

# Rurocio: Informe final

---



## Grupo 17

BARRIO CONDE, JORGE

GARCÍA DIÉGUEZ, JUAN

GONZÁLEZ CALDERÓN, LUCAS MATÍAS

GONZÁLEZ COSGAYA, DAVID

El porcentaje de participación de los miembros del grupo ha sido  
del 25 % cada uno.

Lunes 25 de mayo de 2020

# Índice

<b>1. Presentación del equipo</b>	<b>1</b>
<b>2. Introducción: descripción del sistema web</b>	<b>2</b>
2.1. Problema . . . . .	2
2.2. Idea principal . . . . .	2
2.3. Alternativas existentes . . . . .	3
<b>3. Usuarios objetivo</b>	<b>4</b>
3.1. Descripción de grupos de usuarios . . . . .	4
3.2. Usuario cliente . . . . .	4
3.3. Usuario anfitrión . . . . .	4
<b>4. Especificación de requisitos</b>	<b>5</b>
4.1. Requisitos no funcionales . . . . .	5
4.2. Diccionario de datos . . . . .	5
<b>5. Modelo del dominio</b>	<b>8</b>
<b>6. Mapa del sitio web</b>	<b>9</b>
<b>7. Estructura de la Base de Datos</b>	<b>10</b>
7.1. Modelo del dominio y tipos de datos . . . . .	10
7.2. Esquema Relacional . . . . .	10
<b>8. Descripción de la funcionalidad implementada y justificación de la elección</b>	<b>12</b>
<b>9. Conclusiones</b>	<b>14</b>

## 1. Presentación del equipo

El equipo de desarrollo ha estado formado por cuatro alumnos del grado de Ingeniería Informática de la UVa:

- David González Cosgaya
- Jorge Barrio Conde
- Juan García Diéguez
- Lucas Matías González Calderón

Para el desarrollo ha habido un reparto de trabajo equitativo.

## 2. Introducción: descripción del sistema web

### 2.1. Problema

Un problema recurrente en la sociedad es descubrir un plan o un evento ya pasado, al cual te hubiera gustado asistir pero no sabías de su existencia hasta después de su realización. Este problema no afecta solo a los posibles asistentes, si no también a las instituciones, organizaciones o establecimientos encargados de su planificación, puesto que no reciben la visibilidad y el número de participantes esperado. Esto ocurre muchas veces en los pueblos, gran parte por el éxodo rural y por la baja visibilidad que tienen estos en la sociedad actual, provocando que los pueblos se aíslen de las ciudades e incluso del resto de pueblo.

### 2.2. Idea principal

La idea principal del proyecto consiste en un gestor de eventos y servicios de ocio popular, focalizado principalmente en entornos rurales, que son propuestos y organizados por instituciones, organizaciones o establecimientos donde cualquier persona pueda consultar que eventos tiene cerca, buscar eventos concretos o crear eventos en el caso de ser ofertante. De esta manera, se proporcionará un sitio web donde todos los pueblos puedan mostrar sus acontecimientos más próximos, de forma que favorezca las visitas a estos pueblos, en beneficio del propio municipio y de todos sus establecimientos, como pueden ser bares, tiendas, etc. La plataforma es por tanto, un lugar donde promocionar actividades a todos los potenciales clientes. Estos dispondrán de una gran colección de eventos de todo tipo (gastronómicos, musicales, deportivos, infantiles, culturales y festivos) localizados en entornos especiales lejos del bullicio de las grandes ciudades, pudiendo así, ofrecer a todas las personas que lo deseen, planes mas allá del continuo estrés que caracteriza a las ciudades.

