



UNIVERSIDADE  
DE VIGO

ESCOLA SUPERIOR DE ENXEÑARÍA INFORMÁTICA

Memoria do Traballo de Fin de Máster que presenta

**D. Martín Gil Blanco**

para a obtención do Título de Máster en Enxeñaría Informática

**EventGO: Aplicación de gestión de eventos**



Febreiro, 2024

**Traballo de Fin de Máster N°:** MEI 23/24-6

**Titor/a:** Francisco José Ribadas Pena

**Área de coñecemento:** Ciencias de la computación e IA

**Departamento:** Informática



*A mi familia y amigos por ser el cimiento  
sólido que me sostuvo en cada desafío.*

*A mi tutor de proyecto por entender en todo  
momento mi situación y lo difícil que ha sido  
para mí compaginar los estudios con el trabajo.*

*A todos los que han hecho posible que haya  
conseguido obtener dos másteres en un año y medio.*

*Con cariño y gratitud.*

*Martín*

*La utopía está en el horizonte. Camino  
dos pasos, ella se aleja dos pasos y el  
horizonte se corre diez pasos más allá.*

*¿Entonces para qué sirve la utopía?*

*Para eso, sirve para caminar.*

*Eduardo Galeano*



## Índice de contenido

1. Introducción .....	1
2. Objetivos .....	3
3. Resumen de la solución propuesta .....	4
3.1 Metodología .....	4
4. Planificación y seguimiento .....	5
4.1 Planificación .....	5
4.2 Puntos críticos .....	6
4.3 Seguimiento .....	7
4.4 Desviación .....	8
5. Arquitectura .....	9
5. Tecnologías y herramientas .....	10
6.1 Tecnologías .....	10
6.2 Herramientas de desarrollo .....	11
7. Análisis y especificación de requisitos .....	12
7.1 Requisitos funcionales .....	12
7.1.1 Gestión de usuarios (RF1) .....	12
7.1.2 Gestión de categorías (RF2) .....	13
7.1.3 Gestión de eventos (RF3) .....	13
7.1.4 Gestión de solicitudes (RF4) .....	13
7.1.5 Gestión de suscripciones (RF5) .....	14
7.1.6 Gestión de amistades (RF6) .....	14
7.1.7 Gestión de noticias (RF7) .....	14
7.1.8 Gestión de comentarios (RF8) .....	14
7.2 Requisitos no funcionales .....	15
7.3 Casos de uso .....	15
8. Diseño de Software .....	17
8.1 Diseño estático .....	17
8.1.1 Diagramas de clases de Backend .....	17
8.1.2 Diagramas de clases de Frontend .....	18
8.2 Diseño dinámico .....	19
9. Gestión de información y datos .....	20
9.1 Modelo entidad relación MERE .....	20

9.2 Paso a tablas.....	21
10. Pruebas .....	22
11. Manual de usuario.....	23
11.1 Manual de instalación .....	23
11.2 Manual de uso .....	23
11.2.1 Vista de inicio .....	23
11.2.2 Gestión de usuarios .....	28
11.2.3 Gestión de categorías.....	33
11.2.4 Gestión de eventos.....	33
11.2.4 Gestión de solicitudes y suscripciones .....	35
11.2.5 Gestión de amistades y noticias.....	39
12. Principales aportaciones .....	41
13. Conclusiones.....	42
13.1 Conclusiones técnicas .....	42
13.2 Conclusiones personales .....	42
14. Vías de trabajo futuro .....	43
15. Referencias .....	44
16. Anexo.....	46

## Índice de ilustraciones

Ilustración 1: Diagrama de Gantt de la planificación Inicial.....	6
Ilustración 2: Diagrama de Gantt de las duraciones reales.....	8
Ilustración 3: Arquitectura .....	9
Ilustración 4: Casos de uso de usuario no registrado .....	15
Ilustración 5: Casos de uso de usuario registrado .....	16
Ilustración 6: Casos de uso de gerente .....	16
Ilustración 7: Casos de uso de administrador .....	16
Ilustración 8: Diagrama de la estructura del Backend.....	17
Ilustración 9: Diagrama de clases del Backend .....	18
Ilustración 10: Diagrama de estructura del Frontend .....	18
Ilustración 11: Diagrama de secuencia para insertar una categoría en Backend.....	19
Ilustración 12: Diagrama de secuencia para buscar categorías .....	19
Ilustración 13: Modelo Entidad Relación Extendido (MERE) .....	20
Ilustración 14: Estructura física de la Base de Datos .....	21
Ilustración 15: Login .....	24
Ilustración 16: Validación de datos en cliente .....	24
Ilustración 17: Validación de acciones del servidor .....	24
Ilustración 18: Vista de usuario .....	25
Ilustración 19: Vista de gerente .....	25
Ilustración 20: Vista de administrador.....	26
Ilustración 21: Ver eventos .....	26
Ilustración 22: Registro .....	27
Ilustración 23: Recuperar contraseña .....	27
Ilustración 24: Correo de recuperación de contraseña.....	28
Ilustración 25: Gestión de usuarios .....	28
Ilustración 26: Crear usuario .....	29
Ilustración 27: Acción realizada correctamente.....	29
Ilustración 28: Buscar usuario .....	29
Ilustración 29: Resultado de la búsqueda .....	30
Ilustración 30: Editar usuario .....	30
Ilustración 31: Ver en detalle usuario .....	30
Ilustración 32: Eliminar usuario .....	31
Ilustración 33: Usuario eliminado de forma lógica .....	31

Ilustración 34: Reactivar usuario .....	31
Ilustración 35: Usuario reactivado.....	31
Ilustración 36: Subir imagen de usuario .....	32
Ilustración 37: Nueva imagen de usuario .....	32
Ilustración 38: Nueva imagen de usuario desde Mi Cuenta .....	32
Ilustración 39: Gestión de categorías .....	33
Ilustración 40: Gestión de eventos como administrador .....	33
Ilustración 41: Comentarios de un evento .....	34
Ilustración 42: Gestión de eventos como gerente.....	34
Ilustración 43: Inscripción en evento público.....	35
Ilustración 44: Correo de suscripción a evento .....	35
Ilustración 45: Vista de Mis Eventos .....	36
Ilustración 46: Suscripción a evento.....	36
Ilustración 47: Crear comentario .....	36
Ilustración 48: Comentarios de un evento 2 .....	37
Ilustración 49: Inscripción en evento privado .....	37
Ilustración 50: Solicitud a evento privado .....	38
Ilustración 51: Gestión de eventos del gerente.....	38
Ilustración 52: Aprobación de solicitud.....	38
Ilustración 53: Correo de suscripción a evento 2 .....	38
Ilustración 54: Gerentes a seguir .....	39
Ilustración 55: Gerentes seguidos.....	39
Ilustración 56: Gestión de noticias .....	40
Ilustración 57: Correo de noticia 1 .....	40
Ilustración 58: Correo de noticia 2 .....	40
Ilustración 59: Acreditación de una suscripción en un evento .....	46



## Índice de tablas

Tabla 1: Planificación del proyecto .....	5
Tabla 2: Duraciones reales.....	7

## 1. Introducción

En la actual era digital, la asistencia a eventos se ha convertido en una parte esencial de nuestra vida social. Desde conciertos y festivales hasta eventos deportivos, políticos y sociales, existe una amplia gama de actividades que brindan opciones para que las personas disfruten de su tiempo. Sin embargo, la organización y gestión eficaz de estos eventos plantean un desafío constante tanto para organizadores como para participantes. Ambos buscan una experiencia fluida, atractiva y sin contratiempos en la cual puedan elegir entre las numerosas opciones a su disposición. La popularidad de los eventos masivos ha experimentado un crecimiento notable en los últimos años, lo que ha generado la necesidad de abordar de manera eficiente la organización y gestión de los mismos.

Entendiendo esta problemática, surge la necesidad de desarrollar una solución que facilite la planificación, coordinación y seguimiento de eventos de forma integral. El presente Trabajo de Fin de Máster (TFM) se centra en la creación de una web de gestión de eventos, diseñada para satisfacer las necesidades de organizadores y participantes.

Antes de abordar en detalle las características de este trabajo, es fundamental destacar el papel de plataformas clave en la gestión de eventos, como Eventbrite ([www.eventbrite.com](http://www.eventbrite.com)) o Vivetix (<https://vivetix.com>). Estas herramientas, facilitan la organización y venta de entradas de eventos. Eventbrite, por ejemplo, se destaca por su catálogo de eventos disponibles a cualquier escala, dando a la comunidad la posibilidad de participar en una infinidad de eventos distintos.

Este trabajo tiene como objetivo presentar la aplicación *EventGO* como propuesta sólida y práctica para la gestión de eventos. La herramienta *EventGO* simplifica el proceso de organización y brinda una experiencia fluida tanto para los organizadores como para los asistentes. A través de un análisis exhaustivo de las necesidades y desafíos en la gestión de eventos, se abordan aspectos fundamentales como la funcionalidad, arquitectura de la aplicación o la experiencia de usuario. De esta forma se contribuye a la creación de una nueva comunidad de usuarios interesados en la participación en eventos, aumentando así la popularidad de los mismos.

Para el desarrollo de este proyecto se hizo en un primer lugar un análisis de los diferentes requisitos a nivel de funcionalidad que ha de satisfacer una aplicación como la que se ha presentado anteriormente. A continuación, se lleva a cabo el diseño e implementación de la base de datos donde se aloja la información de nuestra aplicación. Concluido este punto y realizado el análisis y el diseño de las interfaces principales, se lleva a cabo el desarrollo de la aplicación propiamente dicha, que presenta de una interfaz web amigable y fácil de usar para dar soporte a las principales operaciones típicas de gestión de eventos como son: dar de alta un evento, gestionar el evento, ver datos del evento, inscripción, etc.

