el usuario.

Para ello, los objetivos que se buscan lograr en la aplicación son:

- Creación de un sistema de registro de usuarios el cual sea capaz de registrar, enviar correos para confirmar cuentas y poder recuperar contraseñas.
- Desarrollo de las funcionalidades necesarias para implementar la monitorización de tiempos.
- Adaptabilidad tanto a dispositivos móviles como ordenadores.
- Interfaz sencilla y cuidada.

## 1.3. Entorno de la aplicación

Realizando una búsqueda por internet podemos observar cómo hay páginas y aplicaciones con funcionalidades para la gestión de tareas o para la gestión del tiempo, pero hay muy pocos híbridos que combinen ambas características de manera integral.

Otro problema que tienen es que suelen caer en extremos opuestos, o son demasiado sencillas o demasiado complejas estando poco adaptadas a las necesidades de la mayoría de usuarios y requiriendo un alto grado de conocimiento técnico para sacarles el máximo provecho. Lo que lleva a los usuarios más casuales a alejarse de este tipo de aplicaciones.

Muchas de estas páginas tienen en el enfoque en la gestión de proyectos como es el caso de aplicaciones como “Trello” o “Jira” o por el contrario nos encontramos con simples cronómetros y temporizadores que no aportan las funcionalidades necesarias.

A pesar de la dificultad para encontrar aplicaciones las descritas a continuación son las más parecidas con las que realizar comparaciones.

### 1.3.1. TomatoTimers

TomatoTimers podría ser una aplicación parecida a comparar con este TFG, pero, aun así, tiene una gran serie de diferencias que comentaré a continuación.

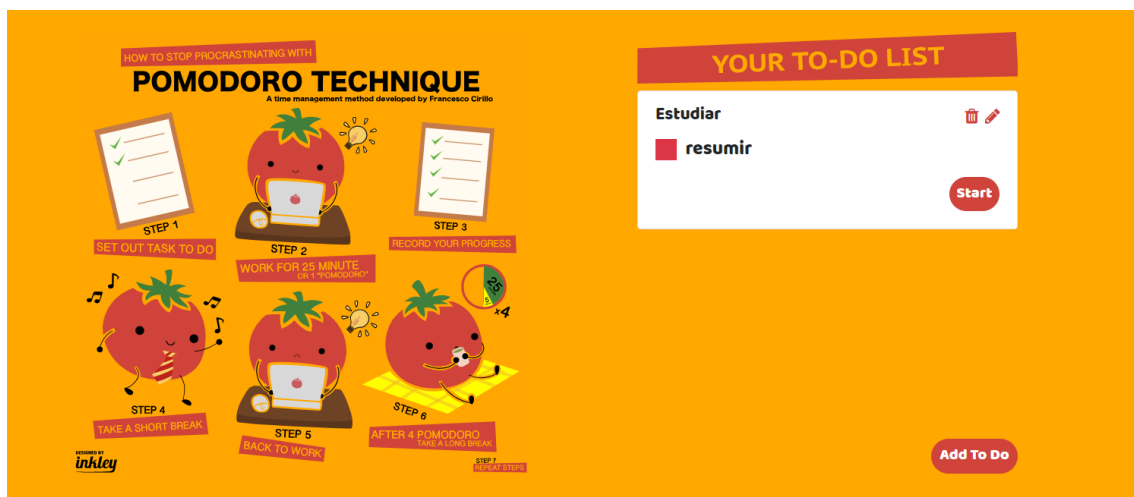
La página web se basa en el uso de la técnica de estudio conocida como Pomodoro, la cual se basa en la utilización de un temporizador para dividir el tiempo en intervalos de 25 minutos llamados “pomodoros” seguidos con pausas más largas cada 4 pomodoros o intervalos.

El funcionamiento es el siguiente:

1. Se decide la actividad a realizar.
2. Se inicia el temporizador.
3. Se trabaja en la tarea de manera intensiva hasta que el temporizador suene.
4. Se anota qué el pomodoro se ha completado.
5. Se toma una pausa breve.
6. Cada cuatro pomodoros, se toma una pausa más larga.

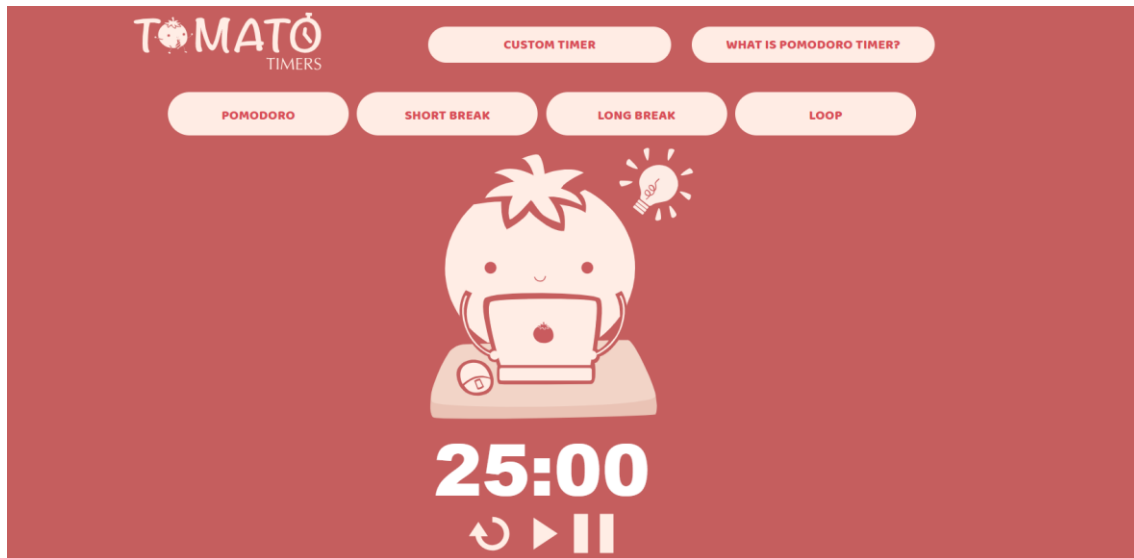
Em la página “tomatotimers” vemos como se lleva a la práctica esta teoría.

El usuario tiene la capacidad de crear actividades y de crear sub-actividades las cuales puede marcar como completadas cuando las realice:



0.1 Página de tomatotimers

Para contabilizar los pomodoros en la parte superior cuenta con un temporizador que aplica la técnica Pomodoro de manera automática.



0.2 Temporizador de tomatotimers

Además, cabe la posibilidad por parte del usuario de iniciar un descanso corto, uno largo, personalizar el tiempo del intervalo o ponerlo en bucle.

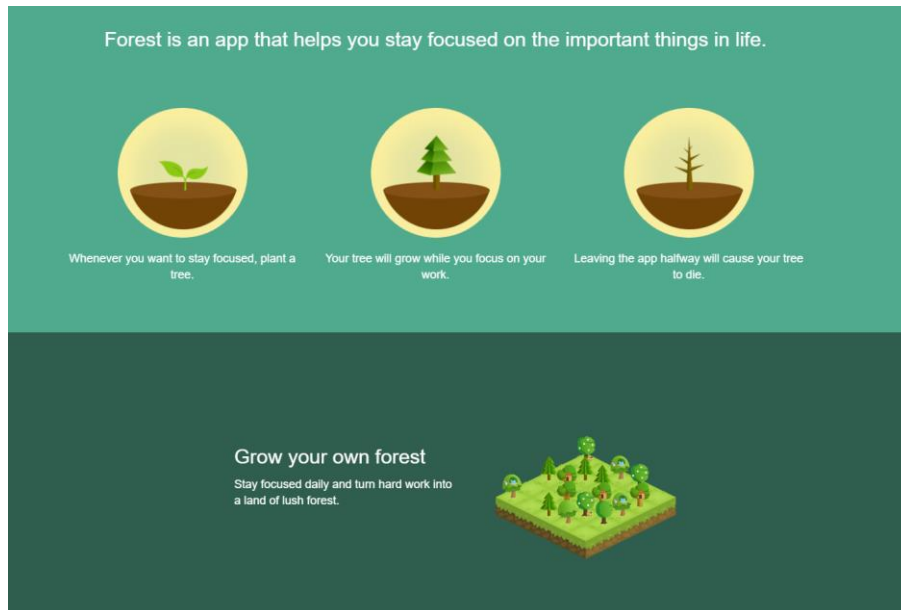
Una vez el usuario complete la tarea puede marcarla como hecha y eliminarla.

### 1.3.2. Forest

Forest es una famosa aplicación móvil que tiene un enfoque similar pero un tanto distinto.

La aplicación también tiene el objetivo de alejar a la gente de las distracciones como es el uso del teléfono. Para ello cada vez que quieras puedes iniciar un temporizador en la app del tiempo que crea conveniente el usuario. Cuando se inicia este temporizador la aplicación “planta un árbol” de manera virtual y si el usuario abandona la aplicación mientras el temporizador está activo el árbol morirá. En cambio, si completa el tiempo propuesto inicialmente, dentro del bosque virtual se plantará un árbol sano.

Cada uno de estos árboles realmente es un registro del tiempo que ha estado el usuario concentrado y de si cumplió su objetivo o no.



0.3 Forest web

Otra funcionalidad muy interesante de la aplicación, mientras Forest tiene un temporizador activo, el usuario puede configurar el bloqueo de otras aplicaciones facilitando que los usuarios no hagan uso del móvil.

Además, cada vez que completas un tiempo propuesto, la aplicación te da monedas en función del tiempo invertido, permitiendo al usuario comprar distintos tipos de árboles para incluir en su bosque virtual.

### 1.3.3. Comparativa

Una vez analizadas las distintas webs y aplicaciones existentes en el mercado y tras seleccionar las más parecidas a Efficient, vamos a realizar un análisis viendo las ventajas y desventajas de las distintas alternativas frente a Efficient.

#### **Ventajas de Efficient:**

Tras analizar TomatoTimers y Forest, en TomatoTimers puedes crear actividades y subactividades, pero en Forest ya no.

Uno de los factores diferenciales a favor de Efficient es que a diferencia de TomatoTimers el usuario puede medir el tiempo invertido en cada una de las subactividades, así como el tiempo invertido en la actividad en general clarificando que subactividades han sido las de mayor tiempo invertido.

Otra gran ventaja frente a Forest (la cual es la única de las aplicaciones que almacena un registro), es que Forest da por hecho que todo el tiempo invertido por el usuario ha sido tiempo efectivo, cuando realmente el usuario a pesar de que el tiempo esté corriendo puede estar teniendo cualquier tipo de distracción que no se contabilizará.

En Efficient el usuario tiene la posibilidad de desgranar el tiempo invertido entre tiempo efectivo y tiempo de descanso, así como ver el tiempo total invertido. Convirtiéndose en la única de las aplicaciones que proporciona tiempos más realistas.

Otra ventaja que ninguna de estas aplicaciones ofrece, es la posibilidad de compartir actividades con otros usuarios de la aplicación. Con Efficient tienes la posibilidad de crear actividades con otros usuarios como podría ser la creación de un proyecto y ver quien ha invertido más tiempo en total, cuánto se ha invertido en total en el proyecto o quién ha aprovechado mejor el tiempo.

A su vez Efficient permite añadir tiempo. Esta funcionalidad es muy importante ya que habrá momentos, como cuando un usuario está en clase, que no podrá estar haciendo uso de la aplicación. En Forest si el usuario quisiera contabilizar ese tiempo tendría que iniciar un temporizador y esperar a que todo el tiempo pase.

En cambio, si un usuario de Efficient quisiera contabilizar el tiempo que le dedica a una asignatura y añadir el tiempo que le ha dedicado estando en clase, podría hacerlo pulsando únicamente un botón.

Además, en Efficient se pueden almacenar los distintos registros en PDF, así como un gran número de funcionalidades secundarias.

**Desventajas de Efficient:**

En ambas aplicaciones de la competencia el usuario completa objetivos, situación que actualmente no es posible en Efficient, pero esta funcionalidad está pensada como una posible mejora o expansión.

Otra diferencia atractiva de Forest es la capacidad de bloquear otras aplicaciones del teléfono. Efficient no podría hacerlo debido a que es una aplicación web, pero si se podría plantear el caso para que si el usuario abandona la página se pierda el progreso.

**Resumen comparativo:**

| Funcionalidad                           | Efficient | Tomatotimers | Forest |
|---|-----------|--------------|--------|
| Creación de tareas                      | Sí        | Sí           | No     |
| Medir tiempo invertido en las tareas    | Sí        | No           | No     |
| Almacenar registros de tiempo invertido | Sí        | No           | Sí     |
| Multiusuario                            | Sí        | No           | No     |
| Completar objetivos                     | No        | Sí           | Sí     |
| Bloquear otras aplicaciones             | No        | No           | Sí     |

Tabla 1 Resumen comparativa entre aplicaciones

## 1.4. Entorno y tecnologías utilizadas

En esta sección se expondrán las tecnologías que se han utilizado para el desarrollo de este trabajo.

### 1.4.1. PHP



0.4 PHP

PHP es un lenguaje de programación interpretado del lado del servidor, es de propósito general lo que es muy conveniente para el desarrollo de aplicaciones web.

El código PHP generalmente se maneja en un servidor web utilizando un intérprete de PHP implementado como un módulo, demonio o ejecutable de interfaz de puerta de enlace común (CGI).

PHP tiene la capacidad para interpretar cualquier tipo de datos, desde imágenes binarias a HTML, así como de generarlos y enviarlos en peticiones HTTP.

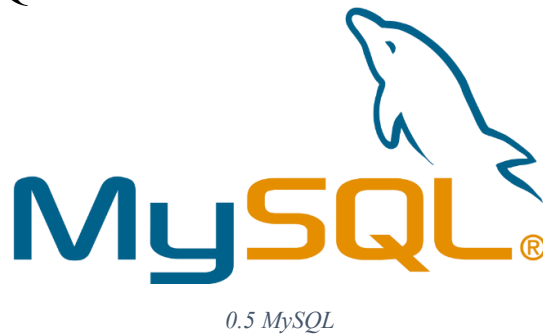
Al ser de código abierto puede implementarse en cualquier servidor web, lo que le ha llevado a que esté instalado en más de 20 millones de sitios web y más de un millón de servidores.

Algunas de las características principales de PHP son:

- Es un lenguaje orientado a la creación de sitios web dinámicos con acceso a bases de datos.
- Sigue el modelo (*single-request-per-script-execution*): cada petición activa un script PHP y el resultado se devuelve en la salida. Por lo que resulta muy difícil que se quede paralizado con una petición inadecuada.
- Al ser el servidor el que se encarga de ejecutar el código y enviar su resultado el código PHP resulta invisible para el navegador.



## 1.4.2. MySQL



MySQL un sistema de gestión de bases de datos relacional, en la actualidad es el más extendido ya que es de código abierto, aunque cuenta con una versión comercial gestionada por Oracle.

Algunas de las características principales de MySQL son:

- Arquitectura Cliente servidor: el funcionamiento de MySQL se basa en esta arquitectura, donde cada cliente puede realizar consultas para la obtención de datos, modificarlos o guardar nuevos.
- Como SQL es uno de los lenguajes más populares de la industria es común que otros motores de bases de datos como PostgreSQL utilicen SQL, por lo tanto, la adaptación de otro motor de búsqueda a este es sencilla.
- Ofrece la posibilidad de crear vistas personalizadas para distintos tipos de usuarios.
- En lugar de procesar directamente las tablas, se pueden crear y ejecutar procedimientos almacenados, lo que aumenta significativamente la eficacia de una implementación.
- MySQL permite la utilización de Triggers que son acciones que se ejecutan automáticamente dentro de nuestra base cuando sucede algún evento.
- Uso de transacciones: lo que aumenta la seguridad frente errores, así como evitar problemas de concurrencia e integridad.

### 1.4.3. JavaScript



0.6 JavaScript

JavaScript es un lenguaje de programación cuya función principal es la de dotar de interactividad a las páginas web y hacerlas más dinámicas.

Es un lenguaje de alto nivel interpretado con un estilo orientado a objetos y basado en prototipos.

Junto a HTML y CSS forma la mayoría de sitios web. Una de sus ventajas es la compatibilidad que tiene con todos los navegadores modernos por lo que puede adaptarse de una manera sencilla a la mayoría de dispositivos y plataformas.

Además, destaca por su capacidad para modificar el Document Object Model (DOM), lo que permite en tiempo de ejecución cambiar estilos y contenidos de la página en función a eventos producidos por el usuario.

### 1.4.4. Gulp



0.7 Gulp

Gulp es una herramienta que forma parte del proceso de desarrollo de software y está específicamente relacionada con la interfaz de usuario. Su principal objetivo es automatizar flujos de trabajo lentos y repetitivos.

Más concretamente es un “Task Runner”, es decir, una herramienta para automatizar tareas.

Algunos de sus usos son:

- Concatenación y minificar archivos;
- Archivos de estilo de preprocesamiento;
- Pruebas unitarias;
- Optimización de imagen.
- Generar el CSS.
- Empaquetar todo, ya sea JavaScript, CSS, HTML y con la estructura de directorios que queremos para ejecutar el script de la página web con el servidor.

Para poder utilizar Gulp necesitamos instalar Node.JS que es un intérprete de JavaScript.

También necesitaremos Node Pacakage Manager (NPM) que permitirá instalar la interfaz de la línea de comandos de Gulp (Gulp-cli).

### GulpFile.JS:

En este archivo estarán las tareas que quieres que Gulp automaticice.

Funciones principales:

- `gulp.task(name, fn)`: registra la función con el nombre, y también le permitirá especificar una dependencia si las tareas deben ejecutarse primero.
- `Gulp.run(tasks...)`: ejecuta todas las tareas con la máxima concurrencia;
- `Gulp.watch(glob, fn)`: busca los archivos que especificó y, cuando se realizan cambios, reinicia la función.
- `Gulp.src(glob)`: devuelve un flujo legible. Como si él fuera el encargado de descargar los archivos.
- `Gulp.dest (directorio)`: devuelve una secuencia grabable o una secuencia liberada para escritura. Es decir, esta es la carpeta de destino donde se guardarán los archivos.

## 1.4.5. Composer



0.8 Composer

Composer es un administrador de dependencias PHP, es utilizado para gestionar las bibliotecas y dependencias de manera automática descargándolas y actualizándolas de manera automática.

Algunas de las funciones principales de Composer son:

- La gestión de versiones: comprueba las versiones existentes y permite instalar las versiones de las dependencias adecuadas.
- Automatización de tareas: se puede definir la ejecución de tareas al instalar o actualizar dependencias para crear documentos o eliminar archivos.
- La generación de archivos de carga automática: se genera un archivo “vendor/autoload.php” desde el cual se gestionan las dependencias. Lo que permite gestionar todas las dependencias de una manera centralizada.

## 1.4.6. HTML



0.9 HTML

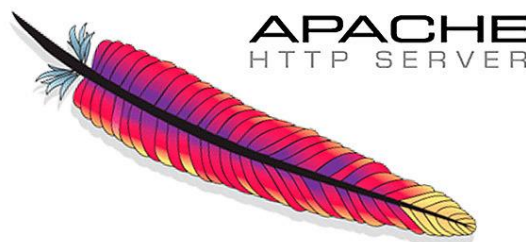
HTML o HyperText Markup Lenguaje es el lenguaje de marcado para la creación de páginas web.

Se ha convertido en un estándar de referencia para la elaboración de páginas web de cualquier tipo definiendo una estructura básica con un código denominado HTML, con el cual se definen contenido de cualquier tipo como texto imágenes o videos entre otros.

Es considerado uno de los lenguajes web más importantes, ya que ha sido uno de los pilares en la creación de internet como lo conocemos hoy en día. Convirtiéndose en el estándar para los navegadores web y la visualización de páginas.

Aunque cabe destacar que HTML no es un lenguaje de programación porque no cuenta con funciones aritméticas, estructuras de control o variables como el resto de lenguajes de programación. Lo que provoca que HTML únicamente sirva para la creación de páginas web estáticas. Pero combinándose con otros lenguajes de programación como PHP o JavaScript se pueden crear páginas web dinámicas como las de hoy en día.

## 1.4.7. Apache



*0.10 Apache*

Apache es un servidor web de código abierto utilizado aproximadamente en el 40% de sitios web del mundo.

Proporciona la capacidad de servir contenidos a través de las páginas web encargándose de recibir las peticiones de los clientes y entregarle los recursos solicitados como HTML, imágenes...

Algunas de las ventajas principales de apache son:

- Es de código abierto y económico.
- Es flexible ya que sigue una estructura basada en módulos.
- Es multiplataforma, se puede implementar tanto para Windows como para Linux.
- Es muy estable con una gran trayectoria.
- Tiene una gran comunidad que se encarga de actualizarlo.

## 1.4.8. SAAS



### *0.11 SAAS*

SAAS es un preprocesador de CSS, por lo que se utiliza para traducir código de hojas de estilo que no siguen los estándares a código CSS que es legible por la gran mayoría de navegadores.

Una de las principales razones para utilizar SAAS es que simplifica la creación de CSS y otras utilidades como las que se nombran a continuación.

Algunas de las ventajas son:

- Permite establecer variables, con lo que se podría reutilizar la información asociada a la variable en cualquier lugar y momento.
- Tiene una estructura más jerarquizada con un anidamiento más simple.
- Permite utilizar “funciones” conocidas como mixins, lo que favorece en gran medida la reutilización del código y la simplificación.
- Soporta la herencia
- Permite que se pueda dividir el código en distintos ficheros haciéndolo más legible.

## 1.5. Estructura de la memoria

La documentación se encuentra dividida en tres partes principales, estas partes contienen capítulos y cada capítulo se divide en múltiples secciones.