El nombre escogido para nuestro servicio web es “Rurocio”



Figura 1: Logotipo del servicio web

### 2.3. Alternativas existentes

Algunos servicios web que abarcan parte de este problema semejantes a nuestra idea serían:

- **Angage** : se trata de una aplicación móvil para reuniones y eventos que cuenta con sistema de votación, audio, web y vídeo-conferencias.  
Lo que nos diferenciaría de esta web es el soporte desde el propio navegador, sin necesidad de instalarse una aplicación móvil y además, este servicio se especializa en entorno de trabajo, que es muy distinto de nuestro entorno de ocio y tiempo libre.  
Además esta aplicación tiene un enfoque más empresarial, a diferencia del nuestro que esta centrada en el ocio y en la expansión turística de los pueblos. En nuestra opinión, esta diferencia significa un gran cambio. En la empresa se centran en una gestión de horarios y actividades de los empleados y no en la promoción de centros de interés.
- **Rural Events** : es una aplicación web que oferta eventos en el ámbito rural para realizar reuniones, convenciones, team building, presentaciones de producto, formaciones, celebraciones, incentivos en masías, hoteles rurales o pequeñas bodegas. Todo ello con el fin de fomentar el compañerismo y el trabajo en equipo.  
El enfoque de esta web es totalmente diferente al nuestro ya que nuestra aplicación busca la promoción de eventos para el ocio y no enfocados en el mundo laboral, para mejorar el trabajo en equipo y el compañerismo.
- **Atrapalo** : es una web de multitud de eventos que, aparte de otras muchas funcionalidades, permite buscar planes de ocio para un lugar específico.  
La diferencia más notoria entre nuestro servicio y el de *atrapalo* es que el nuestro se basa únicamente en eventos creados en zonas rurales con el fin de favorecer la visibilidad y por tanto, la economía de estas zonas. Además, los usuarios cliente de nuestro servicio pueden guardar sus gustos en la web, lo que les permitirá recibir información cuando algún nuevo evento relacionado con sus gustos se haya creado.

## 3. Usuarios objetivo

### 3.1. Descripción de grupos de usuarios

Nuestro servicio va a ir destinado a dos grupos principales de usuarios: *usuario cliente* y *usuario anfitrión*.

### 3.2. Usuario cliente

A este grupo pertenecerán los usuarios destinados a consumir el servicio de los eventos. Esto conlleva la búsqueda de eventos en función de unos filtros, aplicables a gusto de este, consulta de información de los eventos buscados y posterior reserva. El cliente puede visualizar sus reservas y editarlas si lo desea. Una vez hayan asistido, pueden valorar el servicio recibido, lo cual repercutirá en la valoración media de los servicios prestados por el usuario anfitrión correspondiente.

Este tipo de usuario encaja con gran parte de la población que busca encontrar planes de ocio alternativos a los que se ofertan en grandes ciudades, permitiéndoles desconectar por un poco de tiempo de las estresantes vidas que esa gran parte de la población sufre.

### 3.3. Usuario anfitrión

Este grupo de usuarios estará formado por instituciones (ayuntamientos, asociaciones...) o empresas que, tras haber sido verificadas, pueden publicar eventos y servicios de ocio indicando un nombre para el evento, fecha de inicio y de fin, descripción, tipo, dirección del lugar en el que se va a llevar a cabo el acontecimiento, una imagen y aforo máximo si existiese. Este tipo de usuario, al igual que el usuario cliente, puede buscar otros eventos pero no podría hacer ninguna reserva. Como añadido, el anfitrión puede visualizar la lista de sus eventos y editarlos o cancelarlos en el caso de que no haya pasado la fecha de inicio o eliminar eventos antiguos de su lista.

Son organizaciones que necesitan alguna manera de publicitarse y atraer clientes ya que, debido al incremento de la despoblación de los pueblos y zonas rurales, sus ingresos se han visto mermados durante las últimas décadas. El hecho de aparecer reflejado en una web con una gran cantidad de tráfico favorecerá en gran medida la repercusión de los eventos realizados en estos pueblos y por tanto, aumentará el volumen de visitantes a los mismos.

Ante la imposibilidad de crear una infraestructura para la administración continua de nuevos usuarios anfitriones y sus respectivas verificaciones, para este proyecto se asume que todos los usuarios anfitriones reciben una verificación sin comprobación en su registro.