Para implementar la plataforma de asistencia a la gestión de eventos descrita anteriormente se plantea el desarrollo de una aplicación web con dos elementos principales: un back-end responsable del almacenamiento de los datos y de la

implementación de la lógica de la aplicación desarrollado en Java, haciendo uso de las funcionalidades de Spring Framework y un front-end de tipo Single page application o SPA empleando la librería de JavaScript React. Los próximos capítulos de este TFM se pueden clasificar en la descripción del trabajo (objetivos, solución, propuesta, planificación y seguimiento), documentación técnica (arquitectura, tecnologías, análisis, diseño del software, datos, pruebas y manual de usuario) y conclusiones (principales aportaciones, conclusiones, trabajo futuro).

## 2. Objetivos

En base a lo indicado en la introducción, el objetivo principal de este Trabajo Fin de Máster es el de desarrollar una plataforma web destinada a facilitar la gestión de eventos, incrementando la participación de los usuarios en estos actos populares y generando una comunidad de usuarios interesados en participar en ellos. Con el fin de alcanzar este objetivo principal se han identificado los siguientes subobjetivos:

**Obj 1.** El sistema a implementar será una primera versión de un posterior desarrollo final más amplio, en el cual se abarcarán un mayor número de eventos y modalidades de estos.

**Obj 2.** La interfaz web de la aplicación ha de ser amigable al usuario, fácil de emplear y estar destinada para todo tipo de público llegando al mayor número de personas posibles.

**Obj 3.** Autenticación y registro de usuario con varios roles, donde cada uno de ellos tendrá unas acciones concretas en la aplicación web.

**Obj 4.** Creación y publicación de categorías de eventos, donde los usuarios gerentes podrán publicar sus eventos en función de su naturaleza para así poder agruparlos de una forma más intuitiva a los usuarios de la aplicación.

**Obj 5.** Creación y publicación de nuevos eventos por parte de los gestores de eventos, donde diferentes usuarios podrán inscribirse dentro del evento y participar en estos.

**Obj 6.** Los usuarios no registrados pueden visualizar las categorías y eventos del sistema haciendo de *EventGO* una aplicación más abierta al público.

**Obj 7.** Los gerentes del evento podrán diferenciar entre las suscripciones de los usuarios a los eventos que sean públicos, y por lo tanto de libre participación o privados, donde se debe de aceptar la solicitud del usuario para que este pueda estar suscrito al evento.

**Obj 8.** Los usuarios podrán seguir a los usuarios gerentes encargados de crear eventos dentro de *EventGO* lo que les permitirá estar informados de las novedades de sus eventos.

**Obj 9.** Los gerentes podrán publicar nuevas noticias llegándoles un correo informativo a los usuarios que han decidido seguirlos para estar más informados de sus novedades.

**Obj 10.** Los usuarios podrán generar un PDF con los datos principales del evento y propios una vez están suscritos al evento.

**Obj 11.** Publicación de comentarios y valoraciones de los usuarios sobre los eventos a los que están suscritos, pudiendo servir de información a usuarios y al gerente del evento.

**Obj 12.** Creación, visualización, modificación, búsqueda y eliminación de cualquier información dentro de la aplicación por el administrador o usuarios en función de su rol.

Como objetivos complementarios a los mencionados, se pretende contar con una arquitectura lo más modular posible que simplifique la incorporación de nuevas funcionalidades. Asimismo, se ha dado preferencia al uso de herramientas y librerías con licencias libres y se han seguido buenas prácticas recomendadas para el desarrollo software con las tecnologías y herramientas empleadas.

### 3. Resumen de la solución propuesta

Para lograr todos los objetivos anteriormente expuestos, se pretende desarrollar una herramienta web cliente-servidor que dé soporte a la gestión de eventos. En el momento del desarrollo del proyecto, existen dos partes diferenciadas, Backend y Frontend.

- El Backend es la parte de nuestra aplicación encargada de procesar y gestionar la lógica de los datos y del funcionamiento de nuestra web. Es responsable del manejo de la lógica de negocio, la interacción con base de datos, autenticación de usuarios, enrutamiento de solicitudes, entre otras tareas relacionadas con la funcionalidad y la lógica de la aplicación.
- El Frontend es la parte de la aplicación web con la que el usuario final interactúa directamente. Incluye la interfaz de usuario, el diseño visual, la interacción y la experiencia del usuario.

Comúnmente se suele decir que el Frontend es el lado cliente y el Backend es el lado servidor. Para poder desplegar el Backend se hace uso de un servidor Apache Tomcat embebido, respaldado por el gestor de base de datos MySQL. Todo este proceso se implementa gracias al uso de SpringBoot, que es el encargado de la creación, conexión e interacción con la base de datos y desarrollo de la lógica de nuestra aplicación web, exponiendo las funcionalidades del Backend usando un API REST (representational state transfer o REST). El Frontend por su parte, utiliza React que es una biblioteca JavaScript ampliamente utilizada para crear interfaces de usuario interactivas y reutilizables..

#### 3.1 Metodología

En este proyecto TFM se ha tomado la decisión de emplear desarrollo en cascada (Camacho, 2020), ya que desde un inicio se tienen razonablemente bien definidos los requisitos que se pretende alcanzar.

El procedimiento que seguimos se divide en diferentes fases desarrolladas unas detrás de otra. Concluida una, se lleva a cabo un análisis y su correspondiente comprobación antes de seguir con la siguiente, detectando los errores existentes y solucionándolos antes de continuar. Las fases se ejecutarán de forma secuencial y están establecidas en el modelo en cascada como las cinco siguientes:

- **Análisis:** Etapa de preparación del proyecto en la que se determinan los requisitos, los objetivos y las necesidades que se deben de cumplir. A continuación, se reúne toda la información requerida y el análisis de los objetivos.
- **Diseño:** Etapa en la que se diseña de la arquitectura a emplear y plan de diseño, centrándose en componentes principales, las interfaces, etc. Obteniendo un anteproyecto del trabajo que vamos a realizar.
- **Implementación:** Etapa en la que se programa el software de la aplicación, se ponen en práctica los elementos de diseño y requisitos de las etapas anteriores.
- **Verificación:** Etapa de prueba y ejecución del código, en la que se verifica su correcto funcionamiento. Aquí comparamos los objetivos iniciales y su cumplimiento.
- **Mantenimiento:** Etapa de análisis de resultados, donde se realizan las correcciones necesarias para el lanzamiento. Compuesta de la entrega y mejora del software.

## 4. Planificación y seguimiento

En el siguiente capítulo se va exponer el desarrollo de la planificación llevada a cabo para la realización del proyecto, puntos críticos, seguimiento realizado y desviaciones que se han producido a lo largo de la realización del TFM.

### 4.1 Planificación

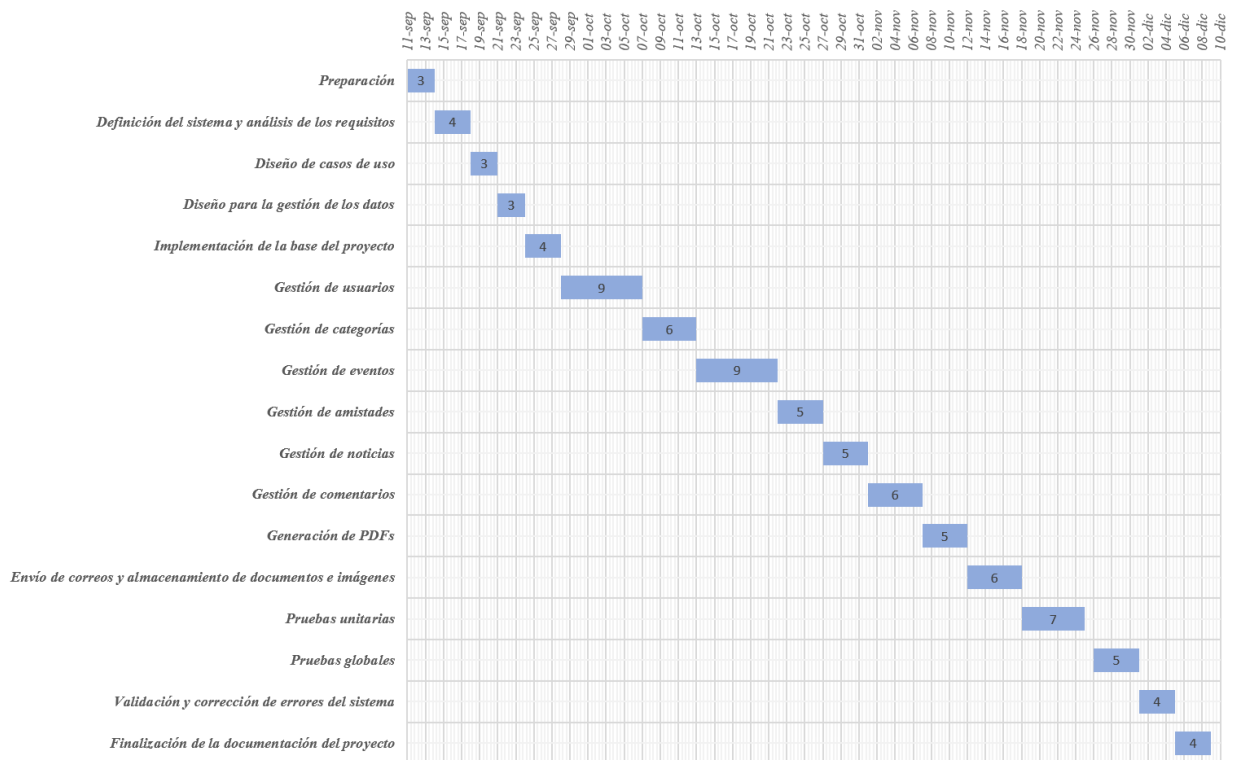
El proyecto presenta una estimación de 225 horas, comenzando su desarrollo en el mes de septiembre y concluyendo en el mes de diciembre. El objetivo es disponer de margen de tiempo que permita la resolución de imprevistos que pudieran retrasar el proyecto. Para el cumplimiento de los plazos se espera realizar un aproximado de 20 horas semanales. Por otra parte, el desarrollo de las funcionalidades a nivel de código se realiza de forma paralela a nivel de Frontend y Backend. La *Tabla 1. Planificación* se ha realizado previamente a la implementación para así tener una especificación de las tareas a realizar y tiempos que cumplir.

