



*Bidaiatzen*  
Aplicación web para viajar

**Jone Oria Abendibar**  
Master en Ingeniería Informática  
Desarrollo de aplicaciones web

**Ignasi Lorente Puchades**  
**César Pablo Córcoles Briongos**

01/2017



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

**Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)**

**A) Creative Commons:**



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](#)

**B) GNU Free Documentation License (GNU FDL)**

Copyright © AÑO TU-NOMBRE.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Soft-

ware Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

### **C) Copyright**

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Bidaiatzen. Aplicación web para viajar</i>
<b>Nombre del autor:</b>	<i>Jone Oria Abendibar</i>
<b>Nombre del consultor/a:</b>	<i>Ignasi Lorente Puchades</i>
<b>Nombre del PRA:</b>	<i>César Pablo Córcoles Briongos</i>
<b>Fecha de entrega (mm/aaaa):</b>	01/2017
<b>Titulación:::</b>	<i>Master en Ingeniería Informática</i>
<b>Área del Trabajo Final:</b>	<i>Desarrollo de aplicaciones web</i>
<b>Idioma del trabajo:</b>	<i>Español</i>
<b>Palabras clave</b>	<i>Viaje, planificador, guía</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b>	
<p>Esta aplicación web se ha desarrollado con la finalidad de ayudar a los viajeros a organizar su viaje y crear un guía personal con las experiencias vividas. Paralelamente, se ha aprovechado para aprender a resolver un proyecto completo. Para ello se ha usado una metodología en cascada que se ha ido completando paso a paso acorde con la planificación. Antes de empezar a desarrollar se ha hecho un análisis de mercado, eligiendo el software más adecuado, optando JSP como lenguaje de programación, Mysql como base de datos y Apache Tomcat como servidor web. Además, se ha hecho un primer diseño usando wireframes y diagramas UML siempre teniendo en cuenta el perfil de los usuarios y siendo viable para una arquitectura cliente/servidor. Como resultado se ha conseguido una aplicación amigable y sencilla que permite hacer todos los requisitos definidos con opciones a mejorar y ampliar definiendo líneas de trabajo futuro.</p>	

**Abstract (in English, 250 words or less):**

This web application has been developed with the purpose of helping the travelers to organize their trips and create personal guides with their experiences. In parallel, it has taken advantage of learn to solve a complete project. For this purpose, a waterfall model has been used according to the planning. Before starting to develop a market analysis has been done, choosing the most appropriate software, selecting JSP as programming language, Mysql as database and Apache Tomcat as a web server. In addition, it has made a first design using wireframes and UML diagrams, always taking into account the profile of the users and being viable for a client / server architecture. As a result we have achieved a friendly and easy application that allows us to make all the requirements defined, always defining future work lines.

# Índice

1.	Introducción.....	1
1.1.	Contexto y justificación del Trabajo.....	1
1.2.	Objetivos del Trabajo.....	1
1.3.	Enfoque y método seguido.....	2
1.4.	Planificación del Trabajo.....	3
1.5.	Breve resumen de productos obtenidos.....	4
1.6.	Breve descripción de los otros capítulos de la memoria.....	4
2.	Análisis.....	5
2.1.	Análisis de software.....	5
2.2.	Análisis de mercado [20-24].....	6
3.	Diseño de la aplicación.....	8
3.1.	Wireframes.....	8
3.2.	Diagramas UML.....	14
3.3.	Usabilidad/UX (DCU).....	18
3.4.	Arquitectura [5].....	18
4.	Desarrollo.....	20
4.1.	Base de datos.....	20
4.2.	Extractos de código.....	23
4.2.1.	Llamadas a servlets.....	24
4.2.2.	Páginas JAVA (Clases).....	24
3.3.3.	Páginas XML.....	27
3.3.4.	Páginas JSP.....	28
3.3.5.	Páginas JAVASCRIPT.....	29
3.3.6.	Páginas CSS.....	31
3.3.7.	Ficheros PROPERTIES.....	32
4.3.	Pantallazos finales.....	33
4.4.	Seguridad.....	37
4.5.	Test.....	37
4.6.	Bugs.....	39
5.	Conclusiones.....	41
6.	Glosario.....	43
7.	Bibliografía.....	44

## **Lista de figuras**

**No se encuentran elementos de tabla de ilustraciones.**



# 1. Introducción

## 1.1. Contexto y justificación del Trabajo

Los últimos años, la gente se está animando cada vez más a viajar por el mundo por su cuenta organizando su viaje según sus intereses.

Cada viaje es especial y aunque parezca sencillo organizar un viaje, siempre hay discusiones y malentendidos entre los viajeros. Además, con la intención de no olvidar el más mínimo detalle, es habitual escribir diarios, guardar folletos, notas, mapas que se amontonan de una forma desorganizada.

Hoy en día, con las tecnologías e internet, parece que todo esto debería de ser más fácil pero a veces gestionar la información adecuadamente no es fácil.

Esta aplicación está pensada para este tipo de personas que les gusta la aventura. Con esta herramienta se quiere facilitar la organización del viaje entre los integrantes del grupo y poder hacer una guía con la propia experiencia para futuros viajantes. Concretamente, esta aplicación web se divide en dos partes diferentes. Primero, el planificador del viaje, con la opción de definir tareas y responsables. Segundo, generador de guías, con opción de guardar todo tipo de detalles (gastos, hoteles, restaurantes, fotos...) y poder sacar un documento para guiar a otras personas.

## 1.2. Objetivos del Trabajo

Este trabajo tiene diferentes objetivos. Por una parte, objetivos personales y por otra objetivos del producto.

Objetivos personales:

- Aprender a resolver un proyecto completo, desde la fase de definición hasta la implementación.
- Aprender a documentar toda la información
- Conocer diferente software y herramientas para aplicaciones web.
- Aprender a planificarse.

Objetivos de la aplicación:

- Ayudar a los miembros del viaje organizarse entre ellos. Estos es, crear un entorno donde puedan definir tareas, fechas y responsables, intercambiando sus opiniones.

- Poder hacer el seguimiento del viaje al máximo detalle. Estos es, crear un entorno donde puedan guardar información sobre los lugares visitados, hoteles, transportes, gastos, fotos... diariamente.

### 1.3. Enfoque y método seguido

Aunque hay diferentes metodologías para el desarrollo de software (incremental, prototipado, espiral, cascada...), la metodología que se ha elegido es la metodología en cascada [1], conocido también como modelo clásico.

Consiste en una secuencia de etapas (definición, análisis, diseño, implementación y pruebas) que permite ordenar de manera rigurosa las etapas de ciclo de vida de software, dado que el comienzo de cada etapa debe esperar a la finalización de la anterior.

**Definición:** se define el proyecto, los objetivos y planificación. Casi siempre su salida es texto.

**Análisis:** consiste en reunir las necesidades del producto y decidir las herramientas y el software que se van a usar. Casi siempre su salida es texto.

**Diseño:** describe la estructura interna del producto y suele representarse con diagramas y texto.

**Implementación:** significa programación. Producto de esta etapa es el código.

**Pruebas:** consiste en chequear la funcionalidad y la usabilidad de la aplicación.

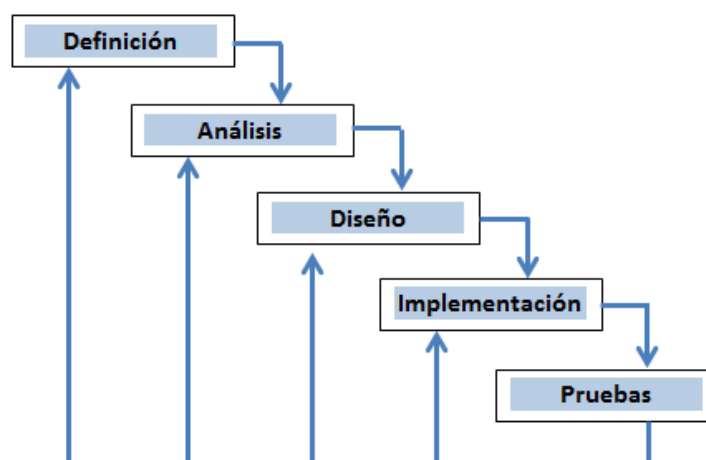


Imagen 1: Metodología Cascada

Se ha optado por esta metodología porque es sencilla y facilita la gestión del proyecto. Esto es importante dado que se va desarrollar un producto nuevo, individualmente y aproximadamente en cuatro meses. Por eso, es necesario tener las fases bien determinadas para poder terminar el proyecto en su tiempo y no perder el tiempo.

#### 1.4. Planificación del Trabajo

En la diagrama de Gantt se definen cuatro fases del proyecto (en naranja) que corresponden a las entregas parciales del TFM.

- **PEC 1:** La primera fase del trabajo. Elegir el tema escribir la memoria (descripción, objetivos, metodología y planificación)
- **PEC 2:** Segunda fase.
  - Analizar el mercado y la viabilidad del proyecto, eligiendo las herramientas y la arquitectura que se van usar.
  - Diseñar la base de datos y la aplicación
  - Preparar el entorno, instalando lo necesario para empezar a programar.
  - Escribir la memoria con toda la información recopilada
- **PEC 3:** Tercera fase.
  - Implementación de la aplicación. Como esta aplicación web está dividido en dos partes (planificador y generador de guías), primero se hará una parte y luego se empezará con la otra.
  - Pruebas del desarrollo.
  - Paralelamente, se irá documentando lo más importante.
- **Entrega final:** Fase final.
  - Se terminará el desarrollo
  - Se terminará con la memoria
  - Se hará la presentación, el video y el autoinforme.

