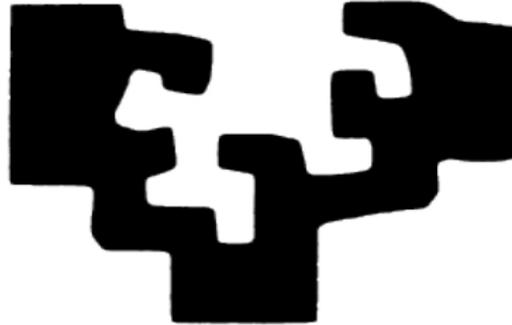


eman ta zabal zazu



universidad  
del país vasco

euskal herriko  
unibertsitatea

**Facultad de  
Informática**

**Informatika  
Fakultatea**

**TITULACIÓN: Ingeniería en Informática**

**Gestión Web de Servicios de Hostelería para  
Alérgicos**

<b>Alumno:</b>	<b>D. Beinat Ruiz de Arcaute Parra</b>
<b>Director:</b>	<b>D. José Ángel Vadillo Zorita</b>

**Proyecto Fin de Carrera, junio de 2015**

# Índice general

<b>1. Introducción</b>	<b>9</b>
1.1. Propuesta . . . . .	13
1.2. Aplicaciones similares . . . . .	15
1.2.1. TripAdvisor . . . . .	15
1.2.2. Boletus . . . . .	16
1.2.3. FourSquare . . . . .	16
1.2.4. Allergychef . . . . .	17
1.2.5. Otras redes sociales . . . . .	17
<b>2. Documento de Objetivos del Proyecto</b>	<b>18</b>
2.1. Introducción . . . . .	18
2.2. Objetivos . . . . .	18
2.3. Alcance . . . . .	24
2.4. Metodología . . . . .	27
2.5. Planificación . . . . .	29
<b>3. Análisis</b>	<b>33</b>
3.1. Casos de Uso . . . . .	33
3.1.1. Actores de los Casos de uso . . . . .	34
3.1.2. Registro de usuario . . . . .	34
3.1.3. Búsqueda de establecimientos . . . . .	36
3.1.4. Detalle de establecimiento . . . . .	37
3.1.5. Escribir una valoración . . . . .	38
3.1.6. Editar datos personales . . . . .	39
3.1.7. Alta de un establecimiento . . . . .	39
3.1.8. Editar un establecimiento . . . . .	40

<b>4. Diseño de la plataforma</b>	<b>42</b>
4.1. Modelo de dominio . . . . .	42
4.1.1. Flujo del problema . . . . .	42
4.1.2. Diagrama del Modelo de dominio . . . . .	43
4.2. Diagramas de secuencia . . . . .	43
4.2.1. Registro de usuario . . . . .	44
4.2.2. Búsqueda de establecimientos . . . . .	45
4.2.3. Detalle de establecimiento . . . . .	47
4.2.4. Escribir una valoración . . . . .	49
4.2.5. Editar datos personales . . . . .	50
4.2.6. Alta de un establecimiento . . . . .	51
4.2.7. Editar un establecimiento . . . . .	53
<b>5. Implementación</b>	<b>55</b>
5.1. Análisis tecnológico . . . . .	55
5.1.1. HTML5 . . . . .	55
5.1.2. RWD . . . . .	56
5.1.3. AJAX . . . . .	57
5.1.4. MySQL . . . . .	58
5.1.5. PHP . . . . .	58
5.1.6. Otras tecnologías valoradas . . . . .	59
5.2. La plataforma Allergic . . . . .	61
5.2.1. Paso de información entre páginas . . . . .	63
5.2.2. Envío de información . . . . .	65
5.2.3. Recogida de información . . . . .	66
5.2.4. Obtención de coordenadas . . . . .	66
5.2.5. Cálculo de distancia entre dispositivo y establecimiento	69
5.3. Arquitectura de Allergic . . . . .	71
5.3.1. Capa de Datos . . . . .	72
5.3.2. Capa de Negocio . . . . .	73
5.3.3. Capa de presentación . . . . .	73
<b>6. Evaluación del sistema</b>	<b>74</b>
6.1. Diseño responsive . . . . .	74
6.2. Velocidad de carga . . . . .	78
6.3. Escalabilidad del modelo de datos . . . . .	80

<i>ÍNDICE GENERAL</i>	4
<b>7. Seguimiento y control</b>	<b>81</b>
7.1. Alcance del Proyecto . . . . .	81
7.2. Control y gestión del tiempo . . . . .	81
7.2.1. Tareas que se planificaron con estimación pesimista . .	82
7.2.2. Tareas que se planificaron con estimación optimista . .	82
<b>8. Conclusiones</b>	<b>83</b>
8.1. Desarrollo del proyecto . . . . .	83
8.2. Trabajo futuro . . . . .	85
8.2.1. Requisitos antes de lanzar la plataforma . . . . .	85
8.2.2. Ampliación de la plataforma . . . . .	86
<b>9. Bibliografía</b>	<b>87</b>

# Índice de figuras

2.1. EDT . . . . .	25
2.2. Planificación . . . . .	30
3.1. Diagrama de casos de uso . . . . .	34
4.1. Modelo de dominio . . . . .	43
4.2. Diagrama de secuencia: registro de usuario . . . . .	44
4.3. Diagrama de clases: registro de usuario . . . . .	45
4.4. Diagrama de secuencia: búsqueda de establecimientos . . . . .	45
4.5. Diagrama de clases: búsqueda de establecimientos . . . . .	46
4.6. Diagrama de secuencia: detalle de establecimiento . . . . .	47
4.7. Diagrama de clases: detalle de establecimiento . . . . .	48
4.8. Diagrama de secuencia: escribir una valoración . . . . .	49
4.9. Diagrama de clases: escribir una valoración . . . . .	50
4.10. Diagrama de secuencia: editar datos personales . . . . .	50
4.11. Diagrama de clases: editar datos personales . . . . .	51
4.12. Diagrama de secuencia: alta de un establecimiento . . . . .	51
4.13. Diagrama de clases: alta de un establecimiento . . . . .	52
4.14. Diagrama de secuencia: editar un establecimiento . . . . .	53
4.15. Diagrama de clases: editar un establecimiento . . . . .	54
5.1. Tecnologías bajo AJAX . . . . .	57
5.2. Barra de navegación de una búsqueda con coordenadas . . . . .	64
5.3. esquema de la arquitectura en tres niveles . . . . .	71
5.4. modelo de Entidad Relación . . . . .	72
6.1. pantalla inicial ancha . . . . .	75
6.2. pantalla inicial estrecha . . . . .	75

*ÍNDICE DE FIGURAS*

6

6.3. pantalla de búsqueda ancha . . . . .	76
6.4. pantalla de búsqueda estrecha . . . . .	77
6.5. pantalla detalle ancha . . . . .	77
6.6. pantalla detalle estrecha . . . . .	78
7.1. Diferencias de tiempo empleadas en las tareas con desvíos . .	82

# Índice de cuadros

5.1. Esquema básico de página HTML . . . . .	62
5.2. Ejemplo de envío de información mediante método GET . . . . .	64
5.3. ejemplo de formulario HTML . . . . .	65
5.4. captura de información en PHP . . . . .	66
5.5. mostrar un mapa en una página HTML . . . . .	67
5.6. obtener coordenadas de una dirección. . . . .	68
5.7. obtener las ubicación de un dispositivo mediante javascript . . . . .	69

# Resumen

Hoy en día existe un gran sector de la población que tiene restricciones alimenticias, bien sea por alergias o por convicciones personales. Para estos usuarios salir a comer fuera de casa acarrea un gran inconveniente, que es: la búsqueda de un establecimiento que ofrezca un servicio de calidad y que tenga en cuenta sus limitaciones a la hora de ofrecerles el servicio.

Muchas veces estos usuarios se ven decepcionados en este sentido, con la consecuente frustración que sienten por el sentimiento de insatisfacción. Dicho sentimiento genera un ansia de comunicación de la experiencia para castigar dicho servicio. Además, esta ansia de comunicación no solo se da en casos negativos, ya que cuando el servicio es bueno el usuario generalmente siente el deseo de expresar su gratitud. Sin embargo, no existe una plataforma efectiva que recoja esas voces críticas o de gratitud que, si fuesen públicas, ahorrarían disgustos a otros usuarios ya que dirigirían el consumo a lugares de con buena calidad de servicio.

En este trabajo presentamos el desarrollo de una plataforma web que permitirá a los usuarios de la misma conocer y dar su opinión para que el resto de usuarios puedan evaluar establecimientos según una serie de criterios, entre ellos los problemas que hayan podido tener con alergias a alimentos. Para ello, esta plataforma se hace uso de la nueva generación de dispositivos móviles que disponen de internet, posicionamiento GPS etc..

La plataforma estará alojada en un servidor web accesible en todo momento desde cualquier parte gracias a que los hoy en día todos los dispositivos disponen de conexión a Internet. La configuración necesaria para el acceso a la plataforma se encuentra en el Anexo I.

**Palabras clave:** plataforma web, geolocalización, PHP, MySQL

# Capítulo 1

## Introducción

Hoy en día existe un gran sector de la población que tiene restricciones alimenticias, bien sea por alergias o por convicciones personales. Para estos usuarios salir a cenar fuera de casa acarrea un gran inconveniente que es la búsqueda de un establecimiento que ofrezca un servicio de calidad y que tenga en cuenta sus limitaciones a la hora de ofrecerles el servicio.

La mayor parte de las personas pueden ingerir todo tipo de alimentos sin problemas. No obstante hay un sector relativamente importante de la sociedad que a los que determinados alimentos o componentes alimenticios les producen reacciones adversas.

Las reacciones adversas a los alimentos no solo pueden ser causadas por alergias, también existen las intolerancias alimentarias. La gran diferencia entre las alergias y las intolerancias es que en el caso de las intolerancias el sistema digestivo no puede digerir el alimento ingerido o el alimento ingerido irrita el sistema digestivo. En cambio en el caso de las reacciones alérgicas, el cuerpo detecta el alimento ingerido como un enemigo y responde ante los alimentos de una forma desproporcionada. Normalmente, una persona con una intolerancia, puede comer una pequeña cantidad del alimento del que es intolerante sin sufrir ningún tipo de problemas, en cambio, en las personas que parecen alergias, el mínimo trazo del alimento o compuesto puede desencadenar una reacción.

## Alergias

Según EUFIC (The European Food Information Council), actualmente, una de cada tres personas tiende a considerarse alérgico aunque la proporción real de alérgicos en realidad sea del 2 % en los adultos y entre el 3 % y 7 % en niños. La razón de esto es debido a la confusión existente entre alergias e intolerancias.

La mayoría de reacciones alérgicas que se suceden por la ingestión de alimentos generalmente es de carácter leve. Muy pocos son los casos que presenten riesgo de anafilaxis, que implica riesgo real de muerte. A continuación se presenta un resumen con los tipos de reacciones que se originan en las alergias alimentarias:

- Respiratorios
  - Moqueo o congestión nasal
  - Estornudos Asma (dificultad para respirar)
  - Tos
  - Sibilancia
  - Trastornos respiratorios
- Cutáneos
  - Inflamación de labios, boca, lengua, cara y/o la garganta (angioedema)
  - Urticaria Erupciones o enrojecimiento
  - Picazón (prurito)
  - Eczema
- Gastrointestinales
  - Dolor abdominal
  - Diarrea
  - Náuseas
  - Vómitos
  - Cólicos
  - Hinchazón
- Sistémicos
  - Shock anafiláctico (shock generalizado grave)

Aunque cualquier alimento puede causar reacciones alérgicas, hay algunas que se dan con una mayor frecuencia. Los alérgenos alimenticios más comu-

nes son: a leche de vaca, los huevos, la soja, el trigo, los crustáceos, las frutas, los cacahuets y los frutos secos, como las nueces.

La alergia a la proteína de la leche de vaca es la más común en niños menores de 3 años. Esta alergia se da entre un 0.5 % y un 4% de los bebés y la prevalencia de la misma disminuye con la edad. Debido a ello, la alergia a esta proteína generalmente remite pronto y la incidencia en niños a partir de 5 años y en adultos es mucho menor.

La alergia a los frutos secos puede estar causada por uno o más frutos secos simultáneamente y comienza a una edad temprana durando para toda la vida. Algunos de ellos pueden provocar reacciones alérgicas mediante inhalación o a través del contacto con la piel, sin necesidad de que haya heridas o brechas en la misma. Una alergia leve a los frutos secos puede limitarse a erupciones, nauseas, dolores de cabeza,... En cambio una alergia grave puede provocar shock anafiláctico. Debido a la gravedad de los posibles síntomas debido a las reacciones, las personas que sufren alergias a los frutos secos reciben indicaciones de evitar cualquier contacto con los mismos y de portar adrenalina en todo momento para contrarrestar las reacciones alérgicas más graves.

## **Intolerancias**

Las intolerancias alimentarias pueden ocasionar síntomas parecidos a los de las alergias, sin embargo, estos suceden sin que entre en juego el sistema inmune de la persona. Una intolerancia se da cuando el sistema es incapaz de digerir un determinado alimento, y a diferencia de las alergias, las personas no necesitan eliminar de su alimentación dichos alimentos. Es más, la personas que sufren intolerancias, generalmente, pueden consumir pequeñas cantidades de los alimentos sin sufrir ningún síntoma, a excepción de aquellas que sufran de intolerancia al gluten o a los sulfitos.

Las causas más comunes de intolerancias alimentarias son el gluten y la lactosa. La intolerancia a la lactosa suele provocar flatulencia, dolores y diarrea debido a que la que no se puede digerir y pasa al intestino, donde se fermenta por las bacterias de la flora intestinal. La lactosa es un azúcar que

se encuentra en la leche. La lactasa, la enzima que se encarga de descomponer la lactosa en azúcares más simples para que puedan ser absorbidos, se encuentra en el intestino delgado. La gente noreuropea generalmente produce suficiente lactasa y debido a eso hay una proporción muy baja de personas con esta intolerancia en esa región. En cambio, la lactosa es un fenómeno común en determinadas zonas geográficas como Oriente Medio, India y zonas de África, así como sus descendientes donde la población adulta alcanza entre un 50 % y un 80 % de intolerantes a la lactosa. En cambio en Europa la deficiencia de lactasa se da cerca del 5 %.

La enfermedad celíaca, que es la que provoca la intolerancia al gluten, se da en un 1 % de las pruebas médicas realizadas. Cuando un enfermo celíaco consume gluten se daña la mucosa del intestino delgado provocando que se vea afectada la capacidad para absorber nutrientes. Entre sus síntomas, destacan: la diarrea, la pérdida de peso, la fatiga, la irritabilidad y el dolor abdominal. En los niños se suelen dar síntomas de desnutrición como problemas de crecimiento. La única forma de ayudar a los pacientes que sufren esta enfermedad es eliminando por completo el gluten de sus dietas. Actualmente en cualquier supermercado se puede acceder a todo tipo de alimentos sin gluten etiquetados a tal efecto. Según se elimina el gluten de la dieta, el intestino se regenera poco a poco y desaparecen los síntomas.

### **Otras restricciones**

A parte de las restricciones orgánicas como las intolerancias y las reacciones alérgicas, también hay otras restricciones alimentarias debido a ideologías o religiones.

El vegetarianismo junto con todas sus vertientes, como el veganismo o el ovolactovegetarianismo establecen sendas restricciones alimenticias a los productos de origen animal. Cada una, a diferente nivel, reprueban la ingesta de los alimentos de origen animal, desde los productos obtenidos gracias a la domesticación de los mismos, como pueden ser los huevos y la leche, incluidos todos los productos lácteos, a la limitación de alimentarse con la carne obtenida de un animal.

Además también existen restricciones religiosas, ya que determinadas religiones impiden la ingesta de ciertos alimentos, como puede ser del cerdo en el caso de los musulmanes. Además algunas también imponen metodologías a la hora de obtener los alimentos, como puede ser la práctica de la matanza según la *sharia* o ley islámica, conocida como *halal*.

## 1.1. Propuesta

En este proyecto se propone una plataforma (Allergic) para proporcionar a los alérgicos una plataforma en la que poder denunciar públicamente el mal trato a los alérgicos en determinados establecimientos. Actualmente en muchos establecimientos, pese a que los clientes avisan de que tienen alguna alergia o intolerancia, no toman las medidas necesarias pese a advertir a los camareros de una restricción alimenticia. Además, en ocasiones, los cocineros de los establecimientos no se molestan en limpiar los utensilios de cocina a la hora de preparar los platos del cliente. Esto provoca que todos aquellos alérgicos que consumen en estos restaurantes sufran reacciones y, en los casos más extremos, peligren sus vidas.

Allergic permite a los usuarios alérgicos dar su opinión respecto a los establecimientos y establecer con qué alergias han tenido problemas. Esto permitirá a los usuarios que tengan alergias filtrar todos aquellos establecimientos que carecen de valoraciones negativas con las alergias que estos padecen. Pese a ello, mediante un único sistema de valoraciones no se proporciona seguridad en los establecimientos que no se hayan sufrido reacciones por las siguientes razones:

1. La ausencia de valoraciones negativas no elimina el riesgo. Pese a que un establecimiento con varias valoraciones no tenga ninguna negativa en cuanto alergias no quiere decir que sea seguro ir padeciendo alergias. Ahora bien sí que es cierto que teniendo un amplio número de valoraciones, si ninguna ha resultado negativa, puede inclinar la balanza a la hora de elegir restaurantes por una persona que sufre alergias.
2. Los establecimientos, generalmente, cuentan con un amplio menú y

las personas con alergias juegan con la selección de los alimentos para reducir su riesgo. Es decir que, por ejemplo, una persona celíaca que haya valorado positivamente un establecimiento no tiene porqué implicar que el establecimiento disponga de pan apto para celíacos, podría haberse limitado a simplemente no comer pan.

Para ello se ofrece, además de una selección de alergias, la posibilidad de introducir una opinión o comentario en la que especificar el trato, los alimentos consumidos, etc... Con esto se pretende informar con mayor exactitud a los usuarios de la plataforma.

Además, también se ofrece la posibilidad de valorar el servicio en función de la calidad, el precio y la variedad. Esto, unido a la selección de alergias y a la opinión o comentario del usuario, permite conocer el servicio del establecimiento con relativa claridad.