Fases	Tiempo	Inicio	Fin
<b>Inicio</b>	<b>10h</b>	<b>11 septiembre</b>	<b>13 septiembre</b>
Preparación	10h	11 septiembre	14 septiembre
<b>Análisis</b>	<b>15h</b>	<b>14 septiembre</b>	<b>17 septiembre</b>
Definición del sistema y análisis de los requisitos	15h	14 septiembre	17 septiembre
<b>Diseño</b>	<b>25h</b>	<b>18 septiembre</b>	<b>27 septiembre</b>
Diseño de casos de uso	10h	18 septiembre	20 septiembre
Diseño para la gestión de los datos	8h	21 septiembre	23 septiembre
Implementación de la base del proyecto	7h	24 septiembre	27 septiembre
<b>Implementación</b>	<b>125h</b>	<b>28 septiembre</b>	<b>17 noviembre</b>
Gestión de usuarios	21h	28 septiembre	6 octubre
Gestión de categorías	14h	7 octubre	12 octubre
Gestión de eventos	23h	13 octubre	21 octubre
Gestión de amistades	13h	22 octubre	26 octubre
Gestión de noticias	12h	27 octubre	31 octubre
Gestión de comentarios	15h	1 noviembre	6 noviembre
Generación de PDFs	13h	7 noviembre	11 noviembre
Envío de correos y almacenamiento de documentos e imágenes	14h	12 noviembre	17 noviembre
<b>Verificación</b>	<b>30h</b>	<b>18 noviembre</b>	<b>30 noviembre</b>
Pruebas unitarias	18h	18 noviembre	25 noviembre
Pruebas globales	12h	26 noviembre	30 noviembre
<b>Mantenimiento</b>	<b>20h</b>	<b>1 diciembre</b>	<b>8 diciembre</b>
Validación y corrección de errores	10h	1 diciembre	4 diciembre
Finalización de la documentación	10h	5 diciembre	8 diciembre
<b>Total</b>	<b>225h</b>	<b>11 septiembre</b>	<b>8 diciembre</b>

Tabla 1: Planificación del proyecto

La *Ilustración 1. Diagrama de Gantt planificado*, podemos ver de una forma gráfica la información recogida en la tabla anterior mediante un diagrama de Gantt. El objetivo de este tipo de diagramas es representar las diferentes fases que presenta el modelo en cascada. La documentación que acompaña al proyecto se ha realizado de forma paralela al desarrollo de código.

## Planificación Inicial



*Ilustración 1: Diagrama de Gantt de la planificación Inicial*

### 4.2 Puntos críticos

El proyecto que se ha desarrollado ha presentado tres puntos críticos a destacar.

Primeramente, la implementación de la gestión de usuarios como parte capital de cualquier aplicación, constituye la base fundamental para la realización de cualquier proyecto y sobre la que se sustentan el resto de funcionalidades de la herramienta.

A mayores, como segundo punto crítico se destaca la gestión de eventos ya que es el núcleo de la herramienta y punto de encuentro entre el resto de funcionalidades. Una correcta gestión de eventos es lo que permite que esta herramienta sea funcional y cumpla su cometido.

Finalmente, tenemos el envío de correos electrónico y el almacenamiento de imágenes y documentos (así como su generación en base a los datos del sistema) dentro del servidor. Este punto es importante ya que permite la transferencia de información entre los usuarios y la correcta realización de estas funcionalidades era capital para el éxito del proyecto.

### 4.3 Seguimiento

Concluido el desarrollo de todas las funcionalidades de la herramienta, se ha creado *Tabla 2. Duraciones reales* que pretende plasmar la verdadera duración de las diferentes fases con la planificación inicial.

Fases	Tiempo	Inicio	Fin
<b>Inicio</b>	<b>10h</b>	<b>11 septiembre</b>	<b>13 septiembre</b>
Preparación	10h	11 septiembre	13 septiembre
<b>Análisis</b>	<b>15h</b>	<b>14 septiembre</b>	<b>17 septiembre</b>
Definición del sistema y análisis de los requisitos	15h	14 septiembre	17 septiembre
<b>Diseño</b>	<b>22h</b>	<b>18 septiembre</b>	<b>27 septiembre</b>
Diseño de casos de uso	8h	18 septiembre	20 septiembre
Diseño para la gestión de los datos	8h	21 septiembre	23 septiembre
Implementación de la base del proyecto	6h	24 septiembre	27 septiembre
<b>Implementación</b>	<b>118h</b>	<b>28 septiembre</b>	<b>19 noviembre</b>
Gestión de usuarios	16h	28 septiembre	2 octubre
Gestión de categorías	12h	3 octubre	7 octubre
Gestión de eventos	23h	8 octubre	16 octubre
Gestión de amistades	13h	17 octubre	21 octubre
Gestión de noticias	12h	22 octubre	26 octubre
Gestión de comentarios	15h	27 octubre	2 noviembre
Generación de PDFs	13h	3 noviembre	7 noviembre
Envío de correos y almacenamiento de documentos e imágenes	14h	8 noviembre	13 noviembre
<b>Verificación</b>	<b>30h</b>	<b>14 noviembre</b>	<b>25 noviembre</b>
Pruebas unitarias	18h	14 noviembre	21 noviembre
Pruebas globales	12h	22 noviembre	26 noviembre
<b>Mantenimiento</b>	<b>20h</b>	<b>27 noviembre</b>	<b>2 diciembre</b>
Validación y corrección de errores del sistema	10h	27 noviembre	30 noviembre
Finalización de la documentación del proyecto	10h	1 noviembre	4 diciembre
<b>Total</b>	<b>215h</b>	<b>11 septiembre</b>	<b>4 diciembre</b>

Tabla 2: Duraciones reales

Al igual que anteriormente, la tabla con la planificación real que se ha llevado a cabo viene acompañada de su correspondiente diagrama de Gantt *Ilustración 2. Diagrama de Gantt de las duraciones reales*.



## Planificación Real

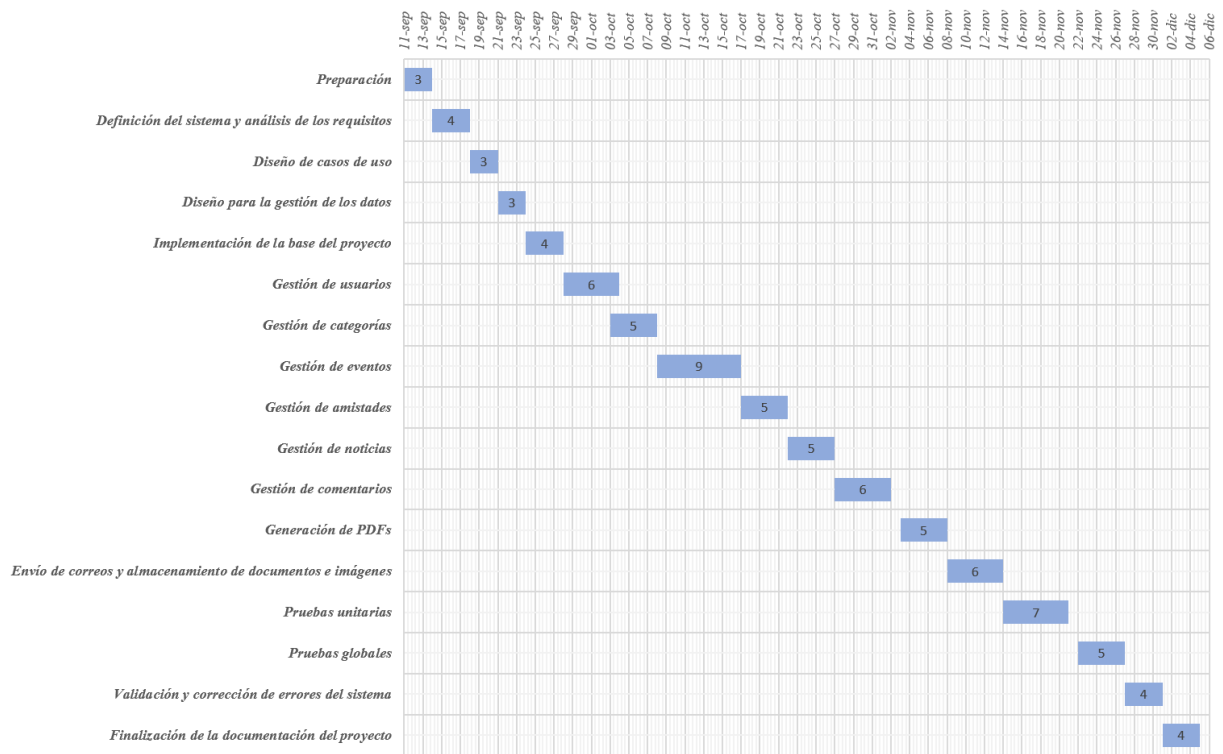


Ilustración 2: Diagrama de Gantt de las duraciones reales

## 4.4 Desviación

Como se puede apreciar en la tabla anterior que refleja la planificación real llevada a cabo en el proyecto, las dos primeras etapas (inicio y análisis) transcurrieron en base a lo previsto. Cabe destacar que la asignación de horas inicial dada para la fase de diseño han sido finalmente menores ya que estaba previsto invertir 25h y finalmente se han consumido en esta fase 22h.