## 4. Especificación de requisitos

Nº	Requisito
RF01	El sistema permitirá al Usuario registrarse por tipo de Usuario en la aplicación
RF02	El sistema permitirá al Usuario identificarse en la aplicación
RF03	El sistema permitirá al Usuario ver su información personal
RF04	El sistema permitirá al Usuario cerrar su sesión
RF05	El sistema permitirá al usuario 'Cliente' buscar eventos
RF06	El sistema permitirá al usuario 'Cliente' consultar detalles de un evento
RF07	El sistema permitirá al usuario 'Cliente' reservar un evento
RF08	El sistema permitirá al usuario 'Cliente' consultar las reservas realizadas
RF09	El sistema permitirá al usuario 'Cliente' cancelar una reserva de un evento
RF10	El sistema permitirá al usuario 'Cliente' valorar un evento reservado cuya fecha de fin es anterior a la actual
RF11	El sistema permitirá al usuario 'Anfitrión' dar de alta un evento
RF12	El sistema permitirá al usuario 'Anfitrión' cancelar un evento cuya fecha de inicio es posterior a la actual
RF13	El sistema permitirá al usuario 'Anfitrión' modificar los detalles de un evento cuya fecha de inicio es posterior a la actual
RF14	El sistema permitirá al usuario 'Anfitrión' borrar de su lista de eventos un evento cuya fecha de fin sea anterior a la actual
RF15	El sistema permitirá al usuario 'Anfitrión' consultar la lista de reservas de un evento propio

**Cuadro 1:** Requisitos funcionales de la aplicación

### 4.1. Requisitos no funcionales

Nº	Requisito
RNF01	La aplicación debe ser intuitiva y fácil de manipular sin conocimientos previos.
RNF02	La aplicación debe ser compatible con los navegadores principales.
RNF03	La aplicación utilizará la API de Google Maps.

**Cuadro 2:** Requisitos no funcionales de la aplicación

### 4.2. Diccionario de datos

Usuario	
Descripción	Persona que utiliza la aplicación web.
Tipos	<ul style="list-style-type: none"> <li>■ Cliente: Puede apuntarse a eventos, no puede crearlos.</li> <li>■ Anfitrión: Puede crear eventos, no puede apuntarse a ellos.</li> </ul>

Información personal	
Descripción	Datos relevantes del usuario de la aplicación.
Parámetros	<ul style="list-style-type: none"> <li>■ Login: Nombre con el que el usuario se identifica en la aplicación.</li> <li>■ Nombre: Nombre que verán el resto de usuarios de la aplicación.</li> <li>■ Contraseña: Cadena de caracteres alfanuméricos para identificar al usuario.</li> <li>■ Correo electrónico: Email asociado a la cuenta.</li> <li>■ NIF: Documento de identificación de un usuario Cliente.</li> <li>■ CIF: Documento de identificación de un usuario Anfitrión.</li> <li>■ Fecha de nacimiento: Necesario para eventos específicos.</li> </ul>