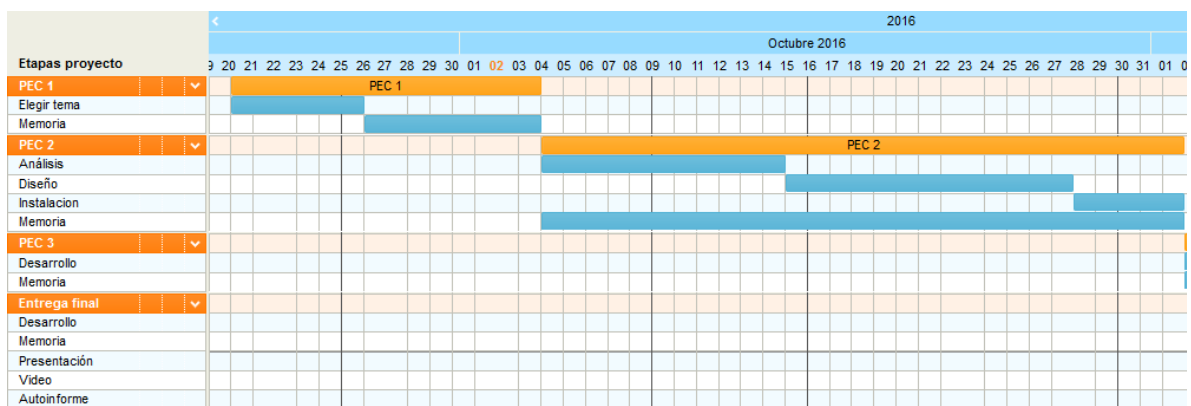


Imagen 2: Diagrama de Gantt 1

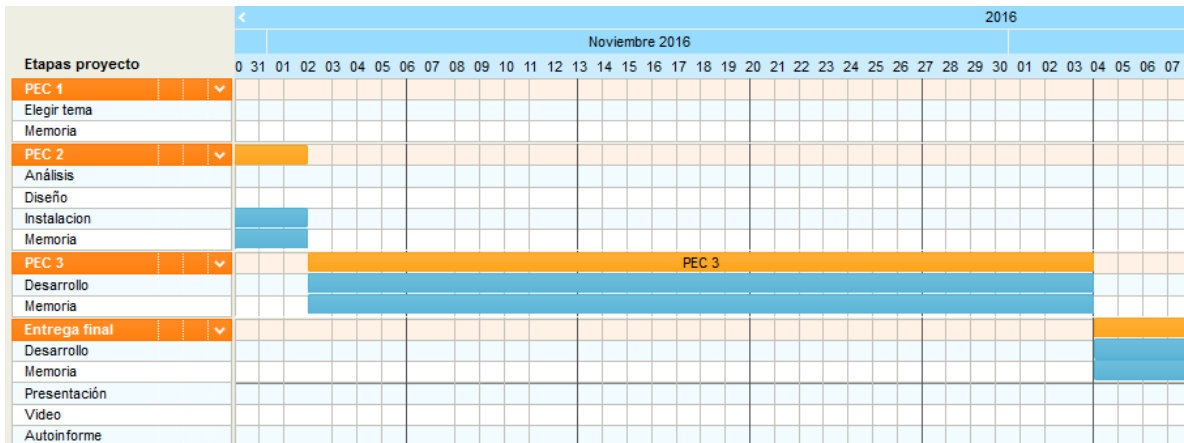


Imagen 3: Diagrama de Gantt 2

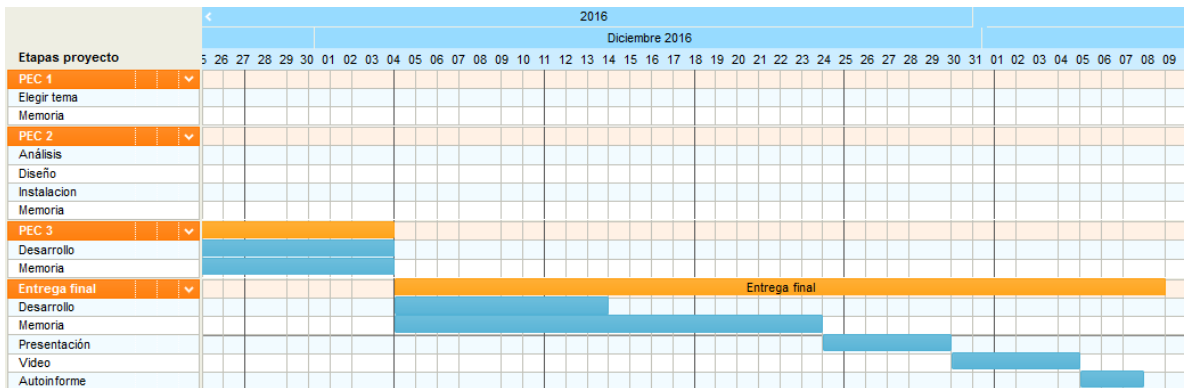


Imagen 4: Diagrama de Gantt 3

### 1.5. Breve resumen de productos obtenidos

Se ha obtenido una aplicación web programa en lenguaje JAVA. Para usar esta aplicación es necesario registrarse y está relacionada con los viajes. Da la opción de planificar un viaje creando diferentes tareas y también para crear un diario con los datos más importantes de cada día. Después se puede generar una guía con la experiencia propia.

Es una aplicación personal, esto es, cada usuario solo podrá ver sus viajes.

### 1.6. Breve descripción de los otros capítulos de la memoria

En esta memoria primero se hace un análisis de software para elegir el software que se va usar durante el desarrollo.

Después de decidir el software, se centra en el diseño de la aplicación siempre pensando en la usabilidad y el usuario final.

Una vez diseñado la aplicación es la hora del desarrollo. En este apartado se citan los puntos más importantes de la aplicación, se mencionan las decisiones tomadas por la seguridad y las pruebas que se han hecho.

Al final se explican las conclusiones que se han sacado.

## 2. Análisis

Este capítulo se hace un análisis de software comentando qué opciones hay en el mercado y cuales han sido escogidos y por qué.

### 2.1. Análisis de software

A la hora de elegir los software que se van a usar en este proyecto, se ha tenido en cuenta que sea un software libre. Teniendo en cuenta que este TFM es de 4 meses y su objetivo no es comercial, se ha decidido que es suficiente usar el software libre. También, la decisión ha sido por el conocimiento o la familiarización de la herramienta ya que en un tiempo limitado es un punto a favor.

#### **Bases de datos [2]**

Aunque hay varios software de bases de datos libres (MySQL, PostgreSQL, Apache Derby, SQLite) se ha elegido usar **MySQL** por sus ventajas y por el conocimiento hacía el software.

Ventajas:

- Velocidad al hacer las operaciones
- Bajo coste en requerimientos para la elaboración de bases de datos
- Facilidad de configuración e instalación



#### **Lenguaje de programación**

Se ha dudado entre dos lenguaje libres, Php que se basa en C y JSP basada en Java. Comparando los dos lenguajes se ha visto que los dos son válidos para este trabajo, pero se ha optado **JSP** por la familiarización y como he comentado antes esto es un punto a favor.

#### **Servidor web [3]**

Como se ha decidido programar en JSP, se ha elegido un servidor web bastante popular para este tipo de páginas, Apache Tomcat.

Tomcat en sí es un contenedor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets y puede funcionar como servidor web por sí mismo.



#### **Plataforma software [4]**

Para desarrollar el proyecto, se ha elegido Eclipse como plataforma porque es un entorno de desarrollo integrado, de código abierto y multiplataforma.



Es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web,

programas en C++ o aplicaciones Java. No es más que un entorno de desarrollo integrado (IDE) en el que se encuentran todas las herramientas y funciones necesarias para el trabajo, recogidas en una atractiva interfaz que lo hace fácil y agradable de usar.

## 2.2. Análisis de mercado [20-24]

En esta era de la tecnología en la cual estamos 24 horas conectados y con el Smartphone en la mano, se han encontrado miles de aplicaciones relacionados con los viajes.

Se ha hecho un estudio del mercado analizando los servicios y funcionalidades que nos dan estas aplicaciones para ver qué necesidades tiene el mercado y cómo podemos destacar en este mundillo con nuestra aplicación web.

Después de hacer el análisis de estas aplicaciones de viajes hemos agrupado en tres diferentes grupos según su funcionalidad y citamos algunas de ellas.

### **Buscadores**

Ayudan a encontrar hoteles, vuelos, restaurantes... de un lugar o zona incluyendo precios, fotos, ranking y comentarios de otros usuarios. Además la mayoría tiene la opción de reservarlos.

- Booking, TripAdvisor, Skyscanner, Hotel Tonight, Atrapalo, Kayak, Minube...

### **Planificador de viajes**

Ayudan a planificar el viaje, sugiriendo las atracciones turísticas, hoteles... de cada país o zona que quieras viajar y algunos permiten archivar los lugares de interés que quieras visitar.

- Way-away, Tripwolf, Minube, Tourist eye, Dopprl, Lonely Planet, Tripomatic, Trabber Planner, diarios de otros usuarios...

### **Diario de viaje**

La mayoría permiten guardar ubicaciones, fotos y ubicaciones en los que has estado por fechas.

- Journi, Travel Diary, Trip Journal, Bonjournal, Day One 2...

Una vez agrupadas y analizadas las aplicaciones vemos que la mayoría de los planificadores de viajes sugieren mucha información de los lugares, precios, hoteles, restaurantes... pero no dejan interactuar entre los miembros del viaje. Este proyecto está enfocado más a un entorno interactivo donde los viajantes puedan proponer y discutir ideas sobre un tema en concreto, más que a sugerir diferentes opciones de un lugar.

Si nos centramos en el diario o guía de viaje, la mayoría de diarios o guías están enfocados a las redes sociales y la información que nos permiten guardar no es muy detallada, normalmente una foto con una ubicación. En este proyecto se quiere ir más allá y se quiere dar la opción de guardar los datos de los transportes, alojamientos, gastos... porque aunque no es tan visual es una información muy útil a la hora de planificar un viaje. Además, esta aplicación permite crear el diario entre todos los miembros del grupo.

También vemos que hay pocas aplicaciones que permiten hacer ambas cosas a la vez, planificar más hacer un seguimiento del viaje.

## 3. Diseño de la aplicación

Este apartado contiene un primer diseño del aspecto y la estructura que va a tener la aplicación web incluyendo wireframes, diagramas UML, usabilidad y arquitectura. También se habla de la usabilidad y la arquitectura.

### 3.1. Wireframes

Los wireframes son planos de sitios web. En ellos podemos ver el esqueleto de la web, qué aspecto va a tener.

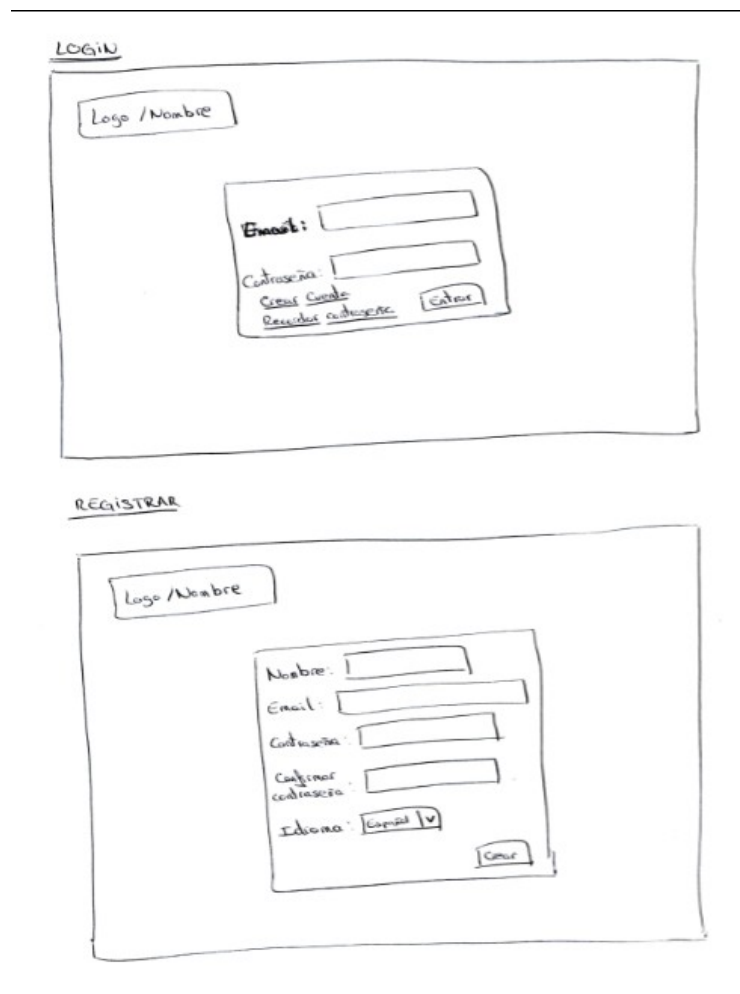


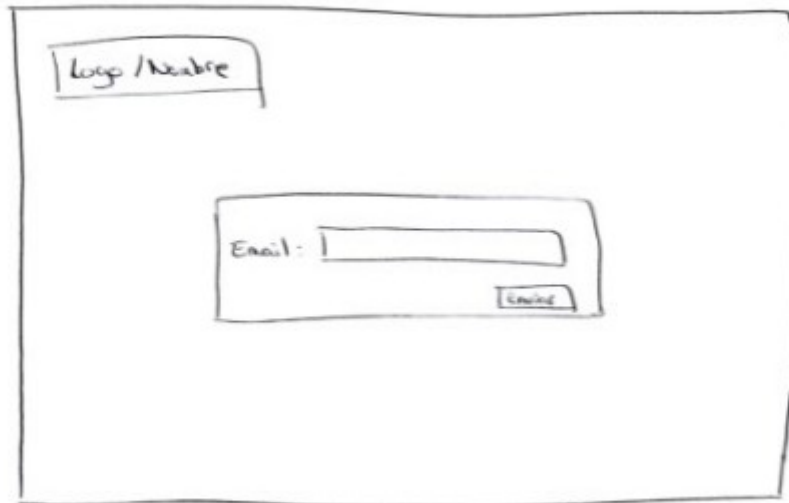
Imagen 5: Wireframe de Login y Registrar

Para entrar en la aplicación habrá un login y opciones para crear una nueva cuenta y recordar la contraseña de la cuenta existente (pantalla de arriba).