Por otra parte, las desviaciones a nivel de fechas se han producido en las siguientes fases, siendo en la fase de implementación donde las fechas previstas para la realización de las tareas se han visto alteradas. Se destacan pues los dos siguientes puntos:

- La gestión de usuarios se ha realizado finalmente en 16h, a diferencia de las 21h estimadas en un inicio, lo que produjo que las fechas variaran con respecto a lo previsto.
- La gestión de categorías, finalmente se ha concluido en dos horas menos, siendo 12h en lugar de las 14h previstas, ya que el proceso terminó siendo más simple de lo esperado en un primer momento.

Si bien el resto de implementaciones y fases se han desarrollado en base a lo previsto en un inicio, estos cambios han producido que la fecha de finalización del proyecto se haya realizado una semana antes.

## 5. Arquitectura

Como ya se dijo anteriormente, este proyecto emplea una arquitectura de tipo cliente-servidor (Gonzales, Lizama, Kindley, & Jeria Morales, 2016). Esta arquitectura permite que múltiples equipos cliente interactúen con un servidor central, el cual ofrece servicios a los distintos clientes. Esta arquitectura se basa en el principio de petición-respuesta, donde un equipo cliente solicita uno o más recursos o acciones al servidor a través de una conexión de red, al mismo tiempo que otros clientes realizan peticiones al servidor. El servidor procesa la solicitud que realiza el cliente, lleva a cabo las operaciones pertinentes y envía una respuesta al cliente, que contiene la información solicitada o el resultado de las operaciones realizadas.

La arquitectura cliente-servidor ofrece varios beneficios, por una parte, tenemos la centralización de la lógica del negocio en el servidor, facilitando la escalabilidad y mantenimiento de nuestro sistema. Por otra parte, permite a los clientes acceder utilizar los servicios ofrecidos por el servidor de forma concurrente, lo que permite una mejor experiencia a los usuarios. Como puntos a destacar tenemos los siguientes:

- **Buena escalabilidad:** La arquitectura cliente-servidor permite una gran escalabilidad al agregar equipos clientes sin degradar el rendimiento del sistema. Se puede adaptar el sistema según las necesidades, aumentando la capacidad del servidor para manejar un número mayor de solicitudes o agregando equipos para satisfacer una mayor demanda.
- **Centralización de la administración:** En la arquitectura cliente-servidor, la mayor parte de la gestión le corresponde al servidor. Los clientes no tienen una gran carga de administración, lo que simplifica la implementación y el mantenimiento. La lógica de negocio se define y se implementa en el lado servidor, permitiendo un control centralizado.
- **Centralización de los recursos:** El responsable y parte central de la red es el servidor encargado de la administración y almacenamiento de recursos. Esto permite una gestión centralizada a nivel de datos y recursos. El cliente se conecta al servidor y accede a los recursos (base de datos, archivos, funcionalidades, etc).

En este proyecto se emplea un diseño SPA (Single page application) permitiendo que la respuesta en tiempo sea más rápida en comparación con otros enfoques. La eficiencia se debe a que gran parte del procesamiento recae en la responsabilidad del cliente. La comunicación cliente-servidor se lleva a cabo a través de peticiones asíncrona, empleando un API REST sobre HTTP sin necesidad de recargar la página completamente, como se puede ver en *Ilustración 3: Arquitectura*. Gracias a esta forma de trabajar, solo se actualizan las partes necesarias de la interfaz con nuevos datos, lo que mejora la experiencia de usuario. La lógica de la aplicación se lleva a cabo en cliente, siendo el servidor quien proporciona y gestiona datos, respondiendo a las peticiones del cliente.

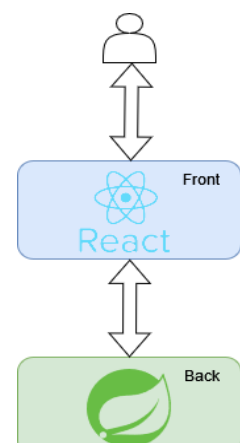


Ilustración 3: Arquitectura

## 5. Tecnologías y herramientas

En este capítulo se van a recoger las principales tecnologías y herramientas de desarrollo que han permitido desarrollar el proyecto.

### 6.1 Tecnologías

**HTML:** Conocido por las siglas de HTML o *Lenguaje de Marcado de Hipertexto* es el lenguaje de marcas estándar para la estructuración del contenido en la web. Los elementos se disponen entre etiquetas, lo que les permite a los navegadores hacer una representación común de la información (Diseño básico de páginas web en HTML, 2009).

**JavaScript:** Lenguaje de script empleado para facilitar la interactividad dentro de la web. Su uso está enfocado en la utilización dinámica de contenido, validación de datos en los formularios y creación de funciones.

**CSS:** Conocido también como hojas de estilos en cascada es un lenguaje de estilo empleado en la presentación y el diseño de las páginas web. Define los estilos utilizados como los colores, fuentes de texto y espaciado, estableciendo un diseño consistente y atractivo en dispositivos de diferentes tamaños de pantalla (Durango, 2015).

**Java:** Lenguaje de programación orientado a objetos (POO) cuyo uso está enfocado en el desarrollo de aplicaciones apps móviles y herramientas. Su filosofía se define por la frase "*escribe una vez, ejecuta en cualquier lugar*" que se refiere a la posibilidad de ejecutar el mismo código en cualquier dispositivo que disponga de una máquina virtual java, la cual otorga independencia del hardware y de plataforma (Eckel, 2010).

**SQL:** Lenguaje específico de gestión y manipulación de información en bases de datos relacionales. Conocido como *Lenguaje de Consulta Estructurada* permite la consulta de datos, actualización de registros y definición de estructuras de bases de datos. Su uso es capital para el almacenamiento, recuperación y gestión datos en aplicaciones y herramientas web o que necesiten almacenar y acceder a información (Escofet, 2002).

**JPA:** Conocida como *API de Persistencia de Java*, esta especificación Java es utilizada para gestionar las relacionales entre datos en aplicaciones Java. Establece una forma de mapeo de objetos Java a tablas en base de datos, lo que simplifica las interacciones con base de datos. Especialmente conocida es Hibernate, como especificación JPA, que permite la interacción con base de datos sencilla (Fernando Pech-May).

**React:** Biblioteca de JavaScript desarrollada por la compañía Meta para el desarrollo de interfaces usuario, muy enfocada en el desarrollo de aplicaciones de tipo SPA donde las actualizaciones son fundamentales para una mejor experiencia de usuario (React, 2016).

**Spring:** Popular framework para el desarrollo de aplicaciones basadas en Java, que permite a los desarrolladores crear aplicaciones complejas, escalables y seguras proporcionando una arquitectura de tipo flexible y modular. Spring se caracteriza por la inyección de dependencias, programación orientada a aspectos y el uso de módulos como son Spring MVC para la construcción de aplicaciones web (Mark Fisher, 2018). En cuanto a los módulos empleados, destacar Spring Boot (proporciona estructura general del

proyecto y simplifica la puesta en ejecución), Spring Data JPA (simplifica el acceso a BD empleando JPA), SpringMVC (para dar soporte a la implementación del API REST) y Spring Security (autenticación autorización usando JWT).

**JWT:** JSON Web Token es una tecnología de código abierto que se utiliza para transmitir información de forma segura entre dos partes como un objeto JSON. Se emplea comúnmente en autenticación y autorización en aplicaciones web y servicios, permitiendo la verificación de la integridad de los datos y la autenticidad de la fuente. JWT es compacto, autocontenido y puede ser firmado y/o cifrado, proporcionando un mecanismo eficiente y seguro para la transferencia de datos entre sistemas. Como URL destacable puede consultarse: <https://jwt.io/>

**PrimeReact:** Biblioteca de componentes prediseñados para React lo que permite crear interfaces de usuario de una forma más estable. Su amplia variedad de componentes listos a usar, como el caso de gráficos, tablas de datos o formularios, facilita el desarrollo de aplicaciones web modernas con que hagan uso de React. Se centra en la estética y funcionalidad simplifica para la creación de interfaces atractivas al usuario final. La URL de esta popular biblioteca es la siguiente: <https://primereact.org/>

**MySQL:** Sistema de gestión de bases de datos relacionales *open source*. Este sistema de gestión goza de una sólida reputación entre la comunidad de desarrolladores por su confiabilidad y rendimiento. MySQL es ampliamente utilizado en desarrollos, herramientas y aplicaciones web. Hace uso del lenguaje SQL para la gestión de los datos y es compatible con múltiples plataformas y lenguajes de programación.

## 6.2 Herramientas de desarrollo

**VisualParadigm:** Herramienta de modelado y diseño de software muy empleado para la representación de diagramas UML hasta modelado de procesos de tipo empresarial. Su uso está ampliamente extendido para facilitar la documentación y visualización de arquitecturas y procesos de software. Su

**Git:** Herramienta para el control de versiones muy popular en el desarrollo de software, que permite a los programadores la gestión de los cambios en código fuente de forma eficiente. Su naturaleza facilita la colaboración y el seguimiento de modificaciones.

**GitHub:** Plataforma web basada en el uso de Git para el control de versiones y facilitar la colaboración en equipo en el desarrollo software. Permite crear y mantener repositorios, gestión de problemas, y realización de solicitudes de extracción.

**IntelliJ:** Entorno de desarrollo integrado (IDE) muy popular para el desarrollo de código en Java y otros Frameworks y lenguajes de programación. Creado por la compañía JetBrains, se destaca por poseer un amplio conjunto de herramientas de desarrollo.