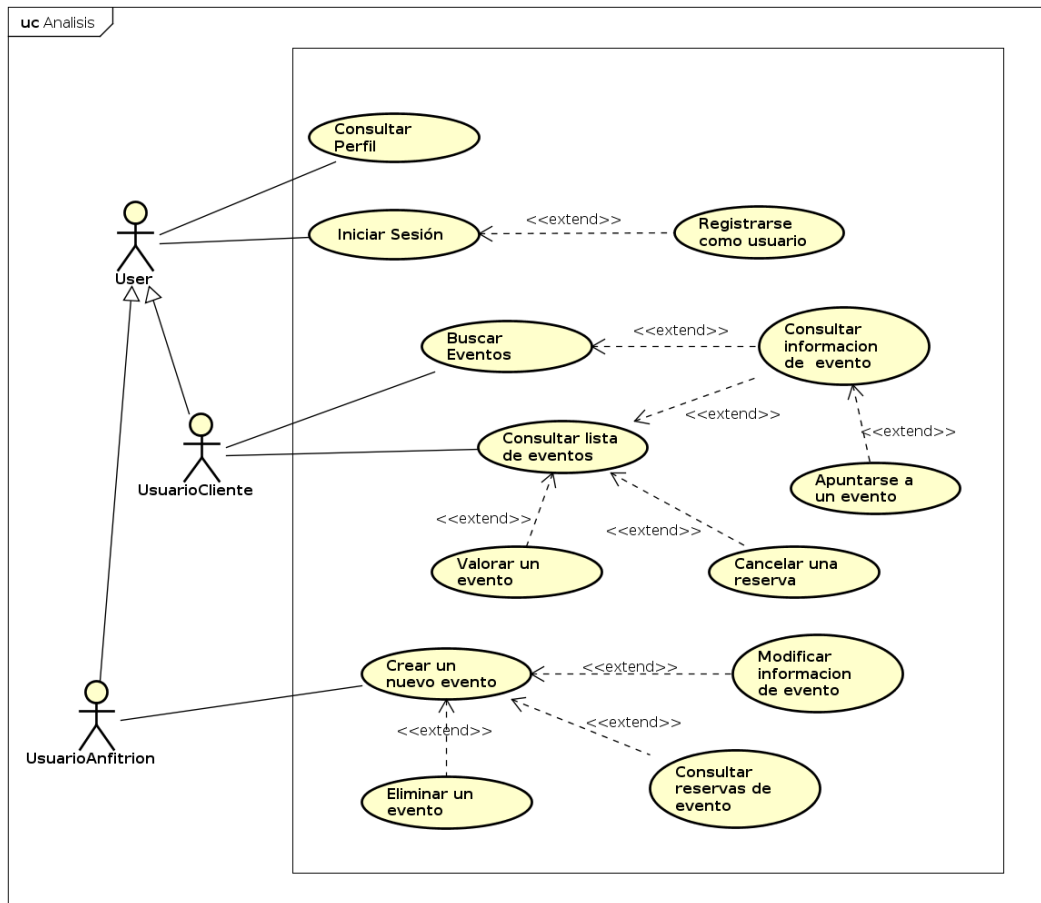
Evento	
Descripción	Actividad a la que un usuario se puede apuntar.
Parámetros	<ul style="list-style-type: none"> <li>■ Nombre: Nombre del evento.</li> <li>■ Descripción: Información detallada sobre el evento.</li> <li>■ Fecha inicio: Fecha a la que se inicia el evento.</li> <li>■ Fecha fin: Fecha a la que finaliza el evento.</li> <li>■ Tipo: Categoría del evento. Puede ser: astronómico, musical, deportivo, infantil, cultural o festivo.</li> <li>■ Localización: Ubicación del evento</li> <li>■ Imagen (opcional): Adjuntos visuales.</li> <li>■ Aforo (opcional): Número máximo de participantes.</li> <li>■ Id: Identificador único asociado a cada evento.</li> </ul>

Reserva	
Descripción	Participación de un usuario a un evento.
Parámetros	<ul style="list-style-type: none"> <li>■ Valoración: Puntuación en el rango de valores 1 a 5 que un Cliente asigna a una reserva si el evento al que referencia ya ha acontecido.</li> </ul>



A continuación se mostrará el diagrama de los Casos de uso que se han descrito en los requisitos del sistema. Se puede apreciar claramente la herencia de los dos tipos de usuarios, Cliente y Anfitrión. Un Usuario tiene casos de uso globales como son el inicio de sesión y consultar su perfil (Ver Figura 2).



**Figura 2:** Casos de uso

## 5. Modelo del dominio

El modelo de dominio obtenido tras el proceso de análisis se puede observar a continuación. Está formado por las clases Usuario, de la cual se especializan los dos tipos de usuarios que va a tener el sistema, Cliente y Anfitrión, y cuya única diferencia consiste en la fecha de nacimiento que se almacena en el Cliente. El Evento se relaciona a través de una clase asociativa (Reserva) con la clase Cliente y esta clase Reserva es necesaria debido a la relación muchos muchos que se genera. Además, aparece relacionado el Evento con su Anfitrión (Ver en Figura 3).