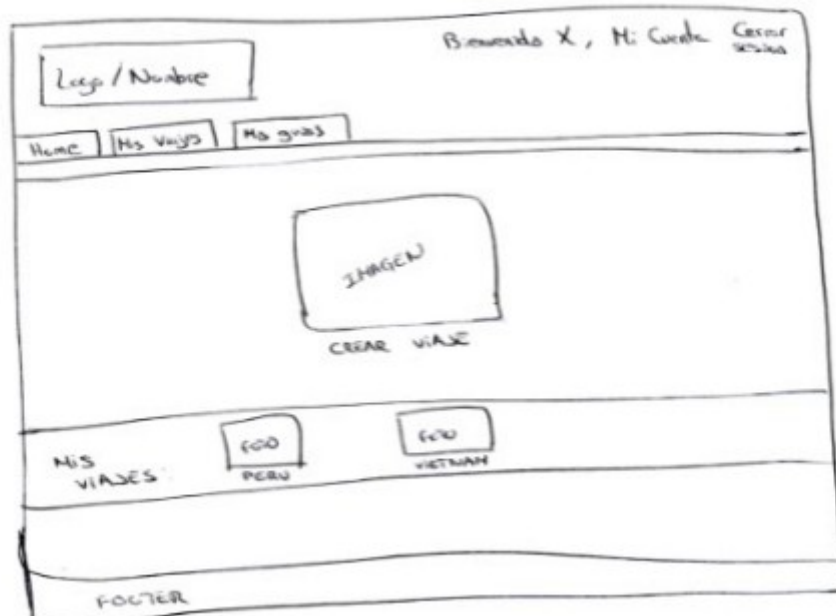
Al clicar el link de crear cuenta, aparecerá un formulario para rellenar datos (pantalla de abajo).



## RECORDAR CONTRASEÑA



## HOME



**Imagen 6: Wireframe de Recordar Contraseña y Home**

Al clicar en el link de recordar contraseña pedirá el email para enviar un email con la nueva contraseña (pantalla de abajo).

La pantalla de abajo será la pantalla inicial de la aplicación. La parte de arriba será común en todas las pantallas, a la izquierda aparecerá el nombre de la aplicación y a la derecha opciones de ir a "Mi Cuenta"(datos del usuario) y cerrar sesión. Además tendrá tres pestañas con las opciones Home, Mis Viajes y Mis Guías. En el centro habrá una imagen de "Crear Viaje" y pinchandola se podrá crear un nuevo imagen. Abajo se listarán los últimos viajes existentes de cada usuario.

## CREAR VIAJE / EDITAR VIAJE

Logo / Nombre

Bienvenido X, Mi cuenta

Cerrar sesión

Home

Mis Viajes

Mis Guías

SUBIR FOTO

Título:

Descripción:

Desde:

Hasta:

Participantes:

Guardar

PLANIFICAR

DIARIO

Imagen 7: Wireframe de Crear Viaje

Esta pantalla será para crear y modificar los datos generales del viaje y habrá acceso para entrar a planificar y al diario.

## MIS VIAJES:

Logo / Nombre		Bienvenido X, M. Gordo, Cerrar Sesión		
Home	Mis Viajes	Mis Guías		
VIAJE	FECHA	EDITAR	GUIA	BORRAR
NUOVA ZEALANDA	05-2017		<input type="checkbox"/>	X
PERU	08-2016		<input type="checkbox"/>	X
VIETNAM	08-2016		<input type="checkbox"/>	X

## MIS ~~VIAJES~~ GUIAS

Logo / Nombre		Bienvenido X, M. Gordo, Cerrar Sesión	
Home	Mis Viajes	Mis Guías	
GUIA	EDITAR	BORRAR	
PERU		X	
VIETNAM		X	

Imagen 8: Wireframe de Mis Viajes y Mis Guías

A estas pantallas se accederá de las pestañas de “Mis Viajes” y “Mis Guías”. En ambas se listarán los viajes y guías existentes de cada usuario y habrá la opción de editar y borrar.

## PLANIFICAR

Logo / Nombre Bienvenido X, Hi: Cerro, Gener session

Home | Mis viajes | Mis viajes

foto

Titulo: \_\_\_\_\_  
Descripcion: \_\_\_\_\_  
Desde: \_\_\_\_\_ Hasta: \_\_\_\_\_  
Participantes: \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_

Editar

TAREAS							
TAREA	DESCRIPCION	STATUS	F.VTO	RESP	OPIT	BOON	
<input type="checkbox"/> HOTEL	Reservar hotel	Pendiente	15/03/2016	Jane	<input type="checkbox"/>		X
<input type="checkbox"/> AVION	Reservar vuelo	Pendiente	-	Mel	<input type="checkbox"/>		X

FICHA TAREA

Tarea:

Descripcion:

F.Vto:

Responsable:

Status:

Guardar

COMENTARIOS

Comentarios:

Imagen 9: Wireframe de Planificar

Esta será la pantalla para planificar un viaje. Arriba del todo se listarán los datos generales del viaje, más abajo habrá una tabla con las tareas creadas. Clicando en cada tarea se podrán modificar en la parte de abajo. Y abajo del todo se podrán escribir comentarios a cada tarea.



**Imagen 10: Wireframe de Diario**

Esta pantalla será para escribir el diario o para generar la guía del viaje. Será muy parecido a la pantalla de planificación. Arriba del todo se listarán los datos generales del viaje. Más abajo habrá una tabla con los días creados del viaje y clicando en cada una de ellas se podrá modificar en la parte de abajo. Cada día tendrá información general, de los alojamientos, del transporte usado y los gastos. Abajo del todo habrá un botón para crear la guía, se generará un pdf.

## CREAR GUIA

Logo Alabe Bienvenido X, M. Corté, Gener 2020

Home Mis Viajes Mis Guías

**GENERAR GUIA**

Fecha: \_\_\_\_\_

Descripción: \_\_\_\_\_

Ubicación: \_\_\_\_\_

Comentarios: \_\_\_\_\_

**HOTEL**

Nombre: \_\_\_\_\_

Ubicación: \_\_\_\_\_

Servicios: \_\_\_\_\_

Comentarios: \_\_\_\_\_

**TRANSPORTE**

Tipo: \_\_\_\_\_

Imagen 11: Wireframe de Generar Guía

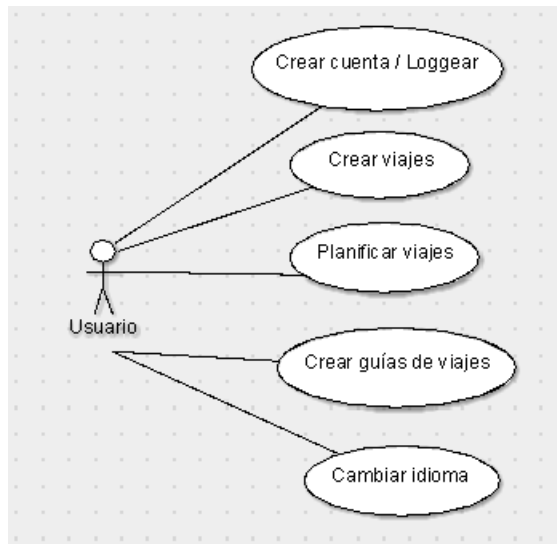
Al clicar el botón de generar guía se podrá elegir que información incluir en la guía, para eso sacará un dialogo.

### 3.2. Diagramas UML

Se hace el diseño de dos diagramas UML, los casos de uso y clases. En el primero se describen las actividades que puede hacer cada actor y en la segunda se muestran las clases del sistema atributos y los atributos.

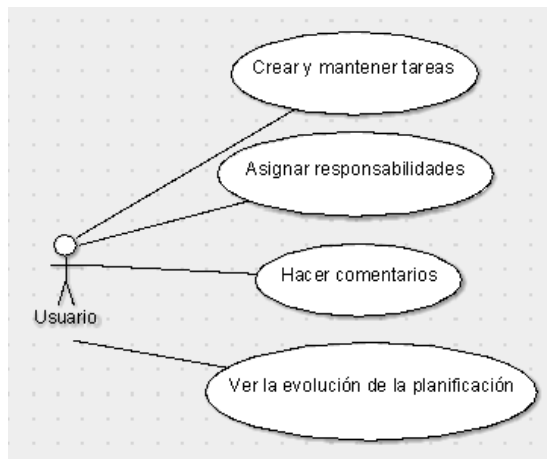
#### Casos de uso

Actividades generales que podrá hacer un usuario.



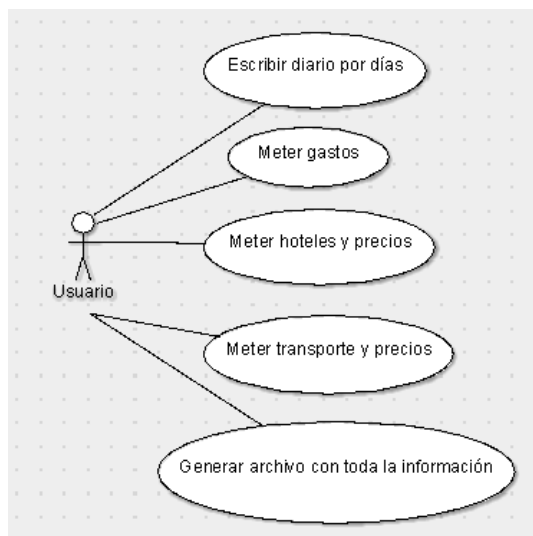
**Imagen 12: Casos de uso generales**

Actividades que podrá hacer un usuario en el planificador.



**Imagen 13: Casos de uso del planificador**

Actividades que podrá hacer un usuario en el generador de guías.



**Imagen 14: Casos de uso del Diario**

Actividades que podrá hacer el administrador de la aplicación.

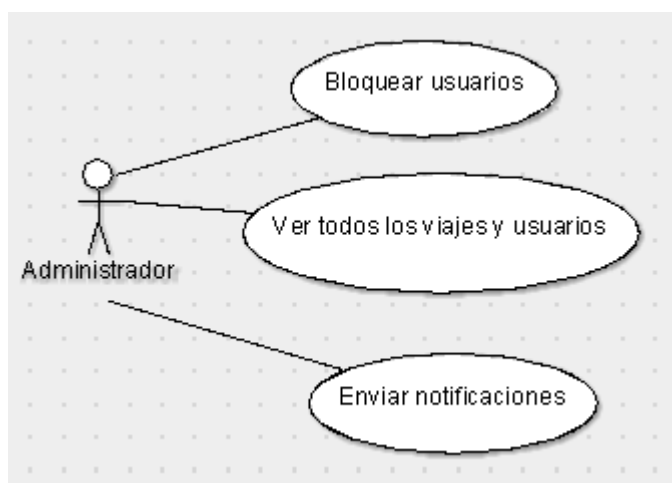


Imagen 15: Casos de uso del administrador

<b>Caso de uso</b>	Crear cuenta
<b>Actores</b>	Cliente
<b>Propósito</b>	Crear una cuenta, para poder entrar en la aplicación
<b>Resumen</b>	Registrarse en la aplicación creando una cuenta con el email y una contraseña
<b>Precondiciones</b>	
Ninguna	
<b>Flujo principal</b>	
Entrar en la aplicación, pinchar en el link de crear cuenta, meter los datos y dar al botón registrar.	
<b>Excepciones</b>	
No dejar registrar si los datos están vacíos.	