**Visual Studio Code:** Conocido como *VS Code* es un editor de código fuente muy popular por su abundante cantidad de extensiones. Desarrollado por Microsoft, es popular por dar soporte a una gran cantidad de lenguajes de programación, amplia gama de extensiones para ampliar sus funcionalidades. Su interfaz intuitiva y su integración con otras herramientas lo convierten en una opción muy común en el desarrollo de código.

## 7. Análisis y especificación de requisitos

En los siguientes puntos se expondrán los requisitos que han de cumplirse en el desarrollo de la aplicación. A continuación, se expondrán los requisitos tanto funcionales y no funcionales.

*EventGO* dispondrá de tres roles diferentes para los usuarios del sistema. En primer lugar, están los *usuarios* que disponen de la posibilidad de ver los diferentes eventos que existe en las categorías del sistema, inscribirse en ellos, comentarlos y seguir a gerentes para ser conocedores de las principales novedades que estos quieran compartir. A continuación, tenemos a los *gerentes*, estos son los encargados de crear nuevos eventos en el sistema y de compartir noticias con el resto de usuarios de la web. Finalmente, está el *administrador* del sistema que tendrá las funcionalidades anteriormente descritas y la posibilidad de gestionar las categorías de eventos de la aplicación. A mayores, los usuarios no registrados podrán ver las categorías y eventos existentes en el sistema.

### 7.1 Requisitos funcionales

Los requisitos funcionales establecen las funcionalidades específicas o comportamiento que presenta el sistema y los componentes que lo integran, además de la manipulación de la información a gestionar. Los siguientes puntos delimitan las operaciones deseadas por parte del sistema y cómo interactúan los componentes de este en relación con el manejo de la información.

#### 7.1.1 Gestión de usuarios (RF1)

- **Login (RF1.1):** Los usuarios que existen en el sistema podrán acceder a este a través de sus credenciales de autenticación.
- **Ver eventos sin hacer login (RF1.2):** Los usuarios si no disponen de una cuenta en el sistema pueden consultar los eventos del sistema sin la posibilidad de inscribirse.
- **Registro (RF1.3):** Los usuarios que no existan en el sistema podrán ingresar su información personal y crear un nuevo perfil para poder logearse en el sistema.
- **Recuperar contraseña (RF1.4):** Un usuario existente en el sistema puede obtener una nueva contraseña de acceso ingresando su login y correo de recuperación.
- **Crear usuario (RF1.5):** El administrador podrá crear nuevos usuarios en el sistema, tanto usuarios como gerentes.
- **Buscar usuario (RF1.6):** El administrador podrá buscar usuarios en el sistema para realizar sobre estas diferentes acciones. Los usuarios y gerentes podrán buscar nuevos gerentes a seguir en el sistema.
- **Editar usuario (RF1.7):** El administrador podrá editar los datos de los usuarios, así como los usuarios y gerentes podrán editar sus datos personales.
- **Ver en detalle usuario (RF1.8):** El administrador podrán acceder a toda la información de un usuario, así como cualquier usuario podrá consultar toda su información.
- **Eliminar usuario (RF1.9):** El administrador podrá eliminar un usuario del sistema.

- **Reactivar usuario (RF1.10):** El administrador podrá reactivar un usuario para que este pueda volver a realizar acciones si no se ha eliminado de forma física del sistema.

#### 7.1.2 Gestión de categorías (RF2)

- **Crear categoría (RF2.1):** El administrador podrá crear nuevas categorías en el sistema.
- **Buscar categoría (RF2.2):** El administrador podrá buscar categorías en el sistema para realizar sobre estas diferentes acciones. Los usuarios y gerentes podrán buscar categorías para ver que eventos existen en su interior.
- **Editar categoría (RF2.3):** El administrador podrá editar los datos de las categorías del sistema.
- **Ver en detalle categoría (RF2.4):** El administrador podrán acceder a toda la información de una categoría.
- **Eliminar categoría (RF2.5):** El administrador podrán eliminar una categoría del sistema.
- **Reactivar categoría (RF2.6):** El administrador podrá reactivar una categoría del sistema que este pueda volver a estar accesible para los usuarios si esta no se ha eliminado de forma física del sistema.

#### 7.1.3 Gestión de eventos (RF3)

- **Crear evento (RF3.1):** El administrador y gerente podrán crear nuevos eventos en una categoría concreta.
- **Buscar evento (RF3.2):** El administrador podrá buscar eventos en el sistema para realizar sobre estas diferentes acciones. Los usuarios y gerentes podrán buscar eventos.
- **Editar evento (RF3.3):** El administrador y gerente del evento podrán editar los datos del evento.
- **Ver en detalle evento (RF3.4):** El administrador, los gerente y usuarios podrán acceder a toda la información de un evento.
- **Eliminar evento (RF3.5):** El administrador y gerente podrá eliminar un evento del sistema.
- **Reactivar evento (RF3.6):** El administrador y gerente del evento podrá reactivarlo para que pueda volver a estar accesible para los usuarios si este no se ha eliminado de forma física del sistema.

#### 7.1.4 Gestión de solicitudes (RF4)

- **Crear solicitud (RF4.1):** Los usuarios podrán crear nuevas solicitudes a eventos privados en los que desean participar.
- **Buscar solicitud (RF4.2):** Los usuarios y gerentes podrán buscar entre sus solicitudes existentes en el sistema.
- **Eliminar solicitud (RF4.3):** Los usuarios podrán eliminar las solicitudes personales anteriormente creadas, los gerentes y administradores podrán eliminar solicitudes de los usuarios en los eventos del sistema.

- **Aceptar solicitud (RF4.4):** Los gerentes y administrador podrán aceptar las solicitudes de participación en un evento, pasando estas a ser suscripciones y generándose un correo electrónico como resultado del proceso que notifique al usuario de la existencia de una nueva suscripción.

#### 7.1.5 Gestión de suscripciones (RF5)

- **Crear suscripciones (RF5.1):** Los usuarios y gerentes podrán crear nuevas solicitudes a eventos públicos en los que desean participar y recibirán un correo electrónico como resultado del proceso.
- **Buscar suscripciones (RF5.2):** Los usuarios y gerentes podrán buscar entre sus suscripciones existentes en el sistema.
- **Eliminar suscripciones (RF5.3):** Los usuarios podrán eliminar las suscripciones personales anteriormente creadas para un evento, los gerentes y administradores podrán eliminar suscripciones de los usuarios en los eventos del sistema.
- **Generación de PDF de la suscripción (RF5.4):** El usuario podrá generar un PDF con sus datos personales y del evento que sirva como prueba de que está suscrito al evento.

#### 7.1.6 Gestión de amistades (RF6)

- **Crear amistad (RF6.1):** Los usuarios y gerentes podrán crear nuevas amistades, comenzando a seguir a un gerente del cual les interesa recibir notificaciones por correo electrónico.
- **Buscar amistad (RF6.2):** Los usuarios y gerentes podrán buscar entre otros gerentes que siguen en el sistema.
- **Eliminar amistad (RF6.3):** Los usuarios o gerentes podrán eliminar otros gerentes a los que están siguiendo para dejar de recibir notificaciones por correo electrónico.

#### 7.1.7 Gestión de noticias (RF7)

- **Crear noticia (RF7.1):** Los gerentes podrán crear nuevas noticias, lo que generará un aviso para cada usuario que siga al gerente.
- **Buscar noticia (RF7.2):** Un gerente podrá buscar entre las noticias que ya haya creado previamente.
- **Eliminar noticia (RF7.3):** Un gerente puede eliminar noticias que haya creado con anterioridad.

#### 7.1.8 Gestión de comentarios (RF8)

- **Crear comentario y valoración (RF8.1):** Un usuario o gerente que tenga una suscripción a un evento puede comentar su opinión sobre este y valorarlo del uno al cinco para que sirva de información a otros usuarios del sistema sobre el evento en cuestión.
- **Ver comentarios y valoraciones (RF8.1):** Los usuarios en el sistema podrán ver los comentarios que otros usuarios hayan creado previamente sobre ese evento y ver en un gráfico sus valoraciones.



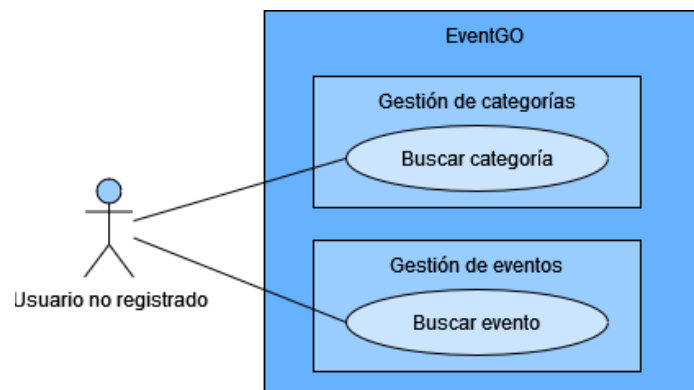
## 7.2 Requisitos no funcionales

Los requisitos no funcionales se centran en las restricciones o características impuestas al diseño de la aplicación. Se focalizan en el comportamiento del sistema sobre las funcionalidades del mismo. Destacamos los siguientes requisitos no funcionales:

- **Desempeño y eficiencia (RNF1):** Se espera que los lapsos de respuesta de las consultas sean los más breves posible, dándole al usuario una experiencia en la web fluida.
- **Adaptabilidad (RNF2):** La aplicación ha de adaptarse a diferentes navegadores web y sistemas operativos, garantizando un correcto funcionamiento.
- **Uso intuitivo (RNF3):** El usuario debe de poder disfrutar en el sistema de una navegación fácil y opciones que permitan perfeccionar las búsquedas.
- **Interfaz intuitiva (RNF4):** La interfaz web ha de ser intuitiva para el usuario, facilitando su manejo, con el fin de proporcionar una experiencia de usuario satisfactoria.
- **Tolerancia a fallos (RNF5):** El sistema debe ser tolerante a fallos, asegurando un correcto desempeño ante fallos en la ejecución o errores del usuario.
- **Multiidioma (RNF6):** La página web es multidioma ya que los usuarios podrán decidir si quieren visualizar el contenido en castellano, inglés y gallego. La traducción se realiza de forma dinámica en front y el idioma seleccionado se mantendrá en los cambios entre página.