El alta de los establecimientos será público, es decir, únicamente será necesario estar registrado para dar de alta nuevos establecimientos. Con ello se pretende dar una mayor difusión a la plataforma, permitiendo que todos los usuarios den de alta establecimientos rellenando una serie de datos básicos, como el nombre, la dirección y unas coordenadas en un mapa.

A su vez, las valoraciones también están pensadas para realizarse de forma rápida, vasta con acceder a la ficha del establecimiento para poder dejar una valoración junto con una opinión del servicio recibido. Todo ello quedará instantáneamente almacenado y podrá ser visible por todos los usuarios de la plataforma.

Una vez más, con el objetivo de conseguir la máxima difusión, todos los usuarios, registrados o no, podrán buscar establecimientos y podrán leer las opiniones y consultar las valoraciones de los establecimientos. Esto, además de buscar difusión de la plataforma, también tiene un segundo objetivo que es permitir al resto de los usuarios conocer los detalles de un establecimiento antes de acceder a él, ya que en función de las valoraciones recogidas sobre el mismo, un cliente podría decidir ir o no ir a un establecimiento concreto.

## 1.2. Aplicaciones similares

A continuación se expondrán una serie de plataformas similares a lo que se plantea el proyecto. Las personas con alergias tienen la necesidad de buscar sitios de confianza para salir a socializar. Si que es cierto que existen plataformas que ofrecen un servicio de búsqueda, pero ninguno de ellos acaba de cubrir sus necesidades al 100 %.

### 1.2.1. TripAdvisor

Es, sin duda alguna, la plataforma referente en cuanto a opinión en la red a día de hoy (2015). Además de ofrecer valoraciones y opiniones sobre establecimientos, también permite valorar viajes, hoteles y una larga lista de contenido relacionado con los viajes y el turismo.

Centrando el foco en la valoración de los establecimientos, dispone de una amplia base de datos, con un número muy alto de usuarios que comparten la opinión sobre el servicio recibido de los establecimientos a los que han ido. Permite realizar búsquedas y conocer la opinión de los usuarios sin necesidad de estar registrado. No obstante, para dar tu opinión necesitas estar registrado.

En cambio, no dispone de ninguna herramienta que permita conocer la satisfacción a los grupos de personas con restricciones alimentarias, por lo que si una persona quiere buscar información sobre el trato del establecimiento con las personas alérgicas, intolerantes o por restricciones voluntarias, no podrá obtener dicha información a menos de que haya algún comentario indicándolo. Esto último además de ser costoso en tiempo puede ser imposible en el caso de carecer de comentarios sobre alergias.

Tras registrarte en la plataforma esta pasa al usuario un cuestionario sobre sus gustos para conocer el gusto gastronómico y estilo de turismo que prefiere. En este cuestionario, la plataforma ofrece como opción el vegetarianismo y, aunque en un principio parece que es para dar información al usuario, no es más que información proporcionada por el usuario a la empresa para futuras estadísticas de mercado y ofrecer ofertas de promociones

impulsadas desde los establecimientos. Tras realizar búsquedas, la plataforma permite filtrar los restaurantes en función de una serie de aspectos definidos y votados por los usuarios.

Para dar de alta un establecimiento no es necesario ser el propietario o algún comercial de la marca, con estar registrado es suficiente. A su vez, para opinar sobre un establecimiento hay que estar registrado y facilita un formulario sencillo con el que se puede dejar una valoración precisa de la experiencia del cliente.

### 1.2.2. Boletus

Se trata de una plataforma que además de proporcionar a los establecimientos una forma sencilla de crear una imagen en la web, permite a los establecimientos publicar promociones en forma de ofertas para atraer clientes.

A diferencia TripAdvisor, permite dar de alta establecimientos, pero únicamente a los propietarios de los mismos, siendo comerciales de la marca Boletus los que dan de alta los comercios o los propietarios de los mismos solicitando acceso a la plataforma.

No ofrecen forma efectiva de comentar el servicio recibido en los establecimientos a usuarios estándares. Tras intentar varias veces publicar la opinión de varios establecimientos, tanto a través de la web, como de la aplicación móvil en Android no parece haber forma de dar una opinión.

De cualquier forma, al igual que TripAdvisor no dispone de ninguna forma de evaluar el establecimiento en función de las alergias.

### 1.2.3. FourSquare

Al igual que TripAdvisor y Boletus, se trata de una plataforma que en 2013 era la número uno de la red en dar opiniones y valoraciones sobre establecimientos. Poco a poco fue dejando paso a TripAdvisor debido a la mayor cantidad de información que permite publicar.

Esta, una vez más, carece de forma de evaluar los establecimientos con alergias, aunque permite a los propietarios del establecimiento crear entradas para promocionar eventos, platos especiales o menús concretos.

#### **1.2.4. Allergychef**

Allergychef es, sin duda, la plataforma más parecida a la que se propone más adelante. En ella se puede consultar la opinión de los clientes de determinados restaurantes y conocer sus problemas con las alergias en los mismos. A la hora del registro la plataforma te permite seleccionar entre una serie de grupos de alergias para, más adelante, poder dar tu opinión respecto a los establecimientos.

Además de las valoraciones, la plataforma ofrece realizar reservas en los restaurantes. No obstante, la oferta de establecimientos es pobre ya que el alta de los establecimientos no está abierta, únicamente se pueden valorar los establecimientos que la plataforma propone limitando mucho el uso informativo de la misma. Esto provoca limitaciones obvias en la difusión de la información ya que pese a recibir un mal servicio en un establecimiento no puedes informar a los usuarios de la plataforma.

#### **1.2.5. Otras redes sociales**

Por último, las redes sociales pueden considerarse aplicaciones similares debido a que, pese a que están pensadas para interconectar personas, también permiten crear páginas para sociedades, grupos de música, establecimientos, etc..

En estas, los usuarios pueden dar su opinión aunque no pueden valorar las aptitudes del establecimiento más allá de lo que pueden describir en un comentario.

## Capítulo 2

# Documento de Objetivos del Proyecto

### 2.1. Introducción

El proyecto trata de una aplicación web autogestionable para compartir y opinar sobre diferentes servicios en la hostelería centrándose sobre todo en la opinión de usuarios alérgicos. Los usuarios con alergias a diferentes alimentos calificarán el grado de aptitud o adecuación del servicio de hostelería que hayan recibido.

La aplicación será de dominio público, con acceso “global” y de forma ininterrumpida para todos aquellos con conexión a Internet. El proyecto está dividido en una serie de fases que se especificarán y justificarán más adelante a lo largo del documento.

### 2.2. Objetivos

El objetivo principal de este proyecto es el desarrollo de una plataforma web accesible desde cualquier dispositivo con conexión a internet para poder consultar y compartir la calidad de servicio así como del trato que han recibido en establecimientos de hostelería.

Allergic se pretende que sea una plataforma web accesible desde cualquier

tipo de navegador con interfaz dinámica y adaptable gracias a RWD (Responsive Web Design). La aplicación permitirá a usuarios de todo el mundo consultar y valorar establecimientos de hostelería así como dar de alta nuevos establecimientos.

Todo ello, aparte de dar a los usuarios una forma de evaluar los servicios de los establecimientos, permite a los establecimientos promocionarse siempre y cuando su servicio sea conforme a lo que los usuarios buscan. Además, como valor añadido y principal distinción del resto de plataformas similares, se permitirá evaluar la calidad del servicio en función del cuidado a los alérgicos y aquellos con restricciones alimenticias.

### **Búsqueda de establecimientos**

Los usuarios tienen que tener un buscador que permita buscar por nombre o por localización. Para ello se utilizan los datos previamente introducidos por otros usuarios sobre los establecimientos.

La búsqueda mostrará dinámicamente los resultados más similares, permitiendo a los usuarios ahorrar tiempo de escritura si han encontrado el establecimiento o localidad que buscaban o recordando el nombre exacto de lo que estaban buscando.

En la búsqueda se pueden filtrar los resultados en función de diversos parámetros, como valoración media, votos positivos en alergias, votos negativos en alergias etc..

Al igual que se podrán ordenar en función de parámetros similares, incluyendo la cercanía del dispositivo. Para ello, se utilizará la posición que da el dispositivo físico y se calcularán cuales son los establecimientos más cercanos.

### **Consulta y valoraciones**

Todos los usuarios podrán consultar la ubicación y la valoración de todos los establecimientos que estén dados de alta. El objetivo principal de esto es dar información precisa de la calidad de los establecimientos y dar a los

establecimientos una forma de anunciarse.

Los usuarios que se registren podrán valorar los establecimientos. El registro para valorar tiene dos razones: en caso de que haya delitos por faltas o vejaciones que dañen el honor o la imagen de establecimientos y personas; y el control de valoraciones de los establecimientos, es decir, establecer un límite de votaciones por usuario.

Las valoraciones podrán ser votadas por el resto de usuarios en función de su interés de tal forma que habrá valoraciones y opiniones que resalten sobre el resto debido a que la comunidad le ha dado más importancia que al resto.

### **Alta de establecimientos**

Los usuarios registrados podrán dar de alta nuevos establecimientos ya sea porque han estado en alguno y quieren compartir su impresión, sea buena o mala. O bien porque son los propietarios del mismo y quieren darse a conocer en este último caso, los usuarios tendrán acceso a mayor espacio y opciones de personalización extra, como imágenes de fondo, posibilidad de insertar cartas y menús. Para ello, hay que comprobar que efectivamente son los propietarios o gerentes del mismo y se les pedirá la documentación necesaria para demostrarlo.

El mínimo de datos de un establecimiento será: Nombre, dirección, situarlo en un mapa, una breve descripción del establecimiento y de sus principales características y una categorización de tipos de restaurante, por ejemplo: comida rápida, de tapas, comida típica de la zona, etc. . .

### **Alergias y/o restricciones alimentarias**

El elemento social de la plataforma reside en el enfoque de clasificación por satisfacción de personas que sufren de alergias y/o tienen restricciones alimentarias.

Esto permitiría a los usuarios buscar o filtrar el establecimiento que cumple con sus peticiones en cuanto a los ingredientes que pueden o no pueden usar en sus alimentos. También se valora la implicación que puedan tener con la higiene del material de cocina usado a la hora de preparar el alimento de esta persona cuando ha dejado constancia de que padece alguna alergia.

De la misma forma, al igual que se pueden incluir todas aquellas intolerancias no voluntarias, es decir alergias e intolerancias alimenticias, también se pueden incluir todas ellas que son voluntarias. En este caso el vegetarianismo y sus diferentes tipos también estarían incluidos.

## **Diseño**

La plataforma debe ser operativa para todos los dispositivos. Para ello es necesario adaptar el contenido en función a la resolución del dispositivo. Es evidente que no cabe la misma cantidad de información en una pantalla de ordenador, que en una de tablet y en un móvil e incluso dentro de cada una de esas tres clasificaciones hay diferentes resoluciones. Para ello se hará uso de RWD, Responsive Web Design o Diseño Web Adaptable y Bootstrap, una base de maquetación web que abarca todas las resoluciones conocidas hasta ahora gracias a ser adaptable.

El usar RWD no sólo tiene como función la correcta visualización en todos los dispositivos, también influirá en la reputación online de la plataforma. El buscador google penaliza todas aquellas webs que no se adapten a la resolución de los dispositivos desde el 21 de Abril de 2015 y tiene planeado eliminar los resultados de todas aquellas webs que no se adapten a la resolución en los dispositivos que no puedan visualizarse correctamente.

## **Arquitectura física**

La arquitectura física será de cliente-servidor. La comunicación de la plataforma con la base de datos es crucial a la hora de devolver resultados y teniendo en cuenta que inicialmente no habrá un gran volumen de datos, no es necesario montar la BD en un servidor externo. Por lo que el servidor,

inicialmente contará con un servidor web y la BD.

### **Arquitectura lógica**

En esta plataforma se hará uso de un desarrollo de tres capas y, además, estará enfocado a maximizar la eficiencia de la aplicación y la rentabilidad de la misma. Para ello, cuanto más eficiente sea, menos recursos necesitará lo que se traduce en servicios de alojamiento más baratos.

Los usuarios que aporten información a la plataforma, dando de alta los establecimientos y valorándolos, son los usuarios de mayor valor. Estos usuarios debido a su actividad los consideraremos usuarios activos, en cambio, aquello que únicamente obtengan información de la plataforma serán considerados pasivos.

Todos los usuarios activos pasarán por un proceso de registro lo más sencillo posible para poder captar usuarios con mayor facilidad. Por tanto, para registrarse, únicamente hará falta un correo electrónico y una contraseña. El proceso contará con un sistema anti bot como por ejemplo un captcha. Una vez registrado, el usuario deberá validar su correo electrónico accediendo a una dirección que se le facilita en un mensaje de correo electrónico que se le ha enviado al registrarse. Una vez validado, el usuario podrá rellenar, de forma opcional, más información sobre él. A su vez, seleccionará las alergias o las intolerancias que tenga de tal forma que la plataforma pueda sugerir establecimientos o el propio usuario pueda filtrarlos más adelante sin tener que volver a facilitar esta información.

Las funcionalidades más básicas de la plataforma son: búsqueda de establecimientos por nombre o localidad, filtrado de la búsqueda en función a unos parámetros preestablecidos, consultar la información, valoración y comentarios de un establecimiento y, por último, valorar y/o comentar un establecimiento. Uno de los parámetros preestablecidos de búsqueda es la cercanía del establecimiento, por lo que el usuario deberá permitir a la plataforma conocer su ubicación.

Toda comunicación entre el navegador y el servidor se hará mediante el

protocolo http o su versión segura https, incluyendo todas aquellas que se lancen mediante ajax.

La obtención de la posición del dispositivo para el ordenado de los establecimientos se realizará mediante la API de geoposicionamiento que proporciona HTML5 que, pese a no ser un estándar, es una tecnología muy extendida. Puede que en algunos dispositivos sea necesaria la triangulación mediante GPS o mediante las torres de comunicación móviles.

### **Gestión de datos**

Toda la información que se almacene de usuarios será en una base de datos que esté en un hosting que satisfaga las necesidades de la plataforma y, además, se hará de acuerdo a los términos que exige la LOPD (Ley Orgánica de Protección de Datos), por lo tanto, antes de almacenar cualquier información:

- Se informará a los usuarios de la información que se está almacenando y el uso que se hará con esa información además de ofrecer la opción de eliminar toda su información almacenada en la base de datos.
- Se Implementarán las medidas básicas para proteger los datos:
  - La base de datos estará protegida con contraseña.
  - Se harán copias semanales de la base de datos.
  - La contraseña de la base de datos se modificará una vez al año.
  - Se redactará un documento de política de seguridad donde se expongan quiénes tienen acceso a la misma además de otros datos como: el tipo de bases de datos, etc.
- Una vez lanzada la plataforma, se registrarán las tablas donde se contenga la información de los usuarios en la AEPD (Agencia Española de Protección de Datos).

### **Sistema de Gestión de la Base de Datos**

Para la plataforma se pretende usar un SGBD libre. Esto permite abaratar costes de licencias tanto a lo largo del desarrollo del proyecto como de mantenimiento una vez lanzado. Se ha estudiado los sistemas de MySQL y PostgreSQL, ambas bases de datos relacionales, y sus ventajas e inconvenientes.

nientes. Las conclusiones que se han extraído son las siguientes:

PostgreSQL ofrece un rendimiento superior a la hora de ejecutar procedimientos almacenados, además de contar con una gran integridad de datos. Cuenta con una gran comunidad de desarrolladores lo que sea más sencillo obtener ayuda con cualquier problema que puedas tener. Además, debido a su estructura, la migración de los datos a sistemas más potentes, aunque no sean libres, como Oracle es más sencilla.

No obstante, a la hora de realizar consultas con operaciones y uniones pesadas, pierde mucho rendimiento lo que hace que pese a ser un SGBD muy estable no sea muy popular en aplicaciones reales.

Entre los puntos débiles de MySQL están el hecho de que cuenta con menos funciones que el anterior, lo cual hace que determinadas consultas tengan que ser más complejas. Además tiene sendos problemas a la hora de manejar la concurrencia de lectura-escritura, provocando deadlocks o interbloqueos con relativa facilidad.

En cambio, pese a lo anterior, MySQL es una herramienta sencilla con la que trabajar, más allá de la experiencia que se tenga con ella. Además dispone de interfaces intuitivas de terceros y una cantidad de controladores muy amplia. Cuenta con una seguridad relativamente alta, lo que facilita el mantenimiento de la base de datos. Y por último, el rendimiento en MySQL es superior en cuanto a operaciones de lectura se refiere.

Debido a lo anterior y a la disponibilidad en los servicios de alojamiento de ambos SGBDs se ha elegido MySQL para este proyecto.

### **2.3. Alcance**

El alcance de este PFC es el desarrollo de las características fijadas en el apartado del alcance mínimo. Igualmente, llegado un punto en el ciclo de vida del proyecto se decidirá sobre la posibilidad de ampliar el alcance con los objetivos opcionales previamente descritos.

- La web debe estar orientada a ayudar a las personas con alergias a alimentos.
- Los usuarios sin registrarse podrán:
  - Buscar establecimientos en base a unos criterios que se le facilitarán para hacer la búsqueda sencilla y práctica.
  - Acceder a la ficha de los establecimientos y leer los comentarios y valoraciones sobre dicho establecimiento.
- Los usuarios registrados podrán realizar las mismas acciones que los que no están registrados, además de:
  - Dar de alta nuevos establecimientos.
  - Publicar comentarios y valoraciones.
- La ordenación de los establecimientos se podrá realizar de una de las siguientes forma, a elección del usuario:
  - Cercanía a la posición detectada por el dispositivo.
  - Alfabéticamente.
  - Valoración media.
  - Compatibilidad de alergias.

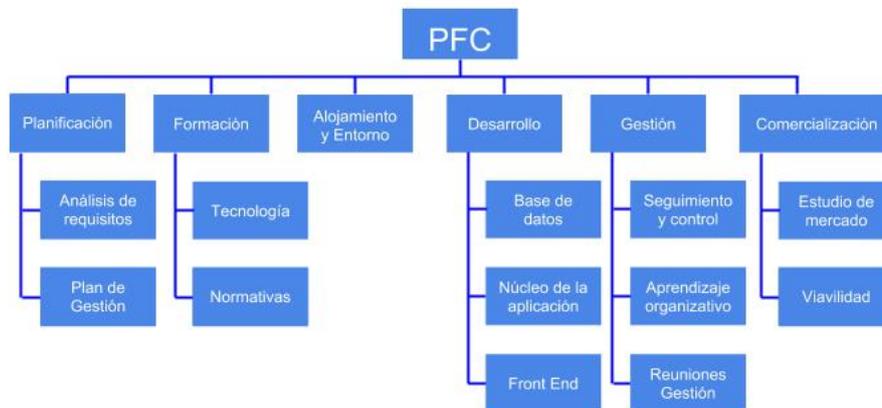


Figura 2.1: EDT

## **Diccionario del EDT**

### **Análisis de Requisitos**

Este paquete de trabajo representa la recogida de diversas fuentes de los requisitos para el proyecto.