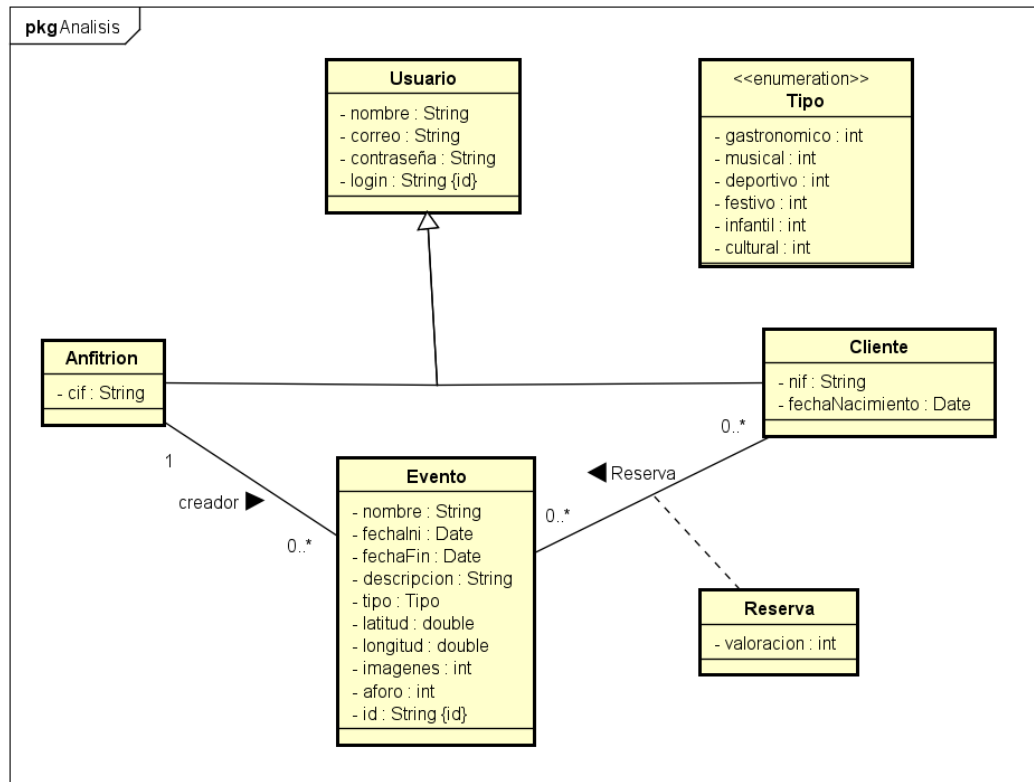


Figura 3: Modelo de dominio

## 6. Mapa del sitio web

En nuestro servicio web contamos con un mapa web bastante complejo. Gracias a nuestro *header* es posible la comunicación entre casi cualquier pagina. La posibilidad de redirección entre paginas varía en función del estado de la sesión. Partiendo de una sesión sin login, desde cualquier vista se puede acceder a la pagina de *Inicio*, *Sobre nosotros* e *Iniciar sesión*. Si nos encontramos con una sesión iniciada por un usuario, sin importar el tipo, se nos mostrará la posibilidad de redirección a dos nuevas ventanas, *Mis Eventos* y *Mi perfil*. Además, en el caso de ser una sesión de un usuario de tipo *Anfitrión*, tendremos la nueva opción de *Nuevo evento*.

En la siguiente figura se muestran el resto de relaciones con mas detalle:

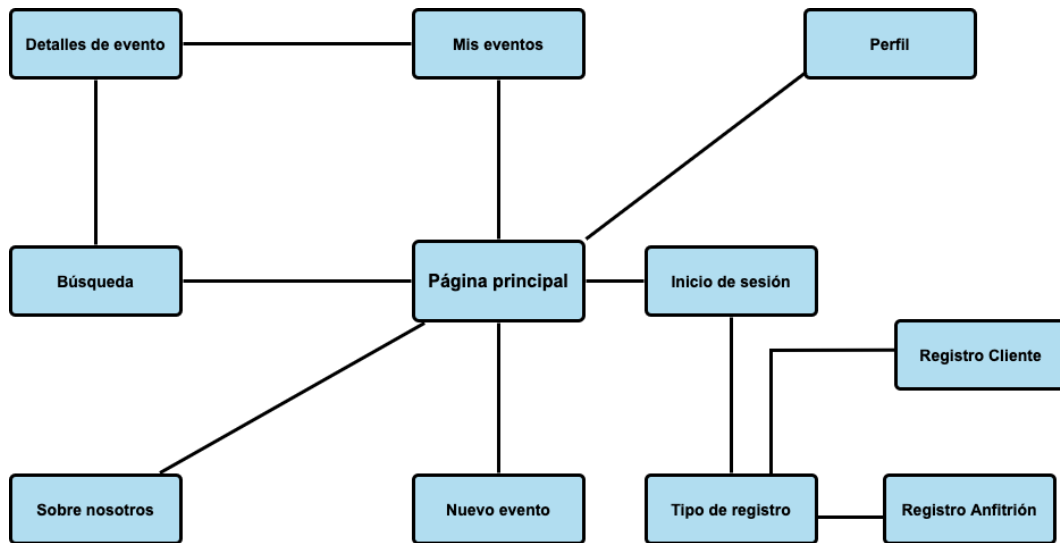


Figura 4: Mapa del sitio web

## 7. Estructura de la Base de Datos

La base de datos se ha decidido implementar en MySQL y siguiendo la estructura premilimar de la entrega anterior pero con algunas modificación debidas a necesidades y limitaciones que se han detectado en esta fase.

### 7.1. Modelo del dominio y tipos de datos

La estructura de la base de datos ha seguido el modelo de dominio propuesto en la entrega anterior (Ver en Figura 3).

Podemos resaltar que para almacenar las imágenes en el modelo se realiza con el tipo de dato *Part* de la biblioteca *javax.servlet.http*, mientras que en la base de datos se almacenan mediante *LONGBLOB*, tipo de *BLOB* (*Binary Large Object*). Optamos por un *LONGBLOB* porque las imágenes que usamos no caben en un *BLOB* normal.

Es necesario aclarar que el script SQL carga las imágenes en un directorio específico por medidas de seguridad, siendo necesario mover las imágenes del proyecto a tal directorio. Este directorio cambia según el sistema operativo instalado por lo que se han generado tres scripts iguales, variando el directorio de lectura de las imágenes. En el caso de *Linux*, de un directorio nombrado como */rurocio* que se encuentra en directorio seguro de MySQL, */var/lib/mysql-files/*.

Debido a los múltiples problemas ocasionados por la inserción de imágenes, hemos generado scripts para cada uno de los S.O. más comunes: Linux, Mac y Windows, siendo este último el mas problemático de todos. En cada script se encuentra detalladamente los pasos a seguir, y en que carpeta se deben depositar las imagenes para su insercion en la base de datos.

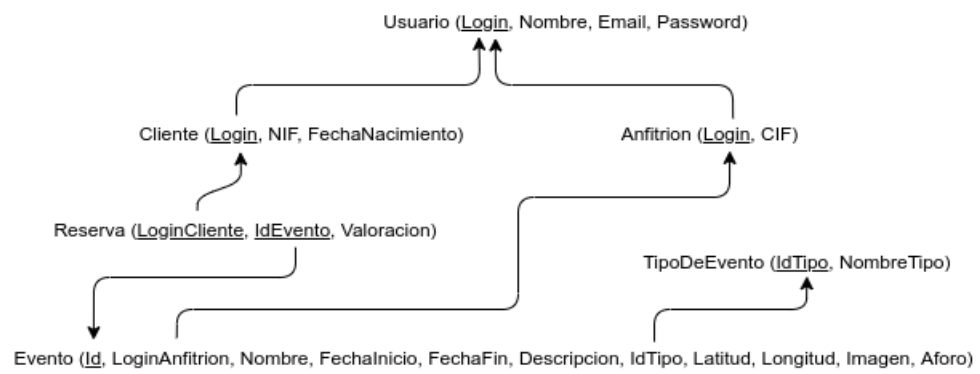
Para los identificadores de los eventos, se ha decidido automatizar su generación mediante la función de MySQL *UUID()*, que te devuelve un valor de 128 bit representado como una cadena utf8 de cinco números hexadecimales.