## 7.3 Casos de uso

En la *Ilustración 4: Casos de uso de usuario no registrado*, podemos ver las acciones que puede realizar un usuario que no esté registrado en el sistema.



*Ilustración 4: Casos de uso de usuario no registrado*

En la *Ilustración 5: Casos de uso de usuario registrado*, podemos ver las acciones que puede realizar un usuario que si está registrado en el sistema.



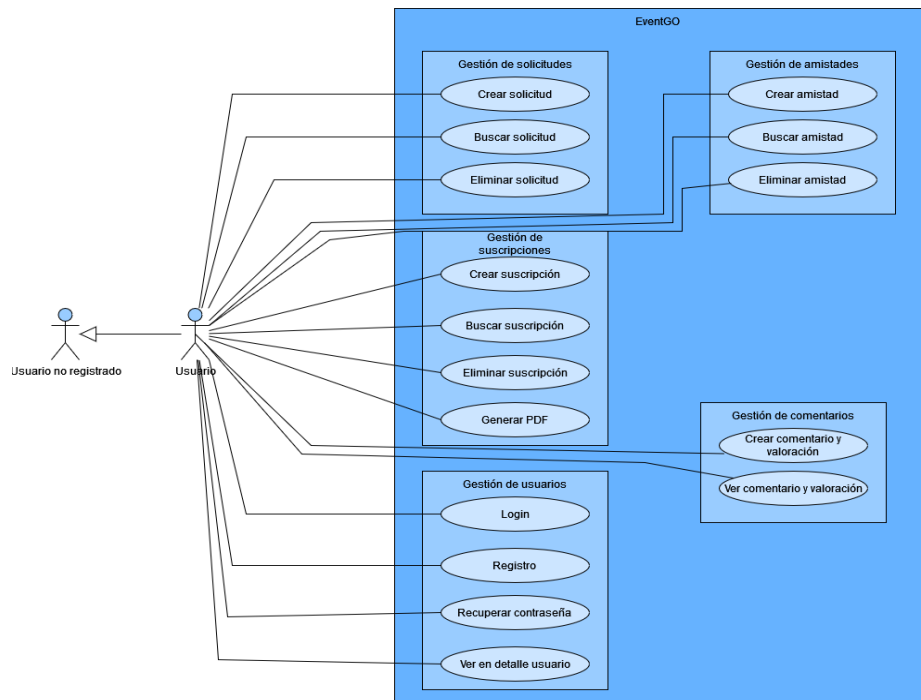


Ilustración 5: Casos de uso de usuario registrado

En la *Ilustración 6: Casos de uso de gerente*, podemos ver las acciones que puede realizar un usuario que si está registrado en el sistema.

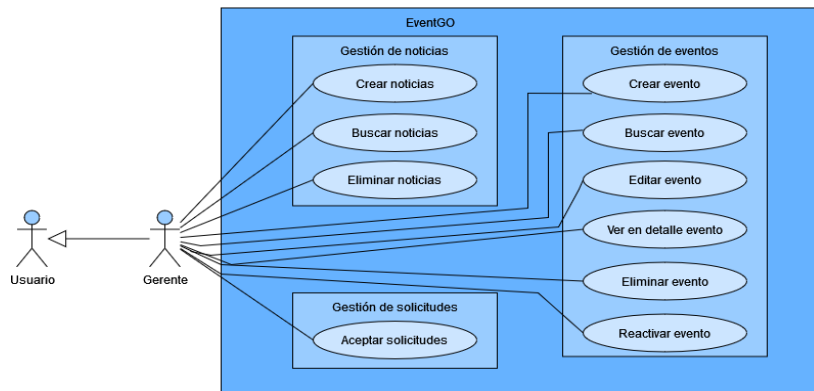


Ilustración 6: Casos de uso de gerente

En la *Ilustración 7: Casos de uso de administrador*, podemos ver las acciones que puede realizar un administrador que si está registrado en el sistema.

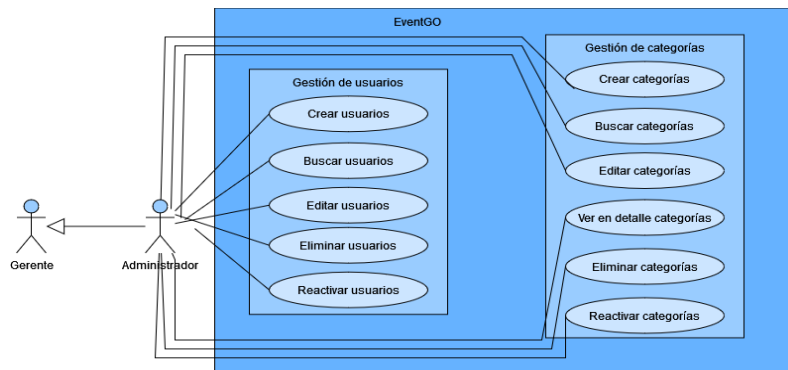


Ilustración 7: Casos de uso de administrador

## 8. Diseño de Software

A continuación, se tratará la representación del diseño del proyecto *EventGO* tanto de forma dinámica como estática. En cada uno de los siguientes puntos se presentarán una serie de diagramas genéricos con el objetivo de reflejar de forma simple el funcionamiento del proyecto y sus partes más destacables.

### 8.1 Diseño estático

En este apartado se realiza una explicación de todos los diseños de tipo estático que representan el funcionamiento a nivel de *FrontEnd* y *BackEnd*.

#### 8.1.1 Diagramas de clases de Backend

A continuación, se presentan la *Ilustración 8: Diagrama de la estructura del Backend* y la *Ilustración 9: Diagrama de clases del Backend*, para servir de apoyo en la explicación del funcionamiento del *Backend*.

En el siguiente diagrama se muestra una representación esquemática de la estructura del *Backend*, para entender de forma lógica su funcionamiento. Podemos apreciar que existen una serie de *Controladores* (también conocidos como endpoints) los cuales reciben las peticiones HTTP que se realizan desde el cliente. Estos *Controladores*, si bien pueden tener más métodos, por regla general, presentarán los mostrados en el diagrama, necesarios para una gestión genérica.

Será desde estos *Controladores*, desde los cuales se llame al método correspondiente de *ValidacionesAtributos*, en el cual se llevará a cabo la validación del formato de los datos. Esto es necesario ya que a nivel de *Backend* no se puede garantizar que la información de las peticiones haya sido anteriormente validada en *Frontend*. Si alguno de los datos incumple las restricciones de formato se producirá una excepción (*atributoException*).

Una vez superada la validación de atributos se realizarán desde los *Controladores* las peticiones a los *Servicios*, en los cuales se lleva a cabo la lógica de negocio. Esto *Servicios* implementarán la interfaz correspondiente a cada uno y es en ellos donde se realiza la validación de acciones. En caso de que se haya detectado algún error a nivel de acciones se producirá una excepción (*accionException*), sino se realizará una petición al *DAO*.

Los *DAO* son los encargados de realizar las consultas a Base de Datos y extienden de JPA Repository. Como ya se mencionó, JPA establece una forma de mapeo de los objetos Java a tablas, esto simplifica las interacciones con base de datos.

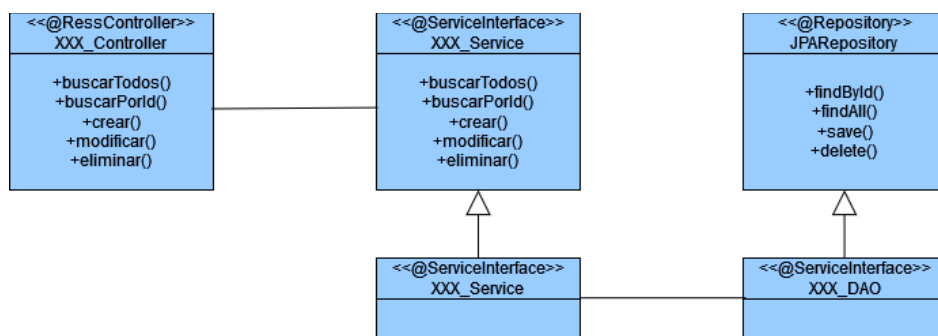


Ilustración 8: Diagrama de la estructura del Backend

A mayores de lo anterior, en el siguiente diagrama podemos apreciar la estructura estática de las clases principales del proyecto a nivel de *Backend*.