- **Parte I:** Introducción. Esta parte contiene los siguientes capítulos:
  - **Introducción:** en este capítulo se presenta el TFG, los objetivos que se quieren conseguir con el mismo y el estado del arte, comparándolo con aplicaciones similares existentes en el mercado
  - **Metodología:** en este capítulo se menciona como se ha realizado el desarrollo de la aplicación.
  - **Planificación:** Esta sección es la encargada de mostrar tanto la planificación previa al proyecto como la real. Además, en esta sección es donde se muestran el análisis de riesgos que pueden surgir durante la realización del mismo como el presupuesto económico
- **Parte II:** Documentación técnica. Esta parte contiene los siguientes capítulos:
  - **Análisis:** La primera fase a realizar en un proyecto software es la de análisis por eso se encuentra en primer lugar. En ella se analizan los distintos casos de uso de la aplicación, así como los requisitos.
  - **Diseño:** En esta fase se definen los componentes del sistema y como interactúan entre ellos. Para ello esta sección se especifican las arquitecturas lógicas y físicas, diagramas de secuencia de la aplicación y el modelado de los datos.
  - **Implementación:** Este capítulo está dedicado a analizar cómo se ha implementado la aplicación, para ello se define la estructura del proyecto y los patrones utilizados junto al Framework creado.
- **Parte III:** Manuales de usuario.
  - **Manual de usuario:** En esta sección se encuentre el manual de usuario explicando el funcionamiento y las distintas funcionalidades de la aplicación





# Capítulo 2

## Planificación

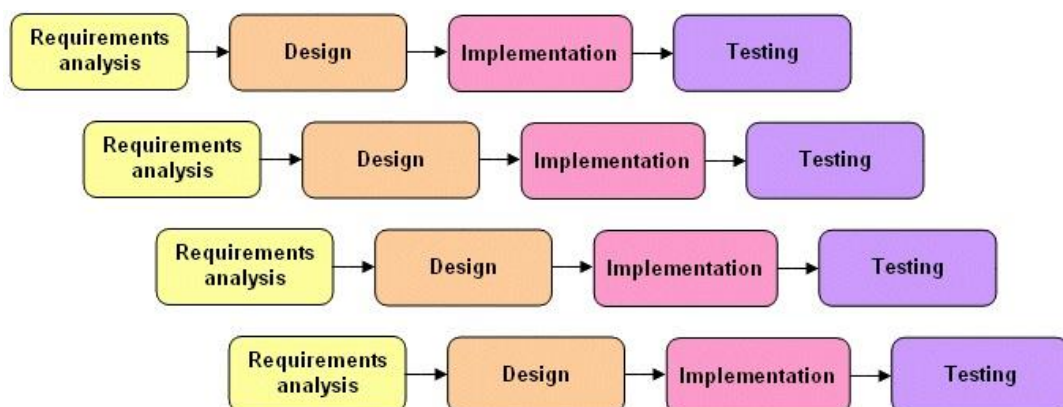
### 2.1. Metodología de trabajo

Tras comenzar el planteamiento del proyecto debido a la naturaleza del proyecto y las características del mismo se llegó a la conclusión que la mejor alternativa disponible iba a ser aplicar una metodología incremental para el proceso de desarrollo.

#### ¿Qué es la metodología incremental?

“El modelo de desarrollo incremental es el ciclo de vida de desarrollo software en el cual un proyecto es descompuesto en una serie de incrementos, cada uno de los cuales suministra una porción de la funcionalidad respecto de la totalidad de los requisitos del proyecto. Los requisitos tienen asignada una prioridad y son entregados según el orden de prioridad en el incremento correspondiente. En algunas (pero no en todas) versiones de este modelo de ciclo de vida, cada subproyecto sigue un “mini-modelo V” con sus propias fases de diseño, codificación y pruebas.”

*(Modelo de desarrollo incremental - Globe Testing, 2013)*



0.1 Gráfico modelo incremental

### Justificación de su uso

Gracias a esta metodología podría dividir fácilmente el proyecto en diversas fases, lo que sería una gran ventaja debido a los cambios constantes (tiempos disponibles, horarios...) y la posibilidad de adaptar las mismas en función de la fase del proyecto.

Otra razón de peso por la que se eligió esta metodología es el carácter del propio proyecto, ya que desde un planteamiento inicial se decidió crear y añadir funcionalidades al proyecto, realizar pruebas y análisis de las mismas y en función de lo realizado y su evaluación posterior implementar unas nuevas características u otras.

También debido a la gran flexibilidad que aporta esta metodología podría recibir una retroalimentación continua pudiendo incluir cambios y mejoras de una manera más sencilla antes de comenzar con un nuevo desarrollo y que fuese más difícil a posteriori realizar cambios.

Además, de esta manera se obtiene un mayor control de riesgos haciendo que estos se puedan identificar y prevenir de una manera mucho más sencilla en cada etapa del proyecto por lo que si detectaba problemas en un incremento en específico se podrían tomar medidas correctivas sin afectar a todo el proyecto.

## 2.2. Planificación del proyecto

En esta sección se comentará cual ha sido la planificación inicial para el desarrollo del proyecto y cuál ha sido la distribución del tiempo final.

### 2.2.1. Planificación inicial

Como se ha mencionado en la sección [2.1. Metodología de trabajo] se ha previsto utilizar una metodología incremental para llevar a cabo todos los requisitos del proyecto en el tiempo propuesto.

Antes de comenzar a añadir funcionalidades al proyecto se realizará una iteración cuyo único objetivo es el aprendizaje de las tecnologías que posteriormente se usarán en el proyecto como es el caso de JavaScript, Gulp o SCSS.

Para ello se realizarán varias prácticas de prueba que aparte de ayudar con el aprendizaje podrán implementar alguna de las funcionalidades futuras del proyecto lo que ahorrará tiempo en la fase de implementación del código.

La realización del proyecto se llevará a cabo entre febrero y junio de 2023 estimando un tiempo total de 5 meses.

En la primera iteración, el tiempo diario para poder realizar el proyecto es limitado, pero se prevé aproximadamente 50 horas, que se dilatarán bastante en el tiempo debido a los cambios de horarios.

## Distribución del tiempo

Según lo mencionado se estima que la primera iteración ocupará desde mediados de febrero hasta finales de abril. Pero existe la posibilidad de que finalice el 7 de mayo debido a circunstancias ajenas al proyecto, por lo que realizaré la estimación en base a esta posibilidad ya que sería el peor de los casos posibles.

En este caso quedarían 57 días para las siguientes fases del proyecto.

Una vez completada esa fase y tras cambiar la situación personal se podrá dedicar mucho más tiempo diario a la implementación del trabajo.

En caso de ser necesario se podría dedicar 8 horas diarias al trabajo, lo que supone un total de 456 horas para la implementación del código y realización de la memoria técnica.

La estimación media de lo que debe durar un TFG es de 300 horas, por lo que, invertidas unas 50 horas en la primera fase, quedarían de margen para solución de errores y posibles problemas unas 206 horas si fuese necesario.

En resumen, se podría seguir la siguiente distribución:

- **Primera iteración:** La duración estimada de esta fase es de **50 horas**. Los objetivos son el aprendizaje de las tecnologías que se implementarán en el proyecto que actualmente se desconocen, y concretar las funcionalidades y estructuras que se implementarán en el futuro.  
Para ello el aprendizaje se realizará realizando pequeños proyectos para la asimilación de conocimientos.
- **Segunda iteración:** La duración estimada de esta fase es de **140 horas**.  
El objetivo principal de esta fase es implementar las funcionalidades principales de la aplicación:
  - Creación de todo el sistema de Login y registro de usuarios.
  - Implementaciones necesarias para la creación y eliminación de actividades y tareas.
  - Creación de históricos, gráficos y temporizadores de la aplicación. Así como las funciones secundarias como añadir más tiempo, exportar el diagrama a PDF....
  - Cuidar el aspecto visual y realizar la adaptación para dispositivos móviles.

En esta fase se va a realizar todas las implementaciones que sean “monousuario” así como decidir la arquitectura que va a seguir el código del proyecto y la configuración de todo el entorno de trabajo.

- **Tercera iteración:** La duración estimada de esta fase es de **90 horas**. En esta fase se va a realizar la implementación de todas las funcionalidades que sean “multiusuario”, es decir:
  - Realización del sistema de invitaciones a actividades grupales: envío de invitaciones, recibir las invitaciones, aceptarlas, eliminarlas.
  - Creación de las funcionalidades para la creación y visualización de históricos, diagramas y actividades y tareas grupales.
  - Resto de funcionalidades asociadas como eliminación o creación de las actividades o tareas, añadir más tiempo...

- **Cuarta iteración:** La duración estimada de esta fase es de **80 horas**. El objetivo es finalizar la documentación técnica del proyecto completando los apartados que no se han podido realizar hasta que el código no se ha encontrado finalizado, y corregir aquellos pequeños errores y detalles que hayan podido surgir en el código.

Como se menciona, la última iteración tiene el enfoque en finalizar y realizar la documentación técnica, aunque no se haya comentado anteriormente se prevé que durante todas las fases se realice parte de la documentación, tiempo que ya está incluido en las horas previstas.

Según estas estimaciones la **duración total del proyecto supondría 360 horas**. Lo que equivaldría aproximadamente a **45 días**.

Como se mencionó anteriormente de las 456 horas totales para realizar las fases 2,3 y 4 y ocupando 310 para la realización de estas fases, quedarían 146 horas para posibles imprevistos. Lo que es equivalente aproximadamente a **18 días de margen**.

## 2.2.2. Análisis de riesgos

Dentro de la planificación inicial pueden surgir muchos problemas que conlleven retrasos en la ejecución del proyecto, incluso que conlleven que el proyecto no se pueda lograr en el plazo establecido.

Para ello es conveniente ver qué posibles riesgos pueden ocurrir, la probabilidad de ocurrencia que tienen, y cuál es el impacto que pueden ocasionar dentro del proyecto.

El costo del proyecto lo vamos a medir en tiempo, calculando el número de días de retraso que supondrían en caso de suceder.

Este análisis de riesgos se va a realizar para las iteraciones 2,3 y 4. Como se ha mencionado con anterioridad, la primera iteración debido a los cambios de horarios tiene un carácter demasiado variable. Al igual que como se ha realizado con la planificación, se va a realizar partiendo desde el peor de los casos posibles para la primera iteración, que salvo riesgos de carácter mayor como problemas de salud severos (los cuales no se van a contemplar) no puede ocurrir nada para que la planificación de la primera iteración se retrase más de lo ya previsto. Pero sí pueden ocurrir riesgos para las siguientes iteraciones.

En primer lugar, vamos a identificar los posibles riesgos que pueden surgir, cuál es su probabilidad de ocurrencia y cuál es el impacto en días que pueden causar en el proyecto.

Para realizar la evaluación del riesgo has tenido en cuenta las escalas de probabilidad e impacto siguientes:

| PROBABILIDAD |   |             |      |      |
|--------------|---|-------------|------|------|
|              |   |             | ≥    | <    |
| A            | 5 | Casi cierto | 0.80 | 1.00 |
| B            | 4 | Probable    | 0.50 | 0.80 |
| C            | 3 | Posible     | 0.20 | 0.50 |
| D            | 2 | Improbable  | 0.10 | 0.20 |
| E            | 1 | Raro        | 0.01 | 0.10 |

0.2 Escala de probabilidad

| IMPACTO |                |                    |
|---------|----------------|--------------------|
| 1       | Insignificante | valor < 1%         |
| 2       | Menor          | 1% ≤ valor < 5 %   |
| 3       | Moderado       | 5% ≤ valor < 10 %  |
| 4       | Mayor          | 10% ≤ valor < 20 % |
| 5       | Catastrófico   | 20 % ≤ valor       |

0.3 Escala de impacto

Según las escalas anteriores obtenemos la siguiente tabla:

| Riesgo | Probabilidad | Clasificación probabilidad | Costo en días | Impacto | Descripción  |
|--------|--------------|----------------------------|---------------|---------|--|
| 1      | 0.95         | 5                          | 3             | 3       | Cambio en las circunstancias personales que impidan priorizar el avance del proyecto (otros trabajos, estudios...)       |
| 2      | 0.05         | 1                          | 4             | 3       | Ponerse enfermo.   |
| 3      | 0.01         | 1                          | 10            | 5       | Perdida de datos por errores en el ordenador o copias de seguridad.  |
| 4      | 0.01         | 1                          | 5             | 4       | Cambios en las funcionalidades iniciales previstas.  |
| 5      | 0.20         | 3                          | 2             | 2       | Problemas surgidos de la propia implementación (Pensar que se podría realizar de una determinada manera, pero no es así) |
| 6      | 0.10         | 2                          | 2             | 2       | Problemas de seguridad que son necesarios corregir   |
| 7      | 0.05         | 1                          | 1             | 2       | Problemas de alojamiento web   |

Tabla 2 Cálculos de probabilidad e impacto

Una vez identificados y tras medir su probabilidad de ocurrencia e impacto, es conveniente medir la exposición del riesgo, la cual se obtiene de multiplicar la probabilidad de ocurrencia por el impacto que tiene el riesgo. Además, para riesgo se va a diseñar un plan de reducción (plan establecido para reducir la posibilidad de que surja el riesgo) y un plan de mitigación (plan establecido para una vez surgido el riesgo reducir el impacto que tiene en el proyecto) como se muestra en la tabla a continuación.

| <b>Riesgo</b> | <b>Exposición</b> | <b>Plan de reducción</b>   | <b>Plan de mitigación</b>   |
|---------------|-------------------|--|---|
| 1             | 15                | Intentar realizar una buena planificación para disminuir contratiempos               | Aprovechar al máximo el tiempo para realizar el máximo trabajo con el tiempo disponible             |
| 2             | 3                 | Seguir buenos hábitos alimenticios, abrigarse...                                     | Descansar y tomar las medicinas correspondientes  |
| 3             | 5                 | Realizar copias continas con programas como Github                                   | Recuperar la última copia y recuperar el trabajo desde ese punto.                                   |
| 4             | 4                 | Realizar una planificación antes de iniciar las fases de implementación del proyecto | Buscar la mejor manera de adaptar los cambios al proyecto actual evaluando el impacto de los mismos |
| 5             | 6                 | Intentar documentarse bien antes de realizar las implementaciones.                   | Buscar en las documentaciones para ver la forma correcta de realizarlo                              |
| 6             | 4                 | Documentarse de las medidas que hay que tomar antes de realizar la implementación    | Documentarse y corregir errores   |
| 7             | 2                 | Asegurarse de realizar las configuraciones de una manera correcta                    | Realizar los cambios pertinentes  |

*Tabla 3 Exposición de riesgos y planes de contingencia*

### 2.2.3. Planificación Final

Tras haber completado la mayor parte del proyecto analizaremos cómo se ha distribuido el tiempo realmente y realizaremos una comparación para saber si las estimaciones iniciales eran correctas.

Las actividades que se tenían pensadas para cada iteración se han ejecutado según lo previsto por lo que las diferencias son únicamente de tiempos de ejecución no de cambios en la planificación.

A continuación, se comentarán los cambios en las iteraciones:

- **Primera iteración:** La duración de la actividad finalmente sí que fue de **60 horas**, siendo un poco mayor que la estimación inicial.

Aunque la duración fuese de 60 horas, el tiempo que se esperaba que se pudiese dedicar en esta fase fue menor al estimado inicialmente además por circunstancias externas se retrasó la planificación inicial un mes.

Lo que supuso que hasta el día 20 de marzo no pudiese empezar la iteración y hasta el 5 de mayo no acabase la fase de aprendizaje, colocándonos en el peor de los casos previstos inicialmente.

- **Segunda iteración:** tras el retraso sufrido por circunstancias externas se empezó desde el día 8 de mayo hasta el día 23 a dedicar una media de 9 horas diarias, lo que supone que en total se dedicaron **145 horas** en esta iteración por lo que no hubo gran diferencia respecto a la planificación inicial.
- **Tercera iteración:** La duración de esta fase fue de 9 días, en los cuales se dedicó una media de 9 horas diarias.

Lo que supone un total de **81 horas** coincidiendo con la estimación inicial.

- **Cuarta iteración:** La duración de esta fase ha sido de **96 horas**, en ella se han completado tanto el código como la documentación técnica.

El total de horas dedicadas al proyecto es de **382 horas** lo que suponen 22 horas más que la estimación inicial.

Viendo los resultados finales y comparándolos con la estimación inicial se deduce que se realizó una estimación inicial muy adecuada ya que únicamente se ha incrementado en un 6%. Esto se debe principalmente a dos factores:

- Antes de realizar el proyecto ya habían realizado proyectos previos en PHP.
- Como se dedicó íntegramente una fase de aprendizaje de nuevas tecnologías los posibles errores por “aprendizaje sobre la marcha” se eliminaron a lo que hay que sumarle que las estimaciones se realizaron con la conjetura de que el desarrollador tenía el conocimiento de las tecnologías.

## 2.2.4. Presupuesto económico:

En esta sección se va a calcular el coste tanto de hardware que ha supuesto la realización del proyecto, como cuál es el coste que supondría el proyecto con el salario medio de un desarrollador. A continuación, se detallan los presupuestos:

### 2.2.4.1. Hardware y software

Para la realización de todo el proyecto se han utilizado programas de uso gratuito, por lo tanto, en el apartado software no ha sido necesario ningún tipo de inversión.

En el apartado hardware el único dispositivo necesario ha sido el ordenador junto a sus complementos (ratón, teclado...).

Para calcular el coste de los dispositivos se ha calculado el número de horas que se han utilizado los mismos. Por lo tanto, se calcula la parte de la inversión relacionada con el proyecto ya que los dispositivos van a tener más vida útil después del proyecto y podrán ser utilizados en otros.

El coste del ordenador junto a sus complementos fue de 800€, y la vida útil media de estos dispositivos es de 5 años. Por lo tanto, al haber utilizado el ordenador un total de 5 meses (aunque únicamente se haya utilizado 382 horas, la vida útil tiene en cuenta el tiempo no el uso, y el proyecto se ha llevado a cabo desde el mes de febrero al de junio), supone que se ha dedicado un 8,33% de la vida útil del dispositivo al proyecto.

Como se mencionó si el coste de los dispositivos fue de 800€, para el desarrollo del proyecto se habrían invertido un total de **66,64€**

### 2.2.4.2. Recursos humanos

En este apartado se hace una estimación en función del número de horas invertidas y el presupuesto medio de un desarrollador PHP en España.

El salario medio de un desarrollador PHP en España es de 2500€ mensuales.

Una jornada completa de trabajo supone 8 horas diarias. Como se comentó la primera iteración fue de aprendizaje, un desarrollador experimentado no necesitaría esta fase, por lo que reduciría la duración del proyecto en 60 horas, lo que supondría un total de 322 horas.

Siendo una jornada completa de 8 horas, se necesitarían un total de 40, 25 jornadas. A lo largo de un mes laboral hay una media de 21 jornadas. Haciendo un pequeño redondeo se necesitarían un total de dos meses para la realización del proyecto. Multiplicando por el salario del desarrollador obtendríamos que se ha de pagar **5000€** aproximadamente.



### 2.2.4.3. Presupuesto total

En esta sección se calculará el presupuesto total estimado del proyecto. Además de incluir los gastos mencionados en las dos secciones anteriores también se incluirán los gastos secundarios como los relacionados con el espacio de trabajo.

Por simplificación supondremos que el desarrollador PHP, con experiencia soy yo mismo en un futuro. En ese supuesto ya se descontaron las horas de aprendizaje y asumiendo que me encuentro en mi residencia actual, los gastos derivados del desarrollo únicamente serían los de la luz e internet.

El gasto medio mensual de luz en mi residencia actual es de **56€** y el de internet de **50€**.

Lo que supondrían respectivamente para los dos meses de desarrollo **112€** y **100€**.

En resumen, el presupuesto total podría resumirse de la siguiente manera:

| <b>Gasto</b>     | <b>Coste</b>    |
|------------------|-----------------|
| Hardware         | 66,64€          |
| Recursos humanos | 5000€           |
| Otros gastos     | 212€            |
| <b>Total</b>     | <b>5278,64€</b> |

*Tabla 4 Presupuesto total*



# Parte II

## Documentación técnica



# Capítulo 3

## Análisis

### 3.1. Requisitos

En esta sección se comentan los requisitos necesarios para la elaboración del proyecto.

Estos requisitos se van a dividir en 3 grupos: Requisitos funcionales, no funcionales y requisitos de información. A su vez los requisitos funcionales al ser tan numerosos van a estar divididos en los distintos “sistemas” o vistas del programa. Los requisitos de información al ser menos numerosos se van a agrupar todos los del proyecto al igual que los requisitos no funcionales.

Los requisitos funcionales son aquellos que describen las funcionalidades del sistema. Cada uno de los requisitos funcionales ha de ser implementado por los desarrolladores para poder satisfacer las necesidades de los usuarios. Estas necesidades se reflejan a través de los casos de uso de los usuarios y se mostrarán detallados en la sección siguiente. Aunque hay distintos subgrupos tanto en los requisitos funcionales como no funcionales, en este caso no se mencionarán estos subgrupos.

Los requisitos no funcionales son los que se encargan de especificar los atributos de calidad del software y los requisitos de información son aquellos que describen el formato de la información transmitida por el sistema y la información en sí misma.

A continuación, se muestran todo este conjunto de requisitos perteneciente a la aplicación.

### 3.1.1. Requisitos funcionales

El conjunto de requisitos funcionales de la aplicación dividido por sistemas o vistas es el siguiente.