### **Plan de Gestión**

Este paquete de trabajo engloba la elaboración del plan para el desarrollo del proyecto.

### **Tecnología**

Este paquete de trabajo se refiere a la formación en la tecnología a utilizar durante el proyecto. Esto incluye, a parte del conocimiento en node.js, conocimiento en express, php, etc....

### **Normativas**

Este paquete de trabajo engloba la formación en las normativas como la LOPD, LSSI o la normativa de accesibilidad web de la WAI. Alojamiento y entorno En este paquete de trabajo se pretende evaluar e investigar diferentes entornos de alojamiento web, como usar los servicios de Amazon EC2, Google App Engine o algún otro hosting.

### **Base de datos**

Este paquete de trabajo cubre el diseño de la base de datos y los posibles procedimientos que se almacenen para mejorar el rendimiento. Además, también cubre la elección entre los diferentes gestores de bases de datos mediante el análisis de los datos obtenidos de un conjunto de pruebas de estrés.

### **Núcleo de la aplicación**

En este paquete de trabajo se realizará la capa de negocio de la aplicación, en la que se procesen los datos, se interactúe con la base de datos a través de la información que se reciba del Front End. Front End Este paquete de trabajo engloba toda la parte visual de la aplicación, tanto la parte visible por los usuarios registrados y sin registrar, como por los administradores de la aplicación que tendrá su propia interfaz.

### **Seguimiento y Control**

Este paquete de trabajo discurre a lo largo de todo el proyecto y consiste en monitorizar el desarrollo del proyecto y compararlo a lo planificado, de forma que se detecten rápidamente los puntos en los que se desvía y se corrijan con el menor coste posible.

### **Aprendizaje Organizativo**

Este paquete de trabajo, al igual que el seguimiento y control se desarrolla en paralelo al proyecto y consiste en gestionar los conocimientos logrados de forma que sean útiles en proyectos posteriores.

### **Reuniones Gestión**

En este paquete de trabajo se agrupan las diferentes reuniones que se realizan a lo largo del proyecto para monitorizar el desarrollo y llegar a acuerdos sobre el proyecto.

### **Estudio de mercado**

Este paquete de trabajo reúne un estudio en el que se determinará las herramientas de las que dispone la competencia y se obtendrá una imagen clara del mercado actual. Viabilidad En este paquete de trabajo se determinará la viabilidad económica del proyecto y se tratará de establecer unas bases de conocimiento de marketing.

## **2.4. Metodología**

Para realizar los objetivos descritos en el apartado anterior, y teniendo en cuenta que las herramientas utilizadas lo permiten, se ha seguido un Proceso Unificado de Desarrollo. El PUD es un proceso de desarrollo de software que cuenta con las siguientes características:

1. Ciclo de vida iterativo e incremental
  - Este ciclo divide el proyecto en varias iteraciones. Cada iteración se trata de manera independiente y va produciendo pequeñas partes del proyecto final. De esta forma se ha conseguido hacer pruebas de la aplicación mientras se seguía con el desarrollo de esta y no solo eso, si no que hace más sencilla la visión global de un proyecto ya que se observa cada iteración de manera independiente.

Además al ser incremental en muchas de las iteraciones se vuelve a repasar las iteraciones anteriores haciendo estas más fiables y ayudando a no olvidar lo aprendido.

2. Guiado por casos de uso
  - Tras una evaluación de los requerimientos del sistema se detallaron las funcionalidades que debía tener.
3. Centrado en la arquitectura
  - La aplicación debe permitir el almacenamiento de los datos generados en una base de datos en la red.
4. Dirigido por el riesgo
  - Por riesgo se entiende que la parte más sensible de la aplicación debe ser la que más seguridad tenga. Se ha de desarrollar un núcleo sólido para evitar futuros fallos.

Dividiremos los métodos en dos tipos de procesos: Tácticos y Operativos, y Formativos.

### **Métodos formativos**

- Reuniones para el control de situación:
  - Se valorará el volumen de trabajo realizado hasta el momento, se detallarán las siguientes tareas y se establecerán fechas para los objetivos e hitos. Será el director quien dicte las tareas futuras ya que cuenta con una visión más global del proyecto y se acordarán fechas de entrega entre el director y el desarrollador. Respecto a temas docentes, se acordarán entregas periódicas con el director para que haga un seguimiento del progreso del proyecto.
- Elección de tecnologías:
  - Antes de adoptar una nueva tecnología se debatirá su viabilidad en una reunión semanal y en el hipotético caso de que ninguno de los participantes la conozca con anterioridad, el desarrollador hará una pequeña prueba y evaluará la tecnología mediante pruebas de estrés.
- Se hará uso de dropbox para compartir documentos. La elección de software de control de versiones, y los servidores que se vayan a usar para el alojamiento web se definirán en base a un estudio previo.
- Al inicio del proyecto se acordará el alcance final del proyecto, todos

sus casos de uso y se establecerán unos límites ampliables por bloques y prioridad.

- Las tareas que corresponden a procesos operativos que deberá hacer el desarrollador son las siguientes:
  1. DOP
  2. Captura de requisitos
  3. Análisis Diseño
  4. Implementación
  5. Memoria
  6. Presentación y defensa

### **Métodos formativos**

Aprendizaje y evaluación de las tecnologías necesarias: Para seleccionar las tecnologías adecuadas para el desarrollo de este proyecto habrá que probar mediante pruebas de estrés los sistemas de BD, la carga de memoria de las diferentes tecnologías que se vayan a usar. Todo ello formará parte de un proceso de aprendizaje de múltiples tecnologías, pudiendo ver de primera mano los puntos fuertes y débiles de las diferentes tecnologías de las que se dispone actualmente.

## **2.5. Planificación**

Dado que el alumno que desarrolla el proyecto tiene que compaginar el trabajo profesional con la realización del proyecto de final de carrera, se hace una planificación con baja carga de horas semanales al proyecto. En contraposición y para cumplir con los objetivos formativos esto repercute en un ciclo de proyecto más largo.

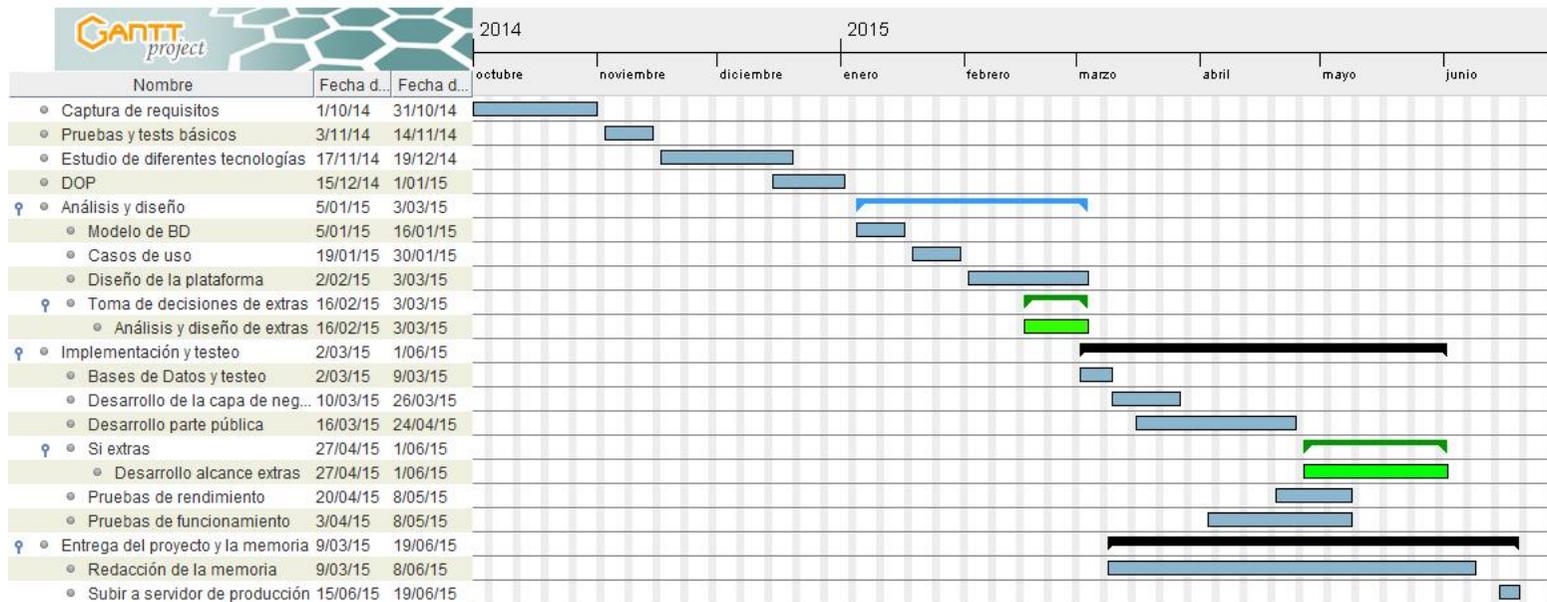


Figura 2.2: Planificación

Tarea	Subtarea	Tiempo Estimado
Captura de requisitos		10 horas
Pruebas y test básicos		20 horas
Estudio de diferentes tecnologías		50 horas
DOP		20 horas
	Modelo de BD	10 horas
Análisis y diseño	Casos de Uso	40 horas
	Diseño de la plataforma	60 horas
	Instalación de plataformas de desarrollo	5 horas
Implementación	Bases de datos y testeo	40 horas
	Desarrollo de la capa de negocio	70 horas
	Desarrollo de la capa visual	70 horas
	Redacción de la memoria	50 horas
Entrega de proyecto y memoria	Pruebas de rendimiento	5 horas
	Pruebas de funcionamiento	5 horas

Se estima una media de 15-20 horas semanales para la realización del proyecto. Se realizarán controles de situación del proyecto bisemanalmente y en el caso de que la realidad se aleje de lo planificado se reestructurará el trabajo futuro para lograr llevar a cabo el proyecto con éxito. Se establecen dos bloques de trabajo opcionales en función del desarrollo del proyecto en el momento, dependiendo del desvío respecto a la planificación inicial, para aumentar el alcance si se considera oportuno.

## Capítulo 3

# Análisis

Los requisitos de plataforma surgieron de una idea inicial de proporcionar un servicio abierto a alérgicos para poder denunciar públicamente el mal servicio y prevenir futuros accidentes por falta de información. En este capítulo se recogen el análisis del proyecto realizado en las fechas establecidas en la Figura 2.2.

Una vez aprobado el alcance del proyecto se inició un proceso de refinado iterativo. Este proceso se realizó de manera sistemática al comienzo de cada iteración. Durante este proceso, se tomó como entradas el alcance del proyecto generándose como salidas del análisis de requisitos el documento de diseño.

Como se podrá observar en la fase de toma de decisión sobre el diseño e implementación del alcance aumentado se acordó no realizar dichas ampliaciones.

### 3.1. Casos de Uso

En este apartado se describen todos los casos de uso implementados en el proyecto. Todos ellos irán acompañados de las figuras de los actores, flujo de eventos y modelo de clases.

### 3.1.1. Actores de los Casos de uso

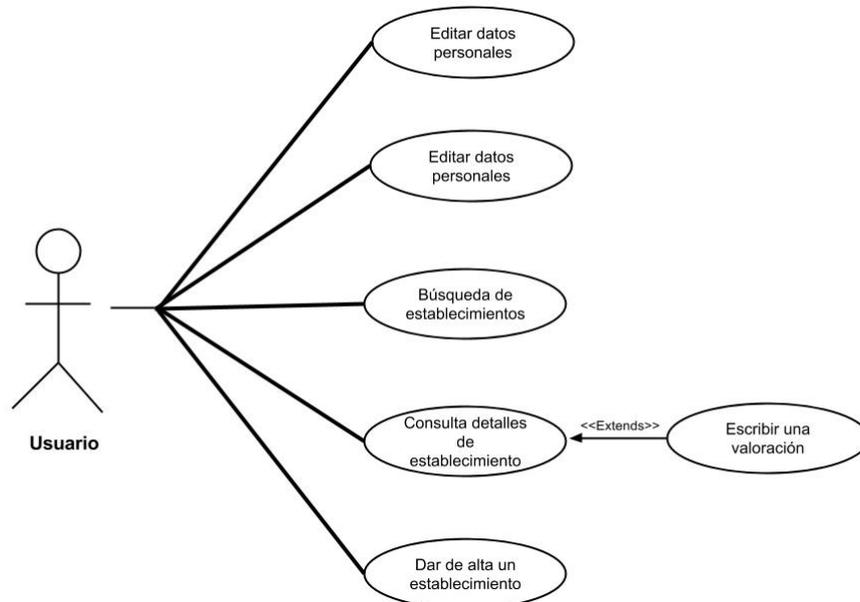


Figura 3.1: Diagrama de casos de uso

### 3.1.2. Registro de usuario

#### Descripción

Para realizar ciertas acciones en la plataforma que se definirán más adelante, será necesario que el usuario esté registrado e identificado en la aplicación para realizarlas. El objetivo de este proceso de registro es informar al usuario de que sus datos van a ser almacenados en una base de datos, que acepte las condiciones del servicio y verificar de que es un usuario real utilizando un correo electrónico personal que le es solicitado para esta comprobación.

Tras la ejecución de este caso de uso el usuario quedará registrado en la base de datos. Pese a ello, cada vez que desee realizar alguna acción en la plataforma deberá iniciar sesión.

### Requisitos funcionales

Disponer físicamente de un dispositivo con conexión a Internet, y una cuenta de correo electrónico hábil.

### Flujo de Eventos

1. El usuario accede a la plataforma a través de un navegador.
2. Pincha sobre un enlace para acceder a la ventana de registro.
3. Se le mostrará una ventana en la que se le solicita rellenar los campos de correo electrónico y contraseña. La contraseña debe ser introducida dos veces para evitar errores tipográficos por parte del usuario. También debe aceptar un acuerdo en el que acepta los términos de registro. Una vez rellenados todos los datos y aceptado el acuerdo y pulse el botón continuar:
  - a) Si el correo electrónico ya existe en la base de datos, se le devolverá al paso 3 mostrándole una alerta en la que se le indicará que el usuario ya existe en la plataforma.
  - b) Si el correo electrónico no existe en la plataforma, este se almacenará en la base de datos junto con la contraseña cifrada y se autogenerará un código aleatorio que será necesario para validar al usuario. El servidor enviará un correo electrónico a la cuenta facilitada para que, de esta forma, demuestre ser el poseedor de la cuenta.
4. Se le solicita al usuario, a través de un correo electrónico, que verifique su cuenta. Para ello en el correo se le envía una URL que contiene el identificador del usuario y el código autogenerado. Esta URL se deberá introducir en un navegador o seguir el enlace que se proporciona.
  - a) Si el código que se envía al servidor concuerda con el código autogenerado para el usuario, se verifica la cuenta para ese correo electrónico.
  - b) Si el código que se envía al servidor no concuerda con el código autogenerado, se muestra una alerta de que no se ha podido verificar el usuario.

### Post-condiciones

Si el proceso de registro se ha completado satisfactoriamente, los datos del usuario quedan almacenados en la base de datos del servidor.

### 3.1.3. Búsqueda de establecimientos

#### Descripción

Este caso de uso permitirá a un usuario registrado o no realizar búsquedas por nombre de establecimiento o por localidad. A este caso de uso se podrá acceder desde la página principal de la plataforma. El usuario deberá rellenar un campo de texto para poder realizar la búsqueda. Mientras el usuario escribe el establecimiento que desea buscar se le mostrarán sugerencias que concuerden con lo que ha escrito hasta el momento a modo de búsqueda proactiva.

Una vez obtenida una serie de resultado además se podrá filtrar el contenido y ordenarlo en función de una serie de parámetros, entre ellos la distancia entre el establecimiento y la posición del dispositivo.

#### Requisitos funcionales

Disponer físicamente de un dispositivo con conexión a Internet.

#### Flujo de eventos

Los pasos 2 y 3 pueden ser ejecutados en orden inverso.

1. El usuario inserta el nombre del establecimiento o la localidad donde quiere buscar establecimientos.
  - a) Si pincha sobre una de las sugerencias realizadas saltará automáticamente al caso de uso *Detalles de Establecimiento*.
  - b) Si pincha sobre el botón buscar se le enviará la consulta al servidor y se le muestra un listado de establecimiento que concuerdan con la búsqueda que ha realizado.
2. El usuario puede seleccionar un orden para los establecimientos.
  - a) Si el usuario selecciona ordenar los establecimientos por cercanía a su posición, el sistema obtendrá su posición y se la mandará al servidor para calcular la distancia entre el dispositivo y la de los establecimientos.

3. Si el usuario está registrado, se filtrarán todos aquellos establecimientos que hayan tenido problemas con las alergias del usuario. También podrá desactivar los filtros para aquellas alergias que quiera.

#### **3.1.4. Detalle de establecimiento**

##### **Descripción**

En este caso de uso se podrá consultar toda la información de un establecimiento concreto. Además de la información del establecimiento, su valoración media en diferentes apartados y una galería de imágenes, también se pueden leer todas las opiniones de los usuarios.

A este caso de uso se puede llegar de tres formas:

1. A través del formulario de búsqueda de establecimientos pinchando en alguno de los establecimientos sugeridos.
2. Tras realizar una búsqueda seleccionando un establecimiento concreto.
3. Introduciendo la URL del establecimiento o a través de algún motor de búsqueda de Internet.

En cualquiera de los tres casos, el tratamiento de la información es el mismo. Al ser una ventana de muestra no se tiene en cuenta la procedencia del usuario a la hora de mostrar la información.

##### **Requisitos funcionales**

Disponer físicamente de un dispositivo con conexión a Internet.

##### **Flujo de eventos**

1. El usuario accede al detalle del establecimiento a través de una URL en la que se introduce como parámetro el identificador del establecimiento.
2. Se le envía al servidor el identificador del establecimiento solicitado.
3. El servidor obtiene todos los datos del establecimiento.
4. El servidor obtiene las opiniones almacenadas del establecimiento.
5. Se le presenta al usuario toda la información en una única página.

