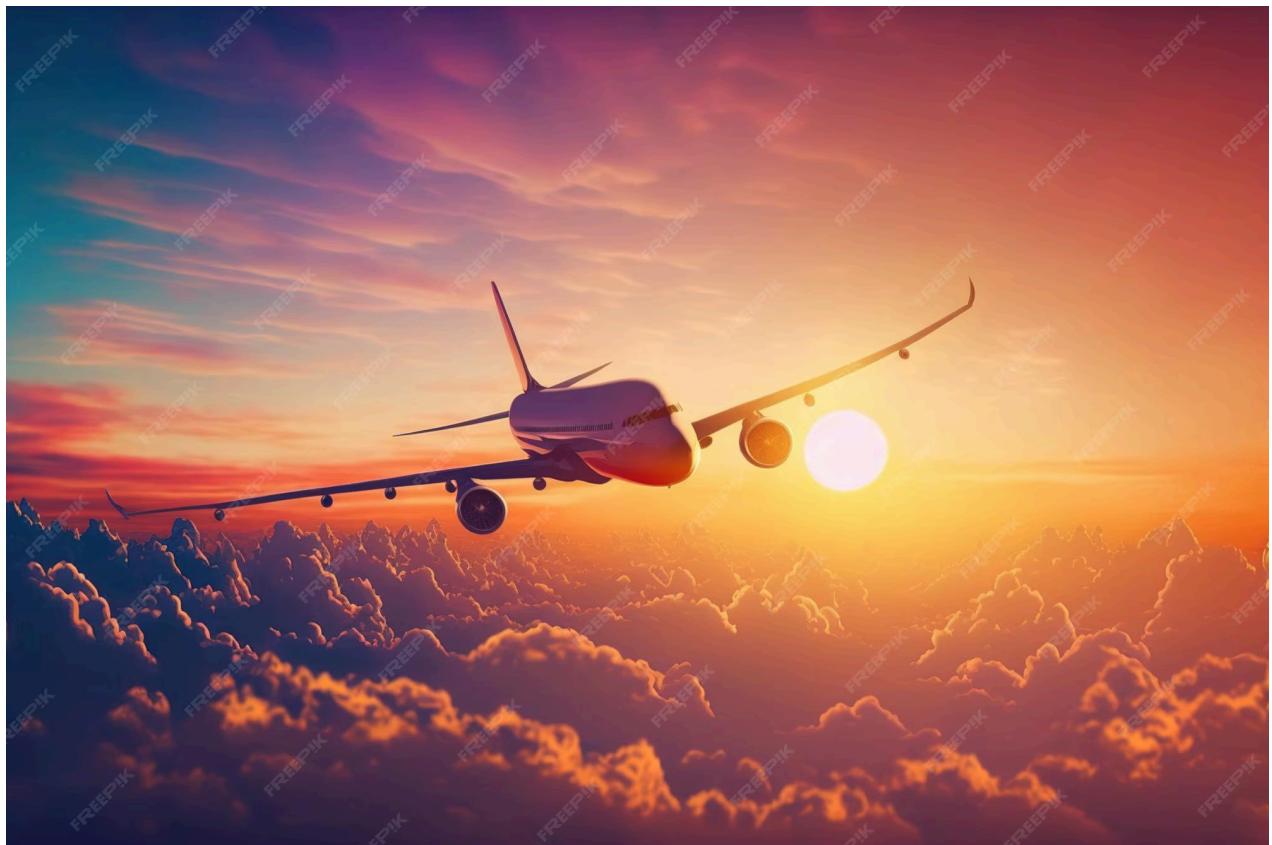


# PLANIFICADOR DE VIAJES



# ÍNDICE DE CONTENIDO

<b>1. Introducción</b>	<b>3</b>
1.1. Objetivo del proyecto:	3
1.2. Justificación del proyecto:	3
<b>2. Requisitos funcionales de la aplicación</b>	<b>3</b>
2.1. Problema que resuelve el sistema	3
2.2. Funcionalidades principales de la aplicación	4
2.3. Tipo de usuarios a los que va dirigido el sistema.	5
<b>3. Análisis y diseño</b>	<b>5</b>
3.1. Diagrama de casos de uso:	5
3.2. Diagrama de clases:	5
<b>4. Metodología de desarrollo</b>	<b>5</b>
4.1. Descripción del ciclo de vida del software utilizado.	5
<b>1. Análisis de requisitos</b>	<b>6</b>
Objetivo principal	6
Requisitos funcionales	6
Requisitos no funcionales	6
Limitaciones del proyecto	6
<b>2. Diseño del sistema</b>	<b>7</b>
Arquitectura	7
Posible estructura de clases	7
Diseño del timeline	7
Diseño de interfaz	7
<b>3. Implementación (pendiente de desarrollo)</b>	<b>8</b>
<b>4. Pruebas (pendiente de desarrollo)</b>	<b>8</b>
<b>5. Despliegue (pendiente)</b>	<b>8</b>
<b>6. Mantenimiento (futuro)</b>	<b>8</b>
4.2. Metodología ágil elegida (por ejemplo, Scrum).	9
4.3. Planificación inicial de tareas y roles del equipo.	9
<b>5. Plan de negocio</b>	<b>10</b>
5.1 Público objetivo del proyecto.	10
5.2 Utilidad social o empresarial.	10
5.3 Posibles costes y beneficios.	11
<b>6. Prevención de Riesgos Laborales</b>	<b>12</b>
6.1 Identificación de riesgos asociados al desarrollo del proyecto	12
6.2 Medidas preventivas aplicables al trabajo con equipos informáticos	12

# 1. Introducción

## 1.1. Objetivo del proyecto:

El objetivo que persigue este proyecto es el de crear una aplicación web que, le de a las personas que viajan por el mundo la opción de poder recomendar y crear ellos mismos las mejores paradas de un lugar que hayan visitado o vayan a hacerlo.

La aplicación estará compuesta de una web, creada usando HTML, CSS y JavaScript, vinculada a una base de datos hecha en SQLite, y gestionada por nuestro programa escrito en Java.

## 1.2. Justificación del proyecto:

Hay dos puntos claves que trata de justificar un proyecto como este :