#### **Sistema de Login:**

| ID    | Nombre                      | Descripción   |
|-------|-----------------------------|---|
| RF_01 | Registro usuario            | El sistema mostrará un formulario en el que una vez introducidos los datos con el formato correcto se registrará al nuevo usuario en la base de datos   |
| RF_02 | Envío confirmación          | El sistema enviará un email a los usuarios dados de alta en la aplicación para que confirmen su cuenta.   |
| RF_03 | Error registro usuario      | En caso de que el usuario escriba un formato de los datos incorrecto, o se intente registrar un usuario ya registrado el sistema no permitirá que se registre el nuevo usuario y mostrará los mensajes de error correspondientes. |
| RF_04 | Iniciar sesión              | El sistema mostrará un formulario para que un usuario ya registrado en la aplicación tras escribir el email y la contraseña correctos inicie sesión en la aplicación  |
| RF_05 | Error inicio sesión         | Si el usuario introduce un usuario o contraseñas incorrectos o datos que no siguen el formato correcto o el usuario no ha confirmado su cuenta el sistema mostrará los mensajes de error correspondientes.                        |
| RF_06 | Reestablecer contraseña     | Un usuario ya registrado en la aplicación podrá recuperar la contraseña   |
| RF_07 | Envío email de recuperación | El sistema cuando un usuario escriba el email del cual quiere recuperar la contraseña estando este registrado y confirmado, enviará un email con un enlace para que pueda recuperar la contraseña                                 |

Tabla 5 Requisitos funcionales sistema de Login

#### **Vista principal:**

| ID    | Nombre                             | Descripción  |
|-------|------------------------------------|--|
| RF_01 | Visualización actividades y tareas | El sistema mostrará las actividades y tareas donde el usuario sea el propietario.                                      |
| RF_02 | Creación actividades y tareas      | El sistema permitirá la creación de actividades y tareas cuando el usuario pulse a los botones correspondientes        |
| RF_03 | Eliminación de actividades         | El sistema permitirá la eliminación de las actividades del usuario cuando el usuario pulse el botón correspondiente    |
| RF_04 | Ver historial actividad            | El sistema mostrará botones asociados a cada actividad que permitirán al usuario navegar al historial de la misma.     |
| RF_05 | Ver temporizadores tarea           | El sistema mostrará botones asociados a cada tarea que permitirán al usuario navegar a los temporizadores de la misma. |
| RF_06 | Ver resumen tarea                  | El sistema mostrará botones asociados a cada tarea que permitirán al usuario navegar al resumen la misma.              |

Tabla 6 Requisitos funcionales vista principal

**Vista estadísticas de una tarea:**

| ID    | Nombre               | Descripción  |
|-------|----------------------|--|
| RF_01 | Mostrar nombre tarea | El sistema mostrará el nombre de la tarea con la que se está interactuando.  |
| RF_02 | Mostrar gráfico      | El sistema representará en un gráfico el tiempo total efectivo de la tarea y el de descanso.   |
| RF_03 | Tiempo total         | El sistema mostrará el tiempo total invertido en la tarea  |
| RF_04 | Porcentaje descanso  | El sistema calculará y mostrará el porcentaje del tiempo descansado respecto al tiempo total.  |
| RF_05 | Añadir tiempo        | El sistema mostrará un botón que al pulsarle salga un modal en el cual el usuario pueda crear un nuevo registro de la tarea con el tiempo efectivo y de descanso elegidos y la fecha actual. |
| RF_06 | Eliminar tarea       | El sistema permitirá una opción para eliminar la tarea   |

*Tabla 7 Requisitos funcionales de la vista de estadísticas de una tarea*

**Vista temporizadores**

| ID    | Nombre                   | Descripción   |
|-------|--------------------------|---|
| RF_01 | Interactuar temporizador | El sistema mostrará los botones correspondientes para que el usuario pueda iniciar pausar y reanudar los temporizadores.                        |
| RF_02 | Parar temporizadores     | Cada vez que se inicie o reanude un temporizador el sistema detendrá el otro.   |
| RF_03 | Almacenar resultados     | El sistema mostrará un botón para que el usuario para almacenar los datos de los temporizadores en el registro correspondiente.                 |
| RF_04 | Mensaje confirmación     | Cuando el usuario pulse en el botón de almacenar los resultados se mostrará un modal con las opciones de confirmar o cancelar.                  |
| RF_05 | Boquear botones          | En función de los botones de los temporizadores que pulse el usuario el sistema bloqueará los correspondientes para que no puedan ser pulsados. |

*Tabla 8 Requisitos funcionales del sistema de temporizadores*

**Histórico de una actividad:**

| ID    | Nombre                   | Descripción  |
|-------|--------------------------|--|
| RF_01 | Mostrar nombre tarea     | El sistema mostrará el nombre de la tarea con la que se está interactuando.  |
| RF_02 | Mostrar histórico        | El sistema mostrará el histórico de la actividad con todos los registros de las tareas asociadas a la actividad indicando el día en el que se trabajó en esa tarea, el tiempo efectivo y de descanso que se realizó y el nombre de la tarea. |
| RF_03 | Suma tiempos             | El sistema mostrará la suma de todos los tiempos de los registros de las tareas asociadas a la actividad.  |
| RF_04 | Suma tiempos efectivos   | El sistema mostrará la suma de todos los tiempos efectivos de los registros de las tareas asociadas a la actividad.  |
| RF_05 | Suma tiempos de descanso | El sistema mostrará la suma de todos los tiempos de descanso de los registros de las tareas asociadas a la actividad.  |
| RF_06 | Exportar a PDF           | El sistema mostrará un botón que cuando el usuario le de 'click' exportará a PDF todo el registro de la actividad y permitirá al usuario almacenarlo   |

*Tabla 9 Requisitos funcionales histórico de una actividad*

**Vista principal actividades grupales:**

| ID    | Nombre                             | Descripción   |
|-------|------------------------------------|---|
| RF_01 | Visualización actividades y tareas | El sistema mostrará las actividades y tareas donde el usuario sea participante.   |
| RF_02 | Creación actividades y tareas      | El sistema permitirá la creación de actividades y tareas cuando el usuario pulse a los botones correspondientes                         |
| RF_03 | Eliminación de actividades         | El sistema permitirá la eliminación de las actividades grupales cuando el usuario que lo va a eliminar fue quien lo creó.               |
| RF_04 | Ver historial actividad            | El sistema mostrará botones asociados a cada actividad que permitirán al usuario navegar al historial de la misma.                      |
| RF_05 | Ver temporizadores tarea           | El sistema mostrará botones asociados a cada tarea que permitirán al usuario navegar a los temporizadores de la misma.                  |
| RF_06 | Ver resumen tarea                  | El sistema mostrará botones asociados a cada tarea que permitirán al usuario navegar al resumen la misma.                               |
| RF_07 | Invitar usuarios                   | El sistema mostrará un botón a los participantes de las actividades grupales para poder invitar a otros usuarios a la actividad grupal. |
| RF_08 | Envío de invitación                | Tras invitar a un usuario a través de su email este recibirá una invitación para formar parte del grupo                                 |

*Tabla 10 Requisitos funcionales de la vista principal de las actividades grupales*



**Vista de estadísticas de una actividad grupal**

| ID    | Nombre               | Descripción   |
|-------|----------------------|---|
| RF_01 | Mostrar nombre tarea | El sistema mostrará el nombre de la tarea con la que se está interactuando.   |
| RF_02 | Mostrar gráfico      | El sistema representará en un gráfico el tiempo efectivo invertido por cada usuario   |
| RF_03 | Tiempo por usuario   | El sistema mostrará el tiempo efectivo total asociado al usuario, así como el porcentaje del total que equivale y el tiempo descansado  |
| RF_04 | Tiempo total         | El sistema mostrará el tiempo total invertido en la tarea   |
| RF_05 | Añadir tiempo        | El sistema mostrará un botón que al pulsarle salga un modal en el cual el usuario pueda crear un nuevo registro de la tarea con el tiempo efectivo y de descanso elegidos, la fecha actual y el usuario que lo añade. |
| RF_06 | Eliminar tarea       | El sistema mostrará una opción para poder eliminar la tarea   |

Tabla 11 Requisitos funcionales de la vista de estadísticas de una actividad grupal

**Temporizadores tarea grupal:**

| ID    | Nombre                   | Descripción   |
|-------|--------------------------|---|
| RF_01 | Interactuar temporizador | El sistema mostrará los botones correspondientes para que el usuario pueda iniciar pausar y reanudar los temporizadores.                        |
| RF_02 | Parar temporizadores     | Cada vez que se inicie o reanude un temporizador el sistema detendrá el otro.   |
| RF_03 | Almacenar resultados     | El sistema mostrará un botón para que el usuario para almacenar los datos de los temporizadores en el registro correspondiente.                 |
| RF_04 | Mensaje confirmación     | Cuando el usuario pulse en el botón de almacenar los resultados se mostrará un modal con las opciones de confirmar o cancelar.                  |
| RF_05 | Bloquear botones         | En función de los botones de los temporizadores que pulse el usuario el sistema bloqueará los correspondientes para que no puedan ser pulsados. |

Tabla 12 Requisitos funcionales de la vista de temporizadores de una actividad grupal

**Barra lateral de navegación:**

| ID    | Nombre                                | Descripción   |
|-------|---------------------------------------|---|
| RF_01 | Cerrar sesión                         | El sistema mostrará en la barra lateral un botón para cerrar la sesión del usuario  |
| RF_02 | Navegar vista principal               | El sistema mostrará en la barra lateral una opción para navegar a la vista principal  |
| RF_03 | Navegar vista de actividades grupales | El sistema mostrará en la barra lateral una opción para navegar a la vista de las actividades grupales  |
| RF_04 | Navegar buzón invitaciones            | El sistema mostrará botones asociados a cada actividad que permitirán al usuario navegar al buzón de invitaciones de las actividades grupales |

Tabla 13 Requisitos funcionales de la barra lateral de navegación

**Histórico actividad grupal:**

| ID    | Nombre                       | Descripción   |
|-------|------------------------------|---|
| RF_01 | Mostrar nombre tarea         | El sistema mostrará el nombre de la tarea con la que se está interactuando.   |
| RF_02 | Mostrar histórico            | El sistema mostrará el histórico de la actividad con todos los registros de las tareas asociadas a la actividad indicando el día en el que se trabajó en esa tarea, el tiempo efectivo y de descanso que se realizó, el nombre de la tarea y el usuario que creó el registro. |
| RF_03 | Suma tiempos                 | El sistema mostrará la suma de todos los tiempos de los registros de las tareas asociadas a la actividad.   |
| RF_04 | Suma tiempos efectivos       | El sistema mostrará la suma de todos los tiempos efectivos de los registros de las tareas asociadas a la actividad.   |
| RF_05 | Suma tiempos de descanso     | El sistema mostrará la suma de todos los tiempos de descanso de los registros de las tareas asociadas a la actividad.   |
| RF_06 | Exportar a PDF               | El sistema mostrará un botón que cuando el usuario le de 'click' exportará a pdf todo el registro de la actividad y permitirá al usuario almacenarlo  |
| RF_07 | Suma tiempo efectivo usuario | El sistema mostrará la suma del tiempo efectivo invertido por el usuario en la actividad  |
| RF_08 | Suma tiempo descanso usuario | El sistema mostrará la suma del tiempo de descanso invertido por el usuario en la actividad   |

Tabla 14 Requisitos funcionales del histórico de una actividad grupal

**Invitaciones:**

| ID    | Nombre                     | Descripción   |
|-------|----------------------------|---|
| RF_01 | Visualización invitaciones | El sistema mostrará las invitaciones a las actividades grupales que se le han enviado al usuario                                  |
| RF_02 | Aceptar invitación         | El sistema mostrará en cada invitación un botón que cuando sea pulsado se introduzca al usuario como participante de la actividad |
| RF_03 | Rechazar invitación        | El sistema mostrará en cada invitación un botón que cuando sea pulsado se elimine la invitación asociada                          |

Tabla 15 Requisitos funcionales del sistema de invitaciones

3.1.2. Requisitos no funcionales:

| ID     | Nombre   | Descripción  |
|--------|--|--|
| RNF_01 | Aplicación responsive                                    | La aplicación debe adaptarse tanto a los tamaños de pantalla móviles como de ordenador                                   |
| RNF_02 | Actividades grupales en tiempo real                      | Se enviarán peticiones constantes cada 3 segundos al servidor para actualizar las tareas y actividades grupales.         |
| RNF_03 | Invitaciones actividades grupales                        | Se enviarán peticiones constantes cada 10 segundos al servidor para ver si se han recibido invitaciones nuevas           |
| RNF_04 | Eliminación actividades grupales                         | Solo el creador de la actividad grupal podrá eliminarla.   |
| RNF_05 | Las comunicaciones de la aplicación deben utilizar HTTPS | Las comunicaciones de la aplicación se cifrarán utilizando certificado HTTPS   |
| RNF_06 | Las contraseñas de los usuarios deben estar cifradas     | Para almacenar las contraseñas de los usuarios en la base de datos estas deben estar cifradas                            |
| RNF_07 | Recuperación de contraseña                               | El usuario podrá recuperar su contraseña utilizando su correo electrónico con el cual se dio de alta                     |
| RNF_08 | Apis privadas  | Para que cualquier api devuelva datos el usuario debe tener una sesión iniciada en la aplicación                         |
| RNF_09 | Manual de usuario  | En la documentación técnica se incluirán manuales de usuario que expliquen el funcionamiento de la aplicación            |
| RNF_10 | Mensajes de error  | El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final                        |
| RNF_11 | Colores  | Los colores utilizados en la aplicación deben ser complementarios o deben provenir de paletas de colores preestablecidas |
| RNF_12 | Aprendizaje  | El tiempo de aprendizaje de un usuario debe ser menor a 1 hora   |
| RNF_13 | Interfaz   | La interfaz debe ser clara para que pueda ser fácilmente interpretable por el usuario                                    |
| RNF_14 | Barra lateral  | La barra lateral se ajustará automáticamente al tamaño de la pantalla para mostrarse contraída o extendida               |

Tabla 16 Requisitos no funcionales de la aplicación

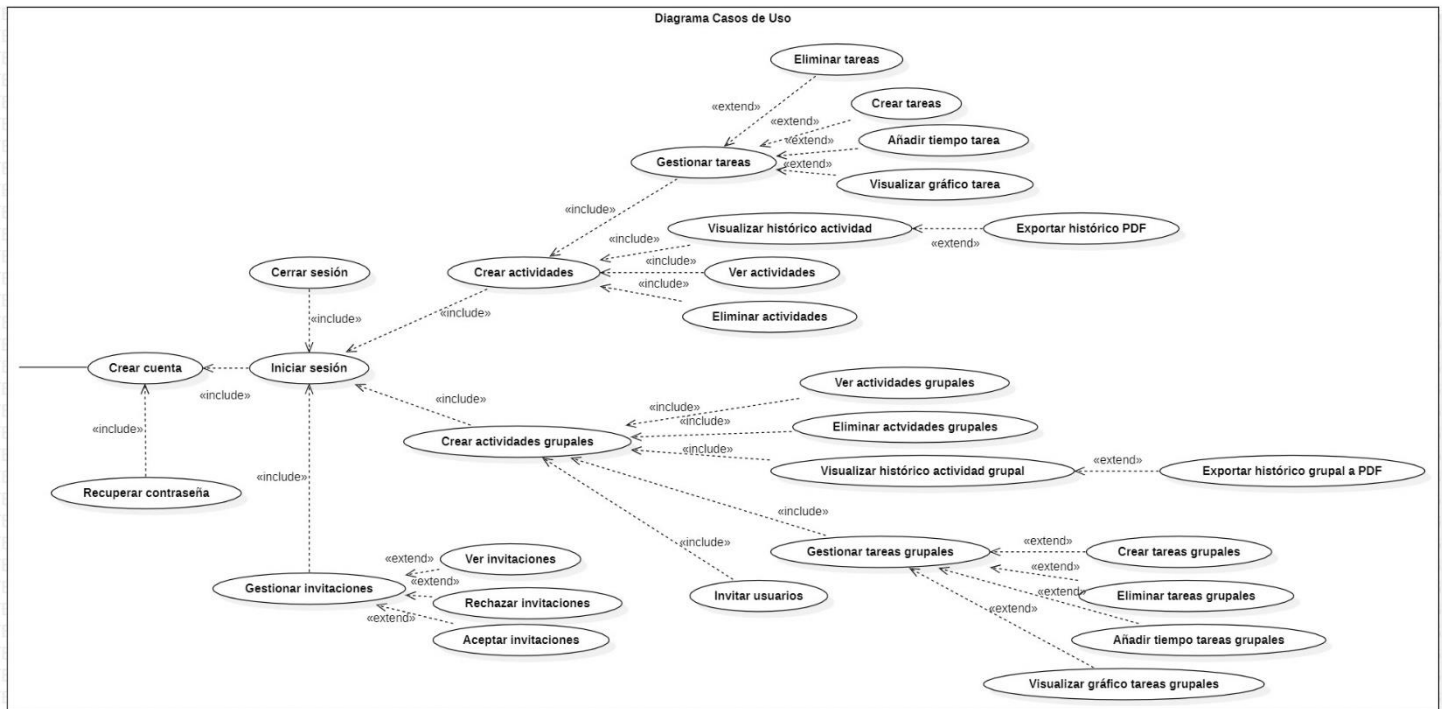
### 3.1.3. Requisitos de información

| ID    | Nombre              | Descripción   |
|-------|---------------------|---|
| RI_01 | Registro usuario    | Un usuario podrá registrarse en el sistema introduciendo nombre, teléfono, email y contraseña   |
| RI_02 | Temporizadores      | Se almacenará el tiempo de los temporizadores con un formato: 'hh:mm:ss'  |
| RF_03 | Fecha actual        | Al añadir tiempo la fecha del sistema tendrá un formato: 'yy:mm:dd'   |
| RF_04 | Id único            | A cada usuario se le asigna un ID único   |
| RF_05 | Registro tarea      | En cada registro de una tarea se mostrará el tiempo efectivo, el tiempo de descanso, el día en el que se guardó, el id de la tarea y el id del usuario que guardó el registro |
| RF_06 | Histórica actividad | Cada actividad mostrará en el histórico los registros de las tareas que les pertenezcan   |

Tabla 17 Requisitos de información de la aplicación

### 3.2. Diagrama de casos de uso

En el siguiente modelo se muestran los casos de uso presentes en la aplicación:



0.1 Diagrama de casos de uso

### 3.3. Historias de usuario

Para una mayor especificación y entendimiento de los distintos casos de uso se mostrará una historia de usuario asociada a cada caso de uso, ya que estas muestran las explicaciones de las diferentes funcionalidades desde la perspectiva del cliente, lo que puede favorecer el entendimiento de personas que no tengan un gran conocimiento técnico.

Cabe destacar que en los casos de uso muy similares como podría ser el caso de “Crear actividades” o de “Crear actividades grupales” únicamente se documenta uno de ellos por no repetir la misma información. Por lo que los casos de las actividades grupales similares a las actividades individuales no se han documentado, los diferentes sí.

|                    |   |
|--------------------|---|
| <b>ID</b>          | <b>HU_01</b>  |
| <b>Nombre</b>      | Iniciar sesión  |
| <b>Prioridad</b>   | Alta  |
| <b>Riesgo</b>      | Alto  |
| <b>Descripción</b> | Como usuario quiero iniciar sesión en la aplicación introduciendo mi email y contraseña |
| <b>Validación</b>  | Quiero poder iniciar sesión en cualquier momento.                                       |

Tabla 18 Historia de usuario del caso de uso "iniciar sesión"

|                    |  |
|--------------------|--|
| <b>ID</b>          | <b>HU_02</b>   |
| <b>Nombre</b>      | Crear cuenta   |
| <b>Prioridad</b>   | Alta   |
| <b>Riesgo</b>      | Alto   |
| <b>Descripción</b> | Como usuario quiero crea una cuenta en la aplicación introduciendo mi email, nombre y contraseña |
| <b>Validación</b>  | Quiero poder crear una cuenta en cualquier momento.  |

Tabla 19 Historia de usuario del caso de uso "Crear cuenta"

|                    |   |
|--------------------|---|
| <b>ID</b>          | <b>HU_03</b>  |
| <b>Nombre</b>      | Recuperar contraseña  |
| <b>Prioridad</b>   | Alta  |
| <b>Riesgo</b>      | Alto  |
| <b>Descripción</b> | Como usuario quiero poder recuperar la contraseña en caso de olvidarla                                    |
| <b>Validación</b>  | - Quiero poder recuperar la contraseña en caso de olvidarla<br>- Quiero poder elegir una nueva contraseña |

Tabla 20 Historia de usuario del caso de uso "Recuperar contraseña"

|                    |   |
|--------------------|---|
| <b>ID</b>          | <b>HU_04</b>  |
| <b>Nombre</b>      | Gestionar actividades   |
| <b>Prioridad</b>   | Alta  |
| <b>Riesgo</b>      | Alto  |
| <b>Descripción</b> | Como usuario quiero poder gestionar mis actividades   |
| <b>Validación</b>  | - Quiero poder visualizar mis actividades creadas<br>- Quiero poder crear nuevas actividades<br>- Quiero poder eliminar mis actividades |

Tabla 21 Historia de usuario del caso de uso "Gestionar actividades"

|                    |  |
|--------------------|--|
| <b>ID</b>          | <b>HU_05</b>   |
| <b>Nombre</b>      | Gestionar tareas   |
| <b>Prioridad</b>   | Alta   |
| <b>Riesgo</b>      | Alto   |
| <b>Descripción</b> | Como usuario quiero poder gestionar mis tareas   |
| <b>Validación</b>  | <ul style="list-style-type: none"> <li>- Quiero poder visualizar mis tareas creadas</li> <li>- Quiero poder crear nuevas tareas</li> <li>- Quiero poder eliminar mis tareas</li> </ul> |

Tabla 22 Historia de usuario del caso de uso "Gestionar tareas"

|                    |   |
|--------------------|---|
| <b>ID</b>          | <b>HU_06</b>  |
| <b>Nombre</b>      | Visualizar histórico actividad  |
| <b>Prioridad</b>   | Alta  |
| <b>Riesgo</b>      | medio   |
| <b>Descripción</b> | Como usuario quiero poder ver los históricos de mis actividades   |
| <b>Validación</b>  | <ul style="list-style-type: none"> <li>-Quiero poder visualizar los tiempos de las tareas y el día que trabajé en ellas</li> <li>- Quiero poder ver el tiempo total de la actividad</li> <li>- Quiero poder ver el tiempo total efectivo de la actividad</li> <li>- Quiero poder ver el tiempo total de descanso de la actividad</li> </ul> |

Tabla 23 Historia de usuario del caso de uso "Visualizar histórico actividad"

|                    |  |
|--------------------|--|
| <b>ID</b>          | <b>HU_07</b>   |
| <b>Nombre</b>      | Exportar histórico actividad   |
| <b>Prioridad</b>   | media  |
| <b>Riesgo</b>      | bajo   |
| <b>Descripción</b> | Como usuario quiero poder descargar un PDF del histórico de la actividad |
| <b>Validación</b>  | Quiero poder mantener el mismo formato                                   |

Tabla 24 Historia de usuario del caso de uso "Exportar histórico actividad"

|                    |   |
|--------------------|---|
| <b>ID</b>          | <b>HU_09</b>  |
| <b>Nombre</b>      | Añadir tiempo tareas  |
| <b>Prioridad</b>   | Alta  |
| <b>Riesgo</b>      | Alta  |
| <b>Descripción</b> | Como usuario quiero poder añadir tiempo a las tareas  |
| <b>Validación</b>  | <ul style="list-style-type: none"> <li>- Quiero poder añadir tiempo efectivo y tiempo de descanso</li> <li>- Quiero poder añadirlo a través de los temporizadores</li> <li>- Quiero poder añadirlo a través de un formulario</li> </ul> |

Tabla 25 Historia de usuario del caso de uso "Añadir tiempo tareas"

|                    |  |
|--------------------|--|
| <b>ID</b>          | <b>HU_10</b>   |
| <b>Nombre</b>      | Visualizar tiempo tareas   |
| <b>Prioridad</b>   | Alta   |
| <b>Riesgo</b>      | medio  |
| <b>Descripción</b> | Como usuario quiero poder visualizar el tiempo invertido en la tarea   |
| <b>Validación</b>  | <ul style="list-style-type: none"> <li>- Quiero poder visualizar en un gráfico el tiempo efectivo y de descanso con sus respectivos porcentajes.</li> <li>- Quiero poder visualizar el tiempo total de la tarea</li> <li>- Quiero poder visualizar el porcentaje de tiempo descansado respecto al total de la tarea</li> </ul> |

Tabla 26 Historia de usuario del caso de uso "Visualizar tiempo tareas"

|                    |   |
|--------------------|---|
| <b>ID</b>          | <b>HU_11</b>  |
| <b>Nombre</b>      | Enviar invitaciones   |
| <b>Prioridad</b>   | Alta  |
| <b>Riesgo</b>      | Alto  |
| <b>Descripción</b> | Como usuario quiero poder invitar a otros usuarios a las actividades grupales en las que participo              |
| <b>Validación</b>  | <ul style="list-style-type: none"> <li>- Quiero poder invitar a otros usuarios escribiendo su email.</li> </ul> |

Tabla 27 Historia de usuario del caso de uso "Enviar invitaciones"

|                    |   |
|--------------------|---|
| <b>ID</b>          | <b>HU_12</b>  |
| <b>Nombre</b>      | Enviar invitaciones   |
| <b>Prioridad</b>   | Alta  |
| <b>Riesgo</b>      | Alto  |
| <b>Descripción</b> | Como usuario quiero poder invitar a otros usuarios a las actividades grupales en las que participo              |
| <b>Validación</b>  | <ul style="list-style-type: none"> <li>- Quiero poder invitar a otros usuarios escribiendo su email.</li> </ul> |

Tabla 28 Historia de usuario del caso de uso "Enviar invitaciones"

|                    |   |
|--------------------|---|
| <b>ID</b>          | <b>HU_12</b>  |
| <b>Nombre</b>      | Aceptar invitaciones  |
| <b>Prioridad</b>   | Alta  |
| <b>Riesgo</b>      | Alto  |
| <b>Descripción</b> | Como usuario quiero poder aceptar o rechazar las invitaciones que me envían   |
| <b>Validación</b>  | <ul style="list-style-type: none"> <li>- Quiero poder aceptarlas para formar parte del grupo</li> <li>- Quiero poder rechazarlas porque no quiero formar parte del grupo</li> </ul> |

Tabla 29 Historia de usuario del caso de uso "Aceptar invitaciones"





# Capítulo 4

## Diseño

### 4.1. Arquitectura Lógica

La arquitectura software es el diseño de más alto nivel de estructura de un sistema.