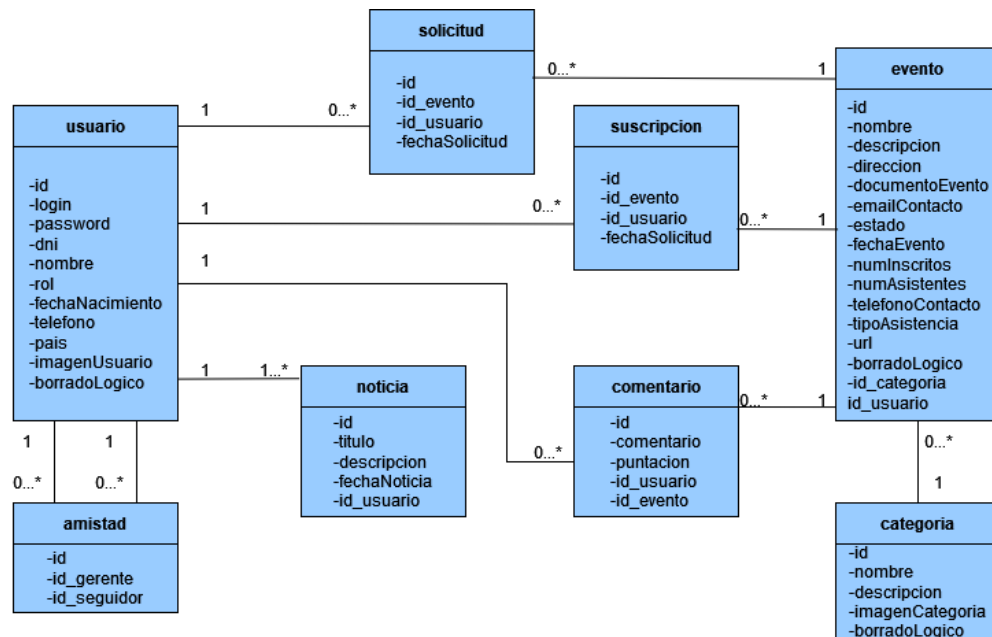


Ilustración 9: Diagrama de clases del Backend

### 8.1.2 Diagramas de clases de Frontend

A continuación, se representa el diseño estático a nivel de *Frontend*, para ello vamos a tomar como ejemplo la gestión de categorías. En primer lugar, tenemos nuestros *Componentes* encargados de la generación de la vista donde interactúa el usuario.

En función de la opción que realice el usuario en la interfaz, se tiene que realizar una petición a nuestro servidor. Estas peticiones se realizan desde las clases *Services* y serán recibidas en los endpoints de los *Controladores* a nivel de *Backend*.

Es cierto que esta estructura puede tener variaciones en algún caso en función de la gestión que estemos realizando, no obstante, todas siguen un desarrollo similar.

- categoriaShowAll: Vista encargada de la gestión de las categorías.
- categoriaLayout: Vista encargada de representar las categorías para ver los eventos.

Podemos ver este proceso en la *Ilustración 10: Diagrama de estructura del Frontend*.

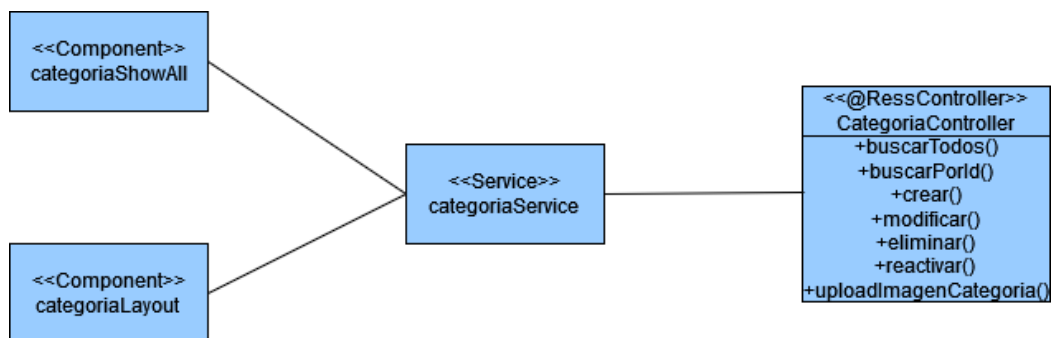


Ilustración 10: Diagrama de estructura del Frontend

## 8.2 Diseño dinámico

En este apartado se va a reflejar el funcionamiento a nivel dinámico para el caso de la inserción de una categoría y buscar categorías. En un primer lugar vamos a reflejar el proceso a nivel de *Backend*, mediante la *Ilustración 11: Diagrama de secuencia para insertar una categoría en Backend* y en la *Ilustración 12: Diagrama de secuencia para buscar categorías de Frontend* vamos a reflejar el proceso a nivel de *Frontend*.

Como podemos apreciar desde *Backend*, desde cliente se envía la petición para crear la categoría a *CategoriaController*, desde aquí se llamará al método de comprobación de atributos, el cual puede producir una excepción si se detecta algún error a nivel de datos. Si no se detecta ningún error se hace la petición al servicio donde se realizará la comprobación de acciones y si no se detecta ningún error se creará la categoría.

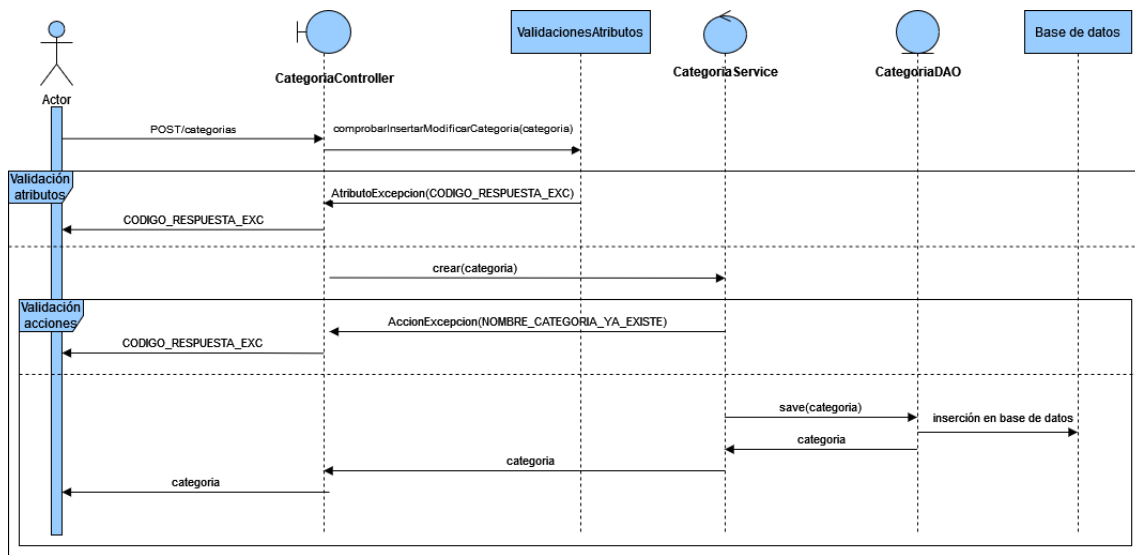


Ilustración 11: Diagrama de secuencia para insertar una categoría en Backend

Como podemos apreciar en el a nivel de *FrontEnd*, el usuario pretende ver las categorías que existen en el sistema. Para ello se llama al método del servicio en *Frontend* para realizar la búsqueda, lo que desencadena la petición al servidor. Como resultado se obtiene las categorías existentes en el sistema.

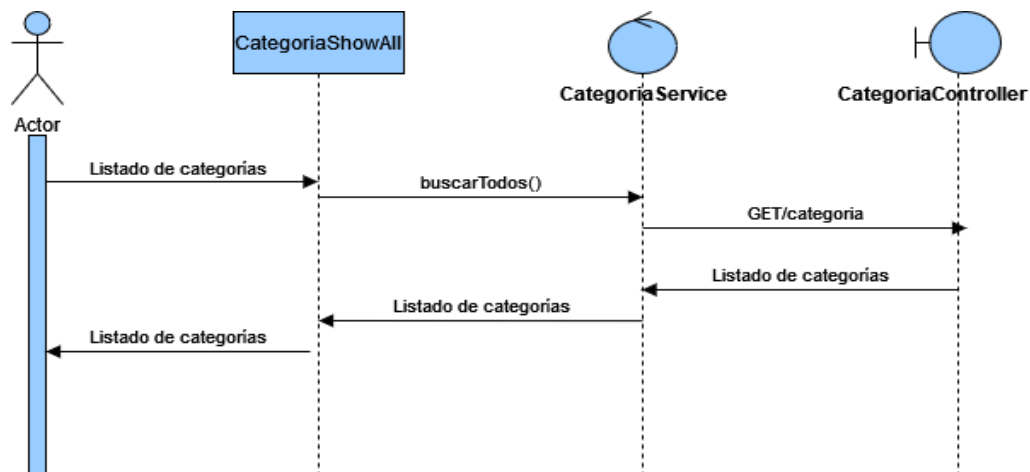


Ilustración 12: Diagrama de secuencia para buscar categorías

## 9. Gestión de información y datos

A continuación, vamos a exponer el Modelo Entidad Relación y paso a tablas que refleja como se estructura el almacenamiento de la información.

### 9.1 Modelo entidad relación MERE

Con el objetivo de mostrar las relaciones existentes entre las entidades de la base de datos, se muestra en la *Ilustración 13: Modelo Entidad Relación Extendido (MERE)*, en el que además se señalan los atributos pertenecientes a cada entidad. Es conveniente recalcar que se hace uso JPA y que todas las entidades estarán mapeadas en clases Java anotadas con `@Entity` y que se hace uso en todos los casos de ID autogenerados.