1. Inmiserir más a los usuarios en las opciones que se dan dentro de la aplicación cuando van a seleccionar un lugar el cual visitar, y las actividades que se van a hacer en ese lugar.

Esto se hará mediante un sistema de reseñas a las actividades practicadas, estas reseñas pertenecerán a la aplicación y no se tomarán como referencia otras reseñas de aplicaciones externas por lo que los usuarios que creen sus viajes y actividades serán los que valoren esos viajes y actividades.

2. Dar una interfaz más creativa y accesible para todas las personas, plenamente enfocada al ámbito de viajes.

# 2. Requisitos funcionales de la aplicación

## 2.1. Problema que resuelve el sistema

Nuestro sistema pretende solucionar la dificultad que tienen muchas personas para planificar correctamente y de manera bien estructurada un viaje. Actualmente, los usuarios utilizan distintas herramientas para organizar sus viajes (notas, mapas, documentos, etc.), al dividir tanto un plan, se vuelve todo menos claro y desorganizado, lo que facilita olvidar ciertos aspectos importantes del viaje.

La solución es ofrecer un software en el que puedas planificar el viaje completo desde una sola plataforma. De esta manera, podrán organizar sus viaje de manera mucho más clara y con menos riesgos a olvidarte de aspectos clave del mismo.

Además, con el uso de una TimeLine interactiva, nuestros usuarios podrán organizar de forma muy sencilla las actividades que van a realizar en cada día, indicando en

los restaurantes, museos, monumentos, o cualquier otro punto de interés al que deseen ir.

## 2.2. Funcionalidades principales de la aplicación

La aplicación contará con las siguientes funcionalidades:

### **Creación y gestión de viajes:**

La web permitirá a los usuarios crear múltiples viajes. Cada viaje podrá ser modificado (cambiar nombre o sus datos) o eliminado.

### **Introducir la información del viaje:**

Al crear un viaje, el usuario deberá de introducir información básica, como a qué país irá, qué ciudades va a visitar, el nombre que le va a dar al viaje, etc. Con todos esos datos, se generará un viaje inicial al que se podrá ya empezar a añadir actividades.