La arquitectura software también es conocida como arquitectura lógica. Esta arquitectura consiste en el conjunto de patrones y abstracciones que definen como se debe interactuar con el código del sistema software.

Para seleccionar una arquitectura adecuada se deben obtener los requisitos del sistema como se realizó en el capítulo 4 además de restricciones y tener en cuenta las características adicionales y contexto del sistema.

Por estas razones es considerado uno de los puntos más críticos del sistema ya que va a definir aspectos como la escalabilidad del sistema o la propia dificultad del mantenimiento de la aplicación.

Debido a todas las características de la aplicación se determinó que la arquitectura más adecuada era la arquitectura “Modelo Vista Controlador (MVC)”.

Esta arquitectura permite una diferenciación clara entre tres capas diferentes: Modelos, vistas y controladores.

Los **modelos** son los encargados de trabajar con los datos y es la encargada tanto de la inserción como de la lectura de datos de la base de datos.

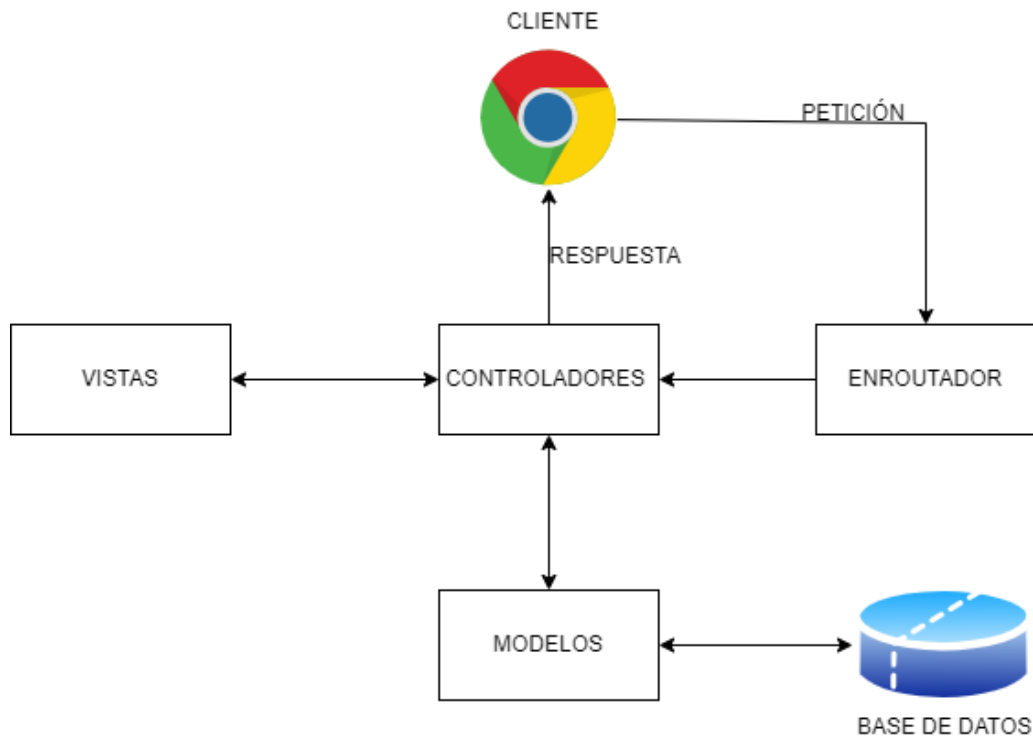
Los **controladores** son el enlace entre las vistas y los modelos, son los encargados de presentar las vistas correspondientes a los usuarios con los datos de los modelos y de transmitir los datos que insertan los datos a los modelos correspondientes.

Las **vistas** contienen el código que produce la visualización de las interfaces de usuario (generalmente código HTML). En ellas no se va a aplicar lógica con los datos ya que su función es la representación de los mismos.

En esta aplicación además de con los elementos anteriores se cuenta con un elemento adicional llamado **enrutador**, este se encarga de recibir y redireccionar las peticiones del usuario a los controladores correspondientes, así como definir las diferentes rutas de la aplicación.

En el diagrama que se muestra a continuación podemos ver cómo es de una manera detallada el flujo de datos a lo largo de todo el sistema.

DIAGRAMA DE ALTO NIVEL

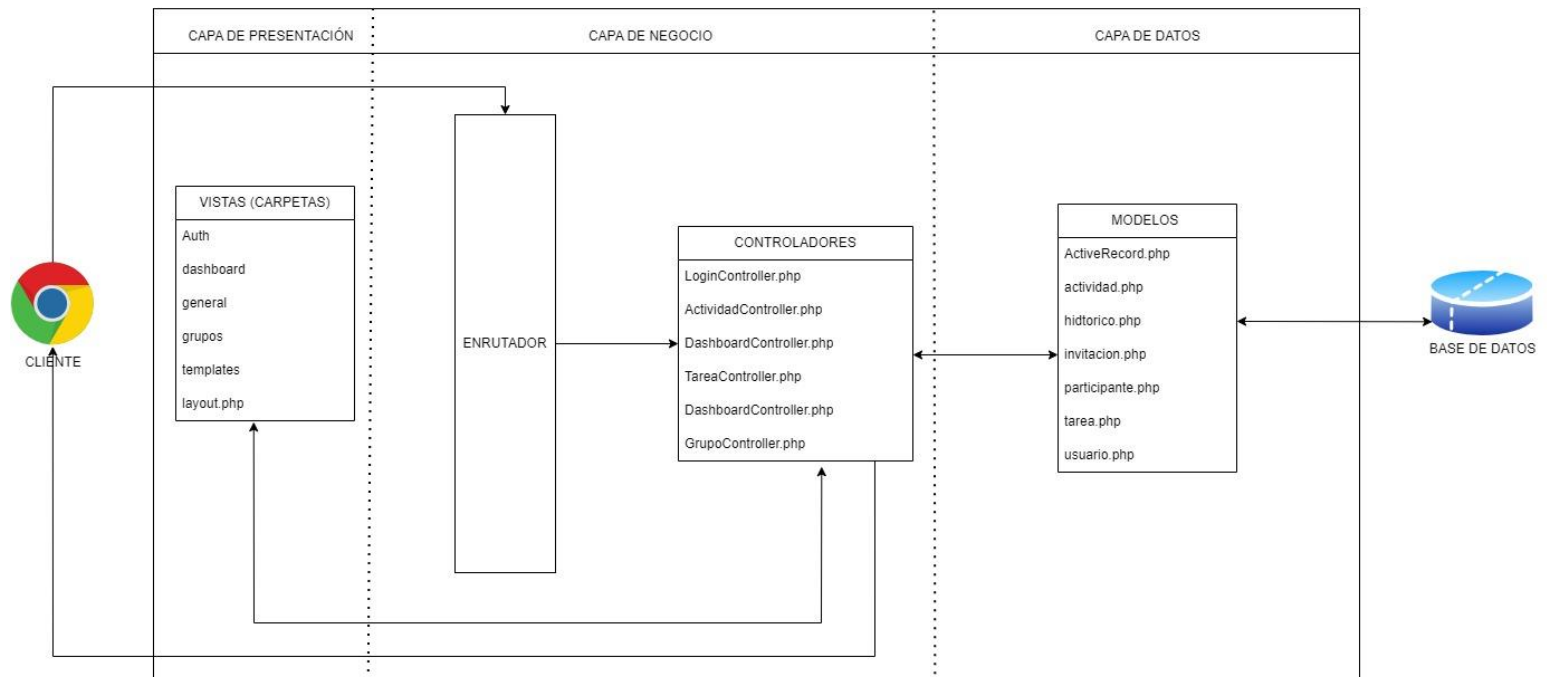


0.1 Arquitectura lógica de alto nivel

En el diagrama anterior vemos de una manera mucho más clara como el usuario se conecta a través de un cliente (un navegador) con la aplicación, el enrutador es el encargado de recibir la petición del cliente y llamar a los controladores correspondientes. Una vez los controladores reciben la llamada estos se comunicarán con los modelos para recuperar los datos correspondientes de la base de datos y los representará junto a las vistas en el navegador del cliente.

DIAGRAMA DETALLADO

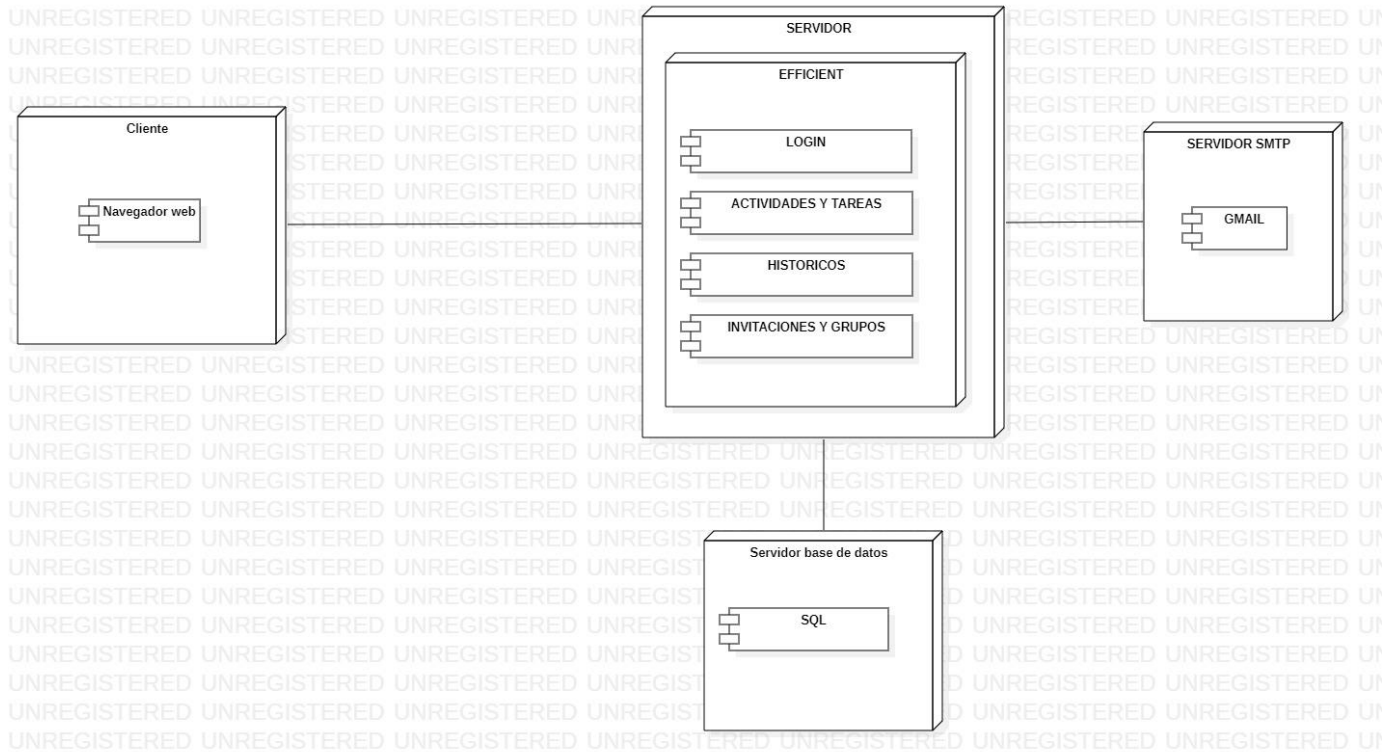
A continuación, se muestra la arquitectura lógica de la aplicación de una manera detallada, pudiendo observar de una manera mucho más clara las diferentes capas de la aplicación.



0.2 Arquitectura lógica detallada

## 4.2. Arquitectura Física

La arquitectura física de una aplicación software es el conjunto de elementos y recursos hardware utilizados por la aplicación. Lo que incluye servidores, redes, sistemas de bases de datos y cualquier componente hardware que forma parte de la infraestructura de la aplicación para el funcionamiento de la aplicación.



0.3 Arquitectura física

### 4.3. Diagramas de secuencia

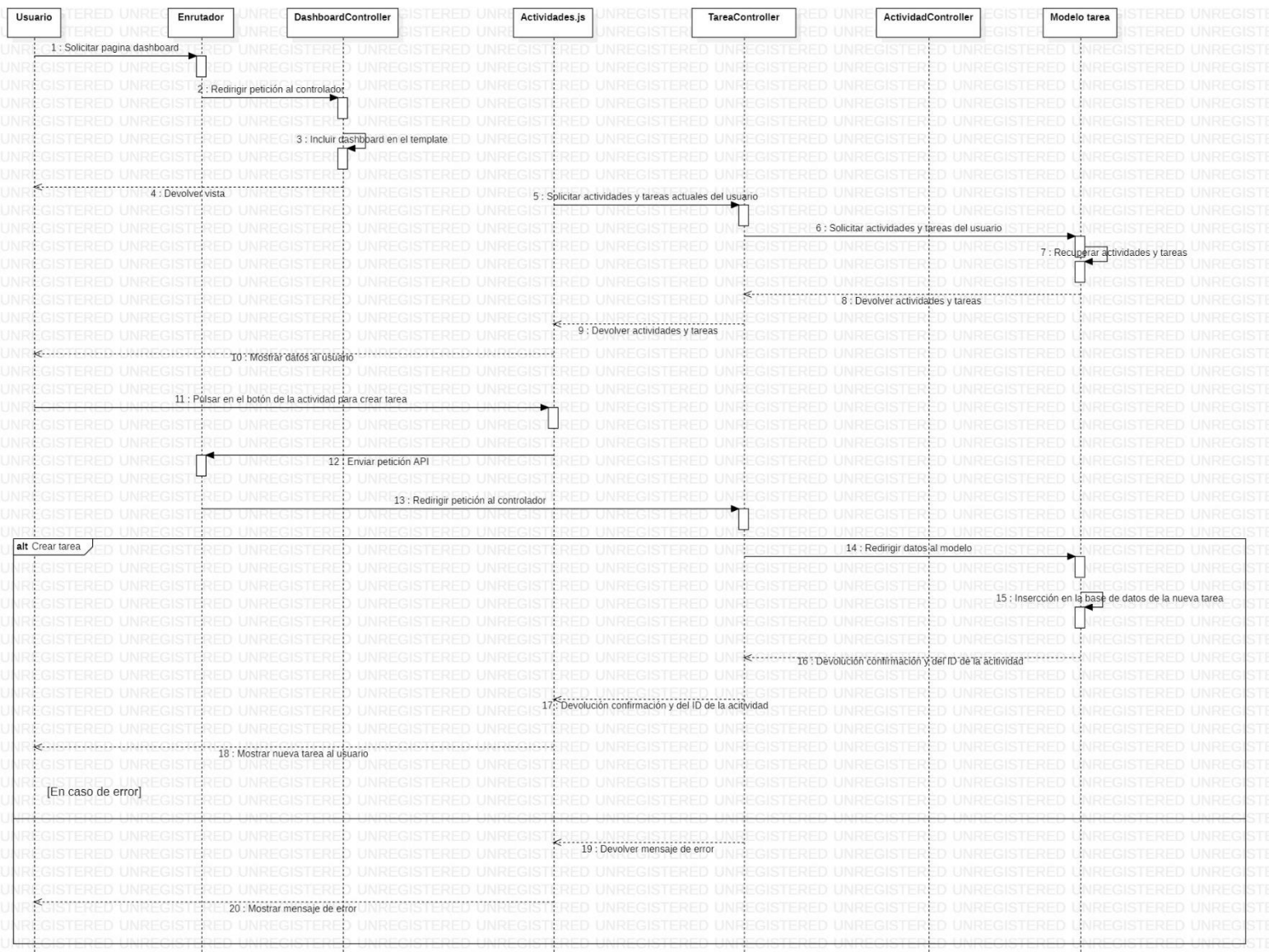
Los diagramas de secuencia son diagramas que muestran la iteración entre los objetos de la aplicación a través del tiempo.

Permiten observar cómo las diferentes partes de un sistema interactúan entre sí para realizar alguna función determinada además de indicar el orden en el que se producen las distintas acciones del sistema.

A continuación, se muestran algunos de los diagramas de secuencia que son más importantes o representativos.

#### 4.3.1. Diagrama de secuencia crear actividad.

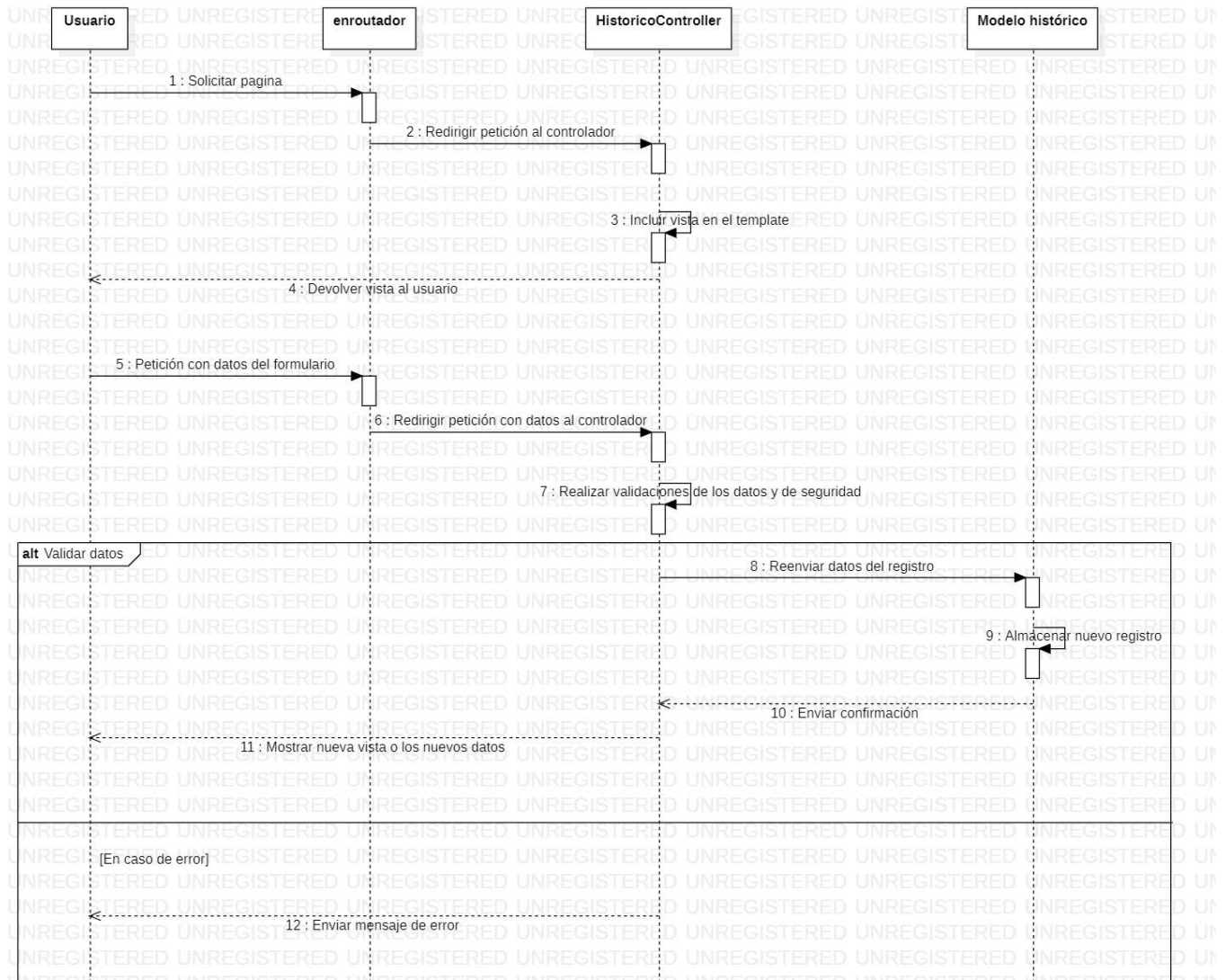
Diagrama de secuencia crear actividad.



0.4 Diagrama de secuencia "crear actividad"

### 4.3.2. Diagrama de secuencia almacenar registro.

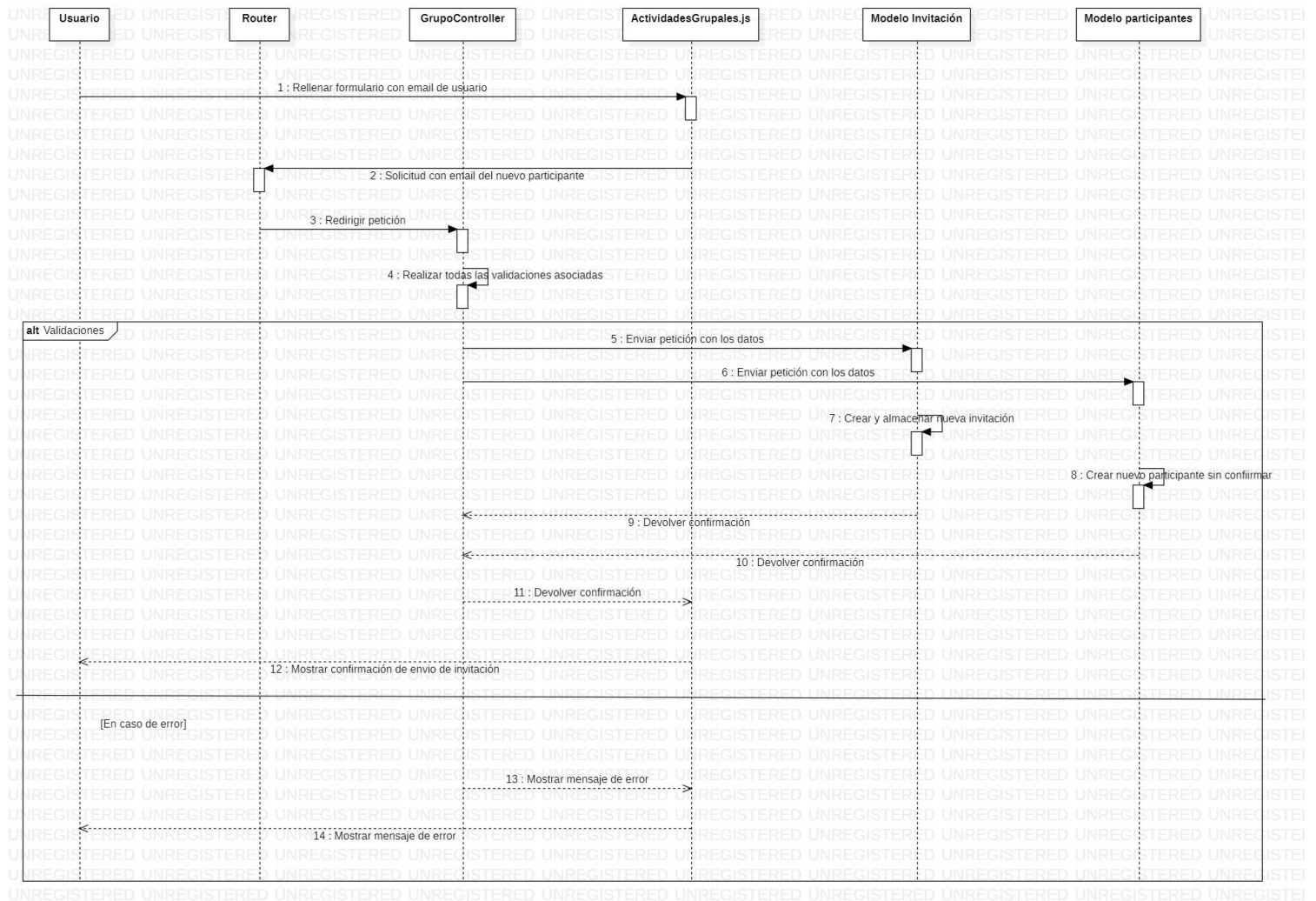
A continuación, se muestra el diagrama de secuencia para almacenar un registro asociado a una tarea.