<b>Caso de uso</b>	Loggear
<b>Actores</b>	Cliente
<b>Propósito</b>	Entrar en la aplicación
<b>Resumen</b>	Meter el email y la contraseña correctas para validar y entrar en la aplicación
<b>Precondiciones</b>	
Estar registrado en el sistema	
<b>Flujo principal</b>	
Entrar en la aplicación, teclear email y contraseña y darle al botón entrar	
<b>Excepciones</b>	
No poder loggear, por meter datos erróneos	

<b>Caso de uso</b>	Crear viaje
<b>Actores</b>	Cliente
<b>Propósito</b>	Crear un viaje para poder planificar o escribir el diario
<b>Resumen</b>	Crear una cabecera con los datos generales del viaje para después poder planificar o escribir el diario
<b>Precondiciones</b>	



Estar logeado
<b>Flujo principal</b>
Entrar en la aplicación, y pinchar al link de crear viaje. Rellenar los datos y guardar.
<b>Excepciones</b>
Ninguno

<b>Caso de uso</b>	Crear guías de viajes
<b>Actores</b>	Cliente
<b>Propósito</b>	Poder generar una guía del viaje para futuros viajantes
<b>Resumen</b>	Se va creando un diario, y al final hay una opción de generar guía con los datos que tú quieras que aparezcan
<b>Precondiciones</b>	Estar logeado, tener creado el viaje y tener escrito el diario.
<b>Flujo principal</b>	Escribir el diario, metiendo los datos del lugar, hotel, transporte, gastos y comentarios, elegir qué datos queréis que aparezcan en la guía y generar un pdf.
<b>Excepciones</b>	No poder generar porque no hay datos

<b>Caso de uso</b>	Planificar viaje
<b>Actores</b>	Cliente
<b>Propósito</b>	Poder planificar el viaje con tus compañeros
<b>Resumen</b>	Se identifican una serie de tareas, con las fechas de vencimiento y responsables para poder planificar el viaje y hacer el seguimiento de ello.
<b>Precondiciones</b>	Estar logeado y tener creado el viaje.
<b>Flujo principal</b>	Crear viaje, darle al botón planificar, crear diferentes tareas asignando el responsable y la fecha de vencimiento. Ir comentado las tareas
<b>Excepciones</b>	No poder planificar el viaje porque no está creado el viaje

## Clases

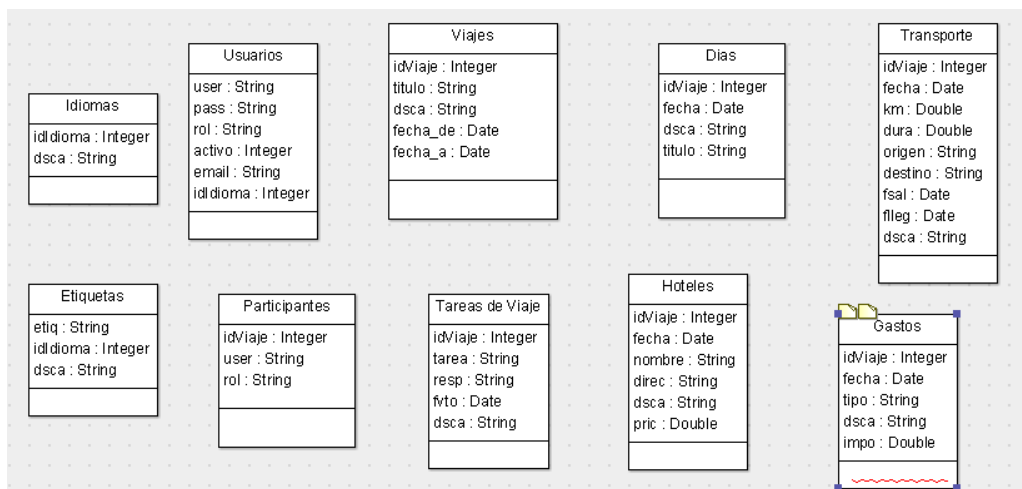


Imagen 16: Diagrama clases

### 3.3. Usabilidad/UX (DCU)

En este apartado se describe a qué personas está orientada la aplicación y se citan algunos requisitos de usuario.

#### **Perfil de usuario:**

Esta aplicación está orientada a personas adultas que les gusta viajar por su cuenta. Aproximadamente desde los 18-20 años hasta 60 años. Se supone que a los menores de 18 años no se les permite hacer un viaje por su cuenta y que los mayores de 60 años prefieren los viajes organizados. Siempre y cuando teniendo en cuenta las excepciones.

En este rango de edad, la mayoría de las personas están familiarizadas con el ordenador y con internet, pero aún y todo se intentará hacer una aplicación sencilla y lógica.

#### **Historias de usuario:**

- 1) Como usuario quiero planificar mi viaje, identificando las tareas y poniendo responsables.
- 2) Como usuario, quiero un sitio donde poder planificar mi viaje con mis compañeros.
- 3) Como usuario quiero hacer el seguimiento de la planificación de mi viaje, pudiendo mirar lo que está pendiente.
- 4) Como usuario, quiero escribir un diario del viaje, incluyendo todos los datos posibles.
- 5) Como usuario, quiero anotar los hoteles y transportes que he usado durante el viaje.
- 6) Como usuario, quiero crear una guía con la experiencia de mi viaje para poder orientar a mis amigos.
- 7) Como usuario, quiero llevar la cuenta de los gastos durante el viaje.

### 3.4. Arquitectura [5]

En este proyecto la arquitectura que se usa es el modelo Cliente/Servidor porque es una aplicación web.

Una aplicación Web es proporcionada por un servidor Web y utilizada por usuarios que se conectan desde cualquier punto vía clientes Web (browsers o navegadores). La arquitectura de un sitio Web tiene tres componentes principales:

- Un servidor Web
- Una conexión de red
- Uno o más clientes

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan.



Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la información en su sistema de archivos (base de datos) y la envía de vuelta al navegador que la solicitó.

### Cliente

Todos los usuarios que se conecten a la web mediante un navegador, sea mediante un pc, Tablet, Smartphone...

**Servidor:** Apache Tomcat

**Base de datos:** MySql

### Librerías de terceros:

Estas son las librerías que se han usado en el desarrollo. Todas son de terceros.

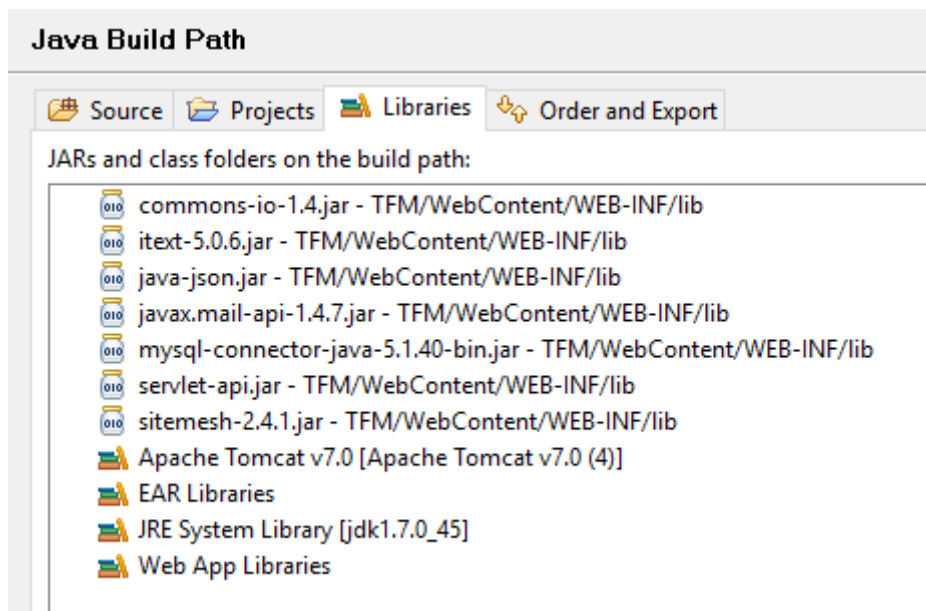


Imagen 17: Librerías

**Common-io-1.4.jar** → Se usa para la manipulación de los ficheros

**Itext-5.0.6.jar** → Se usa para generar el PDF

**Java-json.jar** → Se usa para el tipo de dato json

**Javax.mail-api-1.4.7.jar** → Se usa .para mandar emails

**Mysql-connector-java-5.1.40-bin.jar** → Se usa para hacer la conexión con la base de datos mysql

**Servlet-api.jar** → Se usa para la comunicación con los servlets

**Sitemesh-2.4.1.jar** → Se usa para los decorators. Para sacar a un jsp aparte lo que se repite en muchas jsps y no tener que escribir muchas veces.

## 4. Desarrollo

En este apartado se citan los puntos más destacados del todo el desarrollo. Primero se describen las tablas de la base de datos. Después se habla de diferentes ficheros usados en la aplicación y se pueden ver los puntos más importantes. Al final, se mencionan las decisiones de seguridad que se han tomado y los test que se han hecho para verificar la aplicación.

### 4.1. Base de datos

Se ha creado una base de datos llamada *viajes* y en ella diez tablas.

```
Tables_in_viajes |
+-----+
alojamiento
comentarios
días
gastos
generarguia
tareas
transporte
usuarios
viajeros
viajes
```

Imagen 18: Tablas de la base de datos

#### Alojamiento

En esta tabla se almacenarán los alojamientos utilizados durante el viaje. Se podrá almacenar un alojamiento por día.

```
mysql> describe alojamiento;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idViaje    | int(11)       | NO   | PRI | 0        |       |
| fecha      | date          | NO   | PRI | 0000-00-00 |       |
| nombre     | varchar(60)   | YES  |     | NULL    |       |
| direccion  | varchar(200)  | YES  |     | NULL    |       |
| dsca       | varchar(500)  | YES  |     | NULL    |       |
| precio     | double        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

Imagen 19: Tabla Alojamiento

#### Comentarios

En esta tabla se almacenarán los comentarios de las tareas. Se podrán hacer X comentarios por tarea y se guardará quien y cuando lo ha hecho. Las tareas se crearán en la planificación del viaje.

```
mysql> describe comentarios;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idViaje | int(11) | NO | PRI | 0 | |
| idTarea | int(11) | NO | PRI | 0 | |
| idComent | int(11) | NO | PRI | 0 | |
| texto | varchar(300) | YES | | NULL | |
| fecha | date | YES | | NULL | |
| email | varchar(60) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

Imagen 20: Tabla Comentarios

## Dias

Cada viaje tiene un apartado de diario. En esta tabla se guardarán los datos generales de cada día.

```
mysql> describe dias;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idViaje | int(11) | NO | PRI | 0 | |
| fecha | date | NO | PRI | 0000-00-00 | |
| titulo | varchar(60) | YES | | NULL | |
| dsca | varchar(1000) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

Imagen 21: Tabla Días

## Gastos

En esta tabla se almacenarán los gastos hechos cada día del viaje. Se clasificarán por tipo y se podrán guardar más de uno.

```
mysql> describe gastos;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idViaje | int(11) | NO | PRI | 0 | |
| fecha | date | NO | PRI | 0000-00-00 | |
| tipo | varchar(12) | YES | | NULL | |
| dsca | varchar(60) | YES | | NULL | |
| impo | double | YES | | NULL | |
| idGastos | int(11) | NO | PRI | 0 | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.06 sec)
```

Imagen 22: Tablas Gastos

## Generarguia

Cuando se termine el diario se podrá generar una guía con los datos del diario. Antes de generarlo se podrá elegir que datos publicar y cuales no. En esta tabla se guardarán si son campos para publicar o no.

Field	Type	Null	Key	Default	Extra
idViaje	int(11)	NO	PRI	0	
titulo	tinyint(1)	YES		NULL	
dsca	tinyint(1)	YES		NULL	
participantes	tinyint(1)	YES		NULL	
fechas	tinyint(1)	YES		NULL	
fechaDia	tinyint(1)	YES		NULL	
dscaDia	tinyint(1)	YES		NULL	
tituloDia	tinyint(1)	YES		NULL	
nombreAloj	tinyint(1)	YES		NULL	
direccionAloj	tinyint(1)	YES		NULL	
precioAloj	tinyint(1)	YES		NULL	
dscaAloj	tinyint(1)	YES		NULL	
tipoTrans	tinyint(1)	YES		NULL	
origenTrans	tinyint(1)	YES		NULL	
destinoTrans	tinyint(1)	YES		NULL	
duracionTrans	tinyint(1)	YES		NULL	
dscaTrans	tinyint(1)	YES		NULL	
tipoGastos	tinyint(1)	YES		NULL	
dscaGastos	tinyint(1)	YES		NULL	
importeGastos	tinyint(1)	YES		NULL	
precioTrans	tinyint(1)	YES		NULL	
generado	tinyint(1)	YES		NULL	

Imagen 23: Tabla GenerarGuia

## Tareas

En la parte de la planificación del viaje, se podrán crear diferentes tareas asignando su responsable. En esta tabla se guardarán estas tareas.