### **TimeLine interactiva:**

Cada viaje cuenta con una TimeLine principal en la que se añadirán todas las actividades que el usuario planea realizar en su viaje. La TimeLine se divide por día y hora, permitiendo organizar bien qué actividades irán primero, y qué día del viaje se harán las mismas.

### **Gestión de actividades:**

El usuario podrá crear actividades pertenecientes a un viaje. Cada actividad cuenta con varios datos que el usuario debe de introducir: Se debe poner un título a la actividad, una descripción, una imagen, un emoji que represente lo que harás (por ejemplo, un tenedor para un restaurante), y una fecha de inicio y fecha de fin de la actividad.

### **Drag & Drop:**

Nuestro sistema está pensado para que todas las actividades puedan ser desplazadas en la TimeLine utilizando Drag & Drop (arrastrar y soltar), con esto, hacemos la web más intuitiva.

### **Publicar tu viaje:**

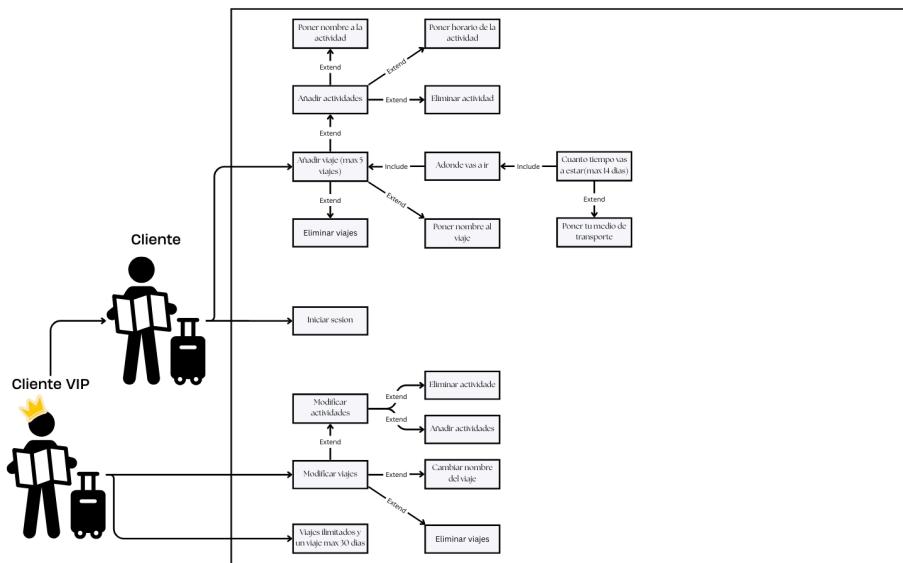
El sistema permitirá al usuario marcar su viaje como público. Los viajes públicos aparecerán en un apartado de viajes de la comunidad, al que todos los usuarios pueden acceder, ayudando a los usuarios a inspirarse con otros viajes, usándolos como referencia.

### **2.3. Tipo de usuarios a los que va dirigido el sistema.**

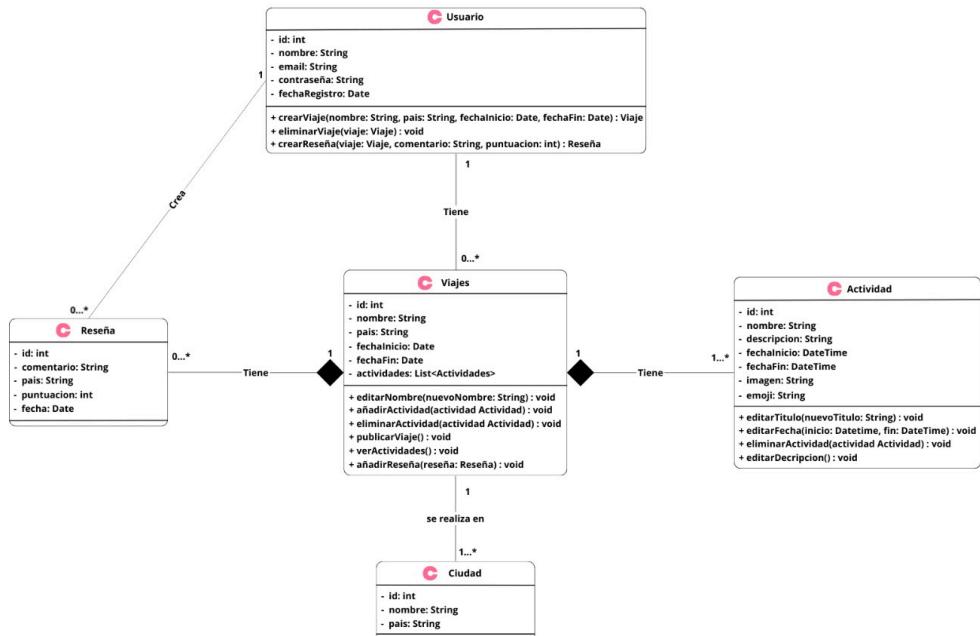
Nuestro sistema va dirigido al público general que desea organizar un viaje de manera eficiente y quiere tener claro qué es lo que hará cada día. La app está pensada para quienes quieren aprovechar al máximo sus visitas a las ciudades a las que viajan, e intentan no perderse sus maravillas.