0.5 Diagrama de secuencia "almacenar registro"

### 4.3.3. Diagrama de secuencia invitar usuario.

A continuación, se muestra el diagrama de secuencia para invitar a un usuario. Se ha omitido la parte para mostrar las actividades grupales y se centra en la propia invitación.



0.6 Diagrama de secuencia "invitar usuario"

### 4.4. Modelado de los datos.

En esta sección se van a determinar las características principales de los datos con los que trabaja la aplicación, así como el modelado lógico de los datos.

Para el desarrollo del proyecto se ha utilizado el sistema de gestión de bases de datos MySQL utilizando la herramienta "MySQL worbench" ya que proporciona un entorno óptimo para el desarrollo de esta aplicación debido a que ha sido implementada utilizando el lenguaje SQL.



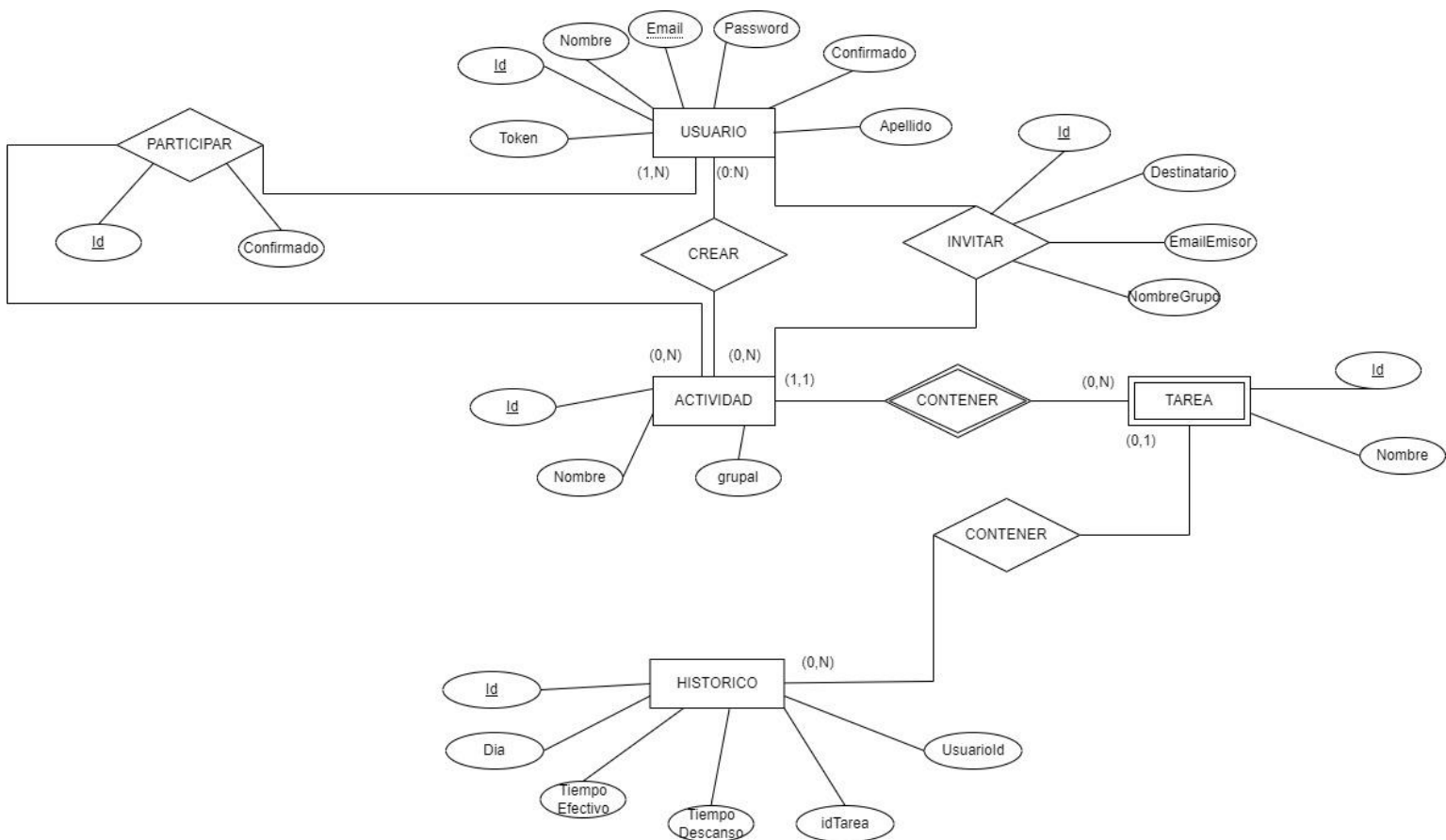
### 4.4.1. Modelo Entidad Relación.

El modelo entidad-relación es un modelo lógico de los datos, que sirve para hacer una representación de los mismo de una manera sencilla.

Su función es la de representar los datos gráficamente y la de diseñar un modelo conceptual que permite ver como los elementos interacciones y se relacionan entre ellos.

Gracias a este modelo se facilita la futura implementación de la base de datos, por lo que es recomendable realizarlo en las etapas tempranas del proyecto.

A continuación, se muestra el modelo Entidad-Relación del proyecto.



0.7 Modelo Entidad-Relación

### 4.4.2. Diccionario de datos.

El diccionario de datos es el conjunto de metadatos que describe las características y estructura de la base de datos, lo que va ayudar a comprender la organización de los datos almacenados.

A continuación, para un mayor entendimiento se define los diccionarios de datos de las entidades del sistema.

| Entidad "actividades" |             |                |          |
|-----------------------|-------------|----------------|----------|
| Atributo              | Tipo        | Único (unique) | Required |
| Id                    | INT         | Sí             | Sí       |
| Nombre                | VARCHAR(45) | No             | No       |
| Propietario           | INT         | No             | No       |
| Grupal                | TINYINT(1)  | No             | No       |

Tabla 30 Diccionario de datos de la entidad "actividades"

| Entidad "histórico" |      |                |          |
|---------------------|------|----------------|----------|
| Atributo            | Tipo | Único (unique) | Required |
| Id                  | INT  | Sí             | Sí       |
| Dia                 | DATE | No             | No       |
| tiempoEfectivo      | TIME | No             | No       |
| tiempoDescanso      | TIME | No             | No       |
| idTarea             | INT  | No             | No       |
| usuarioid           | INT  | No             | No       |

Tabla 31 Diccionario de datos de la entidad "histórico"

| Entidad "invitar" |             |                |          |
|-------------------|-------------|----------------|----------|
| Atributo          | Tipo        | Único (unique) | Required |
| Id                | INT         | Si             | Sí       |
| IdGrupo           | INT         | No             | Sí       |
| Emisor            | INT         | No             | Sí       |
| Destinatario      | INT         | No             | No       |
| emailEmisor       | VARCHAR(45) | No             | No       |
| nombreGrupo       | VARCHAR(45) | No             | No       |

Tabla 32 Diccionario de datos de la entidad "invitar"

| Entidad "participar" |            |                |          |
|----------------------|------------|----------------|----------|
| Atributo             | Tipo       | Único (unique) | Required |
| Id                   | INT        | Sí             | Sí       |
| actividadId          | INT        | No             | Sí       |
| idUsuario            | INT        | No             | Sí       |
| Confirmado           | TINYINT(1) | No             | No       |

Tabla 33 Diccionario de datos de la entidad "participar"

| Entidad "tareas" |             |                |          |
|------------------|-------------|----------------|----------|
| Atributo         | Tipo        | Único (unique) | Required |
| Id               | INT         | Sí             | Sí       |
| idActividad      | INT         | No             | Sí       |
| Nombre           | VARCHAR(45) | No             | No       |

Tabla 34 Diccionario de datos de la entidad "tareas"

| Entidad "invitar" |              |                |          |
|-------------------|--------------|----------------|----------|
| Atributo          | Tipo         | Único (unique) | Required |
| Id                | INT          | Si             | Sí       |
| Nombre            | VARCHAR(45)  | No             | No       |
| Apellido          | VARCHAR(45)  | No             | No       |
| Email             | VARCHAR(45)  | No             | No       |
| password          | VARCHAR(45)  | No             | No       |
| Confirmado        | TINYINT(1)   | No             | No       |
| Token             | VARVCHAR(13) | No             | No       |

Tabla 35 Diccionario de datos de la entidad "invitar"

## 4.5. Diseño de la interfaz de usuario

En esta sección se describirá la estructura de la interfaz de la aplicación junto a sus características principales.

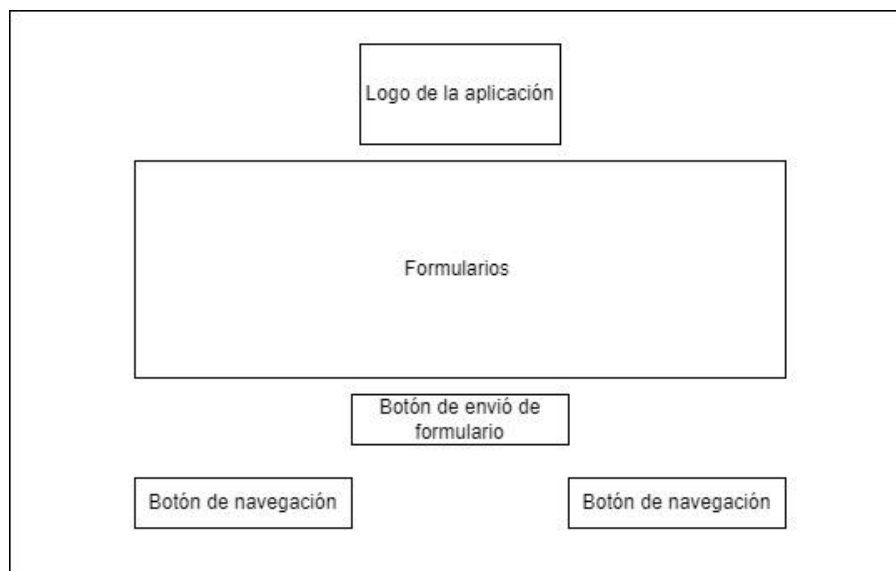
### 4.5.1. Disposición de los elementos

A continuación, se mencionarán las disposiciones de los diferentes elementos en las distintas vistas.

#### **Login:**

Todas las ventanas relacionadas con el sistema de Login siguen la siguiente estructura:

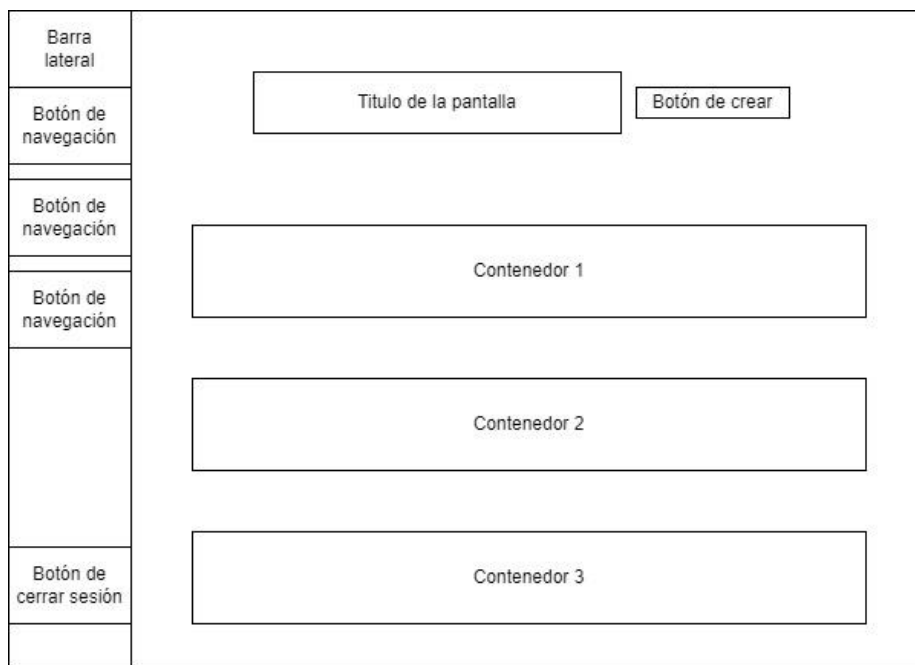
- En la parte superior se muestra el logo de la aplicación en todo momento.
- Debajo del logo se encuentran los distintos inputs de los formularios.
- Debajo de los inputs se encuentra el botón de envío del formulario de manera centrada.
- En la parte inferior se encuentran los dos botones de navegación para viajar entre las distintas vistas del Login.



0.8 Diseño de interfaz vistas de Login

**Vistas de actividades:**

El diseño de interfaz de la figura siguiente se rige tanto en las vistas de actividades como de actividades grupales.



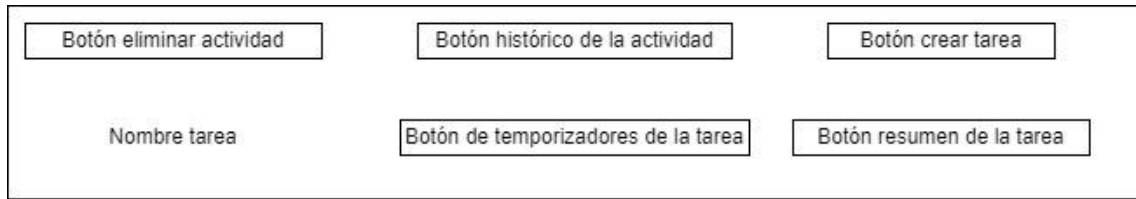
0.9 Diseño de interfaz vistas principales

En este diseño siempre se mantiene la barra lateral a lo largo de toda la aplicación, la cual va a permitir la navegación entre las distintas ventanas y cerrar la sesión del usuario.

En la parte superior se encuentra tanto el nombre de la vista como el botón para crear nuevas actividades ya sean individuales o grupales.

Los contenedores de la parte central serán las distintas actividades con sus correspondientes tareas y opciones.

**Contenedor de una actividad:**



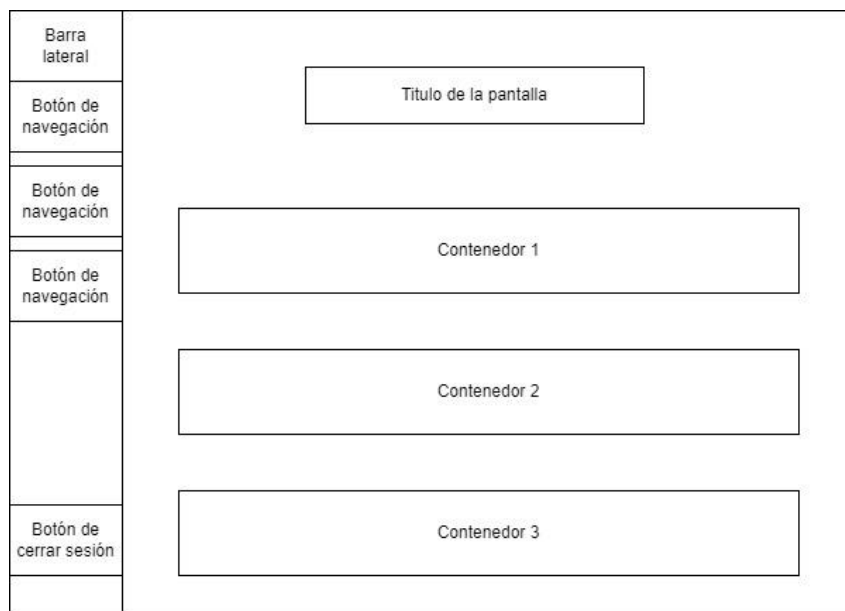
0.10 Diseño de interfaz de una actividad

En cada actividad se sigue la siguiente estructura: en la parte superior se encuentra el botón que redirige al usuario al histórico de la actividad. El botón de eliminar actividad se encuentra en la parte superior izquierda y en la parte superior derecha se encuentra el botón para crear tarea.

Debajo de estos botones se encuentran las distintas tareas, en cada fila se incluye el nombre de la tarea y los botones para redirigir al usuario a los temporizadores de la tarea y el resumen de la misma.

**Vista de invitaciones:**

La vista de las invitaciones mantiene la misma estructura que las vistas de las actividades con la única diferencia de que no contiene el botón de crear en la parte superior.



0.11Diseño de interfaz vista de invitaciones

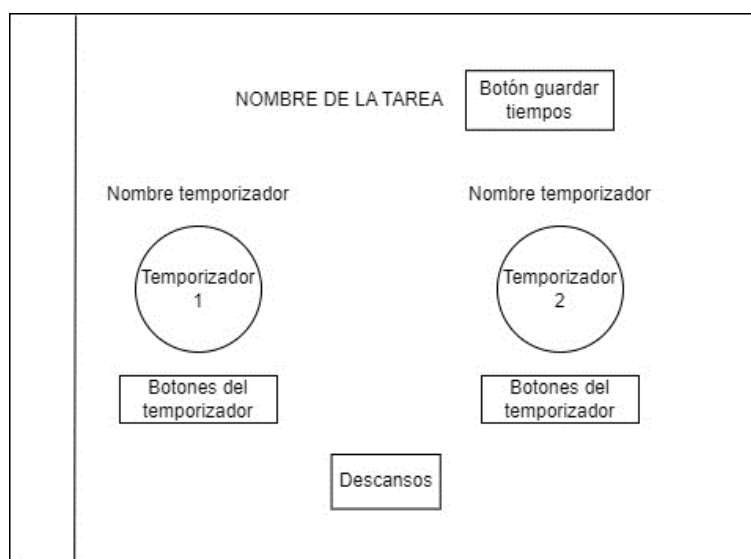
**Vista temporizadores de una tarea:**

En esta vista se sigue el esquema mostrado a lo largo de toda la aplicación, donde el nombre de la tarea o actividad se encuentra en la parte central superior junto a los botones principales de la sección.

En el centro de la vista se encuentran ambos temporizadores junto a sus botones y nombres correspondientes. En la parte inferior de la vista se encuentra un hueco el cual según se vayan iniciando descansos este se llenará de ellos.

En la parte izquierda se muestra la barra lateral de navegación.

Esta distribución se utiliza en las vistas de los temporizadores de tareas grupales e individuales.



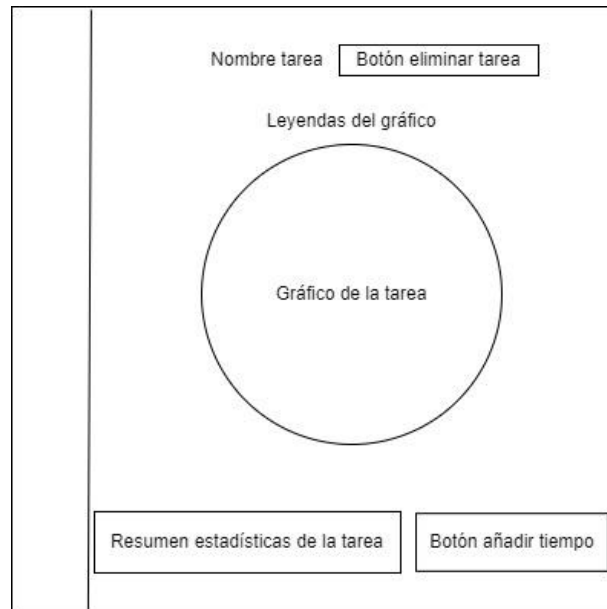
*0.12Diseño de interfaz vista de los temporizadores de una tarea*



### Vista estadísticas de una tarea:

En la parte central superior se encuentra el nombre de la tarea junto al botón de eliminar la tarea.

En el centro de la aplicación visualizamos el contenido principal el cual es el gráfico de la tarea y en la parte inferior el resumen de los datos de la tarea.



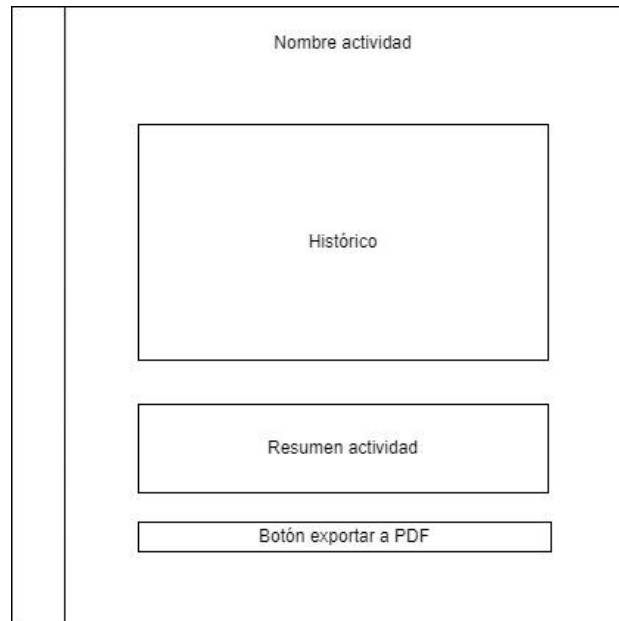
*0.13* Diseño de interfaz vista de estadísticas de una tarea

En las **estadísticas de las tareas grupales** se mantiene el **mismo esquema** con la única diferencia de que el botón de añadir tiempo se encuentra junto al botón de eliminar la tarea y el resumen de la parte inferior ocupa el ancho completo.

### Vista histórico de una de actividad:

En este caso se mantiene el mismo esquema tanto para actividades grupales como individuales.

En la parte superior se encuentra el nombre de la actividad, en la parte central el histórico de la actividad y finalmente en la parte inferior el botón para exportar el histórico.

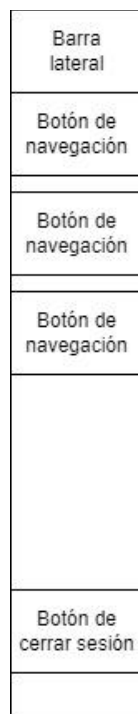


0.14Diseño de interfaz vista del histórico de una actividad

**Barra lateral:**

Como se ha mencionado la barra lateral se encuentra en todas las vistas salvo en las relacionadas con el Login.