```
mysql> describe tareas;
```

Field	Type	Null	Key	Default	Extra
idViaje	int(11)	NO	PRI	0	
idTarea	int(11)	NO	PRI	0	
dsca	varchar(60)	YES		NULL	
resp	varchar(60)	YES		NULL	
fvto	date	YES		NULL	
estado	varchar(3)	YES		NULL	

6 rows in set (0.02 sec)

Imagen 24: Tabla Tareas

## Transporte

En esta parte se guardarán los transportes usados cada día del viaje. Se podrán guardar más de uno y se clasificarán por tipo.

```
mysql> describe transporte;
```

Field	Type	Null	Key	Default	Extra
idViaje	int(11)	NO	PRI	0	
fecha	date	NO	PRI	0000-00-00	
idTrans	int(11)	NO	PRI	0	
origen	varchar(30)	YES		NULL	
destino	varchar(30)	YES		NULL	
duracion	double	YES		NULL	
dsca	varchar(200)	YES		NULL	
tipo	varchar(10)	YES		NULL	
precio	double	YES		NULL	

9 rows in set (0.01 sec)

Imagen 25: Tabla Transporte

## Usuarios

En esta tabla se guardarán los usuarios registrados en la aplicación.

```
mysql> describe usuarios;
```

Field	Type	Null	Key	Default	Extra
pass	varchar(20)	NO		NULL	
email	varchar(60)	NO	PRI	NULL	
nombre	varchar(40)	NO		NULL	
idioma	varchar(2)	YES		NULL	
cambiopass	varchar(1)	YES		NULL	

5 rows in set (0.03 sec)

Imagen 26: Tabla Usuarios

## Viajeros

En esta tabla se almacenarán los participantes del viaje.

```
mysql> describe viajeros;
```

Field	Type	Null	Key	Default	Extra
idViaje	int(11)	YES		NULL	
email	varchar(60)	YES		NULL	

2 rows in set (0.01 sec)

Imagen 27: Tabla Viajeros

## Viajes

En esta tabla se guardarán los datos generales del viaje.

```
mysql> describe viajes;
```

Field	Type	Null	Key	Default	Extra
idViaje	int(6)	NO	PRI	NULL	auto_increment
titulo	varchar(100)	YES		NULL	
dsca	varchar(300)	YES		NULL	
desde	date	YES		NULL	
hasta	date	YES		NULL	

5 rows in set (0.01 sec)

Imagen 28: Tablas Viajes

### 4.2. Extractos de código

Esta aplicación principalmente está creada con páginas JavaServer (JSP) y servlets, pero tienen más de un tipo de fichero.

En este apartado se hace un resumen de todos los ficheros usados mostrando unos ejemplos.

### 1.1.1. Llamadas a servlets

Para hacer las llamadas a servlets generalmente se ha usado el método *post* para evitar que los parámetros se vean en la URL. Estas llamadas se han hecho de dos formas diferentes. Uno, con el formulario normal:

```
<form method="post" action="register">
```

Imagen 29: Form

Y dos, mediante AJAX:

```
$(document).on('click', '#anadirTrans', function() {  
    var idViaje = $("#idViaje").val();  
    var fecha = $("#fecha_datepicker").val();  
    var sizeTrans = $("#sizeTrans").val();  
    $.ajax({  
        type: "POST",  
        url: "diarioServlet",  
        data: "funcion=anadirTrans&idViaje="+idViaje+"&fecha="+fecha+"&sizeTrans="+sizeTrans,  
        dataType: "json",  
        success: function (response) {  
            $(".tablaTrans").append(response.linea);  
            $("#sizeTrans").val(response.sizeTrans);  
        },  
        failure: function (response) {  
            alert("Error");  
        }  
    });  
});
```

Imagen 30: Ajax

### 1.1.2. Páginas JAVA (Clases)

Hay muchas clases generadas. Algunos ejemplos:

- Clases con diferentes consultas a la base de datos, validaciones de datos, inserciones y modificaciones de datos...
- Una clase por cada tabla de la base de datos. Con métodos *get* y *set*.



```
Usuarios.java X
package general;

public class Usuarios {

    public String email;
    public String nombre;
    public String idioma;
    public String cambiopass;
    public String pass;
    public String rol;

    public String getEmail(){
        return email;
    }

    public String getNombre(){
        return nombre;
    }

    public String getIdioma(){
        return idioma;
    }

    public String getCambiopass(){
        return cambiopass;
    }

    public String getPass(){
        return pass;
    }

    public String getRol(){
        return rol;
    }
}
```

Imagen 31: Clase Usuarios

```
public void setEmail(String email){
    this.email = email;
}

public void setNombre(String nombre){
    this.nombre = nombre;
}

public void setIdioma(String idioma){
    this.idioma = idioma;
}

public void setCambiopass(String cambiopass){
    this.cambiopass = cambiopass;
}

public void setPass (String pass){
    this.pass = pass;
}

public void setRol (String rol){
    this.rol = rol;
}
}
```

Imagen 32: Clase Usuarios 2

- Clase para enviar emails *MailSender.java* [9]
- Clase para generar PDFs *GenerarPDF.java* [10]

```

public void createPDF(File pdfNewFile, int idViaje) throws FileNotFoundException, DocumentException {
    // We create the document and set the file name.
    // Creamos el documento e indicamos el nombre del fichero.
    Viajes viaje = ConsultaViajes.getViaje(idViaje);
    LinkedList<String> viajeros = ConsultaViajes.getViajeros(idViaje);
    GenerarGuia generarGuia = ConsultaDiario.getGenerarGuia(idViaje);
    Document document = new Document();

    PdfWriter.getInstance(document, new FileOutputStream(pdfNewFile));

    document.open();
    // Añadimos los metadatos del PDF
    document.addTitle("Viaje a "+viaje.getTitulo());
    document.addSubject("Using iText (usando iText)");
    document.addKeywords("viaje, "+viaje.getTitulo());
    document.addAuthor("Código Xules");
    document.addCreator("Código Xules");

    // Primera página
    Chunk chunk = new Chunk("");
    if (generarGuia.getTitulo()) { chunk = new Chunk(viaje.getTitulo(), chapterFont); }

    Chapter chapter = new Chapter(new Paragraph(chunk), 1);
    chapter.setNumberDepth(0);
    if (generarGuia.getFechas()) { chapter.add(new Paragraph(viaje.getDesde() + " / " + viaje.getHasta(), paragraphFont)); }

    Image image;
    try {
        image = Image.getInstance("C:/TFM/TFM/WebContent/images/viajes/"+idViaje+"/cabecera.jpg");
        image.scaleAbsolute(100, 100);
        image.setWidthPercentage(50);
        chapter.add(image);
    } catch (BadElementException ex) {

```

Imagen 35: GenerarPDF

- UploadFile.java

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    boolean isMultipart = ServletFileUpload.isMultipartContent(request);
    HttpSession session = request.getSession(false);

    String email = (String)session.getAttribute("email");
    String[] parts = email.split("@");
    String part1 = parts[0];
    if (isMultipart) {
        // Create a factory for disk-based file items
        FileItemFactory factory = new DiskFileItemFactory();

        // Create a new file upload handler
        ServletFileUpload upload = new ServletFileUpload(factory);
        try {
            // Parse the request
            List /* FileItem */ items = upload.parseRequest(request);
            Iterator iterator = items.iterator();
            while (iterator.hasNext()) {
                FileItem item = (FileItem) iterator.next();
                if (!item.isFormField()) {
                    String fileName = item.getName();
                    String root = getServletContext().getRealPath("/");
                    File path = new File("C:/TFM/TFM/WebContent/images/viajes/temp");
                    if (!path.exists()) {
                        boolean status = path.mkdirs();
                    }

                    File uploadedFile = new File(path + "/" + part1 + ".jpg");
                    System.out.println(uploadedFile.getAbsolutePath());
                    item.write(uploadedFile);
                }
            }
        } catch (FileUploadException e) {

```

Imagen 36: UploadFile

- Las páginas JSPs son bastante complejas, esto es, la mayoría tienen opción de hacer más de una acción y por no crear un servlet por cada acción, se les pasa un parámetro llamado *función* para que haga una cosa u otra (ver imagen 30).

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String funcion = request.getParameter("funcion");

    String idViaje = request.getParameter("idViaje");
    String fecha = request.getParameter("fecha");

    HttpSession session = request.getSession(false);
    String idioma = ConsultaUsuarios.getUsuario((String) session.getAttribute("email")).getIdioma();
    ResourceBundle labels = ResourceBundle.getBundle(idioma, new Locale(idioma));

    if(funcion.equals("borrar")){
        try {
            borrarDia(Integer.parseInt(idViaje), fecha);
            borrarAlojamiento(Integer.parseInt(idViaje), fecha);
            borrarTransportes(Integer.parseInt(idViaje), fecha);
            borrarGastos(Integer.parseInt(idViaje), fecha);
        } catch (NumberFormatException | SQLException e) {
            e.printStackTrace();
        }
    }
    else if(funcion.equals("borrarTrans")){
        String idTrans = request.getParameter("idTrans");
        try {
            borrarTransporte(Integer.parseInt(idViaje), fecha, Integer.parseInt(idTrans));
        } catch (NumberFormatException | SQLException e) {
            e.printStackTrace();
        }
    }
    else if(funcion.equals("borrarGastos")){
        String idGastos = request.getParameter("idGastos");
        try {
            borrarGasto(Integer.parseInt(idViaje), fecha, Integer.parseInt(idGastos));
        } catch (NumberFormatException | SQLException e) {
            e.printStackTrace();
        }
    }
    else if(funcion.equals("borrar")){

```

Imagen 37: Servlet

### 3.3.3. Páginas XML

- **Web.xml** → Contiene información sobre la estructura y las dependencias externas de los componentes web en el módulo y describe cómo se utilizan los componentes en tiempo de ejecución [8]. Se detallan los servlets de la aplicación.

```

<servlet>
  <servlet-name>login</servlet-name>
  <servlet-class>conexion.Login</servlet-class>
</servlet>
<servlet>
  <servlet-name>logout</servlet-name>
  <servlet-class>conexion.Logout</servlet-class>
</servlet>
<servlet>
  <servlet-name>register</servlet-name>
  <servlet-class>conexion.Register</servlet-class>
</servlet>
<servlet>
  <servlet-name>recordarPass</servlet-name>
  <servlet-class>conexion.RecordarPass</servlet-class>
</servlet>
<servlet>
  <servlet-name>viajesServlet</servlet-name>
  <servlet-class>viajes.ViajesServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>usuariosServlet</servlet-name>
  <servlet-class>general.UsuariosServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>tareasServlet</servlet-name>
  <servlet-class>planificar.TareasServlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>diarioServlet</servlet-name>
  <servlet-class>diario.DiarioServlet</servlet-class>
</servlet>

```

Imagen 38: Web.xml

- **Decorators.xml** → Es para definir los patrones que se van a usar y a qué páginas JSP les tiene que cargar [15]. En esta aplicación se ha creado el patrón *estructura.jsp*.

```

<?xml version="1.0" encoding="UTF-8"?>
<decorators defaultdir="/decorators">
  <excludes>
    <pattern>/index.jsp*</pattern>
    <pattern>/registrar.jsp*</pattern>
    <pattern>/recordar.jsp*</pattern>
  </excludes>
  <decorator name="estructura" page="../estructura.jsp">
    <pattern>/crearViaje.jsp*</pattern>
    <pattern>/planificar.jsp*</pattern>
    <pattern>/diario.jsp*</pattern>
    <pattern>/misViajes.jsp*</pattern>
    <pattern>/misGuias.jsp*</pattern>
    <pattern>/miCuenta.jsp*</pattern>
    <pattern>/tablon.jsp*</pattern>
  </decorator>
</decorators>

```

Imagen 39: Decorators.xml

### 3.3.4. Páginas JSP

Las páginas JSP se ha intentado que sean puro HTML, llevando toda la lógica a los servlet y a otras clases. Para destacar decir que todas están entre un *try-catch* para tener controlados los errores.

La más especial es *estructura.jsp* que es el patrón que se usa en la mayoría de las JSPs.