### 3. Análisis y diseño

#### 3.1. Diagrama de casos de uso:



#### 3.2. Diagrama de clases:



## 4. Metodología de desarrollo

### 4.1. Descripción del ciclo de vida del software utilizado.

Este programa se dedicará a resolver problemas en cuanto a la organización de un viaje, nos dispondremos a ayudar a la planificación de viajes mediante la utilización de un timeline en el que poder añadir actividades y las horas a las que vaya a estar en esa zona visitando los diferentes puntos de interés dentro de la zona.

Nuestros recursos son limitados, y nuestro tiempo no es demasiado pero tenemos la intención de entregar un producto válido para nuestro cliente, estamos buscando una solución eficiente, organizada y viable dentro del plazo establecido.

#### 1. Análisis de requisitos

En esta fase se definen las necesidades del cliente y los objetivos del sistema.

##### Objetivo principal

Desarrollar una aplicación que permita organizar viajes mediante una línea temporal (timeline) donde el usuario pueda planificar actividades según horario y ubicación.

##### Requisitos funcionales

El sistema deberá permitir:

- Crear un viaje.
- Añadir actividades a un timeline.
- Asignar hora de inicio y fin a cada actividad.
- Asociar actividades a una zona o ubicación.
- Visualizar los puntos de interés de cada zona.
- Modificar o eliminar actividades.
- Consultar el itinerario completo organizado por horario.

##### Requisitos no funcionales

- Interfaz sencilla e intuitiva.
- Organización clara de la información.
- Respuesta rápida del sistema.
- Código estructurado y mantenible.
- Uso eficiente del tiempo y recursos de desarrollo.

## Limitaciones del proyecto

- Tiempo reducido.
- Recursos humanos limitados.
- Posible simplificación de funcionalidades avanzadas (por ejemplo, mapas en tiempo real o integración con APIs externas).

En esta fase no se programa, únicamente se define qué debe hacer el sistema.

## 2. Diseño del sistema

En esta fase se define cómo se va a construir la solución.

### Arquitectura

Se propone una estructura modular, posiblemente basada en Programación Orientada a Objetos (POO), donde cada entidad del viaje está representada por una clase.

### Possible estructura de clases

- Clase Viaje
  - nombre
  - fechalinicio
  - fechaFin
  - lista Actividades
- Clase Actividad
  - nombre
  - hora Inicio
  - hora Fin
  - zona
  - descripción
- Clase Zona
  - nombre Zona
  - lista Puntos Interés
- Clase Punto Interes
  - nombre
  - descripción
  - ubicación

### Diseño del timeline

El timeline podría representarse:

- Mediante una lista ordenada por hora.
- Utilizando estructuras como ArrayList ordenadas por hora de inicio.
- Implementando un sistema que valide que no haya solapamiento de actividades.

Diseño de interfaz

Dependiendo del entorno:

- Aplicación de consola (estructura simple).
- Interfaz gráfica básica.
- Visualización cronológica organizada por día y hora.

En esta fase se realizan esquemas, diagramas UML y planificación técnica antes de comenzar a programar.