### 3.1.5. Escribir una valoración

#### Descripción

Este caso de uso permitirá a los usuarios registrados dejar una valoración del servicio recibido en los establecimientos. Permitirá además de dejar constancia de si se ha tenido problemas con una alergia, evaluar otros aspectos del establecimiento como la variedad, la calidad o la relación calidad precio del establecimiento. Todo ello junto a una opinión que se puede emplear para describir con mayor precisión todos los puntos anteriores.

A este caso de uso solo se puede acceder a través del caso de uso previamente descrito Búsqueda de establecimientos.

#### Requisitos funcionales

Disponer físicamente de un dispositivo con conexión a Internet.

#### Precondiciones

El usuario debe estar registrado e identificado en la plataforma.

#### Flujo de eventos

El usuario previamente debe haber buscado o dado de alta un establecimiento y debe haber entrado al detalle del mismo.

1. El usuario rellena un formulario en el que se le solicita un resumen de su valoración, una serie de votaciones de una a cinco estrellas y que exprese si ha tenido problemas con alguna de sus alergias u intolerancias además de dejar una opinión escrita sobre su experiencia en el establecimiento y procede a enviar la valoración.
2. La información se envía al servidor. Se almacena toda la información en la base de datos.
3. Se válvula la valoración media del establecimiento en todos los apartados, al igual que la valoración media global.
4. Se recarga el la ventana y se muestra la valoración del usuario almacenada.

**Post-condiciones**

Se almacena la valoración del usuario en la base de datos y es accesible por todo el mundo.

**3.1.6. Editar datos personales****Descripción**

Este caso de uso permitirá a los usuarios registrados editar sus datos personales, al igual que añadir o quitar alergias asociadas a los usuarios.

**Requisitos funcionales**

Disponer físicamente de un dispositivo con conexión a Internet.

**Precondiciones**

El usuario debe estar registrado e identificado en la plataforma.

**Flujo de eventos**

1. El usuario accede a la sección de edición o consulta de sus datos.
2. El sistema muestra un formulario con toda la información almacenada del usuario visible.
3. El usuario modifica la información del formulario y procede a enviar el formulario.
4. El sistema recibe la información y se almacena en la base de datos.
5. Se vuelve a cargar la ventana de información del usuario con los datos nuevos.

**Post-condiciones**

Se almacena la información del usuario en la base de datos.

**3.1.7. Alta de un establecimiento****Descripción**

Este caso de uso permitirá a los usuarios registrados dar de alta un nuevo establecimiento en el que se supone que han recibido un servicio o quieren tener el establecimiento accesible desde la plataforma.

**Requisitos funcionales**

Disponer físicamente de un dispositivo con conexión a Internet.

**Precondiciones**

El usuario debe estar registrado e identificado en la plataforma.

**Flujo de eventos**

1. El usuario accede a la sección de alta de un establecimiento.
2. El sistema muestra un formulario con los campos necesarios para dar de alta el establecimiento.
3. El introduce la información del establecimiento.
4. Al introducir una dirección y una localidad la plataforma muestra una posición aproximada de la ubicación del establecimiento.
  - a) El usuario define con mayor exactitud la ubicación del puntero sugerido como ubicación del establecimiento si fuese necesario.
5. El usuario envía la información del establecimiento.
6. El sistema crea el establecimiento nuevo y almacena todos los datos. Además habilita la inserción de imágenes para el establecimiento.
7. Se lanza el caso de uso *Editar un establecimiento* para poder subir imágenes del establecimiento.

**Post-condiciones**

Se almacena la información del establecimiento en la base de datos.

**3.1.8. Editar un establecimiento****Descripción**

Este caso de uso permitirá a los usuarios registrados editar un establecimiento que hayan dado de alta ellos previamente.

A este caso de uso se podrá acceder desde la ventana de establecimientos del usuario o a través del caso de uso *Alta de un establecimiento*.

**Requisitos funcionales**

Disponer físicamente de un dispositivo con conexión a Internet.

**Precondiciones**

El usuario debe estar registrado e identificado en la plataforma.

**Flujo de eventos**

1. El usuario accede a la sección de edición de un usuario.
2. La plataforma muestra toda la información almacenada del establecimiento.
3. El usuario modifica la información del establecimiento:
  - a) Al introducir una dirección y una localidad la plataforma muestra una posición aproximada de la ubicación del establecimiento.
    - 1) El usuario define con mayor exactitud la ubicación del puntero sugerido como ubicación del establecimiento si fuese necesario.
  - b) Añade fotos para la galería de imágenes y las ordena según considere oportuno.
    - 1) La plataforma sube las imágenes al servidor y quedan almacenadas.
4. Una vez terminada la edición de los datos e imágenes envía toda la información al servidor.
5. El servidor almacena toda la información del establecimiento.

**Post-condiciones**

Se almacena la información del establecimiento en la base de datos.

## Capítulo 4

# Diseño de la plataforma

En este capítulo se describirá el diseño de la plataforma mediante el modelo de dominio y los diagramas de flujo de cada uno de los casos de uso definidos en el capítulo anterior.

### 4.1. Modelo de dominio

El Modelo de Dominio es una representación visual estática del entorno real del proyecto. Es decir, no se trata de componentes o clases de software, aunque pudieran coincidir. Es una representación de clases conceptuales del mundo real. Representa los objetos del mundo real en un dominio de interés no de clases de software u objetos de software con responsabilidades.

Este modelo de dominio debe servir como un diccionario visual de abstracciones que ayuda a entender el problema e incluye clases de objetos, asociaciones entre ellos y atributos de las clases de objeto.

#### 4.1.1. Flujo del problema

Como ya hemos descrito con anterioridad en este proyecto hay dos elementos que destacan con claridad: los establecimientos y los usuarios. Los usuarios pueden pertenecer a la plataforma sin necesidad de dar de alta algún establecimiento. Sin embargo, los usuarios registrados tienen la opción de dar de alta todos establecimientos que estimen oportuno.

Las valoraciones de los establecimientos serán escritas por los usuarios estando asociadas a un único establecimiento y a un único usuario. Pese a ello, los establecimientos pueden no tener ninguna valoración y los usuarios pueden optar por no dejar ninguna valoración.

Los establecimiento podrán tener imágenes y se podrán ordenar según el criterio del usuario que ha creado el establecimiento. Una imagen pertenece a un único establecimiento, aunque un establecimiento puede tener varias imágenes o no tener imágenes asociadas.

#### 4.1.2. Diagrama del Modelo de dominio

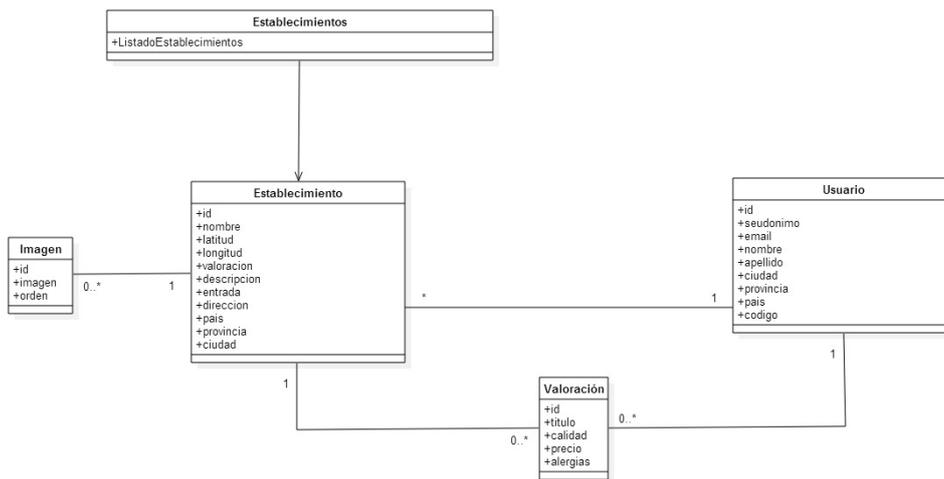


Figura 4.1: Modelo de dominio

#### 4.2. Diagramas de secuencia

En esta sección se definirán los diagramas de secuencia para los casos de uso descritos en el capítulo anterior.

En los diagramas de secuencia, el eje vertical representa el tiempo, y el eje horizontal se colocarán los objetos y actores participantes en la interacción, sin un orden prefijado.

### 4.2.1. Registro de usuario

#### Diagrama de secuencia

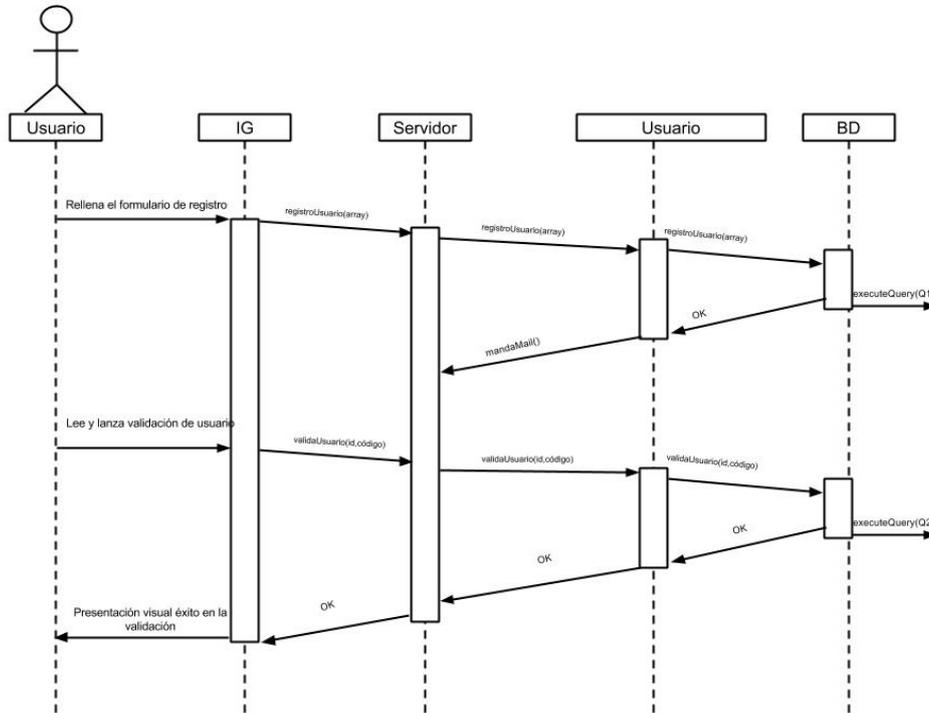


Figura 4.2: Diagrama de secuencia: registro de usuario

#### Diagrama de clases

El cliente ejecuta todas las funciones del servidor a través de envíos de formularios HTML. Se ha elegido representar esta clase con el nombre `ServerConnection`, que representa formulario que ha procesado previamente el servidor a petición del navegador. A continuación se muestran las clases que componen la plataforma del lado del servidor y su reflejo en el lado de los clientes.

Se han implementado dos métodos. El primero es *crearUsuario* que recibirá como parámetros de entrada el correo electrónico y la contraseña que haya elegido el usuario. Como parámetro de salida tendrá un booleano que

indicará si el usuario ya existía con antelación o no. El segundo es *validarUsuario* en el que tendrá como parámetros de entrada el correo electrónico del usuario, su identificador y un código autogenerated. Su parámetro de salida es un booleano indicando si el usuario ha sido validado correctamente o no.

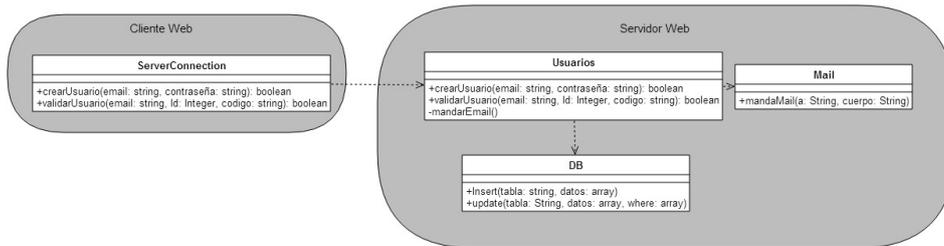


Figura 4.3: Diagrama de clases: registro de usuario

#### 4.2.2. Búsqueda de establecimientos

##### Diagrama de secuencia

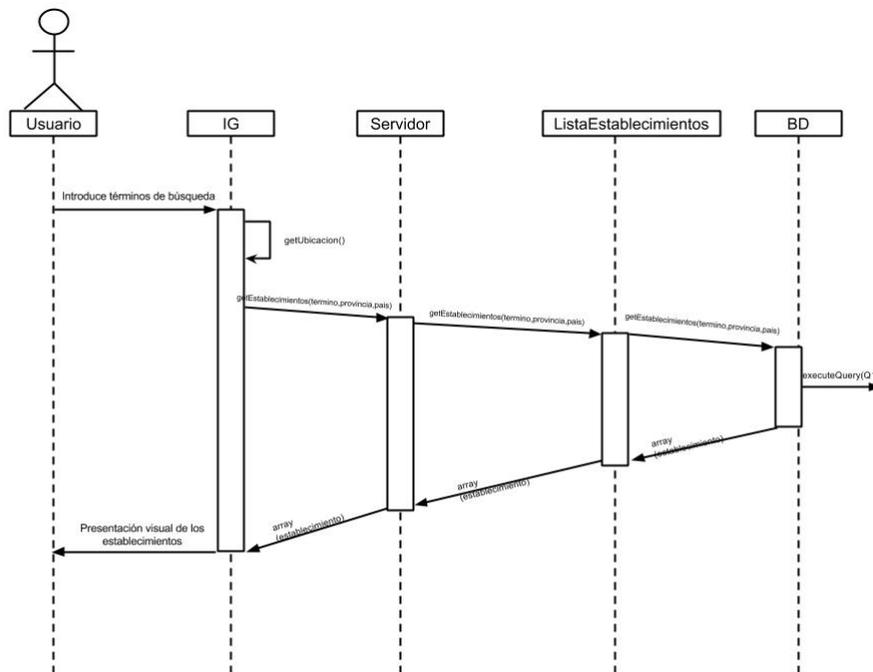


Figura 4.4: Diagrama de secuencia: búsqueda de establecimientos

### Diagrama de clases

Para este caso de uso se han implementado dos métodos. El primero es *obtenerUbicacion* que obtendrá mediante funciones del navegador la posición del dispositivo. Y una segunda *buscarEstablecimiento* que tiene como parámetros de entrada una cadena de caracteres que ha introducido el usuario, el orden que ha seleccionado y las alergias que se deben filtrar. Este método devuelve un listado con todos aquellos establecimientos que cumplen con los términos de búsqueda.

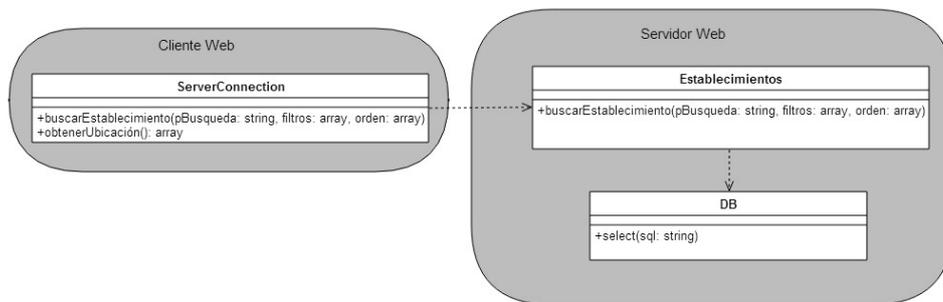


Figura 4.5: Diagrama de clases: búsqueda de establecimientos

### 4.2.3. Detalle de establecimiento

#### Diagrama de secuencia

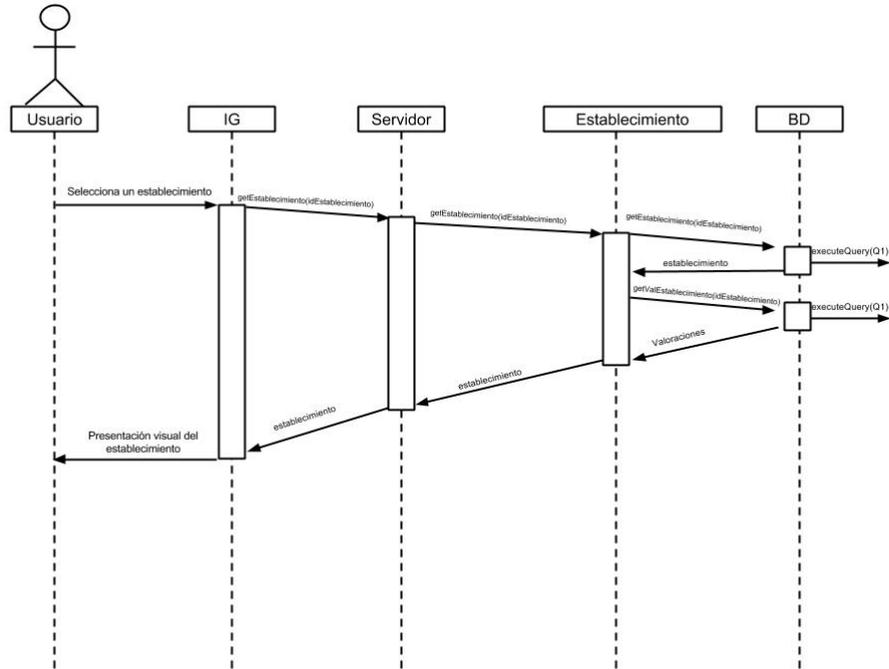


Figura 4.6: Diagrama de secuencia: detalle de establecimiento

#### Diagrama de clases

Para este caso de uso se han identificado cuatro métodos. Dos de ellos ejecutables únicamente en el lado del cliente y otros dos que se comunican con el servidor para obtener la información.

La función `getDetalleEstablecimiento` tiene como parámetro de entrada el identificador de un establecimiento y devuelve toda la información del establecimiento. La función `getValoracionesEstablecimiento` tiene como parámetro de entrada el identificador del establecimiento y devuelve todas las valoraciones del establecimiento.