```

<?xml | JFM/WebContent/estructura.jsp [UTF-8" ?]
<%@ taglib uri="http://www.opensymphony.com/sitemesh/decorator" prefix="decorator" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%try( %>
<html xmlns="http://www.w3.org/1999/xhtml">
<%@ page import = "java.util.*" %>
<%@ page import = "general.*"%>
<head>
  <decorator:head />
  <title>Bidaiaitzen</title>
</head>
<body>
  <%String idioma = ConsultaUsuarios.getUsuario((String)session.getAttribute("email")).getIdoma();
  ResourceBundle labels = ResourceBundle.getBundle(idioma, new Locale(idioma));%>
  <div id="panelwrap">
    <div class="header">
      <div class="title"><a href="#">Bidaiaitzen</a></div>
      <div class="header_right"><%=labels.getString("bienvenido") %> <%=session.getAttribute("email") %>, <a href="miCuenta.jsp" class="settings"><%=lab
    <div class="menu">
      <ul>
        <li><a href="tablon.jsp" class="selected"><%=labels.getString("home") %></a></li>
        <li><a href="misViajes.jsp"><%=labels.getString("mis.viajes") %></a></li>
        <li><a href="misGuias.jsp"><%=labels.getString("mis.guias") %></a></li>
      </ul>
    </div>
  </div>
</div>

```

Imagen 40: Estructura.jsp 1

```

<div class="submenu">
  <ul>
  </ul>
</div>

<div class="center_content" >
  <div id="right_vrsp" >
  <div id="right_content">
    <decorator:body />
  </div><!-- end of right content-->
  </div>
  <div class="clear"></div>
</div><!-- end center_content-->

<div class="footer"></div>
</div>
</body>
</html>
</>catch(Exception e){
  System.out.println("Error estructura *e");
}

```

Imagen 41: Estructura.jsp 2

### 3.3.5. Páginas JAVASCRIPT

Todos los ficheros javascript son de terceros menos uno que se ha creado porque se necesitaba en varios jsp-s.

#### Terceros:

- <https://code.jquery.com/jquery-1.12.4.js> → Para que funcionen llamadas jquery
- <https://code.jquery.com/ui/1.12.1/jquery-ui.js> → Para que funcione llamadas jquery
- <https://cdn.jsdelivr.net/jquery.validation/1.15.0/jquery.validate.min.js> → Para que funcione la funcion validate
- <https://cdn.jsdelivr.net/jquery.validation/1.15.0/additional-methods.min.js> → Para que funcione la funcion validate
- **Avatar.js y Resample.js** → Se usa para pintar la imagen principal del viaje después de subirlo al servidor.
- **Jquery.ajaxfileupload.js** → Se usa para subir el archivo al servidor
- **Jquery.tabify.js** → Se usa para el funcionamiento de las pestañas.

#### Propios:

- **viajeJs.js** → Las funciones para guardar los datos principales del viaje se repiten en varios jsp-s y por eso se ha creado este fichero.

Se ha hecho uso de las funciones de Jquery UI. Se ha utilizado la función *autocomplete*[11], *dialog*[12] y *datepicker*[13]. También se ha usado el jquery validation plugin[14].

#### Autocomplete

```

$(document).on('click', 'input[name=viajero]', function() {
  $('input[name=viajero]').each(function() {
    $(this).autocomplete({
      source: "UsuariosServlet",
      minLength: 2,
      select: function(event,ui) { }
    });
  });
});

```

Imagen 42: Autocomplete

## Dialog

```
$(document).on('click','#borrarDia a',function(){
    var id = $(this).attr('id');
    var fecha = $("#fecha_"+id).html();
    var idViaje = $("#idViaje").val();
    $("#avisoBorrar").dialog({
        modal: true,
        title: "Eliminar dia",
        width: 400,
        height:"auto",
        hide: "scale",
        buttons: {
            "Aceptar": function() {
                $.ajax({
                    type: "POST",
                    url: "diarioServlet",
                    data: "funcion=borrar&idViaje="+idViaje+"&fecha="+fecha,
                    dataType: "html",
                    success: function (response){
                        $("#linea"+fecha).remove();
                    },
                    failure: function (response) {
                        alert("Error");
                    }
                });
                $( this ).dialog( "close" );
            },
            "Cancelar": function() {
                $( this ).dialog( "close" );
            }
        }
    });
});
```

Imagen 42: Dialog

## Datepicker

```
$.datepicker.setDefaults($.datepicker.regional['es']);
$( "#fvto_datepicker" ).datepicker();
$( "#desde_datepicker" ).datepicker();
$( "#hasta_datepicker" ).datepicker();
```

Imagen 43: Datepicker

## Validation

```

jQuery.validator.defaults({
  debug: true,
  success: "valid"
});
$("#myForm").validate({
  rules: {
    fvto_datepicker: {
      required: true,
      date: true,
    },
    resp: {
      required: true,
    },
    dsca: {
      required: true,
    },
    desde: {
      required: true,
      date: true,
    },
    hasta: {
      required: true,
      date: true,
    }
  }
});

```

Imagen 44: Validate

### 3.3.6. Páginas CSS

#### Terceros:

- **960\_rtl.css, Reset.css y Text.css** → Para dimensionar el contenido de la página. Divide la página en X huecos y los elementos se colocan en esos huecos [6]. Se ve claramente usado en *crearViaje.jsp*

```

<div class="container_12">
  <div class="grid_3">
    <div class="loginform_row">
      <a href="#" id="anadirViajero"></a>
    </div>
  </div>

  <div class="grid_6" id="participantes">
    <label><%=Labela.getString("participantes") %></label><span class="nota">{ <%=Labela.getString("no.es.necesario.incluir.su.usuario") %></span>
    <%=for(int i=0; i<viajerosSize; i++)%>
      String atributos = "";
      if(viajeros.get(i).equals(email)){
        atributos = "disabled";
      }
      <div class="loginform_row">
        <input type="text" class="registform_input" name="viajero" id="viajero<i>1 %>" value="<%=viajeros.get(i) %>" size="60" maxlength="100" />
      </div>
    <%=>
    <%=if(viajerosSize == 0){
      viajerosSize = 1;%>
      <div class="loginform_row">
        <input type="text" class="registform_input" name="viajero" id="viajero0" value="" size="60" maxlength="100" />
      </div>
    <%= %>
    <input type="hidden" value="<%=viajerosSize%>" id="corrViajero"/>
  </div>

  <div class="grid_3">
    <input type="file" name="datafile" id="uploader" class="uploader" /><br/>
  </div>
</div>
<div class="clear"></div>
</div>

```

Imagen 45: CrearViaje.jsp