Con respecto a los distintos atributos de las tablas que conforman la base de datos hemos especificado que no puedan ser nulos aquellos pertenecientes a los distintos usuarios por ser estos necesarios para identificar y diferenciar a los usuarios de la aplicación, además campos tales como login, email, nif y cif deben ser únicos para evitar que 2 usuarios de distinto tipo puedan generar con la misma información 2 tipos de cuentas. Los eventos deben de disponer de todos sus campos principales, pudiendo obviar el aforo por no ser tan significativo para algunos eventos, el id no precisa de ser único debido a que se genera mediante *UUID* y esto ya lo garantiza. El resto de los campos del evento no necesitan ser únicos ya todos ellos se pueden repetir en otros eventos. Y con respecto a la reserva, la tabla asociativa ente Evento y Cliente, son obligatorios tanto el login del cliente como el id del evento por ser claves primarias del mismo, la valoración se podría añadir posteriormente pero no es imprescindible.

### 7.2. Esquema Relacional

Respecto al esquema relacional, gracias a los cambios realizados en el anterior prototipo, no ha sido necesario añadir o cambiar nada. Esta nueva versión nos ha permitido cumplir con todos los requisitos.

En el esquema relacional se pueden comprobar las claves primarias de cada tabla siendo estas las que se encuentran subrayadas y las claves foráneas mediante las flechas que apuntan a una clave primaria.



**Figura 5:** Esquema Relacional

## 8. Descripción de la funcionalidad implementada y justificación de la elección

Para la implementación de las funcionalidades han prevalecido las mas importantes. Aún así la implementación realizada deja una larga lista de implementaciones. A continuación se van a explicar los motivos de su importancia y una breve descripción de cada una, intentando ordenarlas de mayor a menor importancia:

1. Búsqueda de eventos: como ya se ha explicado, el motor principal de *Rurocio* no es otro que la búsqueda de eventos en función de unos filtros, aplicables a gusto del usuario. Para su implementación se ha tenido en cuenta la gran importancia que supone una búsqueda adaptada a cada tipo de petición de los usuario, por lo que el filtrado de eventos únicamente se hará con los campos de la búsqueda que el usuario desee completar. Por ultimo, se contempló la situación en la que una búsqueda no obtiene ningún resultados, en ese caso, se le recomendarán al usuario una serie de eventos, principalmente, en función de su localización.
2. Inicio de sesión: nuestro servicio está enfocado a la gestión de eventos, lo que supone que es imprescindible el inicio de sesión de los usuarios para realizar sus acciones pertinentes. Al acceder a la pagina inicial, el usuario accede al apartado de *Inicio de sesión* donde introducirá sus credenciales y, si son correctas, se le redirigirá a la pagina principal, pero ya mostrándole las funcionalidades en el *header* correspondientes a su rol de usuario.
3. Dar de alta evento: para que existan nuevos eventos, necesitamos que usuarios anfitriones den de alta sus próximos eventos. Para ello se ha creado un formulario en el que hay que introducir los datos de evento: nombre, fecha de inicio y de fin, descripción, tipo, una imagen (voluntario) y un aforo (voluntario). A esto se suma una campo para introducir la dirección. Este campo, suministrado por la API de Google Maps, permite comprobar en tiempo real la dirección que se está introduciendo, recomendando en función de la entrada, la dirección exacta. Si alguno de los campos obligatorios está nulo, se le avisará al anfitrión que debe completar el formulario antes de enviarlo.
4. Consultar información de un evento: tras la búsqueda de eventos, el usuario debe tener la posibilidad de visualizar la información detallada del evento. Para ello, debe *clickar* en el nombre del evento. En la descripción detallada se mostrarán todos los datos asociados al evento, junto a un mapa interactivo mostrando la dirección de este.
5. Reservar evento: Una vez que el cliente ha consultado la información del evento de interés, tiene la posibilidad de apuntarse y realizar una reserva, apretando el botón de reservar, entonces, este evento aparecerá en su lista de eventos. Solo podrá acceder a esta funcionalidad un usuario que ha iniciado sesión como Cliente.
6. Consultar mis reservas: Bien sea un usuario Cliente o Anfitrión, ambos tienen eventos. En el caso de un Cliente, se mostrarán los eventos a los que ha realizado una reserva. Si se trata de un evento ya acontecido, se mostrará una opción de votación del evento, y en un evento futuro, la posibilidad de cancelar la reserva. Por otro lado, a un Anfitrión se le mostrarán los eventos que ha creado y, al igual que antes, si se trata de un evento antiguo podrá eliminarlo de su lista, y si es un evento futuro, podrá cancelarlo o editar su información.
7. Cerrar sesión: el cerrar sesión es una necesidad básica para cualquier servicio que gestione usuarios, pero en nuestro caso aún mas. En el caso de que una persona esté al cargo de la gestión de eventos de una organización y desee reservar un evento, necesita tener la posibilidad de cerrar su sesión de anfitrión e iniciar su sesión personal.