Los dos métodos que se ejecutan únicamente en el lado del cliente tienen funciones estéticas. El método *mostrarGalería* modifica la visualización de

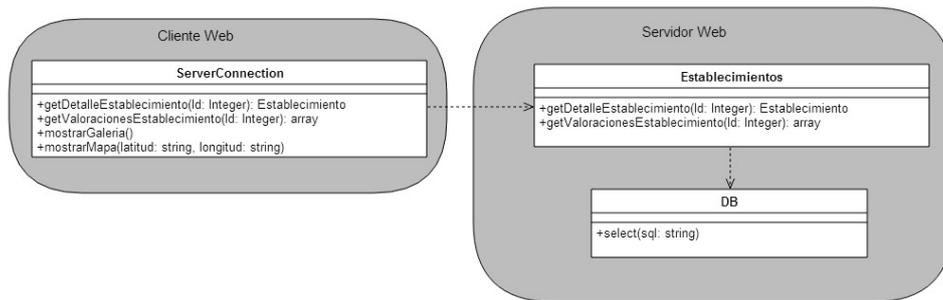


Figura 4.7: Diagrama de clases: detalle de establecimiento

las imágenes para presentarlo en una galería interactiva en la que se pueden ampliar las imágenes. Finalmente, el método *mostrarMapa* toma las coordenadas del establecimiento como entrada y posiciona un puntero representando la localización del establecimiento en un mapa interactivo.

## 4.2.4. Escribir una valoración

## Diagrama de secuencia

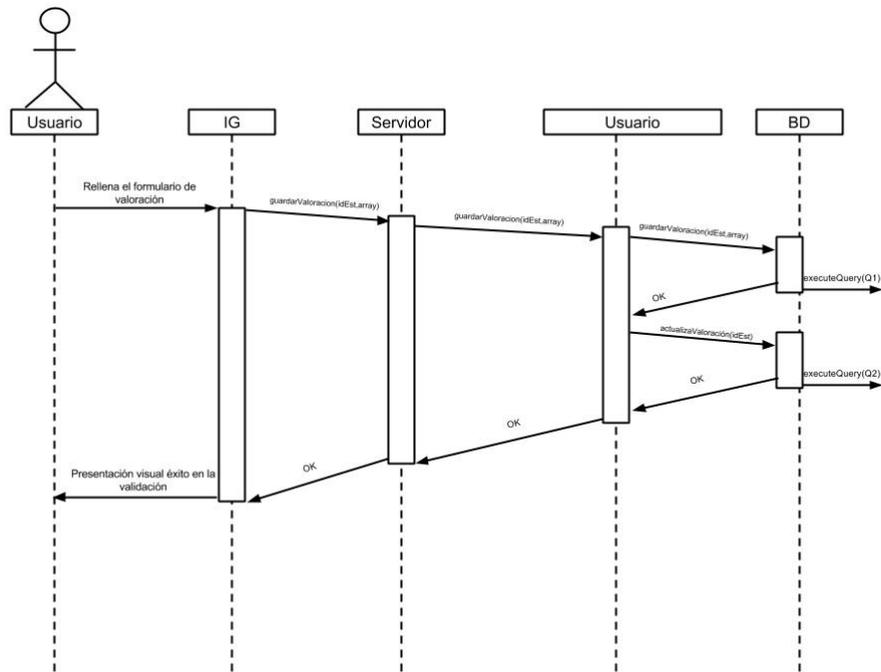


Figura 4.8: Diagrama de secuencia: escribir una valoración

## Diagrama de clases

Este caso de uso ejecuta un único método desde el cliente. No obstante en el servidor se ejecutan dos métodos diferentes:

El método *guardar Valoracion* almacena tiene como parámetro de entrada el identificador del establecimiento y todos los datos que haya introducido el usuario en el formulario y lo almacena en la base de datos.

El método *actualizar ValoracionEstablecimiento* obtiene como parámetro el identificador del establecimiento y calcula la valoración media del establecimiento. Una vez calculada la almacena en la base de datos.

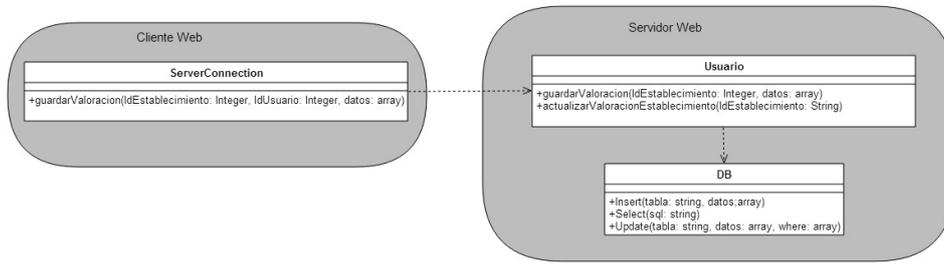


Figura 4.9: Diagrama de clases: escribir una valoración

### 4.2.5. Editar datos personales

#### Diagrama de secuencia

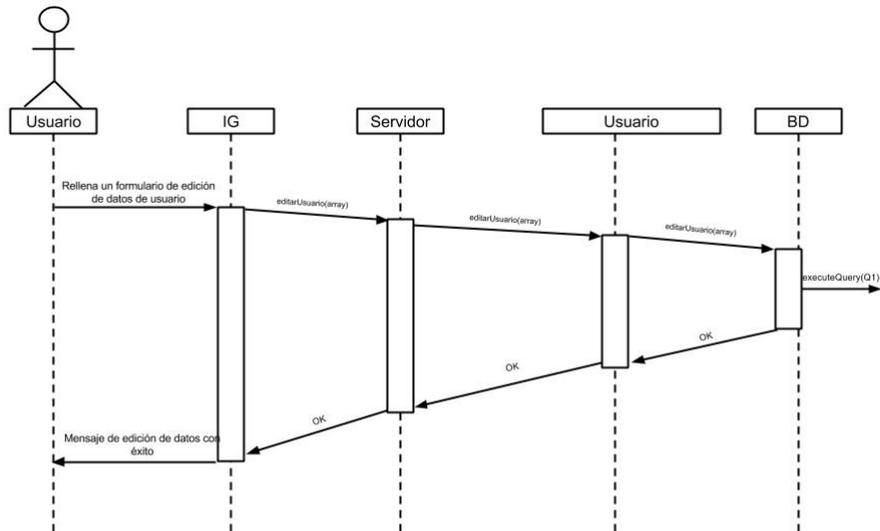


Figura 4.10: Diagrama de secuencia: editar datos personales

#### Diagrama de clases

Este caso de uso cuenta con un único método que se encarga de recoger la información introducida por el usuario y de almacenarla en la base de datos. El método *editarUsuario* recibe como parámetro de entrada todos los datos

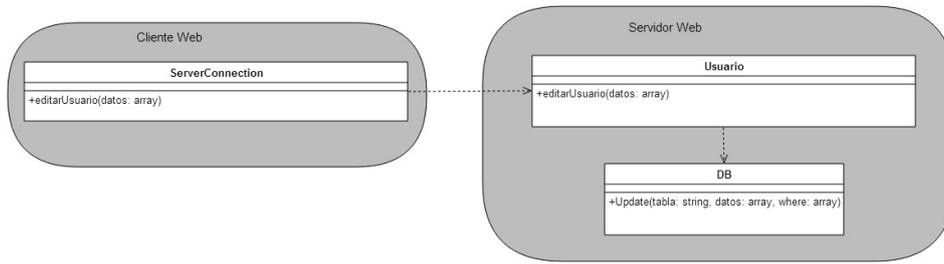


Figura 4.11: Diagrama de clases: editar datos personales

del usuario y lo almacena en la base de datos.

#### 4.2.6. Alta de un establecimiento

##### Diagrama de secuencia

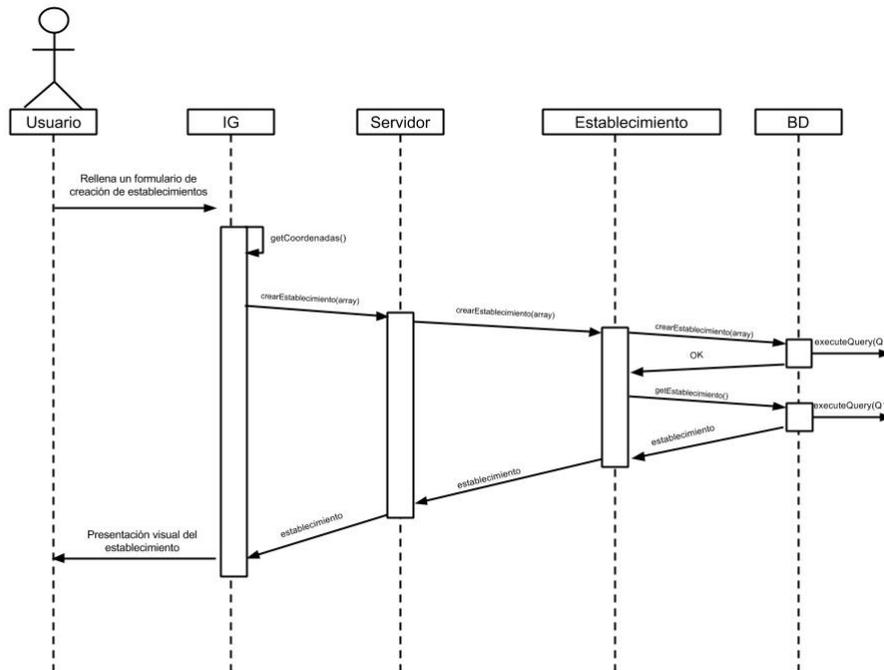


Figura 4.12: Diagrama de secuencia: alta de un establecimiento

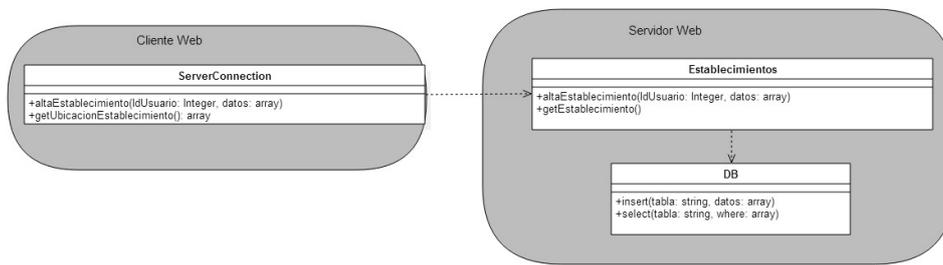


Figura 4.13: Diagrama de clases: alta de un establecimiento

### Diagrama de clases

Este caso de uso cuenta con tres métodos distintos.

El método *getUbicacionEstablecimiento* se ejecuta únicamente en el lado del cliente y sirve para obtener las coordenadas de un establecimiento.

El método *altaEstablecimiento* recoge los datos introducidos por el usuario en el formulario de alta y lo almacena en la base de datos.

Por último, el método *getEstablecimiento* obtiene los datos del establecimiento para mostrárselos al usuario.

### 4.2.7. Editar un establecimiento

#### Diagrama de secuencia

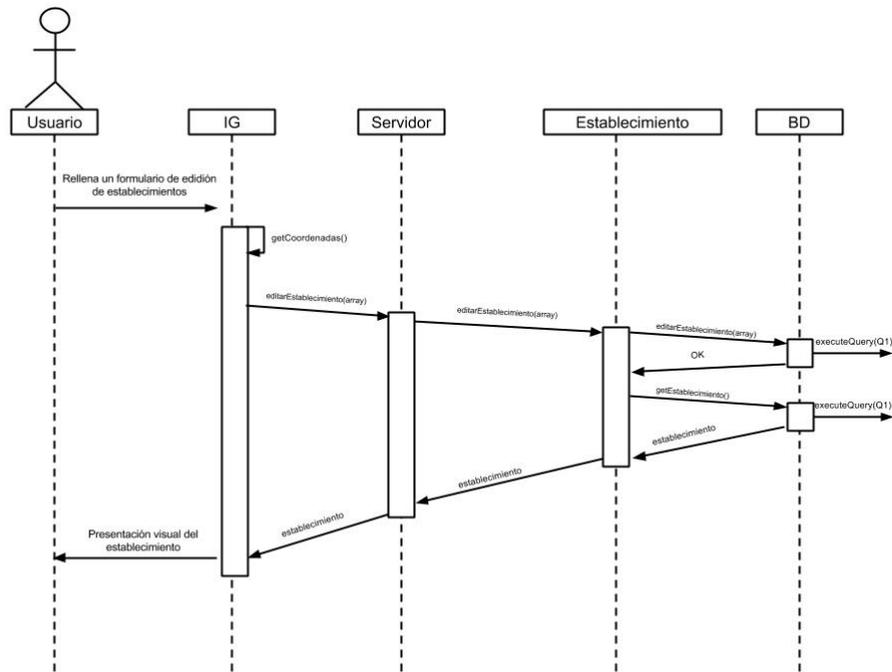


Figura 4.14: Diagrama de secuencia: editar un establecimiento

#### Diagrama de clases

Este caso de uso es muy similar al anterior y cuenta con tres métodos distintos.

El método *getUbicacionEstablecimiento* se ejecuta únicamente en el lado del cliente y sirve para obtener las coordenadas de un establecimiento.

El método *editarEstablecimiento* recoge los datos introducidos por el usuario en el formulario de edición y lo almacena en la base de datos.

Por último, el método *getEstablecimiento* obtiene los datos del establecimiento para mostrárselos al usuario.

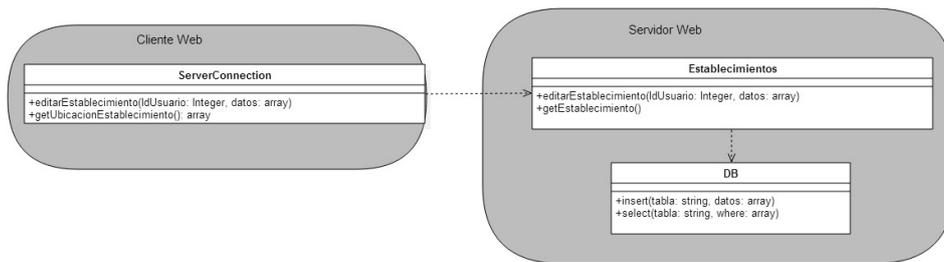


Figura 4.15: Diagrama de clases: editar un establecimiento

## Capítulo 5

# Implementación

En este capítulo se describen todos los detalles relacionados con el desarrollo y la implementación del proyecto, los problemas que han surgido y la forma en la que se han resuelto.

### 5.1. Análisis tecnológico

A continuación se expondrán las tecnologías empleadas para la realización del proyecto y otras que se barajaron y la razón de su descarte.

#### 5.1.1. HTML5

HyperText Markup Language es un estándar para la elaboración de páginas web. Este estándar define una serie de etiquetas, que más adelante son interpretadas por los navegadores y definen el contenido y la estructura de una página web, como imágenes, tablas, texto, enlaces, etc..

La última versión y la más reciente de las revisiones de HTML es HTML5. Actualmente todavía no es un estándar aunque el W3C, el consorcio que se encarga de desarrollar estándares web, ha calificado HTML5 como recomendación; un paso previo a la estandarización. Pese a ello, muchos navegadores modernos soportan las regulaciones descritas para HTML5.

En 2007 las empresas Apple, Opera y la Fundación Mozilla se unieron para formar el grupo de trabajo WHAT (Web Hypertext Application Tech-

nology). Más adelante se unirían Microsoft y Google para iniciar el desarrollo de un nuevo estándar que satisficiera los nuevos requerimiento que surgían con la aparición de los dispositivos móviles. Para ello propusieron una serie de etiquetas nuevas, funciones y métodos nuevos para los navegadores que permitieran satisfacer todas las nuevas necesidades, como reproducción multimedia de audio y video, geoposicionamiento, y presentación de imágenes vectoriales entre otras.

Más adelante, W3C se hará cargo del desarrollo de HTML5 hasta formar el proyecto de estándar que conocemos ahora.

### 5.1.2. RWD

Responsive Web Design, o Diseño web adaptable no es una tecnología en sí. Se trata de un enfoque de diseño para adaptar páginas web a dispositivos con diferentes resoluciones, permitiendo una correcta visualización en todos ellos. El objetivo principal de esta filosofía es facilitar a los usuarios de las páginas webs la visualización de la misma.

Con el crecimiento de la navegación a través de dispositivos móviles, el consorcio W3C propuso en 2008 unas ideas iniciales a las que denominó “One Web”. En ellas se describe la idea de crear una única web que fuera visualmente apta tanto para ordenadores como para dispositivos móviles.

Sobre esta idea, se ha trabajado para facilitar la lectura y visualización de las webs en dispositivos móviles consiguiendo que con una única estructura HTML y CSS se vea correctamente en todos los dispositivos.

### Twitter Bootstrap

Twitter bootstrap es un framework que asiste en la creación de webs bajo la filosofía RWD. El framework cuenta con la licencia de software libre MIT por lo que se podrá catalogar el software desarrollado con esta tecnología tanto como software libre como privado.

Twitter Bootstrap facilita una serie de herramientas, clases, iconos y fun-

ciones de javascript, todas ellas orientadas al diseño de la web. Mediante el uso de las mismas, se compone un diseño basado en bloques que en función de la resolución del dispositivo reorganiza y posiciona el contenido de forma óptima para la lectura y visualización.

### 5.1.3. AJAX

Asynchronous JavaScript And XML es una técnica empleada para enriquecer el comportamiento de las páginas web. AJAX se ejecuta mediante javascript en el cliente, generalmente en un navegador, mientras se mantiene una comunicación asíncrona con el servidor en segundo plano.

Esta técnica no es una tecnología en sí, en realidad es la unión de varias tecnologías independientes. AJAX la forman las siguientes tecnologías:

- XHTML y CSS (Cascade Style Sheet) para la representación de diseño visual de la información.
- DOM (Document Object Model) al que se accede mediante javascript para modificar e interactuar con la información presentada visualmente de forma dinámica.
- XML y JSON como formatos de intercambio de información.
- XMLHttpRequest para el intercambio de información de forma asíncrona.
- JavaScript como unión de tecnologías y lenguaje de programación ejecutable desde el navegador.