### 3. Implementación (pendiente de desarrollo)

En esta fase se procederá a programar el sistema siguiendo el diseño establecido. Se crearán las clases, métodos y estructuras necesarias para que el programa funcione correctamente.

Actualmente esta fase está pendiente de ejecución.

### 4. Pruebas (pendiente de desarrollo)

En esta fase se realizarán pruebas para comprobar:

- Correcto funcionamiento del timeline.
- Validación de horarios.
- Funcionamiento de altas, bajas y modificaciones.
- Control de errores.

Esta fase se llevará a cabo una vez implementado el sistema.

### 5. Despliegue (pendiente)

Consistirá en entregar el producto final al cliente o ponerlo en funcionamiento en el entorno definido.

## 6. Mantenimiento (futuro)

Una vez entregado el producto, se podrán realizar:

- Corrección de errores.
- Mejoras funcionales.
- Ampliaciones (por ejemplo, integración con mapas o recomendaciones automáticas).

### 4.2. Metodología ágil elegida (por ejemplo, Scrum).

Se basa en una metodología ágil y flexible que permite a un equipo solventar problemas complejos y entregar productos funcionales de manera incremental. Se basa en la teoría empírica de procesos y se sustenta en:

#### **Transparencia:**

Todo el equipo tiene acceso a la información clave sobre el proceso y prioridades.

Permite revisar avances y detectar problemas de forma temprana.

No se pueden tomar decisiones correctas sobre datos incompletos o falsos.

Ahora la pregunta es cómo aplicamos esto a nuestro proyecto, primero haremos un brainstorming o backlog el cual sirve para saber qué funcionalidades o mejoras se van a hacer a futuro. Después haremos un tablero (por ejemplo un Kanban o Scrum) con el cual podremos ver que tareas están en proceso, cuales todavía no están empezadas y cuales hemos finalizado. En cada parte de este proceso se hace lo que se llama un sprint en nuestro caso tenemos una serie de entregas prolongadas en el tiempo lo cual hace que no tengamos como tal un sprint en el que tengamos que entregar un producto funcional sino que tenemos que ir entregando el proyecto según los objetivos que se nos imponen. Entonces si que cumpliremos un incremento pero hasta fases finales de este proyecto no podremos ver resultados claros o por lo menos cumplir esos objetivos personales e ideas que hemos pensado.

### 4.3. Planificación inicial de tareas y roles del equipo.

La planificación inicial de tareas y la definición clara de roles son fundamentales para asegurar una ejecución eficiente del proyecto. Hemos ido eligiendo que iremos haciendo mediante forma aleatoria. En esta etapa se establecen responsabilidades, tiempos estimados y mecanismos de coordinación. Este es el tablón mediante el cual hemos ido organizando que hace que.

The screenshot shows a digital Kanban board titled "PROYECTO FINAL". The board is organized into five columns:

- IDEAS**: Contains 9 items, including tasks like "Letras automáticas para canciones" and "Planificador de semana con IA enfocada a los estudios con calendario autorrellenable, con disposición de páginas personalizadas para cada día".
- POR HACER**: Contains 4 items, such as "Establecer paleta de colores de la web" and "Crear diagrama de clases".
- EN CURSO**: Contains 4 items, including "Realizar prototipo del frontend" and "HITO 2 - ASIGNACIONES".
- EN REVISIÓN**: This column is currently empty.
- FINALIZADO**: Contains 1 item, "Crear ficheros con los datos de ciudades y países".

Each card in the columns includes a checkbox, a role indicator (e.g., SCRUM-B, SCRUM-21, SCRUM-20, etc.), and a small circular icon with a number (e.g., 64, 55). A search bar and filter options are visible at the top of the board.

## 5. Plan de negocio

### 5.1 Público objetivo del proyecto.

---

**El público objetivo determina qué tipo de usuarios son específicos para el producto o servicio.**

---

Para definir el público objetivo de nuestro proyecto, se requiere comprender las necesidades del usuario, en relación con la planificación de viajes. Esto permite diseñar la aplicación web con la finalidad de mejorar la navegación del usuario.

El sistema está enfocado a un perfil interesado en viajar y en planificar sus viajes de manera organizada y visual. Así, el usuario puede estructurarlo de forma más eficaz.