En ella en la parte superior se encuentran las distintas opciones de navegación a las vistas principales, y en la parte inferior de la misma se encuentra el botón de cerrar sesión.

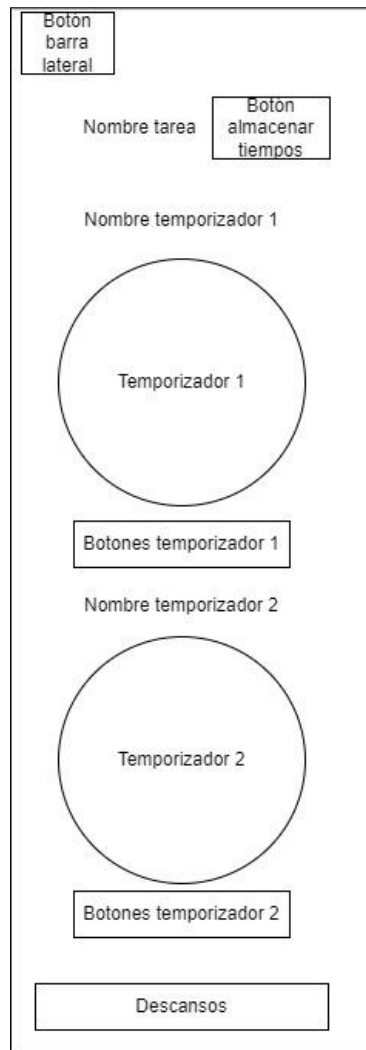


0.15Diseño de interfaz barra lateral

**Vista desde el teléfono móvil:**

Cuando se utiliza la aplicación en el teléfono móvil se muestra exactamente la misma distribución que cuando se visualiza en ordenador.

Las diferencias más notables se encuentran en la vista de los temporizadores en la cual el temporizador de la derecha se mueve a la parte inferior de la vista y los descansos se encuentran debajo de este.



0.16Diseño interfaz vista móvil de la vista de temporizadores de una tarea

La segunda diferencia notable es que la barra lateral se encuentra contraída al máximo por defecto y únicamente aparece el botón para expandirla.

## 4.5.2. Esquemas de colores

A lo largo de toda la aplicación se ha seguido un mismo esquema de color con algunas variaciones:

- Todos los títulos ya sean de actividades, tareas o el mismo logo de la aplicación se encuentran en tonos azules.
- Todos los botones de creación ya sea crear tareas o actividades siguen el mismo color, el cual es una mezcla de azules de tono oscuro asociando la creación de elementos a ese color (Color: #266391).
- En la eliminación de elementos se sigue un rojo granate en todos los botones cuya función es la eliminación de elementos (Color: #c4313d).

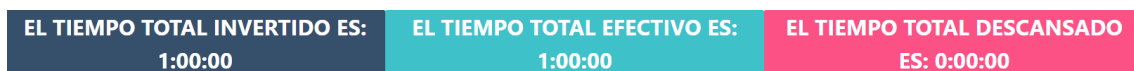


0.17Color de eliminación de elementos



0.18Color de creación de elementos

- En las actividades los botones que permiten navegar hacia las distintas posibilidades que ofrecen se utiliza el siguiente color: #2c9caf.
- Para botones con otras funciones secundarias se ha utilizado el color gris.
- Para distinguir entre tiempos ya sea en los resúmenes del histórico, estadísticas de una tarea, gráficos o los propios temporizadores se sigue el siguiente esquema de colores:
  - En los tiempos totales se utiliza el color #364F6B
  - En el tiempo efectivo: #3FC1C9
  - En los tiempos de descanso y los descansos: #fc5185.



0.19Muestra de colores utilizados en los tiempos

El único lugar donde se rompe la última norma es en las estadísticas de una tarea grupal, donde los colores del gráfico son aleatorios debido a que el número de usuarios que pueden compartir una actividad grupal es variable. En este gráfico la norma implementada es que siempre los colores del gráfico van a ser complementarios entre sí. Por lo tanto, para el resumen de debajo del gráfico se han utilizado colores pastel que combinan con la gran mayoría de colores generados para el gráfico.



# Capítulo 5

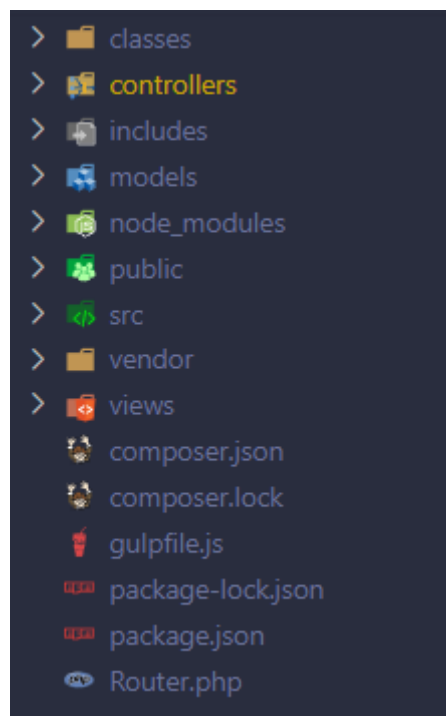
## Implementación

En este capítulo se van a tratar todos los aspectos relacionados sobre cómo se ha construido y desarrollado la aplicación. Se detallarán los aspectos más técnicos relacionados con las tecnologías utilizadas y las funcionalidades implementadas.

### 5.1. Estructura del proyecto.

En esta sección se va a explicar la distribución de los componentes de la aplicación. Para ello se va desgranar las distintas funciones y archivos explicando superficialmente cuál es su función y el porqué de esa estructura.

#### **Estructura completa del proyecto:**

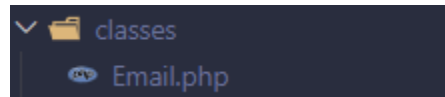


0.1 Carpetas

En la imagen anterior se puede visualizar la estructura de carpetas completa de toda la aplicación.

**classes:**

La función de esta carpeta es la de almacenar clases PHP que van a tener una función auxiliar en el proyecto.

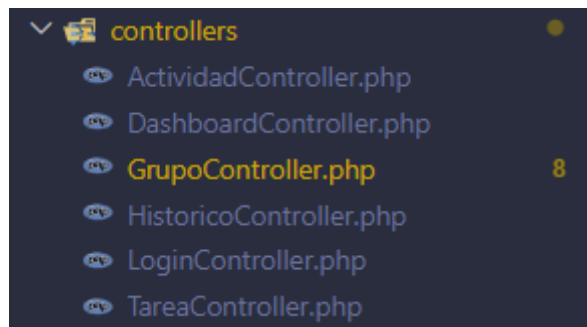
*0.2 Classes*

Dentro de esta carpeta únicamente se encuentra el archivo “Email.php” el cual contiene los métodos para enviar los emails de confirmación de cuenta y de recuperación de contraseña del login.

Para ello se hace uso de la librería “PHPMailer” que va a permitir conectarnos el servidor SMTP y generar y enviar los emails.

**controllers:**

En esta carpeta se almacenan los controladores creados los cuales son un parte fundamental del patrón Model View Controller.

*0.3 Controladores*

-ActividadController: Contiene la lógica necesaria para la creación de nuevas actividades.

-DashboardController: Contiene la lógica para realizar las redirecciones a la vista principal y a la vista principal de grupos.

-GrupoController: Contiene toda la lógica para realizar todas las operaciones relacionadas con las actividades grupales.

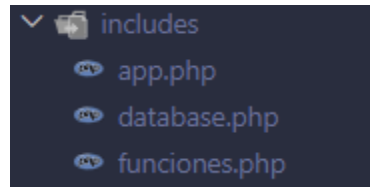
-HistoricoController: Contiene toda la lógica para interactuar con los históricos.

-LoginController: contiene toda la lógica relacionada con el sistema de Login.

-TareaController: Contiene la lógica relacionada con las tareas.

### **includes:**

En esta carpeta se van a incluir todos los archivos que es necesario que estén disponibles en toda la aplicación.



0.4 Carpeta includes

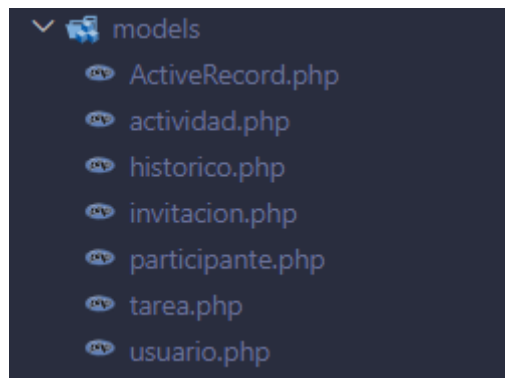
-app: Es el archivo que se va a incluir ya que va a contener el resto de archivos de los includes.

-database: es el archivo en el que se encuentra la lógica necesaria para conectarse al servidor de base de datos.

-funciones: En este archivo se van a encontrar las funciones que son genéricas y se requiere de su uso en cualquier parte y archivo de la aplicación.

### **models:**

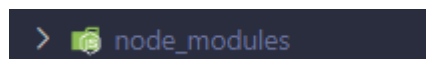
En esta carpeta se encuentran todos los modelos del MVC utilizados por la aplicación.



0.5 Modelos

### **Node modules:**

En esta carpeta se van a almacenar los módulos utilizados por node.js

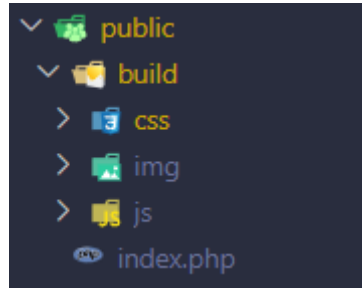


0.6 Módulos de Node



**Public:**

Esta carpeta es una de las más importantes de Framework utilizado, con la explicación del Framework nos detendremos bastante en ella.



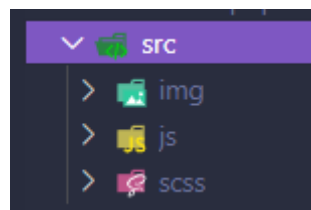
0.7 Carpeta public

En ella se encuentran los archivos “bundle” que son archivos comprimidos que contienen tanto el CSS como el Javascript de la aplicación.

Dentro del archivo index.php se encuentran definidas los Endpoints de la aplicación.

**src:**

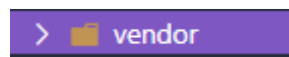
En esta carpeta se encontrarán los elementos que luego Gulp transformará y llevará a la carpeta public. Es la carpeta en la que se va a programar y es donde se encuentran los archivos Javascript, SCSS y las imágenes.



0.8 Carpeta src

**vendor:**

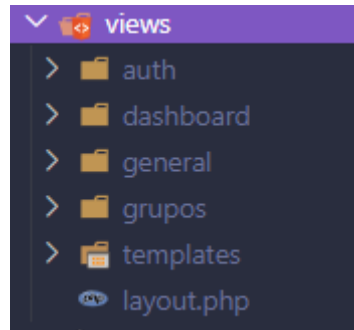
Carpeta de Composer creada automáticamente.



0.9 Vendor composer

## Views:

En esta carpeta se encuentran todas las vistas de la aplicación. Están agrupadas en carpetas en función del “grupo” al que pertenecen.



0.10 Vistas

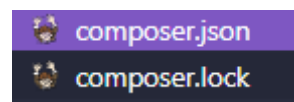
Por ejemplo, en la carpeta “auth” se encuentran todas las vistas relacionadas con el Login y usuarios.

En la carpeta templates se encuentran los elementos HTML que serán comunes a todas las vistas.

El archivo layout.php como su nombre indica es la plantilla en la cual se incluirán el resto de vistas independientemente de cuál sea.

## Composer:

Estos dos archivos forman parte del programa “Composer”.



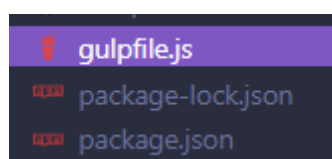
0.11 Archivos Composer

En el archivo composer.json se define el “autoload” que permitirá importar clases de una manera mucho más sencilla a lo largo de la aplicación.

El archivo composer.lock es un archivo generado por Composer que registra las versiones de las dependencias instaladas en el proyecto.

## Gulp:

Archivos de Gulp.



0.12 Archivos Gulp

En el archivo `gulpfile.js` se encuentran definidas las tareas que tiene que realizar gulp.

El archivo `package-lock.json` registra las versiones exactas de todas las dependencias instaladas en un proyecto de Node.js.

El archivo "`package.json`" contiene información esencial como el nombre del proyecto, su versión, una descripción, el autor, las dependencias y otros metadatos relevantes.

## 5.2. Framework creado.

En esta sección se explicará el Framework que se ha creado para la realización del proyecto. Para ello se han utilizado varias tecnologías y funciones propias cuyas que se explicarán más adelante.

### 5.2.1. Patrones de diseño

Para la realización de este proyecto se han utilizado fundamentalmente dos patrones de diseño, estos se muestran a continuación.

#### 5.2.1.1. Active record

Este patrón se basa en el principio de que cada tabla de la base de datos tiene asociada una clase que contiene toda la lógica para manejar los datos almacenados en esa tabla.

Cada clase se mapea directamente a una tabla, estas clases contienen los atributos que son representados en las columnas de la tabla y contiene los métodos necesarios para realizar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) con los datos.

En este patrón se trabaja con objetos en memoria y cualquier operación realizada con ellos se transmite a la base de datos.

Gracias a este patrón obtenemos una interfaz orientada a objetos sencilla con la que interactuar con la que podemos interactuar fácilmente con los datos.

Además, al encapsular la lógica de acceso a datos dentro de clases se simplifica el mantenimiento y legibilidad del código.

Otra razón de peso para la utilización de este patrón es que estas clases van a ser los modelos del patrón MVC, teniendo una gran sinergia entre ambos.

## 5.2.1.2. Modelo vista controlador

El MVC o modelo vista controlador es un patrón de diseño que permite hacer una separación del código en función de sus distintas “responsabilidades” lo que promueve un mayor modularidad del código y que sea más mantenible.

La arquitectura MVC propone la separación de los componentes de una aplicación en tres grupos: modelos, vistas y controladores. Además, describe como interactúan entre ellos, lo que facilita una estructura organizada con bajo acoplamiento entre las distintas capas.

### **Partes del MVC:**

#### Modelo:

Esta capa es la encargada de trabajar con los datos, tiene los mecanismos necesarios para acceder y actualizar la información de la base de datos. Para realizar esta función los modelos contienen todas las funcionalidades para acceder a las tablas de las bases de datos y realizar todas las operaciones CRUD (Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar)).

Debido a la estructura del proyecto se utiliza la librería PDO (PHP Data Objects), lo que va a proporcionar una interfaz orientada a objetos para acceder y gestionar los datos.

#### Vistas:

Los componentes de la vista son los encargados de generar la interfaz de la aplicación.

Componen las pantallas, páginas o cualquier tipo de interfaz utilizable y mostrada al usuario y en ellas se muestran los estados del modelo. Las vistas deben contener los elementos que permitan al usuario interactuar con el sistema, así como enviar información o invocar acciones.

#### Controladores:

Los controladores contienen la lógica necesaria para responder las acciones que se realizan en la aplicación. Es la capa que sirve de enlace entre las vistas y los modelos coordinando las acciones correspondientes. El controlador interpreta las acciones del usuario y decide cuáles son las operaciones que debe realizar el modelo para satisfacer esas acciones y posteriormente se encarga de actualizar la vista con los datos actualizados provenientes del modelo.

## 5.3. Funcionamiento del Framework

Uno de los factores a destacar en esta aplicación es que para la realización de la misma se ha creado un Framework propio.

Un Framework es un conjunto de herramientas bibliotecas y pautas de desarrollo que proporcionan una estructura y funcionalidades predefinidas para facilitar el desarrollo de aplicaciones.

En secciones anteriores se ha explicado cuales son los patrones arquitectónicos utilizados por la aplicación y por el Framework, así como las tecnologías utilizadas.

En esta sección se hablará más concretamente de las funciones que cumplen en la aplicación y de cuál es su enfoque.

### 5.3.1. Funcionamiento en el desarrollo

Como se mencionó en la estructura de carpetas, la carpeta public contiene el conjunto de archivos que son accesibles de manera pública. Entre ellos se encuentran los archivos Javascript, CSS y las imágenes.

Estos archivos Javascript y CSS de la carpeta public son archivos minificados de los archivos Javascript y SCSS que se encuentran en la carpeta “src”.

Esto se consigue gracias a Gulp y a conjunto de task que se han definido para que realicen esa función.

Con esto se consigue que el desarrollador tenga archivos “sourcemap” en la carpeta “src” facilitando el desarrollo y entendimiento de estos archivos, así como obtener archivos minificados que ocupen menos espacio para subirlos a producción.

Un archivo sourcemap es el archivo generado junto al archivo minificado el cual mapea el código fuente original con el código minificado.

Además, como se ha mencionado en la carpeta “src” se trabaja con código SCSS debido a que permite estructurar el código CSS de una manera más sencilla además de facilitar la reutilización de código.

Para trabajar con SCSS se necesita un conversor de código CSS, de esto se encarga Gulp en otro task que se ha definido.

En este task Gulp, genera los archivos minificados, con los prefijos necesarios de CSS para que sean entendibles por los distintos navegadores en la carpeta public de los archivos SCSS que se encuentran en la carpeta src. También genera los archivos sourcemap en esta última carpeta.

Para la gestión de librerías PHP se utiliza Composer lo que facilita la portabilidad de las librerías utilizadas y la actualización y portabilidad de las mismas.

Pero otra funcionalidad muy importante de Composer en el proyecto es la función “autoload” que permite la carga automática de clases en el proyecto.

Con esto se consigue eliminar la necesidad de incluir manualmente los archivos de clase por lo que cada vez que se quieran usar estos archivos no es necesario utilizar las funciones ‘require’ o ‘include’ las cuales necesita la ruta completa del archivo y bastaría con únicamente incluir ‘use’ lo que facilita enormemente el desarrollo. Por ejemplo: use Model\Actividad.

Para cumplimentar la funcionalidad anterior se utiliza la función de PHP ‘namespace’ que permite realizar distinciones entre clases, aunque estas tengan el mismo nombre.

Como ya se mencionó en la sección [5.1. Estructura del proyecto.] otra de las ayudas que proporciona el Framework a la hora del desarrollo es la de otorgar una separación clara de archivos, así como proporcionar unas guías para la creación de nuevas funcionalidades.

Para añadir una nueva funcionalidad desde cero habría si no existe, que crear el modelo asociado a la tabla de la base de datos. Una vez creado el modelo se crearía el controlador asociado a ese modelo, así como los métodos que implementen la lógica de la funcionalidad que se quiere crear. Simultáneamente se crearían las vistas que van presentar esta nueva funcionalidad y finalmente se crearía el nuevo Endpoint que llame al método correspondiente del controlador el cual será el encargado de transmitir los datos hacia y desde el modelo, seleccionar la vista correspondiente y presentarla al usuario.

## 5.3.2. Funcionamiento general.

En esta sección se explicará cómo funciona el Framework, así como las funciones principales. Anteriormente hemos explicado cuáles son sus beneficios en el desarrollo de aplicaciones, pero en esta sección se mencionará más detalladamente funciones y funcionalidades concretas, así como el funcionamiento general cuando el usuario interactúa con el sistema.

### 5.3.2.1. Tratamiento de una petición

En la carpeta public como ya se mencionó se encuentran todos los archivos accesibles por los usuarios. Aquí también se encuentra el archivo ‘index.php’ este archivo es una parte fundamental y es el encargado de gestionar todas las peticiones HTTP del sistema. Contiene 3 características fundamentales.

En primer lugar, incluye el archivo ‘app.php’ de la carpeta ‘includes’ el cual contiene todas aquellas funciones que queremos que sean accesibles desde cualquier lugar de la aplicación y la conexión a la base de datos.

En segundo lugar, en este archivo se declaran los Endpoints, que son todas aquellas URLs específicas dentro de la aplicación a la cuales se les puede realizar una solicitud HTTP para interactuar con la aplicación y recibir una respuesta.

La estructura de estas rutas es la siguiente:

```
$router->get('/', [LoginController::class, 'login']);  
$router->post('/', [LoginController::class, 'login']);
```

#### 0.13 Endpoints de la aplicación

Primero, se crea un objeto de tipo router del cual se explicará su funcionamiento a continuación. Se define el tipo de petición HTTP que se va a realizar para ello se utilizan los métodos del router 'get' en el cual se incluirán todos aquellos Endpoints que envíen solicitudes de tipo get y el método 'post' en el cual se incluirán todos aquellos Endpoints de que envíen solicitudes de tipo 'post'.

Estos métodos reciben 2 parámetros.

- El primer parámetro indica la URL de la petición.
- El segundo parámetro es un array con 2 valores:
  - El primer valor indica al controlador al cual se debe redirigir la petición asociado a esa URL y método de petición.
  - El segundo valor indica el nombre de la función a la cual se ha de llamar dentro de ese controlador asociado.

Según este ejemplo cuando un usuario se encuentre en la URL '/' enviará una petición de tipo GET la cual llamará al método 'login' del controlador 'LoginController' que contendrá la lógica necesaria para tratar los datos y presentar las vistas correspondientes.

Finalmente, una vez definidos todos los Endpoints se llama al método 'comprobarRutas' del router.

Este método comprueba cual es la URL en la que se encuentra el usuario, y el método de la petición HTTP.

Si el método es de tipo 'GET' comprobará en el array que se ha creado gracias al método get() de la misma clase si existe esa ruta, y si es de tipo 'POST' realizará la misma comprobación con el array que contiene las rutas de tipo 'POST'.

Si esas rutas no existen se mostrará al usuario una pantalla con el error 404 indicando que la página no existe.

En caso de que exista la ruta acorde a la solicitud, con la función de PHP 'call\_user\_func()' se llamará al controlador y métodos asociados al Endpoint. Para ello recibe como parámetro el array definido en el Endpoint que como se mencionó contiene el controlador y el método que se deben de llamar.

Una vez llamado al controlador este es el encargado de interactuar con los modelos y las vistas.

Para interactuar con los modelos los controladores crean objetos de estos modelos y llaman a funciones de los mismos. Luego actualizados los objetos los modelos con sus correspondientes métodos son los encargados de actualizar los datos de la base de datos.

Como se mencionó explicando el patrón Active Record existe una clase correspondiente a cada tabla en la base de datos, pero además existe una clase auxiliar llamada 'ActiveRecord' la cual va a ser el padre del resto de modelos y estos extenderán esta clase.

Esta clase contiene métodos generales y comunes a todos los modelos, así los modelos salvo casos específicos únicamente se encargan de crear los objetos, contener los atributos correspondientes a sus tablas de la base de datos y de contener métodos para realizar las validaciones a los datos.

En resumen, en la clase Active record se encuentran métodos genéricos que contienen las sentencias para interactuar directamente con la base de datos y el resto de modelos heredarán estos métodos y únicamente deberán ejecutarlos sin realizar cambios, lo que incrementa en gran medida la escalabilidad de la aplicación.