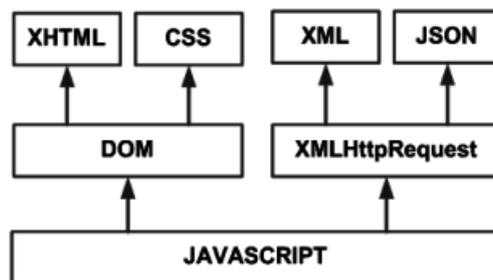


Figura 5.1: Tecnologías bajo AJAX

#### 5.1.4. MySQL

MySQL es un Sistema de Gestión de Bases de Datos relacional, multiusuario y multihilo libre bajo licencia GPL. MySQL fue desarrollado en 1995 por MySQL AB. Más adelante, en 2008 fue adquirida por SUN Microsystems que a su vez pertenece a ORACLE. Este SGBD, pese a su licencia GPL tiene derechos reservados para el uso de parte de su código, por lo que si se quiere realizar un desarrollo privado es necesario adquirir una licencia.

Cuenta con interfaces de conexión para muchos lenguajes de programación, entre ellos, C, C#, PHP y Java además de una interfaz ODBC para todos aquellos que no dispongan de una interfaz nativa en su lenguaje de MySQL para comunicarse con las bases de datos.

Es el SGBD más utilizado en aplicaciones web y se encuentra en la mayoría de los servicios de alojamiento instalado ya que hay plataformas muy comunes como Wordpress (sistema de gestión de contenidos número uno), phpBB (software de foros más utilizado), o Magento (plataforma ecommerce más utilizada) hacen uso de MySQL.

#### 5.1.5. PHP

Las siglas PHP son un acrónimo recursivo de PHP:Hypertext Preprocessor y, actualmente, es el lenguaje de programación, de código libre, más popular en el desarrollo web. Es un lenguaje interpretado, orientado a objetos, que a diferencia de JavaScript, está centrado en la ejecución de scripts en el lado del servidor en vez de en el navegador del cliente. Este lenguaje permite crear ficheros HTML e incrustar código PHP para alterar el funcionamiento de la web en función de los parámetros o los comportamientos que se tengan programados.

PHP es muy simple para el principiante en este lenguaje aunque, según se adquieren conocimientos en la sintaxis y se profundiza creando objetos, ofrece características muy avanzadas. Ofrece soporte para una gran cantidad de bases de datos, entre ellas, MySQL. Además también ofrece la posibilidad de utilizar una capa de abstracción como PDO, utilizar el

estándar de Conexión abierta a Bases de Datos ODBC, o a través de cURL o sockets.

Además de su soporte para un gran abanico de bases de datos, a parte de la generación de ficheros HTML, PHP también incluye la posibilidad de crear imágenes, ficheros PDF, mandar correos electrónicos, y una larga de características que son de utilidad para el proyecto, o pueden serlo para el trabajo futuro.

### 5.1.6. Otras tecnologías valoradas

A continuación se describirán dos tecnologías estudiadas y probadas, pero finalmente descartadas por diferentes motivos. No obstante, es posible que en el futuro se pueda volver a plantear implementar o utilizar alguna de las siguientes tecnologías.

#### Node.js

Node.js es un entorno de programación lanzado en 2009, que al igual que PHP es, para el lado del servidor. Está basado en el lenguaje ECMAScript, muy similar a JavaScript, cuyas operaciones de Entrada/Salida está orientada a eventos.

Node.js permite construir aplicaciones web escalables de forma relativamente sencilla. Node.js, a diferencia de PHP, no genera un hilo de procesamiento para cada solicitud que se realiza al servidor. Las peticiones en vez de generar un hilo, lanza la ejecución de un evento para resolver la solicitud. Gracias a esta forma de trabajar, en Node.js se pueden responder a un gran número de solicitudes sin tener problemas de memoria.

No obstante, esta tecnología no usa un servidor web que gestiona las solicitudes, como podrían ser Apache o Tomcat. En Node.js aparte de desarrollar las funcionalidades de la web, también se debe crear su propio servidor web.

Actualmente, la cantidad disponible de servicios de alojamiento que permitan la utilización de Node.js es escasa y económicamente poco viables para

un proyecto sin financiación. Además de que a bajas cargas de trabajo, con pocos usuarios que soliciten peticiones, consume más memoria que otras tecnologías como Apache.

### **Amazon Web Services**

AWS es la plataforma de computación en la nube que ofrece la compañía Amazon.com. Dentro de esta plataforma se ofrecen una gran cantidad de servicios, entre ellos los de alojamiento web, servidores DNS, bases de datos, y un largo etcétera.

La gran diferencia entre los productos que ofrece Amazon y el resto de servicios de alojamiento es la escalabilidad del sistema. AWS ofrece sus servicios a un coste según el uso que se haga del mismo. De tal forma que un servicio que consuma pocos recursos o sea poco utilizado generará menos costes que uno menos eficiente u otro que tenga muchas peticiones. Con ello, a parte de la reducción de costes, también se consiguen productos con una escalabilidad casi ilimitada gracias a que se pueden aumentar o reducir las capacidades del servicio gracias a un panel de control de administración que ofrece Amazon.

Gracias a todas estas características, AWS permite desplegar nuevas aplicaciones, plataformas o servicios en el momento, sin necesidad de esperar a un aumento de hardware, por ejemplo, que tardaría horas o días.

Además de lo anteriormente mencionado, sus servicios cuentan con sistemas con múltiples capas de seguridad operativos y físicos. Cuenta con un sistema de supervisión de seguridad que se puede gestionar para aquellos casos en los que el sistema detecte falsos positivos.

Cuenta con servidores en todos los continentes, permitiendo montar la plataforma en el servidor más cercano o en aquel que no viole las leyes de protección de datos de sus respectivos países. Por ejemplo, en España, la LOPD sólo permite almacenar los datos de los usuarios en servidores que se encuentren en la Unión Europea, además de Suiza, Hungría, Argentina y Brasil.

Para este proyecto es un sistema a valorar en el futuro, pero hasta que sea necesario debido a una gran cantidad de usuarios, se valorarán otros servicios de alojamiento más baratos.

## 5.2. La plataforma Allergic

La plataforma ha sido desarrollada en PHP. Se ha elegido este lenguaje por dos razones principales: la facilidad de uso para plataformas de este estilo, siendo muy sencillo iniciar un desarrollo desde cero y la alta disponibilidad en los servicios de alojamiento.

Para almacenar los datos se ha hecho uso de MySQL, a parte de por las mismas razones que se ha elegido PHP, por las prestaciones que ofrece para este tipo de plataformas web.

### Interfaz y persistencia de la información

Toda la interfaz de la aplicación se ha realizado con HTML5 y CSS3, dos tecnologías que interpretan la mayoría de los navegadores actuales. La interfaz se ha diseñado para que sea lo más cómoda posible para los usuarios y para conseguir un buen posicionamiento en los diferentes motores de búsqueda que se pueden encontrar en internet.

A la hora de crear una página en HTML se deben seguir una serie de reglas. HTML es un lenguaje de maquetado que sigue una estructura de validación por XML, es decir, todas las etiquetas que se abren deben tener su correspondiente etiqueta de cierre, de lo contrario, la visualización de la página puede no ser como uno espera. A parte de eso, todas las páginas deben tener una estructura similar a la que se muestra en el cuadro 5.1.

Como puede verse en el cuadro 5.1, toda la página se encuentra dentro de la etiqueta `<html>`. Dentro de esta se aprecian cuatro bloques distintos:

```
<html>

  <head>

    <!-- Zona de carga de meta datos, título, hojas de estilo -->

  </head>
  <header>

    <!-- Cabecera de la página. Menús, título, etc... -->

  </header>
  <body>

    <!-- Aquí se encuentran todos los componentes visibles de la página -->

  </body>
  <footer>

    <!-- Pie de la página. -->

  </footer>

</html>
```

Cuadro 5.1: Esquema básico de página HTML

### <head>

Este elemento es el contenedor de todos los meta datos del documento HTML. Los meta datos no contienen información que se muestre visualmente a excepción de la etiqueta <title> que contiene el título que se muestra en la cabecera del navegador. Los meta datos que se muestran suelen ser:

- <style>: estilos en la propia cabecera del documento.
- <link>: enlaces a hojas de estilos.
- <script>: contiene código o hace referencia a un fichero donde se encuentra el código javascript.
- <meta>: es una etiqueta muy amplia que puede contener todo tipo de datos, como información relacionada con el autor, palabras clave del documento, descripción del contenido de la página, etc,...

- `<base>`: etiqueta que define la localización base para todas aquellas imágenes que van a ser mostradas.

### **`<header>`**

Esta etiqueta representa el contenedor para contenido introductorio de la página como el título, subtítulo y una breve descripción, o el menú de navegación. Este uso es el más típico ya que realmente, en esta etiqueta, se puede introducir prácticamente lo que sea.

### **`<body>`**

Contiene todo el cuerpo de la página, todos los elementos a mostrar y toda la información relevante. En el estándar HTML5 no es necesaria la presencia de esta etiqueta. Tampoco lo son las etiquetas `<header>` y `<footer>`, aun así, su uso es recomendable ya que definen claramente las secciones de la página.

### **`<footer>`**

Este elemento contiene el pie de la página. Generalmente es un espacio reservado para mostrar la autoría y los derechos de la página, poner una serie de enlaces a páginas relevantes o presentar la información de contacto. No obstante, al igual que sucede con la etiqueta `<header>` se puede introducir todo tipo de información en esta sección si los requisitos de diseño visual lo requieren.

## **5.2.1. Paso de información entre páginas**

En las páginas web, para pasar información de una página a otra, o de un fichero PHP a otro se usan dos métodos distintos: POST y GET.

### **GET**

Este método se utiliza para obtener información del servidor. Mediante este método, se pueden apreciar los parámetros que se le solicitan en la URL. Como puede verse en el cuadro 5.2 se solicita la información al fichero `search.php` y se envía el parámetro `search` con valor `vitoria`.

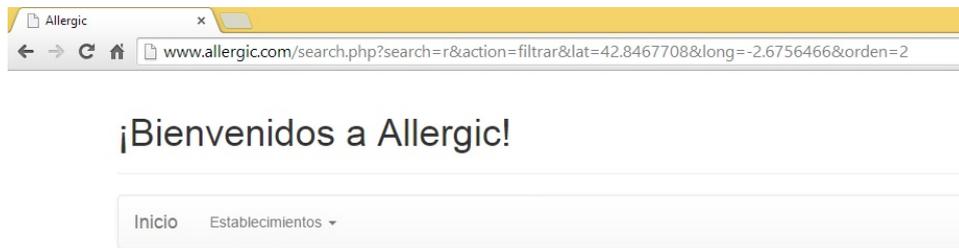


Figura 5.2: Barra de navegación de una búsqueda con coordenadas

```
http://www.allergic.com/search.php?search=vitoria
```

Cuadro 5.2: Ejemplo de envío de información mediante método GET

No hay límite de parámetros de entrada para el método GET aunque si que hay un máximo de caracteres para las URLs. Además es muy importante tener en cuenta que los parámetros de entrada deben ser para obtener datos. Al ser parámetros que se introducen en la URL, los navegadores guardan esas peticiones en el historial de navegación y guardan cache de las páginas. Por otro lado, este método permite compartir las URLs o almacenarlas en los marcadores para, por ejemplo, volver a consultar los resultados de una búsqueda previa.

## POST

Este método envía la información oculta en las cabeceras HTTP. Generalmente es usado para enviar información al servidor aunque también se puede utilizar para solicitar información de forma oculta. Utilizando este método, los parámetros que se envían son transparentes para el usuario, no ve nada en la URL a diferencia de lo que sucede con el método GET. Además los navegadores no guardan cache ni almacenan los parámetros enviados en el historial de navegación.

Por lo tanto, todos los envíos de información sensible se realizan a través de este método. El alta de usuarios, alta de establecimientos, etc... utiliza el método post. En cambio, las búsquedas de establecimientos y la visualización de los mismos se realiza a través del método GET.

### 5.2.2. Envío de información

En HTML, la forma de obtener y enviar información del usuario es a través de formularios. Se hace uso de la etiqueta `<form>` a la que se le puede especificar un método concreto. En el cuadro 5.3 se puede observar un ejemplo de un formulario con varios campos de entrada.

```

<form action="registrar" method="post">

  <input type="hidden" name="accion" id="accion" value="registro"
  />
  <input type="text" name="email" id="email" />
  <input type="password" name="pass" id="pass" />
  <input type="submit" name="enviar_registro"
  id="enviar_registro" value="enviar"/>

</form>

```

Cuadro 5.3: ejemplo de formulario HTML

En el cuadro 5.3 pueden verse dos etiquetas diferentes aunque en una de ellas hay cuatro tipos distintos. La primera de todas es la etiqueta `<form>` a la que a través del parámetro `action` le establecemos a dónde debe mandar la información y a través del parámetro `method` le establecemos el método.

Las etiquetas `<input>`, por otro lado, contienen la información a enviar. Hay muchos tipos diferentes de etiqueta `<input>`. En el cuadro 5.3 se aprecian cuatro diferentes, *hidden*, *text*, *password* y *submit*. A continuación se muestran los tipos diferentes de etiquetas que se pueden utilizar:

- `<input type="hidden">`: entrada oculta, usada para enviar información desconocida por el usuario, como acciones, índices, etc,...
- `<input type="text">`: entrada de texto simple de una única línea.
- `<input type="password">`: igual que la entrada de texto normal solo que no se muestra el contenido al usuario. El contenido que se introduce permanece ofuscado al usuario.
- `<input type="submit">`: botón que acciona el envío del formulario.
- `<input type="file">`: permite subir ficheros. Para habilitar la carga de archivos al servidor en necesario utilizar el método POST y añadir a la etiqueta `<form>` el siguiente parámetro: `enctype="multipart/form-data"`.

- `<textarea></textarea>`: area de texto. a diferencia de la entrada de texto, esta permite saltos de línea y un mayor numero de caracteres.

### 5.2.3. Recogida de información

La captura de información se realiza mediante PHP. Todos los elementos enviados a través de cualquiera de los métodos queda almacenado temporalmente en sus respectivas variables. Como puede verse en el cuadro 5.4, cada uno de los métodos cuenta con su propia variable. En PHP no hay el concepto de listas con punteros que hay por ejemplo en java, todos los listados de elementos están almacenados en arrays.

- Captura de información del método GET:  
`$search = $_GET['search'];`
- Captura de información del método POST:  
`$pass = $_POST['password'];`
- Captura de información de cualquier método:  
`$param = $_REQUEST['parametro'];`

Cuadro 5.4: captura de información en PHP

En PHP, a parte de las variables que contienen únicamente los parámetros de sus respectivos métodos, hay una variable que contiene los parámetros de todos los métodos. Esta variable solo es recomendable usarla cuando no importa a través de qué método llega la información.

### 5.2.4. Obtención de coordenadas

Para ubicar los dispositivos y los establecimientos, es necesario la obtención de las coordenadas del mismo. El resultado de la obtención de coordenadas es el mismo. la latitud y la longitud en grados decimales. Las coordenadas se obtienen en este formato para simplificar el calculo de distancias entre un dispositivo y el establecimiento.

Pese a que el resultado de la obtención de coordenadas es el mismo, el método de obtenerlos es completamente distinto. Para obtener la posición del establecimiento es necesario ubicar un puntero sobre un mapa que devuelve las coordenadas del puntero. En cambio, para obtener las coordenadas del

dispositivo, es necesario acceder a funciones del navegador para geolocalizar el mismo.

En el cuadro 5.5 se puede ver un ejemplo de visualización de un mapa en una página web. En este caso se hace uso de la API de Google Maps que tiene una gran cantidad de posibilidades. en el pude verse la creación de un mapa partiendo de unas coordenadas, y el posicionamiento de un puntero partiendo de esas mismas coordenadas.

```
//creamos una variable con las coordenadas centrales del mapa.
var latLng = new google.maps.LatLng(coordenadaX,coordenada);
/*Definimos algunas opciones del mapa a crear:
 * center: coordenadas donde se centra el mapa
 * zoom: aumento del mapa
 * mapTypeId: tipo de mapa: carreteras, satélite o híbrido.
 */
var myOptions = {
    center: latLng,
    zoom: 15,
    mapTypeId: google.maps.MapTypeId.ROADMAP
};
//creamos el mapa con las opciones anteriores y le pasamos el elemento div
map = new google.maps.Map(document.getElementById("map_canvas"),
    myOptions);
/*creamos el marcador en el mapa
 * map: el mapa creado anteriormente
 * marker: el mapa creado en el paso anterior
 * position: objeto con latitud y longitud
 * draggable: el puntero se puede arrastrar o no
 */
marker = new google.maps.Marker({
    map: map,
    position: latLng,
    draggable: true
});
```

Cuadro 5.5: mostrar un mapa en una página HTML

Para facilitar al usuario el posicionamiento del establecimiento, al introducir la dirección completa del mismo, se le sugiere una posición. Un ejemplo de esta funcionalidad se puede ver en el cuadro 5.6. Una vez más se hace uso

de la API de Google Maps. Además, en el mismo ejemplo, se puede ver el código que obtiene las coordenadas del puntero cuando es arrastrado.

```

//creo una variable de búsqueda para la dirección completa
var búsqueda = direccion + ', ' + provincia + ', ' + país;
//hago la llamada al geocoder
geocoder.geocode( { 'address': búsqueda}, function(results, status) {
    //si el estado de la llamado es OK
    if (status == google.maps.GeocoderStatus.OK) {
        /*en la variable results[0].geometry.location se encuentran
        * almacenadas las coordenadas.
        */
        //centro el mapa en las coordenadas obtenidas
        map.setCenter(results[0].geometry.location);
        //coloco el marcador en dichas coordenadas
        marker.setPosition(results[0].geometry.location);
        /* Añado un listener para cuando el marcador se termine
        * de arrastrar actualize el formulario con las
        * nuevas coordenadas
        */
        google.maps.event.addListener(marker, 'dragend', function(){
            //llamo a función que almacena las coordenadas
            updatePosition(marker.getPosition());
        });
    } else {
        //si no es OK devuelvo error
    }
});

```

Cuadro 5.6: obtener coordenadas de una dirección.

Con el código expuesto en los cuadros 5.5 y 5.6 ya se puede obtener las coordenadas de un establecimiento. Ahora nos queda obtener las coordenadas del dispositivo.