### 5.2 Utilidad social o empresarial.

#### **Utilidad social:**

La aplicación web desde el punto de vista social, facilita al usuario planificar sus viajes de forma organizada, a gestionar actividades mediante una línea temporal y a compartir cada itinerario con una comunidad con intereses similares, mejorando la experiencia en el destino.

El sistema está diseñado para estructurar los viajes de forma visual y organizada, reduciendo el estrés que puede generar no saber por dónde comenzar ni cómo planificar las actividades, y optimizando el tiempo dedicado a la organización durante el propio viaje.

Al publicar los viajes en la comunidad los usuarios pueden acceder a los itinerarios de otros e inspirarse para crear los suyos.

#### **Utilidad empresarial:**

La aplicación web desde el punto de vista empresarial, está en un sector con mucha demanda, ya que, el tipo de usuario interesado en viajes busca siempre el poder planificar y organizar sus viajes de manera efectiva.

El sistema crea su propia comunidad de usuarios que comparten sus itinerarios, de esta manera el valor de la aplicación incrementa.

El proyecto ofrece diferentes planes, el gratuito y el premium, que contiene funcionalidades avanzadas.

### **5.3 Posibles costes y beneficios.**

#### **Costes:**

El proyecto requiere costes de desarrollo, que se asocian principalmente al tiempo dedicado a la programación, diseño de la aplicación web, diseño de la base de datos y pruebas del sistema. Estos costes pueden variar según la funcionalidad implementada y la dificultad de la misma.

Además, al ser una aplicación web, se requiere costes de infraestructura, ya que es necesario tener un hosting.

#### **Beneficios:**

El tiempo invertido en la programación permitirá que la aplicación sea interactiva y funcione de manera fluida, permitiendo a los usuarios crear viajes y gestionar de forma intuitiva la TimeLine y actividades. Además de permitir que la información se guarde en la nube, conectándose a todos los viajes a una Base de Datos.

El diseño web nos permite que la interfaz sea más clara, intuitiva y atractiva para los usuarios, facilitando organizar de forma muy visual sus viajes.

Implementar pruebas al sistema asegura que nuestra aplicación sea fiable y que cuente con la menor cantidad de errores posible. Con esto evitamos errores que afecten a la experiencia de usuario, como que sus viajes se borren, se muestre de forma incorrecta la información, errores visuales en la web, etc. Realizar pruebas hará que tengamos una aplicación web mucho más pulida y trabajada.

Contratar un hosting nos va a traer un claro beneficio: Hacer que cualquier persona pueda acceder a la aplicación de forma sencilla esté donde esté, guardando sus viajes entre dispositivos.

## **6. Prevención de Riesgos Laborales**

### **6.1 Identificación de riesgos asociados al desarrollo del proyecto**

#### **Organización del trabajo.**

Los riesgos que provoca tener una mala organización, son:

- El burnout
- Estrés

#### **Carga de trabajo.**

Los riesgos que provoca tener carga de trabajo, son:

- Fatiga física y mental por descansos insuficientes y jornadas largas de trabajo

#### **Uso prolongado de equipos informáticos.**

Los riesgos que provoca el uso prolongado de equipos informáticos, son:

- Fatiga visual
- Dolor muscular debido a malas posturas

### **6.2 Medidas preventivas aplicables al trabajo con equipos informáticos**

#### **Pausas periódicas.**

- Descansar cada 50 min para así evitar tanto la fatiga visual como la muscular.

#### **Ergonomía en el puesto de trabajo.**

- Uso de sillas ergonómicas para evitar la fatiga muscular
- Ajustar la altura e inclinación del monitor para evitar la tensión cervical
- Ajustar el brillo del monitor para evitar la fatiga visual
- Ajustar la distancia del trabajador con la del monitor

- Mantener los brazos y muñecas en una posición natural, para no forzarlos, a la hora de utilizar el teclado y el ratón
- Gafas especiales para pantallas si lo requiere el trabajador
- Uso de luz natural combinada con luz artificial para evitar fatiga visual

### **Planificación realista del tiempo.**

- Distribución de tareas equitativamente
- Jornadas y descanso adecuados para evitar la fatiga física y mental