Además, como se explicó en el patrón Active Record cada tupla de la base de datos va a generar un objeto de la clase correspondiente para ello la clase ActiveRecord también contiene los métodos para generar un objeto de cada tupla leída de la base de datos.

Para mostrar las vistas `call_user_func()` recibe un segundo parámetro que es un objeto de la clase router. Esto es debido a que las vistas se mostrarán al usuario gracias al método `render()` de la clase router.

Este método `render` recibe dos parámetros:

- La dirección de la vista que se ha de renderizar.
- Un array con los datos que se quieren pasar a la vista.

Además, recorre el array de datos y crea una variable con para cada dato facilitando su acceso posterior.

Posteriormente inicia un almacenamiento en memoria para incluir la vista y los datos en la variable `$contenido` la cual se incluirá en el archivo 'layout.php' de las vistas, el cual es el template de las mismas y se muestra ese contenido al usuario.

## 5.4. Dificultades en el desarrollo.

Esta última sección del capítulo se dedicará a comentar de manera general cuales han sido las mayores dificultades que han surgido a lo largo de la creación del proyecto y a explicar cuál ha sido su evolución a lo largo del mismo.

Las principales dificultades surgen del desconocimiento previo de algunas de las tecnologías.

Antes de enfrentarme a este proyecto desconocía cual era el funcionamiento de JavaScript, como se implementaba y que era capaz de realizar. Uno de los objetivos en la implementación del proyecto era que este tuviese una buena interfaz y que no fuese 'pesada'.

Para lograr esto tenía que evitar en gran medida que continuamente se estuviesen recargando las páginas cuando se hacía 'click' o se realizaban cambios con los datos.



Sabía que este comportamiento lo dotaba JavaScript por lo tanto antes de la realización del proyecto tuve que realizar una fase de aprendizaje del lenguaje.

Otro de los problemas que surgieron derivados de este mismo era como conectar PHP con JavaScript, y la manera más tradicional de hacerlo era a través de APIS creadas en PHP las cuales recibirían peticiones de JavaScript. Lo que me llevo a tener que realizar cambios en algunas partes de la estructura del proyecto.

Finalmente estoy muy contento con los resultados y gran parte del “front” del proyecto está realizado con JavaScript y APIS lo que me ha aportado grandes conocimientos que se extrapolan a otro tipo de proyectos y tecnologías.

Un segundo problema surge de que además de tener querer tener una buena interfaz mi idea inicial de back-end era conseguir un código escalable, fácilmente mantenible y de calidad. Esto me llevó a la creación de un Framework propio. Gran parte del aprendizaje para realizarlo proviene de la realización del curso previo en el que también aprendí JavaScript.

En este apartado también estoy muy contento con los resultados ya que me ha proporcionado un gran conocimiento de la sinergia de los distintos patrones entre sí, y como Frameworks del mercado como Laravel funcionan por lo que sería muy fácil adaptarme a ellos.



# Capítulo 6

## Pruebas

En este capítulo se mencionan las pruebas realizadas en el sistema con el objetivo de garantizar un funcionamiento correcto y óptimo frente a situaciones inesperadas.

Para ello, a continuación, se muestra algunas de las pruebas más representativas realizadas junto a los resultados de la prueba.

### 6.1. Pruebas de caja blanca

Las pruebas de caja blanca son un conjunto de técnicas software centradas en examinar el código interno de un programa. En este conjunto de pruebas el tester tiene acceso al código fuente el cual es utilizado para realizar las pruebas.

El objetivo principal de estas pruebas es evaluar la lógica del software y asegurarse de que tanto la lógica como condiciones se prueben de manera exhaustiva. Para ello se ha de verificar la ejecución de todas las instrucciones decisiones y bucles además de las excepciones y errores.

Estas son algunas de las pruebas de caja blanca que se han realizado en el proyecto:

- Prueba de código: se ha comprobado que todas las líneas de código se han ejecutado al menos una vez
- Prueba de conexión con la base de datos: Se ha verificado la correcta conexión con la base de datos
- Prueba de inserción de datos: Se ha comprobado que todos los datos introducidos en los distintos formularios se ingresan adecuadamente en la base de datos
- Prueba de eliminación de datos: Se ha comprobado que todos los datos se eliminen sin ocasionar problemas
- Prueba de actualización de datos: Se ha comprobado que todos los datos se actualicen correctamente en la base de datos
- Pruebas de consulta de base de datos: se ha comprobado que las peticiones retornen los datos esperados.
- Pruebas de datos de APIS: se ha comprobado que las APIS retornan los datos esperados.
- Pruebas de seguridad de APIS: Se ha comprobado que las APIS solo devuelven los datos que deben y solo ingresan los datos que deben.
- Prueba inyección JavaScript: Se ha comprobado que la aplicación no sea vulnerable contra XSS injection.
- Prueba inyección SQL: Se ha comprobado que la aplicación no sea vulnerable contra inyecciones SQL

- Pruebas de autenticación: Se ha comprobado que tanto el registro como el login funcionen correctamente.
- Pruebas de autenticación: Se ha comprobado que los usuarios no logueados no puedan acceder a determinadas páginas.
- Pruebas de manejo de errores: Se ha comprobado que los errores de la aplicación se muestren de una manera correcta indicando el tipo de error al usuario.

## 6.2. Pruebas de caja negra

Las pruebas de caja negra son un conjunto de pruebas las cuales tienen como objetivo verificar las funcionalidades de la aplicación sin tener en cuenta el código interno ni los aspectos referentes a la implementación.

Estas pruebas se llevan a cabo con desconocimiento de la implementación enfocándose en la entrada y las salidas que originan la aplicación, teniendo como base sus requisitos.

Por lo tanto, este conjunto de pruebas está relacionado con la validación de los recursos funcionales de la aplicación.

A continuación, se muestran algunas de las pruebas de caja negra más representativas realizadas en este proyecto.

| <b>PCN-01: Registro de un usuario</b> |   |
|---------------------------------------|---|
| <b>Propósito</b>                      | Comprobar que el registro de usuarios se hace correctamente y el email de confirmación de cuenta llega al correo indicado.  |
| <b>Prerrequisitos</b>                 | Ninguno   |
| <b>Datos de entrada</b>               | Email y datos del usuario   |
| <b>Resultado esperado</b>             | Al introducir todos los datos de forma correcta por el usuario, que se envíe un email de confirmación de cuenta al email introducido y que se registre al usuario |
| <b>Resultado obtenido</b>             | Correcto  |

Tabla 36 Prueba de caja negra "registro de un usuario"

| <b>PCN-02: Iniciar sesión</b> |  |
|-------------------------------|--|
| <b>Propósito</b>              | Comprobar que un usuario puede iniciar sesión en la aplicación correctamente   |
| <b>Prerrequisitos</b>         | El usuario debe estar registrado   |
| <b>Datos de entrada</b>       | Email y contraseña del usuario   |
| <b>Resultado esperado</b>     | Al introducir el usuario la contraseña y el email de forma correcta se le redirigirá a la página principal de la aplicación. |
| <b>Resultado obtenido</b>     | Correcto   |

Tabla 37 Prueba de caja negra "Iniciar sesión"

| <b>PCN-03: Recuperar contraseña</b> |  |
|-------------------------------------|--|
| <b>Propósito</b>                    | Comprobar el correcto funcionamiento para recuperar la contraseña de un usuario    |
| <b>Prerrequisitos</b>               | El usuario debe estar registrado en la aplicación                                  |
| <b>Datos de entrada</b>             | Email al que se debe enviar el correo de recuperación                              |
| <b>Resultado esperado</b>           | Al introducir el email se enviará un correo con el enlace para recuperar la cuenta |
| <b>Resultado obtenido</b>           | Correcto   |

Tabla 38 Prueba de caja negra "Recuperar contraseña"

| <b>PCN-04: Crear actividad</b> |   |
|--------------------------------|---|
| <b>Propósito</b>               | Comprobar el correcto funcionamiento de la creación de actividades  |
| <b>Prerrequisitos</b>          | El usuario debe estar registrado  |
| <b>Datos de entrada</b>        | Nombre de la actividad  |
| <b>Resultado esperado</b>      | Se creará una nueva actividad con el formato correcto y el nombre introducido por el usuario en caso de ser este válido |
| <b>Resultado obtenido</b>      | Correcto  |

Tabla 39 Prueba de caja negra "Crear actividad"

| <b>PCN-05: Eliminar actividad</b> |  |
|-----------------------------------|--|
| <b>Propósito</b>                  | Comprobar el correcto funcionamiento de la eliminación de actividades y de las tareas asociadas              |
| <b>Prerrequisitos</b>             | La actividad debe estar creada de antemano   |
| <b>Datos de entrada</b>           | Ninguno  |
| <b>Resultado esperado</b>         | Al pulsar en el botón de eliminar de una actividad del usuario, debe eliminarse junto a sus tareas asociadas |
| <b>Resultado obtenido</b>         | Correcto   |

Tabla 40 Prueba de caja negra "Eliminar actividad"

| <b>PCN-06: Crear tarea</b> |  |
|----------------------------|--|
| <b>Propósito</b>           | Comprobar el correcto funcionamiento de la creación de tareas                        |
| <b>Prerrequisitos</b>      | Haber creado la actividad que contendrá la tarea.                                    |
| <b>Datos de entrada</b>    | Nombre de la tarea   |
| <b>Resultado esperado</b>  | Si la tarea tiene un nombre con un formato adecuado la tarea se creará correctamente |
| <b>Resultado obtenido</b>  | Correcto   |

Tabla 41 Prueba de caja negra "Crear tarea"

| <b>PCN-06: Eliminación de tarea</b> |   |
|-------------------------------------|---|
| <b>Propósito</b>                    | Comprobar que las tareas se eliminan adecuadamente  |
| <b>Prerrequisitos</b>               | Haber creado una actividad y la correspondiente tarea asociada                                  |
| <b>Datos de entrada</b>             | Ninguno   |
| <b>Resultado esperado</b>           | Al hacer 'click' en el botón de eliminar y tras confirmar que se elimine la tarea correctamente |
| <b>Resultado obtenido</b>           | Correcto  |

Tabla 42 Prueba de caja negra "Eliminación de tarea"

| <b>PCN-07: Añadir tiempo a una tarea</b> |  |
|--|--|
| <b>Propósito</b>                         | Comprobar que los tiempos registros asociados a las tareas junto a sus registros se almacenan correctamente  |
| <b>Prerrequisitos</b>                    | Tener creada la tarea  |
| <b>Datos de entrada</b>                  | Ninguno  |
| <b>Resultado esperado</b>                | Al pulsar en el botón para almacenar los tiempos de los temporizadores o al añadir más tiempo para una actividad que se almacenen los tiempos junto al resto de datos asociados a la tarea |
| <b>Resultado obtenido</b>                | Correcto   |

Tabla 43 Prueba de caja negra "Añadir tiempo a una tarea"

| <b>PCN-08: Visualizar gráfico tarea</b> |   |
|---|---|
| <b>Propósito</b>                        | Comprobar la correcta visualización del gráfico de las tareas en función de los registros almacenados de la misma         |
| <b>Prerrequisitos</b>                   | Haber creado una tarea y haber guardado registros de la misma   |
| <b>Datos de entrada</b>                 | Ninguno   |
| <b>Resultado esperado</b>               | Al hacer 'click' para visualizar el diagrama de una tarea que se muestre el gráfico en función de los tiempos almacenados |
| <b>Resultado obtenido</b>               | Correcto  |

Tabla 44 Prueba de caja negra "visualizar grafico tarea"

| <b>PCN-09: Visualizar histórico actividad</b> |   |
|---|---|
| <b>Propósito</b>                              | Comprobar que la visualización del histórico es correcta  |
| <b>Prerrequisitos</b>                         | Haber creado una actividad, tareas y haber almacenado registros de las tareas   |
| <b>Datos de entrada</b>                       | Ninguno   |
| <b>Resultado esperado</b>                     | Al hacer 'click' en el histórico de una actividad que se muestre correctamente el histórico en función de los registros almacenados |
| <b>Resultado obtenido</b>                     | Correcto  |

Tabla 45 Prueba de caja negra "visualizar histórico actividad"

| <b>PCN-10: Exportar histórico a PDF</b> |   |
|---|---|
| <b>Propósito</b>                        | Comprobar que el histórico se exporta correctamente   |
| <b>Prerrequisitos</b>                   | Haber almacenado algún registro de las tareas asociadas a la actividad.   |
| <b>Datos de entrada</b>                 | Ninguno   |
| <b>Resultado esperado</b>               | Al hacer 'click' en el botón para exportar que se genere un PDF almacenable por el usuario con el contenido del histórico |
| <b>Resultado obtenido</b>               | Correcto  |

Tabla 46 Prueba de caja negra "exportar histórico a PDF"

| <b>PCN-11: Invitar usuario</b> |  |
|--------------------------------|--|
| <b>Propósito</b>               | Comprobar que las invitaciones de usuarios a actividades grupales funcionan correctamente  |
| <b>Prerrequisitos</b>          | Ser participante de una actividad grupal   |
| <b>Datos de entrada</b>        | Email del usuario a invitar  |
| <b>Resultado esperado</b>      | Al invitar un usuario y escribir correctamente el email de un usuario registrado en la aplicación le llegará una invitación para poder formar parte de la actividad grupal |
| <b>Resultado obtenido</b>      | Correcto   |

Tabla 47 Prueba de caja negra "invitar usuario"

| <b>PCN-12: Aceptar invitación</b> |  |
|-----------------------------------|--|
| <b>Propósito</b>                  | Comprobar el correcto funcionamiento de las actividades  |
| <b>Prerrequisitos</b>             | Tener invitaciones sin aceptar   |
| <b>Datos de entrada</b>           | Ninguno  |
| <b>Resultado esperado</b>         | Al hacer 'click' en el botón de aceptar de una invitación el usuario formará parte de la actividad grupal y podrá interactuar con ella |
| <b>Resultado obtenido</b>         | Correcto   |

Tabla 48 Prueba de caja negra "aceptar invitación"





# Capítulo 7

## Conclusiones

Esta sección está dedicada a modo de retrospectiva para comentar tanto las experiencias personales obtenidas tras el trascurso de la realización del proyecto, como el grado de satisfacción con el mismo y las posibles mejoras que se podrían implementar.

Una vez finalizado el proyecto se confirma que los objetivos iniciales se han cumplido de una manera satisfactoria. Uno de los puntos de los cuales estoy más orgulloso es que al plantear el proyecto inicialmente definí superficialmente algunas de las funcionalidades de la aplicación actual, las cuales se plantearon como “casi imposibles” por tiempos, dificultad y desconocimiento de las tecnologías, pero finalmente todas esas funcionalidades están implementadas tal cual se definieron además del resto de funcionalidades planteadas.

Otro aspecto con el que estoy muy contento, además del resultado final obtenido en el proyecto, es el aprendizaje que he obtenido con la realización del mismo. He aprendido tecnologías nuevas de las cuales partía con un total desconocimiento. Hoy en día no puedo considerarme un experto ni mucho menos, pero sí que tengo unos conocimientos bastante sólidos de las mismas.

En resumen, puedo afirmar que la realización del proyecto ha sido muy satisfactoria. El hecho de no saber inicialmente como iba a realizar el proyecto y que finalmente los bocetos que se plantearon, estén implementados, me aporta una gran seguridad. Y a diferencia del principio del proyecto donde no sabía cómo iba a realizar las funcionalidades ahora se me ocurren mejoras para la aplicación que sé que con tiempo soy totalmente capaz de llevar a cabo.

### 7.1. Líneas de trabajo futuras

En esta sección se presentan un conjunto de funcionalidades que se podrían implementar a futuro en caso de querer ampliar la aplicación.

- **Calendario:** una posible funcionalidad que podría ser interesantes sería mostrar en el calendario en que tareas se trabajó y en el momento en el que se pulse cualquiera de estos registros del calendario se redirija al usuario al historial de la actividad.
- **Establecimiento de objetivos:** Una funcionalidad a implementar podría ser el establecimiento de objetivos por parte del usuario. Antes de comenzar una “sesión” es decir, antes de comenzar con la zona de temporizadores que miden el tiempo efectivo y tiempo de descanso cuando se usa la aplicación, se podrían

establecer una serie de objetivos o porcentajes de efectividad y tiempo por el usuario y registrar si el usuario va cumpliendo con sus objetivos completándolo con anotaciones para que pueda indicar el porqué de esos resultados.

- **Modificación de perfil:** actualmente esta funcionalidad no se ha implementado porque por el propio carácter de la aplicación no se ha considerado necesario, pero en un futuro se podrían implementar funcionalidades para añadir fotos de perfil o modificar el nombre de usuario.
- **Administradores de aplicaciones grupales:** actualmente el creador de la actividad grupal es el único que puede eliminar tanto la actividad como actividades que se haya creado por los distintos participantes cumpliendo el papel de administrador. En un futuro se podría implementar niveles de permisos como que el administrador pueda hacer otros administradores o solo determinadas personas puedan invitar a otros usuarios.
- **Abandonar actividad grupal:** Una futura implementación y mejora es que los usuarios puedan abandonar actividades grupales en cualquier momento para dejar de formar parte de ellas sin eliminar la actividad.

Estas son solo algunas sugerencias aparte de las implementaciones técnicas como mayores medidas de seguridad o pequeños cambios de interfaz. Antes de implementar cualquier tipo de mejora se buscaría conocer la opinión de los usuarios y en función de esas opiniones implementar unas funcionalidades u otras.



# Parte III

## Manuales de usuario



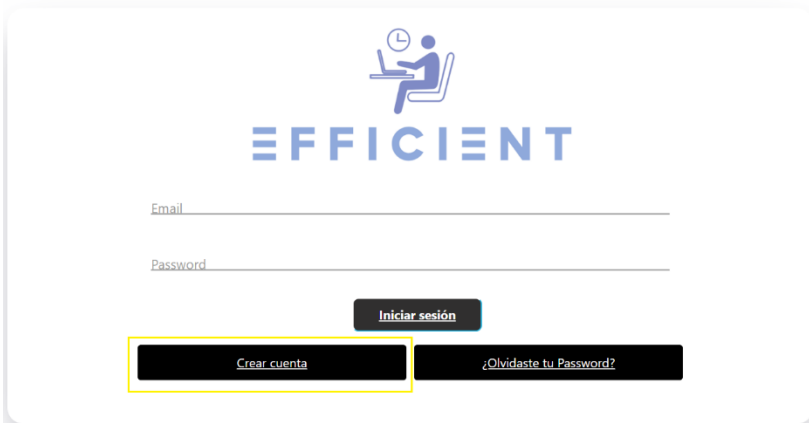
# Capítulo 8

## Manual de Usuario

En esta sección se definirá el manual de usuario para la utilización de la aplicación.

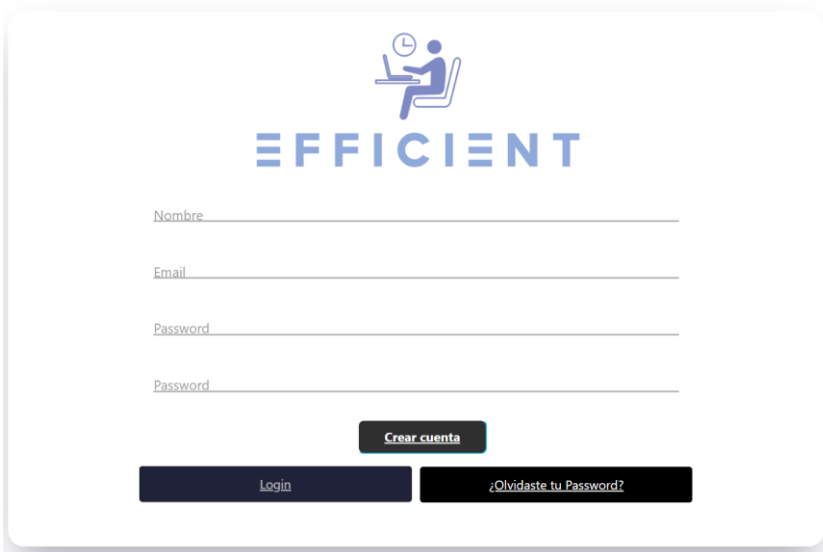
### **Creación de cuenta:**

En primer lugar, para utilizar la aplicación el usuario debe registrarse en la aplicación para ello se ha de rellenar el siguiente formulario:



The screenshot shows the login interface for the Efficient application. At the top center is the Efficient logo, which consists of a blue icon of a person sitting at a desk with a clock above their head, and the word 'EFFICIENT' in blue capital letters below it. Below the logo are two input fields: 'Email' and 'Password'. Underneath these fields is a dark blue button labeled 'Iniciar sesión'. At the bottom of the form, there are two dark blue buttons: 'Crear cuenta' (highlighted with a yellow border) and '¿Olvidaste tu Password?'.

0.1 Formulario de Login de Efficient



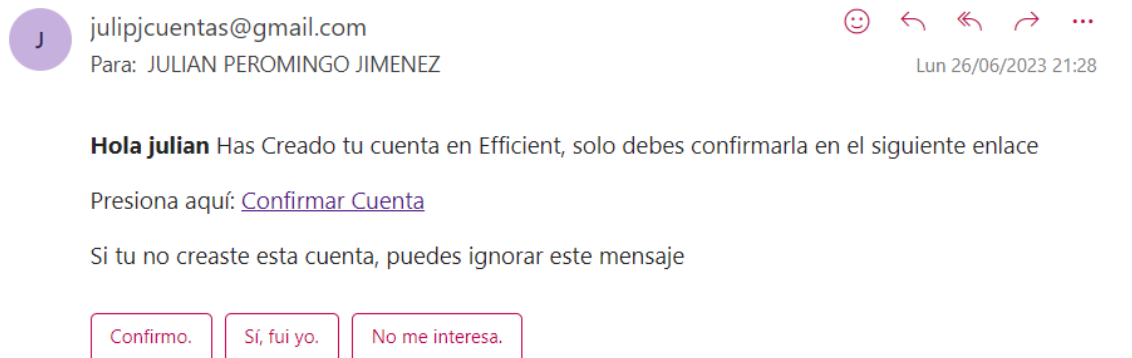
The screenshot shows the account creation interface for the Efficient application. At the top center is the Efficient logo, which consists of a blue icon of a person sitting at a desk with a clock above their head, and the word 'EFFICIENT' in blue capital letters below it. Below the logo are four input fields: 'Nombre', 'Email', 'Password', and a second 'Password' field. Underneath these fields is a dark blue button labeled 'Crear cuenta'. At the bottom of the form, there are two dark blue buttons: 'Login' and '¿Olvidaste tu Password?'.

0.2 Formulario de creación de cuenta Efficient

En él se debe escribir el nombre de usuario, un email existente y una contraseña que se ha de repetir.

Una vez rellenados estos datos llegará un email de confirmación al email escrito en el formulario.

Tras recibir un email con el formato como el siguiente se ha de pulsar en el enlace de “confirmar cuenta”.