Para obtener las coordenadas de un dispositivo, el navegador del mismo debe ser compatible con HTML5 y sus diversas funciones. Si el navegador no sigue dicho estándar no se podrá obtener la ubicación del mismo y, a su vez, el usuario necesitará proporcionar permisos a la plataforma para que esta obtenga la ubicación. En el cuadro 5.7 se puede ver un ejemplo de código para obtener las coordenadas del dispositivo.

```
//primero compruebo si el navegador permite obtener la ubicación del dispositivo.
if(navigator.geolocation) {
    // llamo a una función del navegador para obtener la ubicación.
    navigator.geolocation.getCurrentPosition(onSuccessGeolocating,
        onErrorGeolocating, {
            enableHighAccuracy: true,
            maximumAge: 5000,
            timeout: 20000
        });
} else {
    //el navegador no permite la geolocalización.
}
```

Cuadro 5.7: obtener las ubicación de un dispositivo mediante javascript

La función *getCurrentPosition* del navegador que se puede ver en el cuadro 5.7, tiene como parámetros de entrada dos funciones javascript. La que se ha definido en el ejemplo como *onSuccessGeolocating*, es la función a la que se llama cuando se consigue geoposicionar el dispositivo con éxito. La función recibe como parámetro las coordenadas del dispositivo en un objeto de tipo *Position*, que además de las coordenadas proporciona la fecha exacta en la que se ha obtenido la ubicación.

La segunda función, *onErrorGeolocating*, se le llama cuando no ha sido posible posicionar el dispositivo ya sea por falta de conectividad con los satélites o repetidores, porque el usuario no ha dado permiso a la plataforma para obtener la ubicación o porque ha expirado el tiempo de espera (*timeout*).

### 5.2.5. Cálculo de distancia entre dispositivo y establecimiento

En la sección anterior hemos definido la obtención de las coordenadas de los establecimientos y del dispositivo. Las coordenadas del dispositivo tienen un margen de error muy variable desde los 30 metros de margen de error máximo que proporciona el posicionamiento GPS a los 100 metros que proporciona la triangulación mediante redes de telefonía.

No obstante, para el cálculo de la distancia entre el dispositivo y el esta-

blecimiento se ha obviado el margen de error ya que pese a que en algunos casos sea elevado, no es posible mejorar el posicionamiento a través de una plataforma web.

El cálculo de la distancia entre dos puntos del planeta, por muy cercanos que sean, no es el cálculo entre dos puntos en un espacio plano. Evidentemente, en un margen de pocos kilómetros de distancia, de podría calcular sin tener en cuenta la curvatura de la tierra. Pese a ello, se ha optado por calcular la posición entre dos puntos de la superficie de una esfera.

Para el cálculo de la posición se ha hecho uso de la *Fórmula del Haversine*, un modelo matemático que permite conocer la distancia entre dos puntos del globo conociendo la longitud y latitud de ambos puntos. La Formula contempla que para cualquier par de puntos sobre una esfera:

$$\text{havrsin}\left(\frac{d}{R}\right) = \text{havrsin}(\varphi_1 - \varphi_2) + \cos(\varphi_1) \cos(\varphi_2) \text{havrsin}(\Delta\lambda).$$

Donde  $\text{havrsin}(\theta) = 2\text{sen}^2(\frac{\theta}{2}) = (1 - \cos(\theta))$ ,  $\varphi$  representan las latitudes de cada uno de los puntos,  $\Delta\lambda$  es el diferencial entre longitudes de los puntos,  $R$  es el radio de la esfera y  $d$  es la distancia entre los dos puntos.

En este caso necesitamos la distancia entre dos puntos por lo que tenemos que aislar la variable  $d$ . Para aplicar esta fórmula es necesario realizar los cálculos con las formulas en radianes y, una vez aislada y aplicando los cambios oportunos, nos quedamos con la siguiente fórmula:

$$d = R * 2 \arcsin\left(\sqrt{\sin^2\left(\frac{(\Delta\varphi)\pi}{180 * 2}\right) + \cos\left(\frac{\varphi_1 * \pi}{180}\right) * \cos\left(\frac{\varphi_2 * \pi}{180}\right) * \sin^2\left(\frac{(\Delta\varphi)\pi}{180 * 2}\right)}\right)$$

En este caso, tenemos la constante  $R$  el radio de la tierra que tiene una media de 6371 que es la que se va a aplicar en esta fórmula para la plataforma.

Dado que un grado de error, en una escala del tamaño de la Tierra es de entorno a 111 kilómetros no se pueden perder decimales reduciendo partes de la ecuación a excepción de las multiplicaciones entre enteros. Pese a ello, la ejecución de consultas en MySQL no es demasiado costoso, entorno a 0.01 segundos por cada par de coordenadas procesada. esto hace que una consulta con 10 resultados se realice en aproximadamente 0.1 segundos.

### 5.3. Arquitectura de Allergic

La arquitectura seleccionada para la realización de este proyecto ha sido la arquitectura en tres niveles. A la hora de diseñar una plataforma web hay que tener en cuenta la importancia de simplificar lo máximo posible los cambios de apariencia o funcionamientos básicos de la página. Esta arquitectura permite la modificación de cualquiera de las tres capas sin afectar a las otras dos. Esta modularidad permite hacer, por ejemplo, un cambio de SGBD sin necesidad de modificar toda la plataforma o hacer cambios en la parte visual sin necesidad de cambiar nada más.

A continuación se presentarán las tres capas desarrolladas en este proyecto.

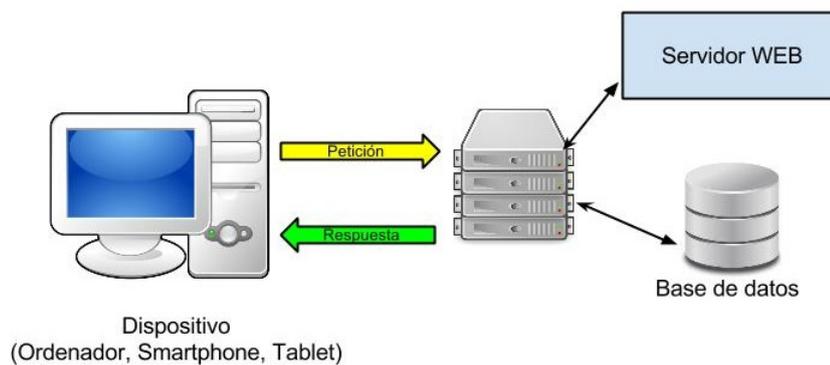


Figura 5.3: esquema de la arquitectura en tres niveles

### 5.3.1. Capa de Datos

Esta capa es en la que residen los datos. En este caso, se trata de un SGBD MySQL, como se ha comentado antes. Esta capa en la plataforma cobra una amplia importancia ya que debe estar optimizada para una gran cantidad de consultas. Además el almacenaje de datos es una de las claves de la plataforma, ya que los datos son los que van a dar valor a la misma. Por ello además de diseñarse para una gran cantidad de accesos simultáneos, se ha diseñado para que las consultas sean ágiles y pese a que haya millones de registros no exceda en utilización de memoria y tiempo.

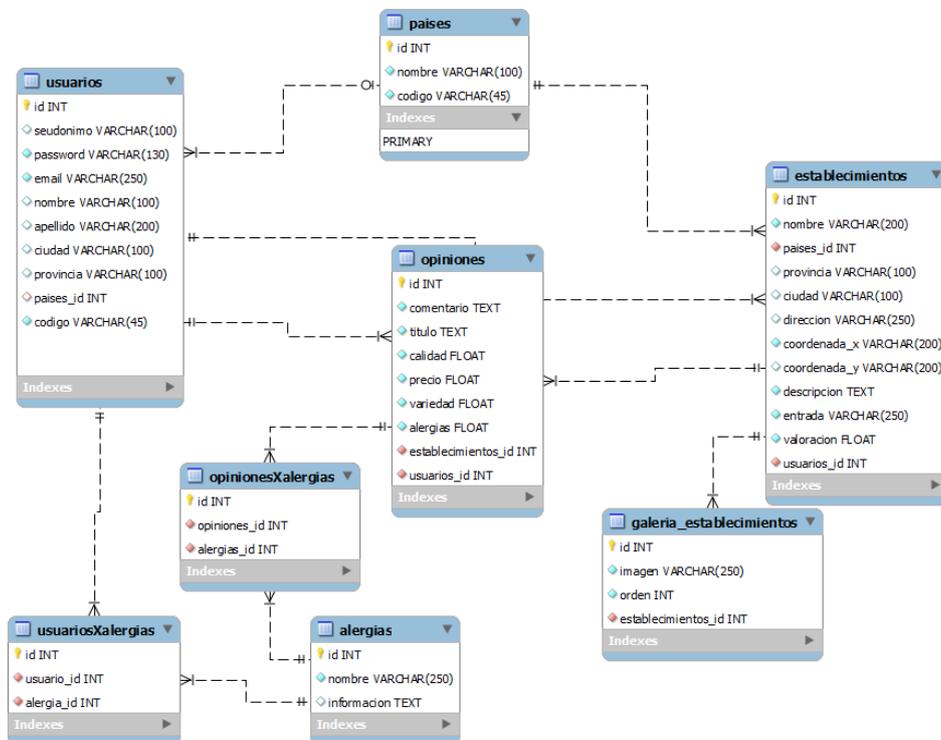


Figura 5.4: modelo de Entidad Relación

En la figura 5.3 se presenta el modelo de Entidad Relación para este proyecto. Se ha diseñado teniendo en cuenta lo anteriormente nombrado. Como puede apreciarse todas las tablas están relacionadas entre sí, además, se ha intentado prevenir la inconsistencia de datos añadiendo restricciones con claves privadas en las tablas referenciando valores de otras.

### 5.3.2. Capa de Negocio

Esta es la capa en la que se tratan y manejan todos los datos antes de almacenarse en la capa de datos y antes de mostrarse en la capa de presentación.

El desarrollo de esta capa se ha realizado de forma modular, tratando de forma independiente cada una de las funcionalidades de la plataforma. Además, se ha tenido en cuenta la posible reutilización de alguna de las funciones para el futuro, por lo que se han desarrollado de tal forma que cualquier dato que se obtiene, es accesible desde cualquier punto de la plataforma. Esto se ha hecho así para que en un futuro no sea necesario replantear el modelo de negocio ni modificar funciones ya desarrolladas.

### 5.3.3. Capa de presentación

La capa de presentación de esta aplicación, engloba todas las partes visuales, maquetación y estilos de la plataforma. Además de esto, se encarga de la captura de los datos para ser procesados por el servidor. Estos datos, antes de ser capturados son validados con javascript en el lado del cliente. Además, para los datos más sensibles, se realiza una segunda validación en el lado del servidor, ya que la validación del lado del cliente puede ser saltada si el usuario tuviera conocimientos suficientes.

La capa de presentación cobra especial importancia en esta plataforma ya que cuenta con un diseño adaptativo a cualquier dispositivo. Además se ha hecho especial hincapié en cumplir con las normas de la W3c en cuanto a accesibilidad web se refiere. No obstante es un apartado a abordar con mayor rigor en un futuro ya que pese a que se obtiene una buena calificación, no es la óptima para aquellos usuarios con problemas de accesibilidad.

## Capítulo 6

# Evaluación del sistema

A continuación se recogen una serie de pruebas visuales y de rendimiento que se han realizado para evaluar las propiedades de la plataforma. Las pruebas se realizan con intención de optimizar la experiencia de usuario y conseguir una plataforma fluida independientemente de la carga de usuarios que accedan a la misma.

### 6.1. Diseño responsive

Desde el 2008 el diseño responsive ha ido cobrando más fuerza hasta el punto de ser necesario para obtener la puntuación más alta en los análisis oficiales de la W3C en cuanto a experiencia del usuario. Un gran impulsor de esta tecnología es Twitter que puso a disposición de todo el mundo bajo licencia MIT Bootstrap, pero el impulsor en la sombra de las buenas prácticas en cuanto a diseño y maquetación web siempre ha sido Google.

Para obtener una buena puntuación y, por lo tanto, un posicionamiento mejor en los buscadores, los motores de búsqueda han empezado a exigir que las páginas tengan un alto nivel de satisfacción en la experiencia de usuario.

A continuación se van a exponer una serie de ventanas que teniendo la misma maquetación, el contenido se va a adaptar automáticamente a la anchura de la pantalla del dispositivo.

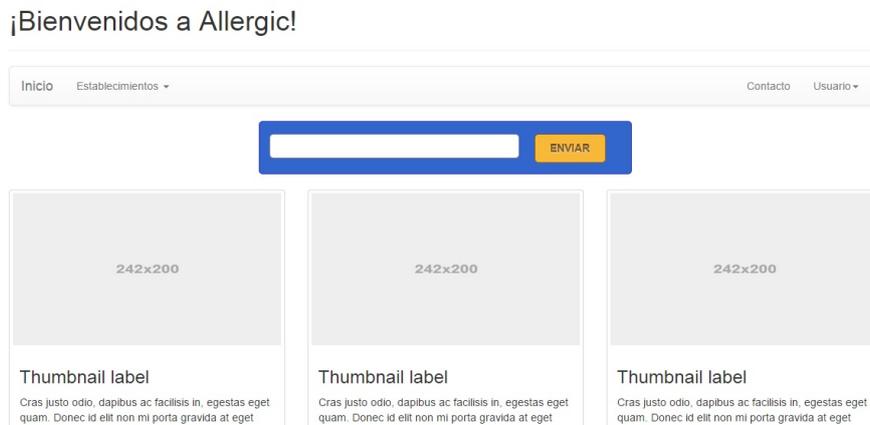


Figura 6.1: pantalla inicial ancha

Como puede verse en las figuras 6.1 y 6.2, en la versión ancha el menú ocupa toda la pantalla abarcándola de lado a lado, mostrando contenido en ambos extremos de la misma. En cambio, en la versión estrecha, enfocada a dispositivos móviles, el menú se agrupa en una pestaña desplegable de tal forma que la navegación en la página es la misma pero la visualización de los elementos es más compacta y a su vez, visualmente relajada.



Figura 6.2: pantalla inicial estrecha

En la figura 6.3 se muestra el resultado de una búsqueda cualquiera, en la que aparecen los resultados en bloques mostrados en filas de tres elementos. En resoluciones anchas se ha limitado a 3 elementos por fila ya que se considera que un tamaño menor para la visualización de los resultados de búsqueda puede resultar incómodo.

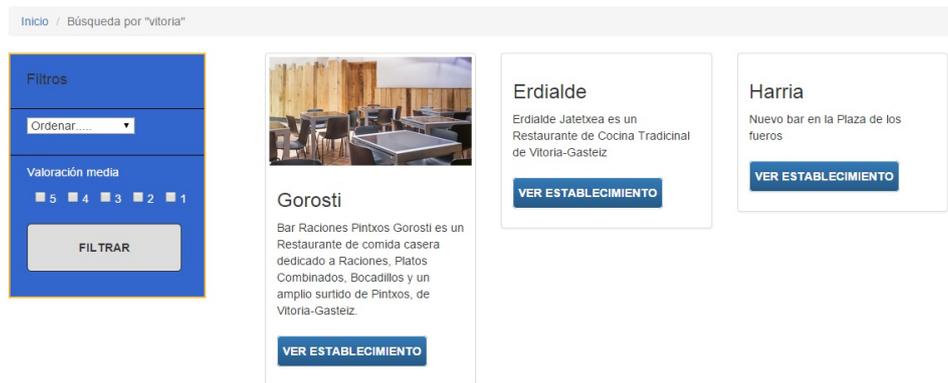


Figura 6.3: pantalla de búsqueda ancha

La siguiente figura 6.4, muestra los mismos elementos que se han buscado en la figura 6.3 visualizándose a través de un dispositivo móvil. En ella se puede ver que los resultados han conservado su tamaño pero no la posición. Se han ordenado de tal forma que ahora aparecen uno debajo de otro verticalmente ordenados.



Figura 6.4: pantalla de búsqueda estrecha

Por último, a continuación se mostrarán la visualización de el detalle de un establecimiento.

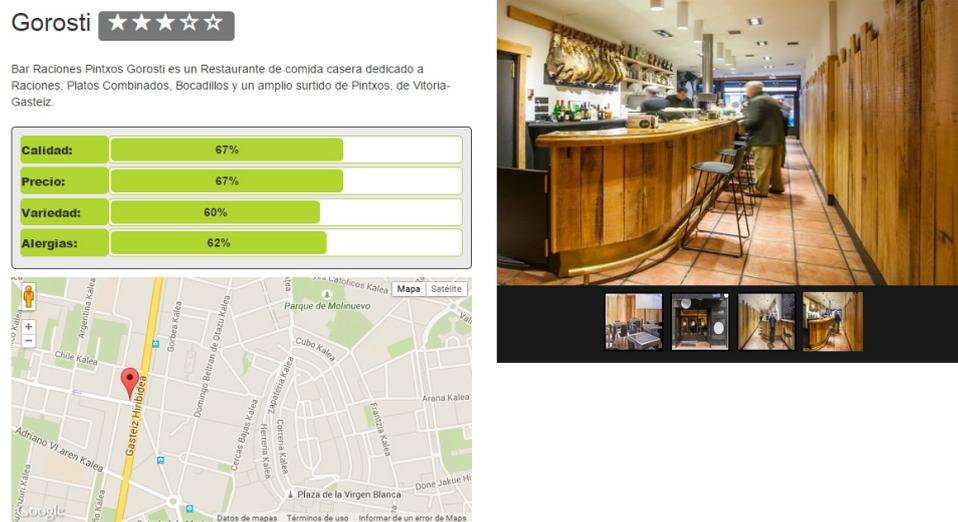


Figura 6.5: pantalla detalle ancha

Como puede verse en la figura 6.5, los distintos elementos de la página aparecen representados de tal forma que aprovechan el ancho de la pantalla para ocupar la menor altura posible y evitar desplazamientos verticales en

la página para visualizar la información.



Figura 6.6: pantalla detalle estrecha

En cambio, en la figura 6.6 se puede ver cómo en un dispositivo móvil, la información no cabe en el tamaño de una pantalla. Pese a ello, todo el contenido de la página se ha reubicado para mostrarse desplazando únicamente la pantalla de forma vertical. Evidentemente, la información que cabe en una pantalla de menor tamaño es menor, por ello, se ha considerado priorizar la información del establecimiento, la valoración y la posición para beneficio de los usuarios. Desplazando verticalmente hacia abajo la página se seguiría disponiendo del resto de información que se muestra en la pantalla ancha.

## 6.2. Velocidad de carga

Al tratarse de una plataforma web, el tiempo de carga de la página debe ser óptimo. Esto no solo afecta a la experiencia del usuario, también afecta al posicionamiento y unos tiempos de carga altos pueden evidenciar código poco depurado y poco eficiente.