```

<div class="container_12">
  <div class="suffix_10_grid_1_prefix_1">
    <div class="loginform_row">
      <input type="submit" class="loginform_submit" value="{%=labels.getString("guardar") %}" id="guardarViaje"/>
    </div>
  </div>
  <div class="clear"></div>
</div>
<br/><br/>
<div class="container_12" id="plani_diario_imagen" >
  <div class="grid_4_prefix_1">
    <a href="#" id="diario"></a>
  </div>
  <div class="suffix_3_grid_4">
    <a href="#" id="planificar"></a>
  </div>
  <div class="clear"></div>
</div>
<div class="container_12" id="plani_diario_texto">
  <div class="suffix_2_grid_3_prefix_1">
    <label style="text-align:center">{%=labels.getString("diario") %}</label>
  </div>
  <div class="suffix_4_grid_2">
    <label>{%=labels.getString("planificar") %}</label>
  </div>
  <div class="clear"></div>
</div>

```

Imagen 46: CrearViaje.jsp 2

- **Style.css** → Para hacer la página web se ha basado en una plantilla descargada por internet [7]. Esta es la hoja de estilo de esa plantilla, con unas modificaciones propias y es la hoja de estilo principal de esta aplicación.
- <https://jqueryvalidation.org/files/demo/site-demos.css>
- [//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css](https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css)

### 3.3.7. Ficheros PROPERTIES

Se han creado tres ficheros *properties*, uno por cada idioma para hacer la aplicación multiidioma. *EN.properties* (ingles), *ES.properties* (español) y *EU.properties* (vasco). Los usuarios registrados en la aplicación pueden cambiar el idioma.

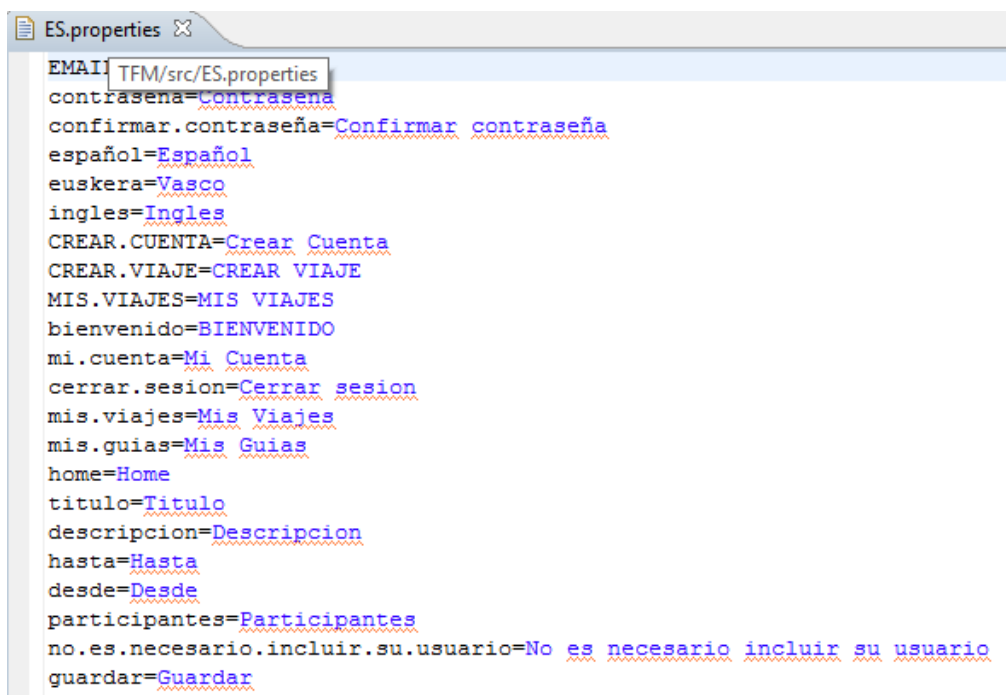


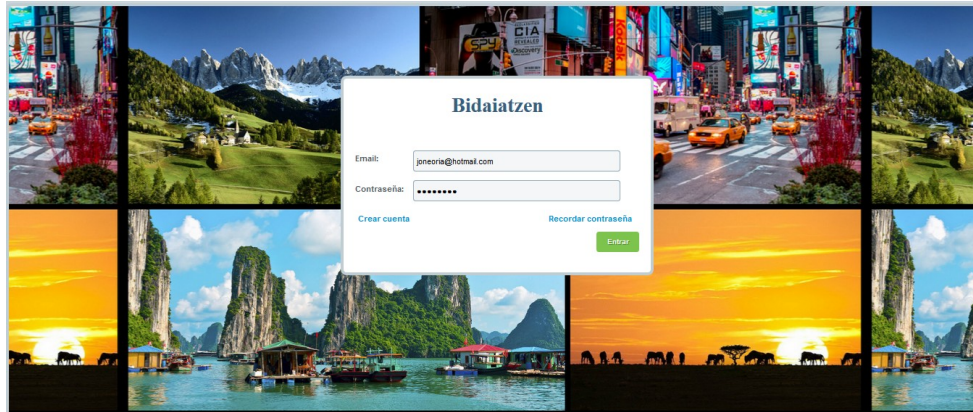
Imagen 47: Properties



### 4.3. Pantallazos finales

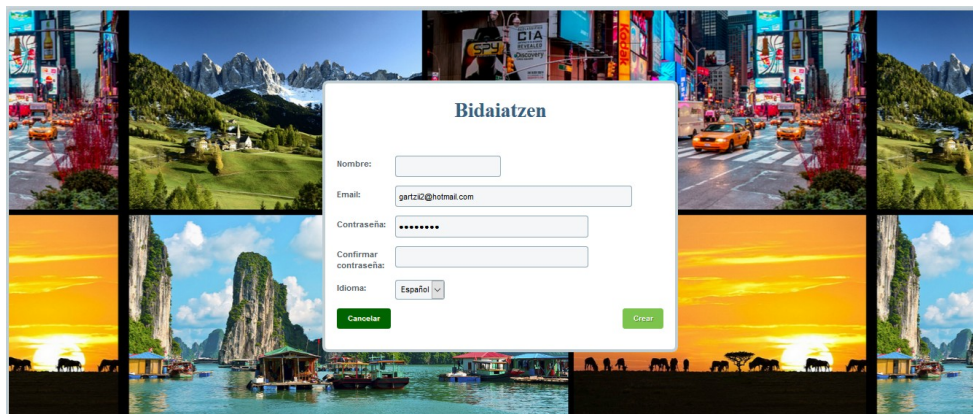
Abajo los pantallazos correspondientes a cada wireframe. Se puede ver claramente el aspecto final de la aplicación web.

#### ***Login***



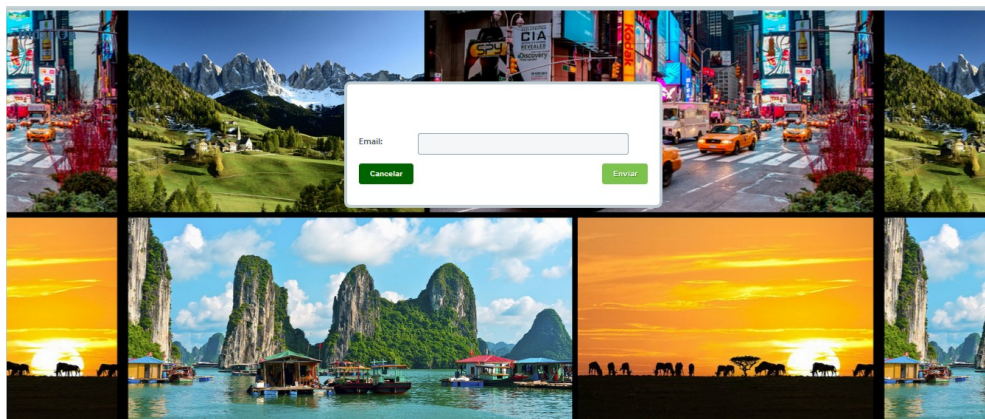
**Imagen 48: Login**

#### ***Registrar***



**Imagen 49: Registrar**

#### ***Recordar contraseña***



**Imagen 50: Recordar Contraseña**

## Home

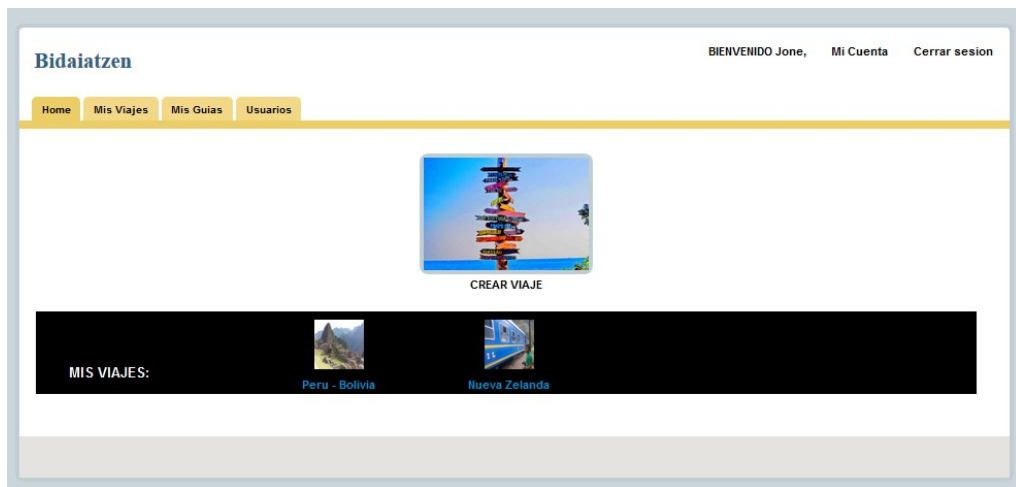


Imagen 51: Home

## Crear Viaje

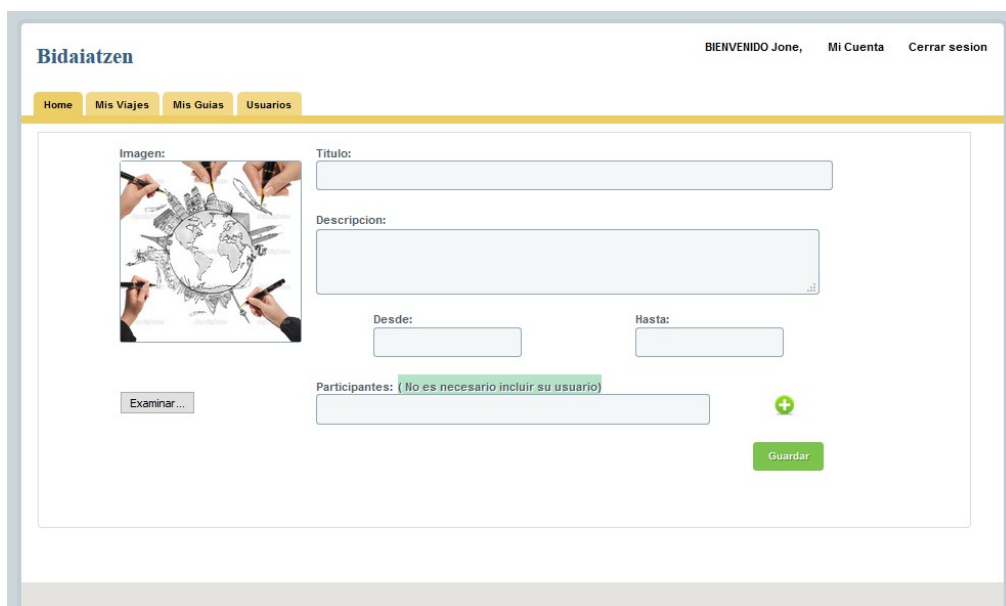



Imagen 52: Crear Viaje

## Editar Viaje

Editar Viaje

Imagen: 

Título: Peru - Bolivia

Descripcion: Mikel eta Jone 2016ko Abuztuan

Desde: 2016-08-06 Hasta: 2016-08-27

Participantes: ( No es necesario incluir su usuario )

- gartzia@hotmail.com
- joneoria@hotmail.com

Examinar... Guardar

Imagen 53: Editar Viaje

## Mis Viajes

Bidaiatzen BIENVENIDO Jone, Mi Cuenta Cerrar sesion

Home Mis Viajes Mis Guias Usuarios




ID	Titulo	Descripcion	Desde	Hasta	Planificar	Diario	Borrar
2	Peru - Bolivia	Mikel eta Jone 2016ko Abuztuan	2016-08-06	2016-08-27			
3	Nueva Zelanda	Astesantutan 2017	2017-04-10	2017-04-30			

Imagen 54: Mis Viajes

## Mis Guias

Bidaiatzen BIENVENIDO Jone, Mi Cuenta Cerrar sesion

Home Mis Viajes Mis Guias Usuarios








ID	Titulo	PDF	Editar	Borrar
2	Peru - Bolivia			
3	Nueva Zelanda			

Imagen 55: Mis Guías

## Planificar

**Bidaiatzen** BIENVENIDO Jone, Mi Cuenta Cerrar sesion

Home Mis Viajes Mis Guias Usuarios



### Peru - Bolivia (2016-08-06 / 2016-08-27)

Mikel eta Jone 2016ko Abuztuan

Viajeros: [gartzi2@hotmail.com, joneoria@hotmail.com]

[Editar](#)

**Tareas**

Tarea	Descripcion	Resp	F.Vto	Estado	Editar	Borrar
1	Reservar vuelo	joneoria@hotmail.com	2016-06-01	CER		
2	Reservar hotel	gartzi2@hotmail.com	2016-07-20	CER		
3			null	PEN		

[Crear Tarea](#)

**Imagen 56: Planificar 1**

**Tarea 1**

Descripcion:

Responsable:

Fecha Vencimiento:

Estado:

[Guardar](#)

**Comentarios**

Comentario:


[Enviar](#)

**Imagen 57: Planificar 2**

**Diario**

**Bidaiatzen** BIENVENIDO Jone, Mi Cuenta Cerrar sesion

Home Mis Viajes Mis Guias Usuarios



### Peru - Bolivia (2016-08-06 / 2016-08-27)

Mikel eta Jone 2016ko Abuztuan

Viajeros: [gartzi2@hotmail.com, joneoria@hotmail.com]

[Editar](#)

**Diario de viaje**

Fecha	Titulo	Editar	Borrar
2016-06-07	Cuzco		
2016-06-08	Valles Sagrados		

[Crear dia](#)

**Imagen 58: Diario 1**

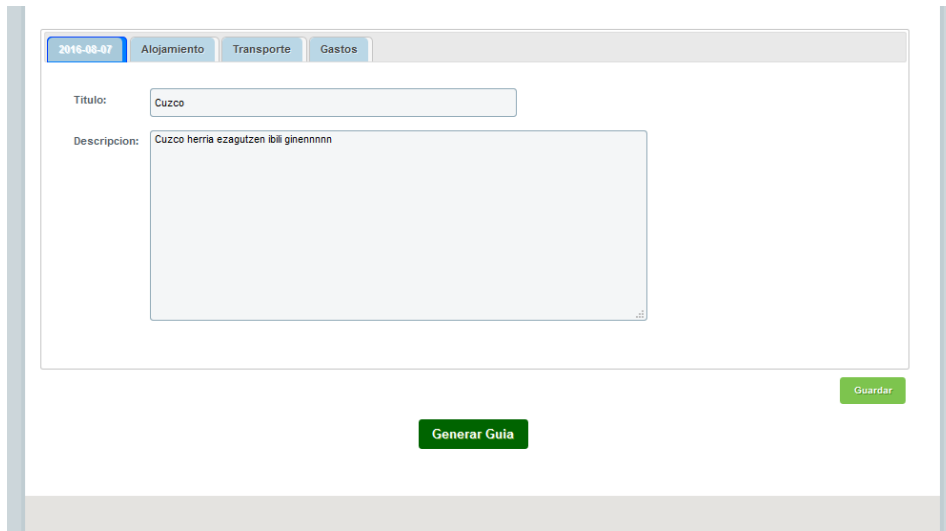


Imagen 59: Diario 2

## Crear Guia

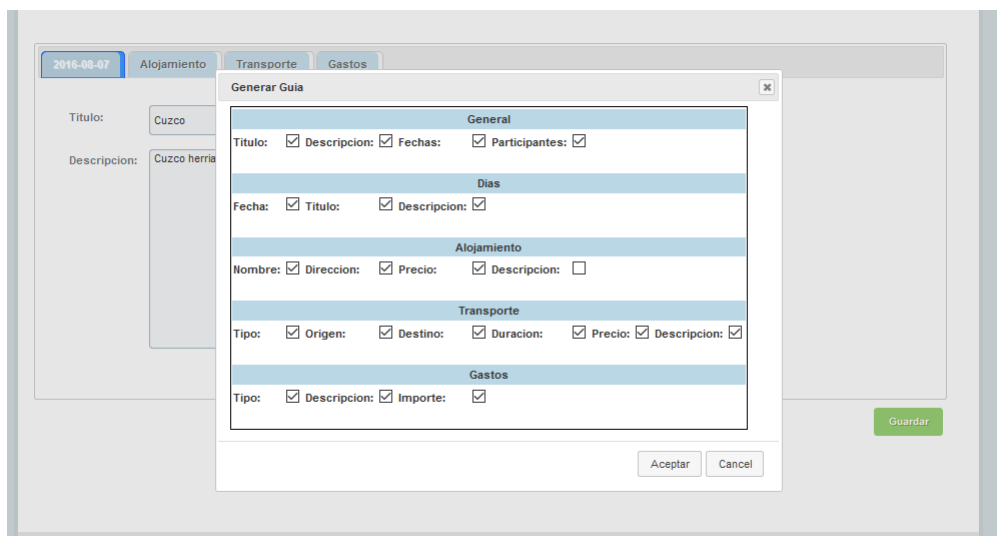


Imagen 60: Crear Guía

## 4.4. Seguridad

Principalmente, las decisiones de seguridad que se han tomado son a nivel de inserción de datos y permisos de usuario. No se ha visto por ejemplo, la necesidad de hacer copias de seguridad porque esta aplicación todavía no está en marcha y solo se puede acceder localmente.

Estas son las medidas que se han tomado:

- Controlar el tipo de dato que se mete (número, fecha, string...) en todas las queries de la base de datos.
- Controlar los permisos de cada usuario para que cada uno pueda ver solo lo suyo.

## 4.5. Test

El test principal que se ha hecho es el de usuario para ver la funcionalidad, rendimiento, utilidad, seguridad... de la aplicación. Se han escogido tres personas con perfiles diferentes, se les ha dado una lista con las funcionalidades que deben probar y se les ha dejado manejar la aplicación sin dar ninguna explicación. Mientras, se han anotado los comentarios que han hecho con la intención de encontrar errores y mejoras.

En las siguientes tablas se pueden ver un resumen de los comentarios de cada usuario y la lista de funcionalidades con el estado de correcto si o no.

Nombre	Mikel Oria
Edad	21 años
Perfil	Estudiante
Comentarios	
<ul style="list-style-type: none"> <li>- Con el tabulador va a la fecha hasta. Hay que coger el raton para situarse en desde.</li> <li>- ¿Por qué siempre el calendario se sitúa en diciembre?</li> <li>- La imagen no se carga</li> <li>- Estaría bien poder ver visualmente mediante colores el estado de las tareas</li> <li>- Hay que avisar o notificar los campos numéricos como importe o duración.</li> <li>- El formato de las guías hay que trabajar más</li> <li>- No deja borrar las guías. No funciona.</li> <li>- No se abre el pdf.</li> </ul>	

Nombre	Mikel Garciandia
Edad	29 años
Perfil	Trabajador (Tornero)
Comentarios	
<ul style="list-style-type: none"> <li>- El calendario siempre se sitúa en diciembre. Debería de acotar las fechas.</li> <li>- Habría que haber la opción de buscar los viajeros por nombre y no por email.</li> <li>- La primera vez que rellenas la fecha vencimiento de las tareas, siempre dice que está sin rellenar. Hace mal.</li> <li>- La pregunta de si quieres borrar la tareas aparece más abajo, hay desplazarse hacia abajo.</li> <li>- En la pestaña transportes no avisa si el campo es obligatorio</li> <li>- Deja meter letras en el importe aunque no grabe.</li> <li>- Estaría bien visualizar con un color más fuerte en qué pestaña estas ("home", "mis viajes", "mis guías")</li> </ul>	

Nombre	Belen Abendibar
Edad	61 años
Perfil	Trabajador (Profesora de autoescuela)
Comentarios	
<ul style="list-style-type: none"> <li>- Estaría bien un apartado de guía de usuario explicando cómo crear un viaje desde cero. Por los menos la primera vez.</li> <li>- ¿Por qué no se carga la imagen?</li> <li>- ¿Por qué no deja borrar la tarea? Hay que desplazarse hacia abajo para ver la pregunta y poder borrar.</li> <li>- No graba transportes. Si dejas vacío sí. Hay algún error.</li> <li>- No se abre el pdf</li> </ul>	

Con los resultados obtenidos de estos tests y a partir de los casos de uso se ha definido una tabla determinando que funcionalidades se han probado con éxito y cuáles no.

Funcionalidad	¿Correcto?
Crear cuenta	Si
Loggear	Si
Recordar contraseña	Si
Crear/Modificar viaje	Si
Subir imagen	A medias. No actualiza al momento.
Crear/Modificar tareas	Si
Hacer comentarios	Si