0.3 Email de confirmación de cuenta de Efficient

Tras pulsar en el enlace se redirigirá al usuario a la aplicación y aparecerá el siguiente mensaje:



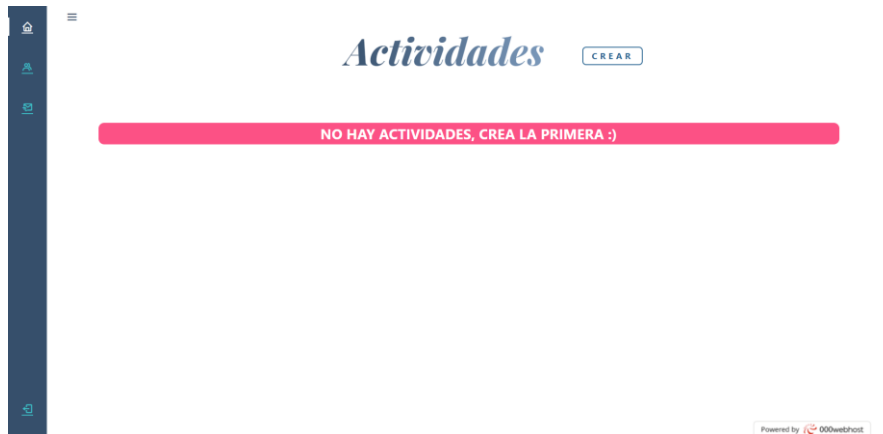
0.4 Vista de confirmación de cuenta Efficient

### **Utilización de la aplicación:**

Una vez confirmada la cuenta tras pulsar en el botón de “iniciar sesión” se redirigirá al usuario a la pantalla de Login. En ella tras rellenar el email y contraseña con los cuales se registró podrá acceder a la aplicación.

Tras haber iniciado sesión aparecerá la siguiente vista:





0.5 Vista principal de Efficient

Para crear una nueva actividad ha de pulsarse en el botón “crear” de la parte superior derecha. Tras pulsar aparecerá un pequeño formulario para introducir el nombre de la nueva actividad

0.6 Formulario de creación de una nueva actividad

Una vez introducido el nombre y tras pulsar en el botón de “Crear actividad” la actividad se creará correctamente.

Para empezar a trabajar con la actividad han de crearse tareas asociadas a la actividad, para ello se ha de pulsar en el botón de “+” de la actividad:

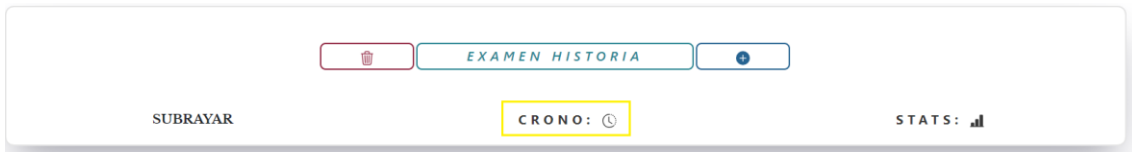


0.7 Actividad de Efficient

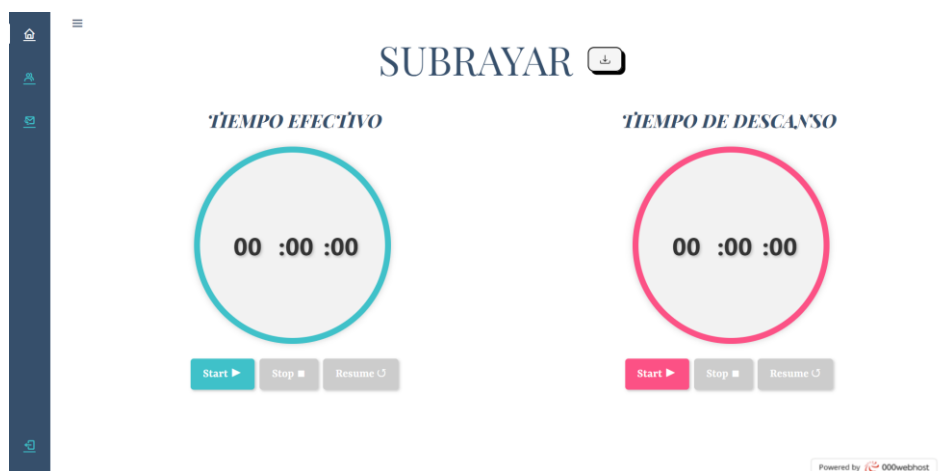
Al pulsarle aparecerá un formulario similar al de crear actividad para crear el nombre de la tarea.

Tras crear la tarea tendremos varias opciones:

Al pulsar en crono nos llevará a la zona de temporizadores para medir los tiempos



0.8 Actividad de Efficient con botón de los temporizadores resaltado



0.9 Temporizadores de una tarea

Gracias a estos temporizadores podremos medir el tiempo efectivo y de descanso que decíamos a las tareas. Para utilizarlos únicamente se ha de pulsar en los botones correspondientes a los temporizadores. Cuando se inicia uno el otro se detendrá.

Cada vez que se inicie el temporizador de tiempo de descanso y se detenga aparecerá un cuadro indicando el número de descanso y la duración del mismo:



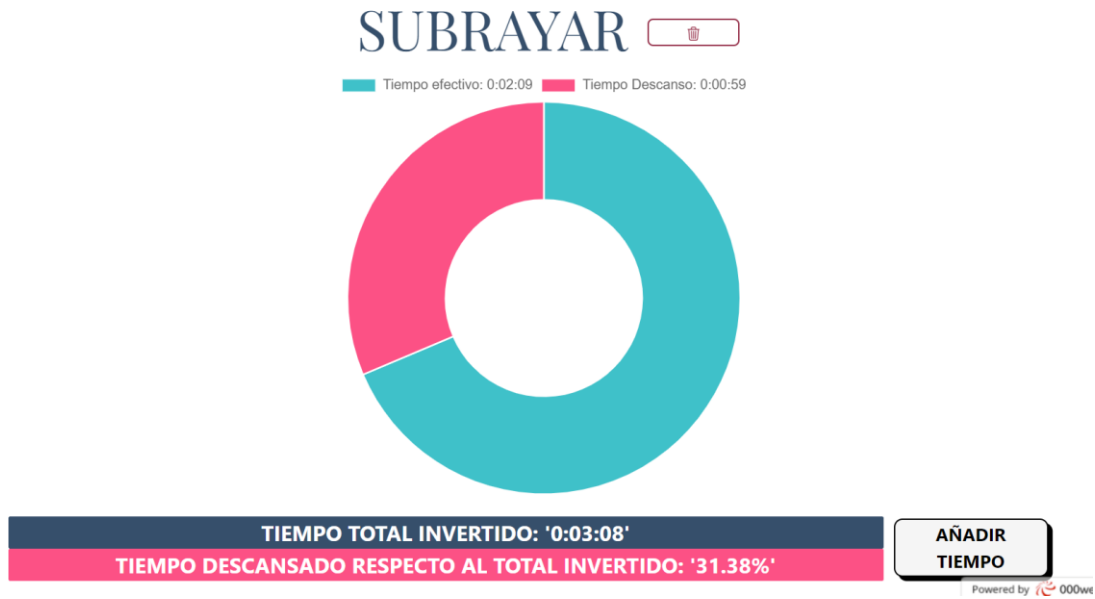
0.10 Muestra de un descanso en el apartado de temporizadores

Una vez terminamos la tarea y queremos guardar los tiempos registrados pulsaremos en el botón de guardar (Botón gris contiguo al nombre de la tarea). Aparecerán mensajes de confirmación y tras pulsar en aceptar se almacenará el tiempo total efectivo y el tiempo total descansado, los cuales parecen en los temporizadores correspondientes.



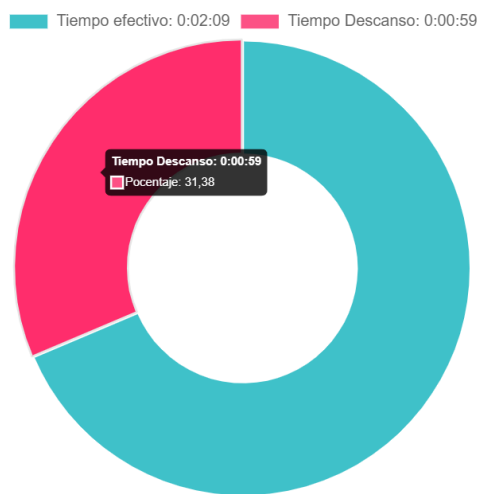
0.11 Actividad de Efficient con botón de estadísticas resaltado

Una vez tenemos registros de tiempo almacenados podemos pulsar en el botón de “stats” asociado a la tarea. Tras pulsarle seremos redirigidos a la página en la cual se encuentra el grafico con los tiempos totales invertidos en la tarea.



0.12 Vista de estadísticas de una tarea

Además de las opciones visibles, si mantenemos el ratón sobre el gráfico indicará los porcentajes de los tiempos.

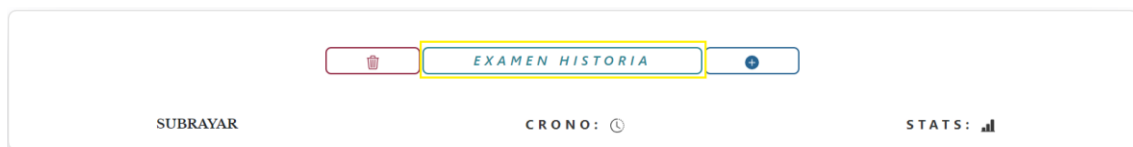


0.13 Muestra de porcentajes del diagrama

Además de visualizar los tiempos en esta pantalla tenemos las opciones para eliminar la tarea pulsando al botón superior de la derecha, o la posibilidad de añadir más tiempo a la tarea por si se hubiese dedicado tiempo a la tarea sin usar la aplicación y se quiere mantener constancia del mismo.

0.14 Formulario para añadir más tiempo a una tarea

Ahora que hemos creado dos registros podríamos visualizar el tiempo dedicado a la actividad en total, para ello pulsamos en el título de la actividad que nos llevará a la página con todos los registros.



0.15 Actividad con botón para mostrar el histórico resaltado

En esta pantalla se muestra el tiempo total dedicado por la actividad además de estar desgranado por tiempo efectivo y tiempo total.

### EXAMEN HISTORIA

| DIA        | TIEMPO EFECTIVO | TIEMPO DESCANSO | TAREA    |
|------------|-----------------|-----------------|----------|
| 2023-06-29 | 00:02:09        | 00:00:59        | SUBRAYAR |
| 2023-06-29 | 00:01:00        | 00:00:00        | SUBRAYAR |

**EL TIEMPO TOTAL INVERTIDO ES:**  
0:04:08

**EL TIEMPO TOTAL EFECTIVO ES:**  
0:03:09

**EL TIEMPO TOTAL DESCANSADO ES:**  
0:00:59

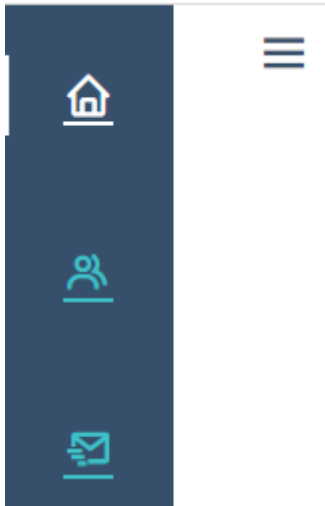
Exportar a PDF

0.16 Histórico de una actividad

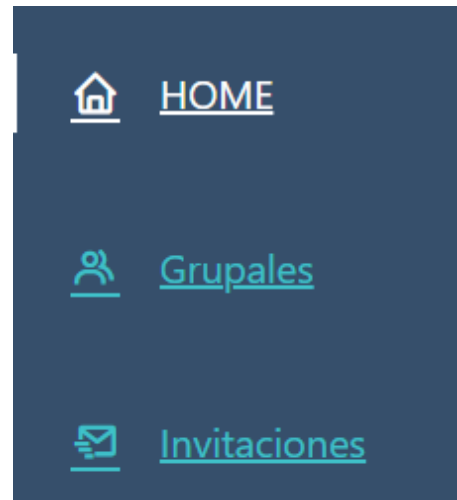
También, se indica en que tarea se trabajó en qué fecha y que tiempos se realizaron.

La última opción que nos permite esta pantalla es la de exportar el histórico a PDF, pulsando en el botón inferior “Exportar a PDF” se creará un PDF almacenable por el usuario.

Para la navegabilidad entre pantallas en todo momento se encuentra una barra lateral la cual permite esta navegación:



0.17 Barra lateral contraída

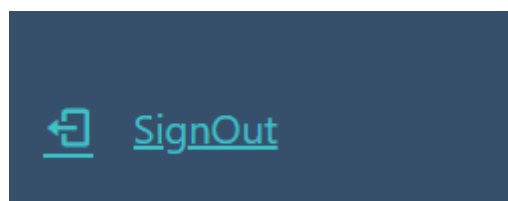


0.18 Barra lateral expandida

En la parte superior de la barra aparecen 3 iconos cuando se encuentra plegada, para desplegarla se ha de pulsar en las 3 barras horizontales que se encuentran junto a ella.

Pulsando en la opción de “HOME” seremos redirigidos a la pantalla principal donde se encuentran las actividades y tareas del usuario. Pulsando en “Grupales” seremos redirigidos a la pantalla donde se encuentran las actividades y tareas grupales en las cuales el usuario es participante. Y pulsando en “Invitaciones” se redirigirá al usuario a la pantalla donde se encuentra el buzón de invitaciones a actividades del usuario.

En la parte inferior de la barra lateral se encuentra el botón para cerrar la sesión de la aplicación.



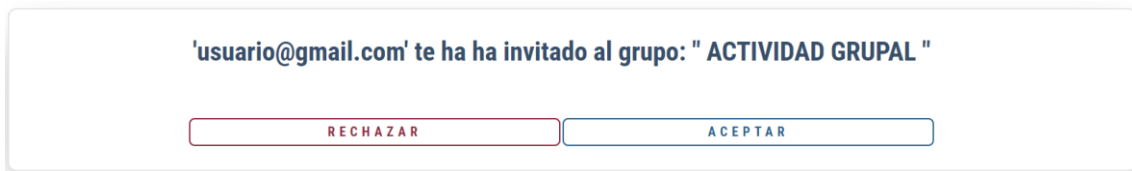
0.19 Botón cierre de sesión

Cuando pulsamos en actividades grupales las actividades tienen exactamente el mismo funcionamiento que las actividades individuales, con la diferencia de que se pueden compartir con otros usuarios. Tras pulsar en el botón de compartir se ha de escribir el email del destinatario y si es correcto, este recibirá una invitación para ser participante de la actividad.



0.20 Actividad grupal con botón para compartir actividad grupal resaltado

Una vez escrito el correo adecuadamente el usuario recibirá una nueva invitación con un formato como este:



0.21 Ejemplo de invitación

En el momento que se acepta la invitación el usuario podrá interactuar con la actividad y aparecerá en sus actividades grupales.

En estas actividades grupales las diferencias en el histórico de la actividad es que aparecerá los tiempos desgranados por usuarios y además se mostrará quien creó el registro de la tarea, y en los gráficos de las tareas que únicamente aparecerá los tiempos efectivos desgranados por usuarios, no se mostrarán los tiempos de descanso en el diagrama:

## ACTIVIDADGRUPAL

| DIA        | TIEMPO EFECTIVO | TIEMPO DESCANSO | TAREA         | USUARIO   |
|------------|-----------------|-----------------|---------------|-----------|
| 2023-06-20 | 00:00:00        | 00:00:04        | TAREA PRIMERA | julian    |
| 2023-06-20 | 00:00:01        | 00:00:00        | TAREA PRIMERA | julian    |
| 2023-06-20 | 01:00:00        | 00:00:00        | TAREA PRIMERA | julian    |
| 2023-06-20 | 01:00:00        | 00:00:00        | TAREA PRIMERA | julian    |
| 2023-06-20 | 01:00:00        | 00:00:00        | TAREA PRIMERA | Estefania |

TOTAL INVERTIDO: 3:00:05

TOTAL EFECTIVO: 3:00:01

TOTAL DESCANSO: 0:00:04

JULIAN

EFECTIVO: 2:00:01

DESCANSADO: 0:00:04

ESTEFANIA

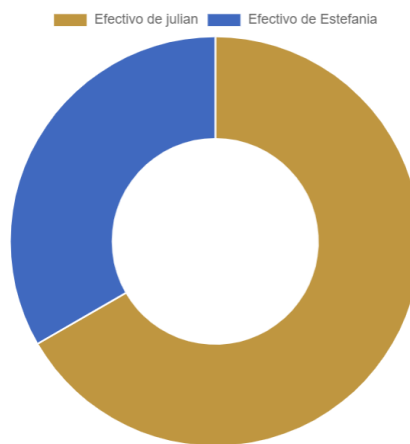
EFECTIVO: 1:00:00

DESCANSADO: 0:00:00

Exportar a PDF

0.22 Ejemplo de histórico actividad grupal

## TAREA PRIMERA



JULIAN

EFECTIVO: '2:00:01'

DESCANSADO: '0:00:04'

ESTEFANIA

EFECTIVO: '1:00:00'

DESCANSADO: '0:00:00'

0.23 Muestra vista estadísticas tarea grupal

En este caso el botón de añadir tiempo se corresponde con el botón azul de la parte superior derecha.



## Webgrafía

1. colaboradores de Wikipedia. (2022). Pruebas de caja blanca. *Wikipedia, la enciclopedia libre*. Última vez visitado: 26/06/2023.  
[https://es.wikipedia.org/wiki/Pruebas\\_de\\_caja\\_blanca](https://es.wikipedia.org/wiki/Pruebas_de_caja_blanca)
2. colaboradores de Wikipedia. (2023a). Arquitectura de software. *Wikipedia, la enciclopedia libre*. Última vez visitado: 22/06/2023.  
[https://es.wikipedia.org/wiki/Arquitectura\\_de\\_software#:~:text=Una%20arquitectura%20de%20%5Bsoftware%5D%2C,el%20c%C3%B3digo%20fuente%20de%20software.](https://es.wikipedia.org/wiki/Arquitectura_de_software#:~:text=Una%20arquitectura%20de%20%5Bsoftware%5D%2C,el%20c%C3%B3digo%20fuente%20de%20software.)
3. colaboradores de Wikipedia. (2023b). Técnica Pomodoro. *Wikipedia, la enciclopedia libre*. Última vez visitado: 12/06/2023.  
[https://es.wikipedia.org/wiki/T%C3%A9cnica\\_Pomodoro](https://es.wikipedia.org/wiki/T%C3%A9cnica_Pomodoro)
4. colaboradores de Wikipedia. (2023c). PHP. *Wikipedia, la enciclopedia libre*. Última vez visitado: 12/06/2023. <https://es.wikipedia.org/wiki/PHP>
5. colaboradores de Wikipedia. (2023d). HTML. *Wikipedia, la enciclopedia libre*. Última vez visitado: 12/06/2023. <https://es.wikipedia.org/wiki/HTML>
6. *Composer, gestor de dependencias para PHP*. (2022, 23 diciembre). Última vez visitado: 12/06/2023. <https://www.cursosdesarrolloweb.es/blog/composer-gestor-dependencias-php#:~:text=Composer%20es%20el%20gestor%20de,terceros%20para%20a%C3%B1adir%20funcionalidades%20adicionales.>
7. @denqbit. (2018, 3 junio). Entendiendo M de MVC y sus problemas - @denqbit - Medium. *Medium*. Última vez visitado: 22/06/2023.  
<https://medium.com/@denqbit/entendiendo-m-de-mvc-y-sus-problemas-ebc0cbf518ec>

8. *Forest*. (s. f.). Última vez visitado: 12/06/2023. <https://www.forestapp.cc/>
9. *Gulp: la herramienta que automatiza tus tareas*. (s. f.). Última vez visitado: 12/06/2023. <https://www.hostgator.mx/blog/gulp-la-herramienta-que-automatiza-tus-tareas/>
10. *Modelo de desarrollo incremental - Globe Testing*. (2013, 22 mayo). Globe Testing. Última vez visitado: 13/06/2023. <https://ahorasomos.izertis.com/globetesting/modelo-de-desarrollo-incremental/>
11. *PHP: ¿Qué es PHP? - Manual*. (s. f.). Última vez visitado: 12/06/2023. <https://www.php.net/manual/es/intro-what-is.php>
12. *Qué es MVC*. (s. f.). DesarrolloWeb.com. Última vez visitado: 25/06/2023. <https://desarrolloweb.com/articulos/que-es-mvc.html>
13. Robledano, A. (2023, 13 abril). Qué es MySQL: Características y ventajas. *OpenWebinars.net*. Última vez visitado: 12/06/2023. <https://openwebinars.net/blog/que-es-mysql/>
14. *Salario para Desarrollador Php en España - Salario Medio*. (s. f.). Talent.com. Última vez visitado: 30/06/2023. <https://es.talent.com/salary?job=desarrollador+php>
15. Team, K. (2023a, abril 11). ¿Qué son las pruebas de caja blanca? *KeepCoding Bootcamps*. Última vez visitado: 26/06/2023. <https://keepcoding.io/blog/que-son-pruebas-de-caja-blanca/>
16. Team, K. (2023b, mayo 4). ¿Qué son las pruebas de caja negra? | *KeepCoding Bootcamps*. *KeepCoding Bootcamps*. Última vez visitado: 28/06/2023. <https://keepcoding.io/blog/que-son-las-pruebas-de-caja-negra/#:~:text=Las%20pruebas%20de%20caja%20negra%2C%20conocidas%2>

- Otambi% C3%A9n% 20como% 20black% 20box, la% 20informaci% C3%B3n% 20r  
eferente% 20a% 20la
17. *TomatoTimers - Custom Pomodoro Timer and To-do List*. (s. f.). Última vez visitado: 12/06/2023. <https://www.tomatotimers.com/>
  18. Unir, V. (2023, 9 enero). ¿Qué es SASS y cómo aprenderlo? *UNIR*. Última vez visitado: 13/06/2023. <https://www.unir.net/ingenieria/revista/que-es-sass/#:~:text=SASS%20es%20un%20preprocesador%20de,la%20mayor% C3%ADa% 20de% 20los% 20navegadores>.
  19. Urrutia, D. (2021). Qué es el JavaScript - Definición, significado y ejemplos. *Armetrics*. Última vez visitado: 12/06/2023. <https://www.armetrics.com/glosario-digital/javascript>
  20. *What is Apache - Javatpoint*. (s. f.). [www.javatpoint.com](http://www.javatpoint.com). Última vez visitado: 12/06/2023. <https://www.javatpoint.com/what-is-apache>
  21. Wikipedia contributors. (2022). Active record pattern. *Wikipedia*. Última vez visitado: 24/06/2023. [https://en.wikipedia.org/wiki/Active\\_record\\_pattern](https://en.wikipedia.org/wiki/Active_record_pattern)
  22. Juan Pablo De la torre Valdez. Udey.com. Última vez visitado: 30/06/2023. <https://www.udemy.com/share/1013ea3@3953omOZkG5THXiKDOiRZNLZXQYjIKKRk61J9MctN56bKG-cDqJ04yTd8PqsVcck5Q==/>