8. Consultar reservas de un evento propio: en el caso de un Anfitrión, el numero de asistentes y sus datos es una información muy relevante pues esto puede determinar la preparación final del evento. A su vez, un anfitrión puede necesitar la lista de *emails* para mandar alguna información extra sobre el evento a sus futuros asistentes.
9. Previsualización de eventos cercanos: la posibilidad de que un usuario reciba una primera visualización en la pantalla principal de un *top 3* de los eventos mas cercanos obtendría una buena aceptación, análogo a la ventana de inicio del conocido *Youtube*, donde recibes unas recomendaciones en función de unos parámetros para que sea mas sencilla la búsqueda de nuevos servicios que consumir. Para ello se ha representado en un mapa las ubicaciones de los tres eventos recomendados (el mapa está centrado en el baricentro del triangulo formado por las tres ubicaciones) y justo debajo, se encuentra una tabla con una breve descripción de cada uno, pudiendo acceder a sus detalles si se desea, *clickando* en el nombre.
10. Valoración de eventos completados: una vez pasado un evento al cual un Cliente haya asistido, se habilitará la posibilidad de valorarlo. Esta opción esta disponible en la sección "Mis Eventos"del Cliente. Se ha implementado un sistema de votación por estrellas, de 1 a 5, donde el usuario escoge el número de estrellas que desea otorgarle a la actividad. Una vez seleccionado, se refresca la página mostrando la selección del usuario y sin permitir modificarlo. Estas votaciones servirán en el motor de búsqueda de actividades a los Anfitriones para destacar frente al resto.

Todas las funcionalidades cuentan con una adaptación al estado de la sesión, mostrando u ocultando componentes, de forma que un usuario, en función de su rol, si han iniciado sesión o incluso de la fecha de acceso, puede tener diferentes vistas, restringiendo así, funcionalidades que no le corresponden.

Por ultimo, aclarar que para gestionar de localizaciones y los mapas, se ha utilizado la API de Google Maps.

## 9. Conclusiones

Este proyecto, aunque sencillo a primera vista, ha resultado ser más ambicioso de lo que al inicio se había planteado. La necesidad de implementar una API externa como la de Google Maps para poder localizar eventos o buscarlos, ha supuesto un reto, pero ha permitido una aproximación a la solución que se buscaba mucho mas grande que si se hubiera optado por una vía más sencilla, como por ejemplo, permitir al Anfitrión escribir la ubicación. Además el uso de herramientas como Bootstrap han supuesto una ayuda a la hora de desarrollar, al igual que Mockup para diseñar, agilizando el proceso. Sacamos en claro que el proyecto podría continuarse hasta ser una página bien pulida, con mas implementaciones incluso aplicación web, llevando el producto más allá de lo que se plantea en esta asignatura. El uso de las tecnologías como Servlets y JSPs nos han ayudado a comprender como funciona una página web, sus disparadores, acciones e interacciones, la conexión entre Front y Back y la necesidad de blindar la aplicación tanto en funcionalidad como en seguridad.