Borrar comentarios	No
Asignar responsables	A medias. No mira si es participante del viaje. Busca todos los usuarios dados de alta
Crear días del diario	Si
Guardar/Modificar alojamiento	Si
Guardar/Modificar transporte	A medias. Deja meter letras en el campo duración y precio y luego no guarda. No debería dejar meterlo o por lo menos avisar.
Guardar/Modificar gastos	A medias. Deja meter letras en el campo duración y precio y luego no guarda. No debería dejar meterlo o por lo menos avisar.
Generar guía	A medias. Guarda bien pero no abre al generar hay que ir a la pestaña "Mis Guías" para abrirlo.
Cambiar contraseña	Si
Cambiar idioma	Si
Bloquear usuarios (admin)	No
Ver/Borrar todos los usuarios (admin)	No
Enviar notificaciones (admin)	No

A partir de estos tests se han detectado algunos errores que se mencionan en el apartado siguiente y algunas mejoras que están escritas en el apartado de conclusiones en líneas de trabajo futuro.

#### 4.6. Bugs

A partir de los tests de usuarios y la tabla de funcionalidades conseguidas, se han detectado una serie errores:

- Al cambiar el icono del viaje, no actualiza al momento. Guarda bien, pero no se actualiza hasta refrescar la aplicación.
- Al crear el viaje, si vas de un campo a otro con el tabulador primero va a la fecha "hasta" y después a "desde". Debería de ser al revés.
- En los formularios no entra al dar al botón "intro". Hay que clicar el botón.

- Según donde está posicionado en la página, el dialog sale a la vista o a veces escondido.
- En las tareas al intentar meter como responsable una persona que no es del viaje lo permite y no lo debería de hacer.
- No debería de dejar planificar tareas a fechas anteriores de hoy. Ni planificar viajes que se han hecho.
- El calendario se posiciona siempre en el día de hoy. Sería más cómodo si se posicionara en las fechas del viaje, por ejemplo.
- En la pestaña "Mis Guías" no abre el PDF hasta refrescar la aplicación.
- En los campos "duración" y "precio" no avisa al intentar meter letras. Debería de avisar que son campos numéricos.



## 5. Conclusiones

### Lecciones aprendidas

Durante el desarrollo de esta aplicación la lección más importante que se ha aprendido es que es muy muy importante el trabajo previo del desarrollo; los objetivos, la planificación y el diseño. Cuando empiezas a trabajar en un proyecto quizá no parezca que sea tan importante pero a lo largo del desarrollo te das cuenta que has ahorrado un montón de tiempo. Aun así, es muy difícil seguir a pie de la letra lo definido previamente porque van surgiendo diferentes problemas que no los tienes previstos.

A nivel de programación también se han aprendido nuevas herramientas como decorators, generador de pdfs, maquetación con grids...

### ¿Hemos logrado los objetivos planteados?

En cuanto a los objetivos se han logrado todos los objetivos planteados, tanto a nivel personal y a nivel de aplicación.

A nivel personal, se ha aprendido a resolver un proyecto completo, a planificar y a documentar.

A nivel de aplicación se puede planificar un viaje creando diferentes tareas definiendo fecha y responsables y se puede crear un diario metiendo todos los detalles.

### Seguimiento de la planificación y metodología

La metodología prevista ha sido la adecuada y durante la TFM se ha seguido como se definió al principio.

En cuanto la planificación podemos decir que los inicios y fines de las cuatro fases se han seguido correctamente porque coincidían con las entregas y eran obligatorios. Pero en el medio de cada fase, no ha sido muy preciso. Mayormente en la fase tres porque se definía que el desarrollo y la memoria iban a ir paralelamente y no ha sido así. Se le ha dedicado mayor tiempo al desarrollo que a la memoria.

### Líneas de trabajo futuro

- Corregir los bugs
- Poder subir fotos
- Crear roles entre participantes de un viaje. Ahora tienen todos los mismos permisos.
- Crear un perfil administrador de la aplicación que pueda ver toda la información que se mueve y si es necesario eliminar información o usuarios. El rol administrador está creado y ahora puede ver los usuarios pero habría que añadir más funcionalidades.

- En el diario crear otro apartado actividades.
- Crear alertas de notificación si algún compañero de viaje hace modificaciones en el viaje.
- Poner más visual cuando la fecha de vencimiento de una tarea se está acercando y todavía está pendiente
- Una vez generado el PDF poder publicar para poder ver otros usuarios que nos son miembros del viaje.
- Crear el PDF más bonito.
- Abrir el PDF cuando se genera. Ahora hay que ir a la pestaña "Mis Guías".
- Clasificar las guías generadas por país y poder buscarlos (un buscador)
- Controlar las fechas para las tareas (no tiene sentido planificar un viaje que ya se ha hecho).
- Poder encontrar los participantes por nombre y no por email.
- Hacer un guía de usuario explicando cómo crear un viaje desde cero.
- Crear planificador de rutas
- Crear versión móvil

## 6. Glosario

- **Aplicación web:** Herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de Internet mediante un navegador [18].
- **Wireframe:** un esquema de página o plano de pantalla, es una guía visual que representa el esqueleto o estructura visual de un sitio web [16].
- **JSP:** JavaServer Pages (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML y XML, entre otros tipos de documentos. Se usa el lenguaje de programación Java [17].
- **JAVA:** Es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo [19].

## 7. Bibliografía

- [1] <http://metodologiaencascada.blogspot.com.es/>. 02/10/2016
- [2] <http://es.slideshare.net/nipas/10-sgbd>. 19/10/2016
- [3] <https://es.wikipedia.org/wiki/Tomcat>. 23/10/2016
- [4] [https://www.ecured.cu/Eclipse.\\_entorno\\_de\\_desarrollo\\_integrado](https://www.ecured.cu/Eclipse._entorno_de_desarrollo_integrado). 23/10/2016
- [5] <https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/>. 19/10/2016
- [6] <http://960.gs/>
- [7] [http://csstemplatesmarket.com/freecsstemplates/panelo\\_admin\\_template/](http://csstemplatesmarket.com/freecsstemplates/panelo_admin_template/)
- [8] [https://www.ibm.com/support/knowledgecenter/es/SS7K4U\\_8.5.5/com.ibm.websphere.zseries.doc/ae/twbs\\_jaxrs\\_configwebxml.html](https://www.ibm.com/support/knowledgecenter/es/SS7K4U_8.5.5/com.ibm.websphere.zseries.doc/ae/twbs_jaxrs_configwebxml.html)
- [9] <http://www.desarrolloweb.com/articulos/2243.php>
- [10] <http://codigoxules.org/java-itext-pdf-creando-pdf-java-itext/>
- [11] <https://jqueryui.com/autocomplete/>
- [12] <https://jqueryui.com/dialog/>
- [13] <https://jqueryui.com/datepicker/>
- [14] <https://jqueryvalidation.org/>
- [15] <http://wiki.sitemesh.org/wiki/display/sitemesh/Start+Using+SiteMesh+in+10+Minutes>
- [16] [https://es.wikipedia.org/wiki/Wireframe\\_\(dise%C3%B1o\\_web\)](https://es.wikipedia.org/wiki/Wireframe_(dise%C3%B1o_web))
- [17] [https://es.wikipedia.org/wiki/JavaServer\\_Pages](https://es.wikipedia.org/wiki/JavaServer_Pages)
- [18] [https://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)
- [19] [https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
- [20] <http://www.abc.es/viajar/20130128/abci-redes-sociales-para-planificar-201301221016.html>
- [21] <https://www.trabber.es/viajes/>
- [22] <http://www.travelandexchange.com/5-aplicaciones-imprescindibles-para-tu-viaje/>
- [23] [http://tecnologia.elpais.com/tecnologia/2011/02/02/actualidad/1296640862\\_850215.html](http://tecnologia.elpais.com/tecnologia/2011/02/02/actualidad/1296640862_850215.html),
- [24] <https://houtinee.com/day-one-2-la-app-perfecta-para-llevar-tu-diario-de-viaje/>