Es por ello por lo que se ha considerado los tiempos de carga un punto importante del proyecto. Para optimizar el proceso de carga ha sido necesario optimizar el código de ejecución, tanto en el servidor como la pequeña parte

que se ejecuta en el cliente. Con objetivo de asegurar que se está evaluando el rendimiento de la plataforma, para evitar latencias de red se ha hecho la prueba en un servidor local. Un servidor con 512MB de memoria y un procesador ARM de un único núcleo a 1.2GHz. El servidor a parte del web sirve otras peticiones a otras aplicaciones. La carga del servidor se asume como parte del procesamiento ya que en otros servidores sin dedicación exclusiva, se comparten recursos con otros procesos de la misma manera.

Con ello se han conseguido los siguientes tiempos de carga:

Página	peso(Kb)	T. 1.Carga	T. 2.Carga
Inicio	663,6	312 ms	275 ms
Búsqueda	750.9	362 ms	341 ms
Establecimiento	2257	9620 ms	2670 ms

Tabla 6.1: Tiempos de carga de la web

Como puede verse, tanto en la ventana de Inicio como en la de búsqueda los tiempos de carga son muy reducidos. En la página de inicio no se realiza ninguna operación a parte de la comprobación de si el usuario está validado o no. en cambio, en la ventana de búsqueda se ejecuta la consulta a la base de datos, y pese a ello se han conseguido resultados muy similares.

La excepción en este caso es en la ventana de Establecimiento. Tras un análisis más exhaustivo del tiempo de carga, midiendo rigurosamente los tiempos de carga de todos los componentes para determinar la razón de estos tiempos de carga se ha llegado a determinar la razón de los elevados tiempos. El servidor, al tratarse de una red local responde entre 10 y 20 ms y el contenido del resto de la página, con imágenes, texto etc.. se carga en una media de 531 ms.

La utilización de Google Maps para representar el establecimiento en un mapa hace que sean necesarias llamadas a servidores externos una vez cargada la web. Esto provoca que una vez cargada la web, la página desde el lado del cliente realiza una serie de llamadas a los servidores de google para obtener imágenes, y una serie de datos necesarios para la representación del

mapa.

Concluido este análisis de puede considerar como bueno el rendimiento de la plataforma.

### 6.3. Escalabilidad del modelo de datos

Lo primero a la hora de evaluar la escalabilidad de un producto es tener en cuenta que la base de datos no es la que determina al completo la escalabilidad de la misma. No obstante, sí que es una de las partes más importantes. El SGBD elegido por motivos de comodidad, disponibilidad y economía no cumple con el requisito principal para considerar un producto escalable al completo, que es la distribución del sistema. MySQL, a diferencia de SGBDs más potentes como Oracle, no dispone de la opción de bases de datos distribuidas. Se podría simular realizando diferentes conexiones en función del lugar de donde se conecta un usuario, pero no sería un sistema distribuido en sí.

En términos de la estructura del modelo de datos, como puede verse en la figura 5.3, no existen redundancias de datos, las tablas cuentan con claves privadas para evitar la inconsistencia de datos y el SGBD tiene habilitada la cache de consultas para reducir el tiempo de respuesta. Gracias a ello y a una optimización realizada en la capa de negocio para el acceso a los datos, en los que únicamente se obtienen los estrictamente necesarios, hacen que la base de datos tenga respuestas rápidas que consumen pocos recursos.

Con todo ello se consigue a parte de facilitar la escalabilidad al consumir menos recursos, ejecutar una consulta con uniones entre 3 tablas con números de tuplas que superan el millón en menos de 0.489 segundos la primera vez. Las consultas posteriores, debido a la cache del SGBD se reduce más de diez veces.

Además, gracias a la arquitectura en tres capas, no se necesita un esfuerzo muy grande para migrar de MySQL a Oracle, permitiendo de esta forma la distribución de SGBDs en diferentes localizaciones consiguiendo de esta forma la escalabilidad optima para la plataforma.

## Capítulo 7

# Seguimiento y control

En este capítulo se mostrarán los procesos de seguimiento y control empleados para garantizar la calidad y la finalización del proyecto dentro de los plazos establecidos dentro de la planificación inicial del proyecto. A su vez, se mostrarán las desviaciones que ha habido y las razones de las mismas.

### 7.1. Alcance del Proyecto

La elaboración de todos los objetivos planificados ha sido cumplida correctamente, no obstante, el alcance ampliado no ha podido realizarse. Los plazos de entrega y los hitos internos del proyecto han sido cubiertos con éxito habiendo surgido desviaciones en las dedicaciones a cada tarea, que se especificarán más adelante.

### 7.2. Control y gestión del tiempo

En esta sección se recogen los cambios que se han ido realizando para reasignar recursos a las tareas que en un principio se planificaron con demasiado tiempo y utilizar ese tiempo para las tareas que se han complicado. A continuación se describen, primero las tareas que se han realizado con menores costes de los planificados y posteriormente las tareas que han requerido mayor dedicación de la prevista.

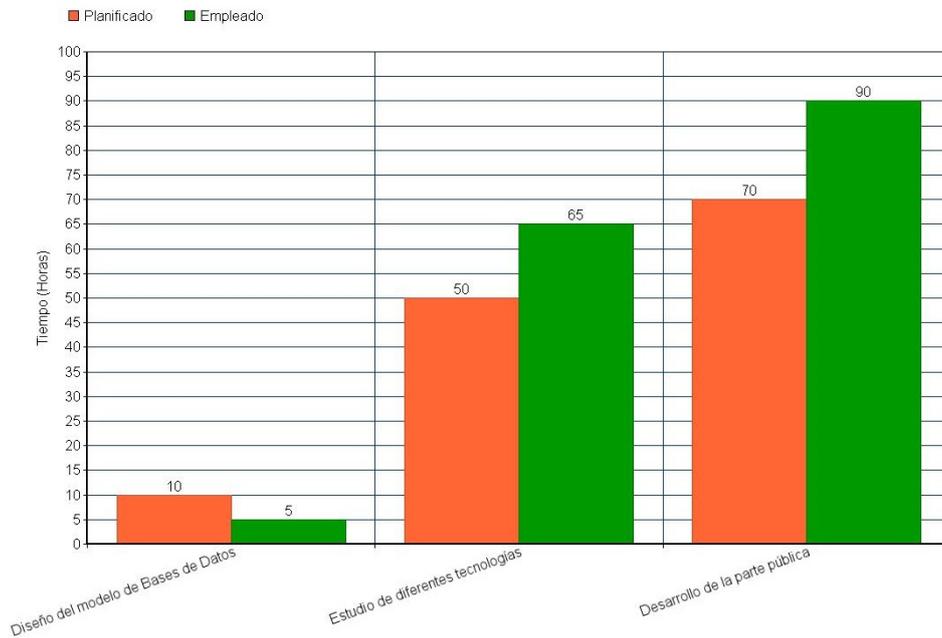


Figura 7.1: Diferencias de tiempo empleadas en las tareas con desvíos

### 7.2.1. Tareas que se planificaron con estimación pesimista

Estas son las tareas que se realizaron con un coste inferior al estimado inicialmente.

- Diseño del modelo de Bases de Datos

### 7.2.2. Tareas que se planificaron con estimación optimista

Estas son las tareas que se realizaron con un coste superior al estimado inicialmente.

- Estudio de diferentes tecnologías
- Desarrollo de la parte pública

Todos los desvíos han sido solventados añadiendo una mayor dedicación por parte del alumno para conseguir alcanzar el éxito en las fechas previstas con la menor repercusión posible. Esto se ha hecho de esta forma para evitar tener que replanificar el grueso del proyecto y para prevenir futuros agobios en fechas de entrega.

En la Figura 7.1 se puede ver un gráfico en el que se muestran la diferencias de tiempo empleadas.

## Capítulo 8

# Conclusiones

En este capítulo se expondrán las conclusiones finales del proyecto, las posibilidades de ampliación y/o mejora que cuenta la plataforma y una pequeña valoración personal.

### 8.1. Desarrollo del proyecto

El proyecto se ha desarrollado en un total de ocho meses, del 1 de octubre de 2014 al 8 de mayo de 2015. La dedicación al proyecto ha sido de 5 días a la semana 3-4 horas al día de media, resultando en un proyecto realizado en aproximadamente 450 horas.

Para la realización del proyecto se tuvo que recordar todos los procedimientos y tecnicas que se emplean para definir el diseño y el análisis del proyecto. Tras una serie de correcciones de forma proactiva se consiguió llegar al diseño actual que condujo a la finalización del proyecto en los plazos definidos. La abstracción que da el lenguaje UML permite ver el problema de forma global y especificar claramente todos los requerimientos técnicos que debe implementar un proyecto.

El proyecto, debido a su naturaleza de plataforma web podría ser desarrollado con varias tecnologías, todas ellas, completamente válidas. Para el desarrollo del mismo se estudiaron y analizaron varias tecnologías y servicios que al final fueron descartados por diversas razones quedándonos, finalmente, con las tecnologías empleadas para este proyecto, es decir, principalmente:

PHP, MySQL, Tweeter Bootstrap y un alojamiento con un servidor Apache o similares.

Durante el desarrollo del proyecto se adquirió el conocimiento necesario para la mejora del posicionamiento en los motores de búsqueda, principalmente en Google. Es ahí cuando tras leer y analizar una serie de distintas maquetaciones se obtienen resultados que pueden indicar el peso de Google en el mundo de Internet. Actualmente existe una guerra contante entre empresas, páginas de ocio, blogs, etcétera.. para aparecer en los resultados de las búsquedas en Google los primeros. Este tipo de guerra permite a Google establecer las reglas. Esto, pese a que pueda parecer una conclusión mala, actualmente no lo es. Google analiza las páginas web teniendo en cuenta la experiencia del usuario, buscando palabras clave dentro del contenido y una larga serie de análisis para determinar la calidad y el valor de la página para los usuarios. Esto, teniendo en cuenta que es una empresa privada cuyo principal objetivo es el lucro, es muy positivo actualmente, pero en cualquier momento, puede cambiar las reglas del posicionamiento.

La elección del lenguaje PHP fue principalmente por el conocimiento básico que se tenía al inicio del proyecto. Una vez finalizado el proyecto, se puede considerar que el conocimiento de la estructura y la sintaxis a la hora de llevar buenas prácticas en la programación en este lenguaje ha mejorado considerablemente, en parte, debido al diseño enfocado para un lenguaje orientado a objetos.

Por otro lado, el conocimiento adquirido en la maquetación y diseño de páginas web ha sido sin partir de ninguna base. El conocimiento que se tenía sobre estos temas, incluido HTML5 o CSS3 se limitaba a conocer la existencia de ambos. Tras la finalización del proyecto, se puede considerar que se tiene una base sólida.

Al punto anterior hay que añadir la utilización de la maquetación por capas adaptativas de Twitter Bootstrap. Una tecnología inicialmente desconocida que permite la creación de páginas RWD de manera ágil y sencilla. Durante el desarrollo del proyecto se tomaron varias páginas web con un buen diseño responsive y sorprendió la cantidad de páginas que usan la tecnología de Twitter. Además se puede utilizar para convertir antiguas webs

estáticas no adaptables a webs adaptables de forma relativamente sencilla, y debido a los nuevos requisitos de Google en este tema, actualmente hay un gran mercado de adaptación de páginas a RWD.

Otra de las tecnologías nuevas que se emplearon fue AJAX, una unión de tecnologías que a primera vista parecía compleja, tras estudiarla y comprender su funcionamiento se adquirieron los conocimientos para desarrollar pequeñas funcionalidades y para valorar la integración de nuevas funcionalidades con esta técnica.

El entorno de desarrollo seleccionado, Eclipse para PHP, dado que ya se conocía con anterioridad no hizo falta ningún proceso de adaptación. Además sorprendió conservando algunas de las funcionalidades que proporcionaba a otros lenguajes como a java, con una amplia documentación en funciones propias de la tecnología.

Finalmente, para la evaluación del rendimiento de la plataforma, se diseñó un estudio básico de rendimiento para el cual se estudiaron los factores relevantes en el rendimiento de una web, que son: velocidad de carga y utilización de recursos del sistema. Se han simulado una serie de pruebas con distintas herramientas para medir el rendimiento y se ha llegado a la conclusión de que el rendimiento es óptimo. No obstante, la elección de MySQL como SGBD provoca que la escalabilidad de la plataforma está limitada.

## 8.2. Trabajo futuro

Al tratarse de una plataforma desarrollada de forma modular, el desarrollo de nuevas funcionalidades es sencillo. Además, antes de considerar el lanzamiento de la plataforma hay ciertos apartados que hay que realizar.

### 8.2.1. Requisitos antes de lanzar la plataforma

Estos son los requisitos mínimos necesarios antes de lanzar la plataforma a un entorno de producción:

- Revisión de maquetación y rediseño de secciones de la plataforma. Dar una imagen más vistosa a la plataforma.
- Redacción de documentos legales para cumplimentar la LOPD.

- Pasar una auditoría de seguridad a la plataforma y auditar el hosting seleccionado. Al tratar datos de establecimientos y de usuarios es necesario auditar todo el desarrollo realizado y el sistema donde esté alojado.
- Programación de copias de seguridad de base de datos y de datos almacenados en la plataforma.
- Alta en herramientas de posicionamiento web. Creación de ficheros con el mapa de la web conteniendo todas las páginas visibles de la misma
- Desarrollo de una administración visual para la plataforma en la que se puedan modificar los elementos de la web.

### 8.2.2. Ampliación de la plataforma

Aquí se expondrán las ideas de mejora y de ampliación para la plataforma.

- Creación de contenido estático. Páginas de información para los visitantes de la página.
- Monetización de la página, es decir, conseguir rentabilidad económica a través de la plataforma. Para ello se han pensado los siguientes puntos:
  - Mantener contacto con los establecimientos ofreciendo mejoras de contenido y publicidad en la página. Esto nunca puede perjudicar a los usuarios.
  - Iniciar contacto con asociaciones de alérgicos para la difusión de la plataforma. Ofrecer publicidad gratuita a las asociaciones y secciones de donaciones a las mismas a través de la plataforma.
  - Permitir a los establecimientos mejorar su cuenta para poder introducir menús, ofertas, precios y eventos en el detalle de su establecimiento en la plataforma.
- Creación de un gestor de reservas para usuarios *Premium*. Ofrecer a los establecimientos un sistema de gestión de reservas descentralizado completamente configurable para todo tipo de establecimientos. Estos usuarios tendrán una cuota mensual que abonarán a Allergic a cambio de acceso al sistema de gestión de reservas. Además se podrá asociar el sistema de reservas al perfil de un establecimiento concreto para permitir a los usuarios poder efectuar reservas online. También se podrá establecer un pago simbólico para certificar la reserva.

## Capítulo 9

# Bibliografía

Wikipedia [<http://www.wikipedia.org>] - Enciclopedia libre online.

Movable Type Scripts [<http://www.movable-type.co.uk/scripts/gis-faq-5.1.html>]  
- Great circle distance between 2 points.

W3Schools [<http://www.w3schools.com/>] - Etiquetas de maquetación, diseño y documentacion de javascript.

Mozilla Developer [<https://developer.mozilla.org/es/docs/Web/JavaScript>]  
- Documentación de javascript.

XAMPP [<https://www.apachefriends.org/>] - Servidor web básico para entorno de desarrollo local.

Google [<https://static.googleusercontent.com/media/www.google.com/es//intl/es/webmasters/doc>]  
- Pasos básicos de optimización para Google.

jQuery [<https://jquery.com/>] - Página oficial de jQuery.

MySQL [<https://dev.mysql.com/doc/refman/5.1/en/c-api.html>] - API de MySQL 5.1.

PHP [<http://php.net/>] - Página oficial de PHP. Contiene documentación de todas las funciones que proporciona el lenguaje y foros de discusión.

Pupload [<http://www.plupload.com/>] - Herramienta de carga de archivos.

Amazon AWS [<http://aws.amazon.com/es/>] - Servicio de Amazon para computación en la nube.

Node.js [<https://nodejs.org/>] - Página oficial de Node.js.

PostgreSQL [<http://www.postgresql.org.es/>] - Página oficial de PostgreSQL.

Bootstrap [<http://getbootstrap.com/>] - Página oficial de Bootstrap con documentación y ejemplo de aplicación.

PHP the right way [<http://www.phptherightway.com/>] - Manual de buenas prácticas para PHP.

JS the right way [<http://jstherightway.org/>] - Manual de buenas prácticas para JS.

# Anexo I

## Despliegue de la plataforma

El despliegue inicial de esta plataforma se ha realizado en un servidor con características muy limitadas situado en el domicilio del alumno. El servidor se trata de un NAS Synology DS213j, con un procesador ARM MARVELL Armada 370 a 1.2GHz y 512MB de memoria.

La conexión actual a internet es una línea ADSL con dirección IP fija limitada a un ancho de banda máximo de 10 Mb de bajada y 2Mb de subida. Al tratarse de un servicio que sirve información, y, debido a la velocidad de subida es muy escasa, es posible que la carga de la plataforma se demore en el tiempo. Aún y todo la carga de la plataforma no debería tardar más de 5 segundos, ya que, pese a las limitaciones técnicas, no existe una carga de trabajo suficiente para saturar el sistema.

No se ha adquirido ningún nombre de dominio para la plataforma por lo que, para poder acceder la plataforma, es necesario añadir el siguiente registro en el en el fichero hosts del sistema:

80.33.245.28    www.allergic.com
----------------------------------

## Modificar el fichero hosts en Windows

Para modificar el ficheros hosts en Windows, es necesario abrir el Bloc de Notas o algún otro editor de texto plano con privilegios de administrador. Una vez abierto, necesitamos abrir el fichero hosts que normalmente se encuentra en la carpeta `C:\Windows\System32\drivers\etc`. Es posible que, dependiendo de la intalación del sistema, la unidad varíe. Una vez abierto, añadimos la entrada al final del fichero y, tras guardar el fichero, la plataforma será accesible.

## Modificar el fichero hosts en sistema UNIX

El fichero hosts en los sistemas UNIX más comunes se encuentra en `/etc/hosts`. Para editarlo es necesario abirlo con permisos de root y, al igual que en Windows, vasta con añadir la entrada al final del fichero y guardar para que la plataforma sea visible.

## **Modificar el fichero hosts en OS X**

El fichero hosts en OS X se encuentra ubicado en `/private/etc/hosts`. Al igual que en UNIX, para editarlo, es necesario abrirlo con permisos de root y basta con añadir la entrada al final del fichero y guardar para que la plataforma sea visible.