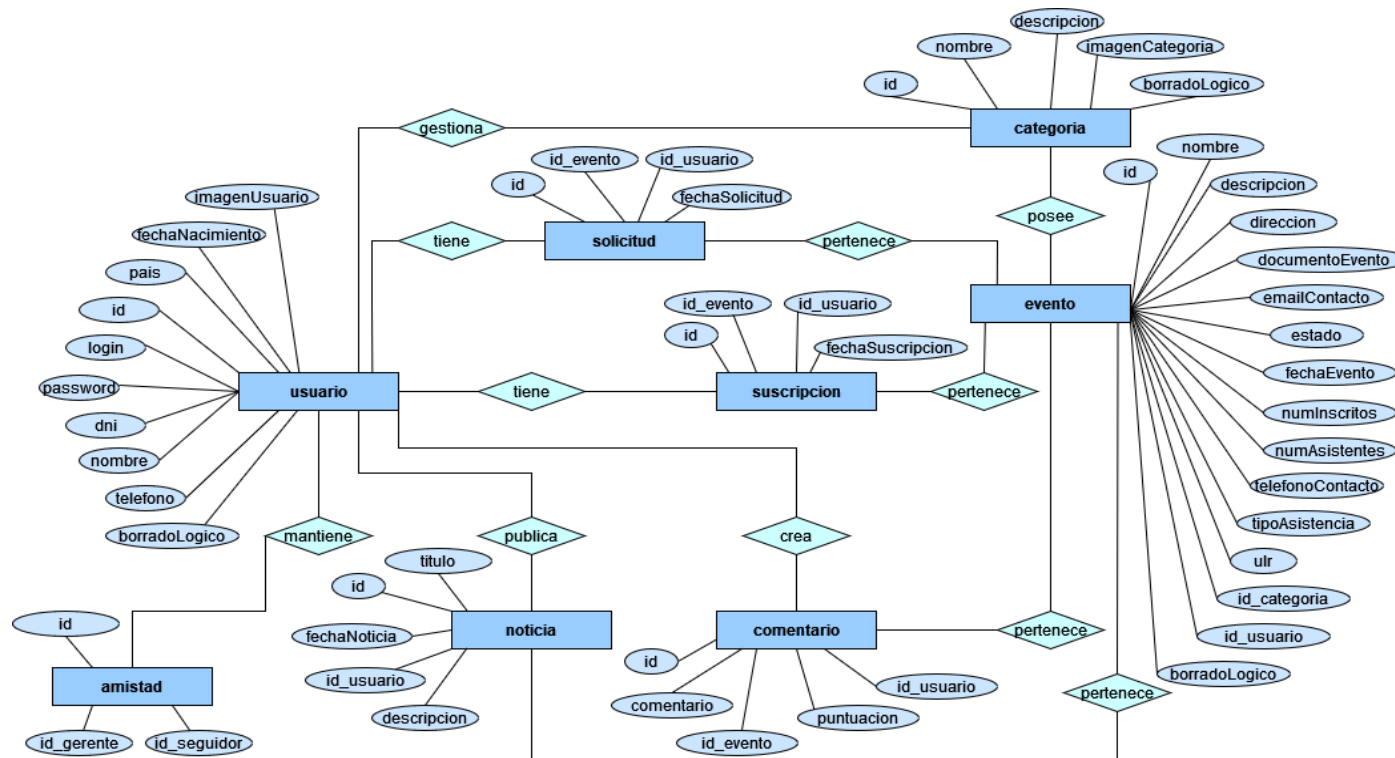


Ilustración 13: Modelo Entidad Relación Extendido (MERE)

## 9.2 Paso a tablas

Además, se adjunta la estructura física de la base de *EventGO* a continuación en la *Ilustración 14: Estructura física de la Base de Datos*, indicando las relaciones entre las entidades del sistema.

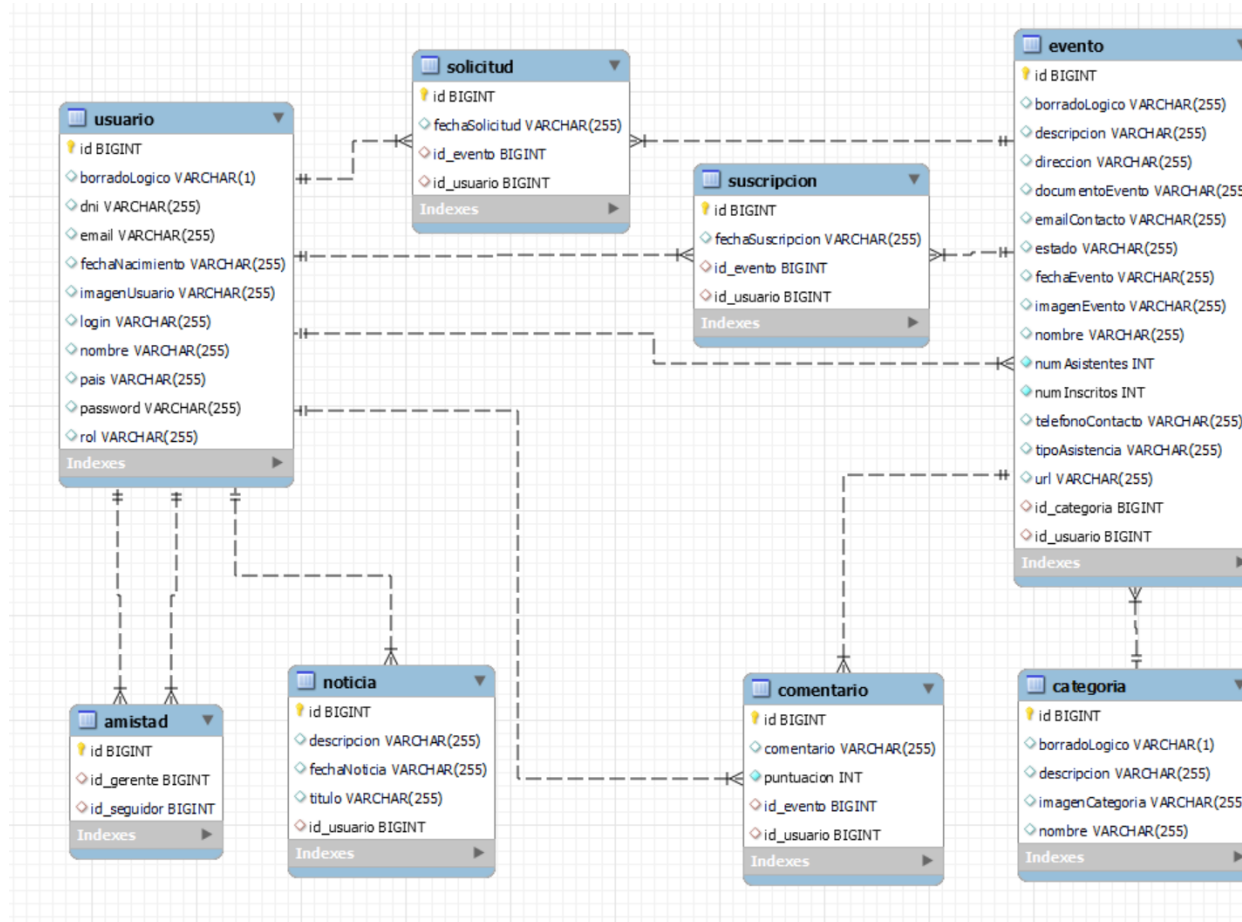


Ilustración 14: Estructura física de la Base de Datos

## 10. Pruebas

En esta sección se recoge una explicación breve de las pruebas llevadas a cabo sobre el proyecto a lo largo de su realización. Las pruebas que se han realizado sobre el código han sido tests de interacción manuales sobre el API REST haciendo uso de postman. Todas estas pruebas se han ido realizando a medida que se realizaban nuevas funcionalidades en el sistema.

Sobre el proyecto se han realizado dos tipos de pruebas. Las primeras han ido enfocadas en la validación del funcionamiento de los diferentes endPoints del API REST. El objetivo de ello era ver si el comportamiento desencadenado al realizar una petición a una URL determinada era el correcto.

El segundo tipo de pruebas han ido enfocadas en la validación del funcionamiento de la aplicación. Se ha probado las diferentes acciones que presenta el sistema y comprobando si el resultado esperado era el correcto o no.

Estas pruebas están recogidas en un archivo situado en /documentación/TFM.postman.json el cual se puede importar y cuenta en su interior con un conjunto de carpetas en función del endPoint determinado y una salida de cada una de las peticiones correctamente realizadas.

## 11. Manual de usuario

En esta sección de la documentación se recogen en forma de guía el proceso de instalación y puesta en funcionamiento de la aplicación, así como las acciones de funcionamiento dentro del sistema.

### 11.1 Manual de instalación

Para la puesta en marcha de la aplicación, será necesario contar con un gestor de bases de datos MySQL, donde estarán almacenados los datos del sistema. En este trabajo, se ha tomado la decisión de hacer uso de WorkBench como herramienta visual para la gestión de los datos. Como paso inicial es necesario la creación de la base de datos y un usuario con permisos. Para ello se puede hacer uso de los siguientes comandos:

```
create user root@localhost identified by "root";  
  
create database eventgo;  
  
grant all privileges on eventgo.* to root@localhost;
```

Como siguiente paso es necesario poner en funcionamiento el Backend del proyecto. Para poder realizar este paso es necesario contar con una versión de JDK igual o superior a 11. Para la puesta en marcha del Backend, se puede hacer uso de los siguiente comandos Maven, una vez situados en el directorio bases de nuestro proyecto.

```
mvn clean  
  
mvn spring-boot:run
```

Finalmente, es necesario poner en marcha el *Frontend*. Para su ejecución debemos instalar las dependencias del proyecto que hace uso de Node.js versión 16 mediante los siguientes comandos una vez situados en el directorio base de nuestro proyecto.

```
npm install  
  
npm start
```

Si el proceso ha sido exitoso, la interfaz del proyecto estará disponible en la URL por defecto en el puerto 3000, esta URL es la siguiente: <http://localhost:3000>

### 11.2 Manual de uso

En el siguiente punto se recogen una serie de explicaciones sobre el funcionamiento de *EventGO* por parte de los usuarios dentro del sistema.

#### 11.2.1 Vista de inicio

##### 11.2.1.1 Login

Si nos dirigimos a esa URL deberíamos encontrarnos en la vista de login, desde la cual vamos a poder acceder al sistema introduciendo las credenciales correspondientes. En esta vista, tal y como podemos apreciar en la siguiente imagen *Ilustración 15 Login*. En caso de que las credenciales de acceso fueran incorrectas se le notificaría por pantalla al usuario. Además, tenemos la posibilidad de cambiar entre los tres idiomas de la aplicación, así como ver los eventos del sistema, registrarnos o recuperar la contraseña.



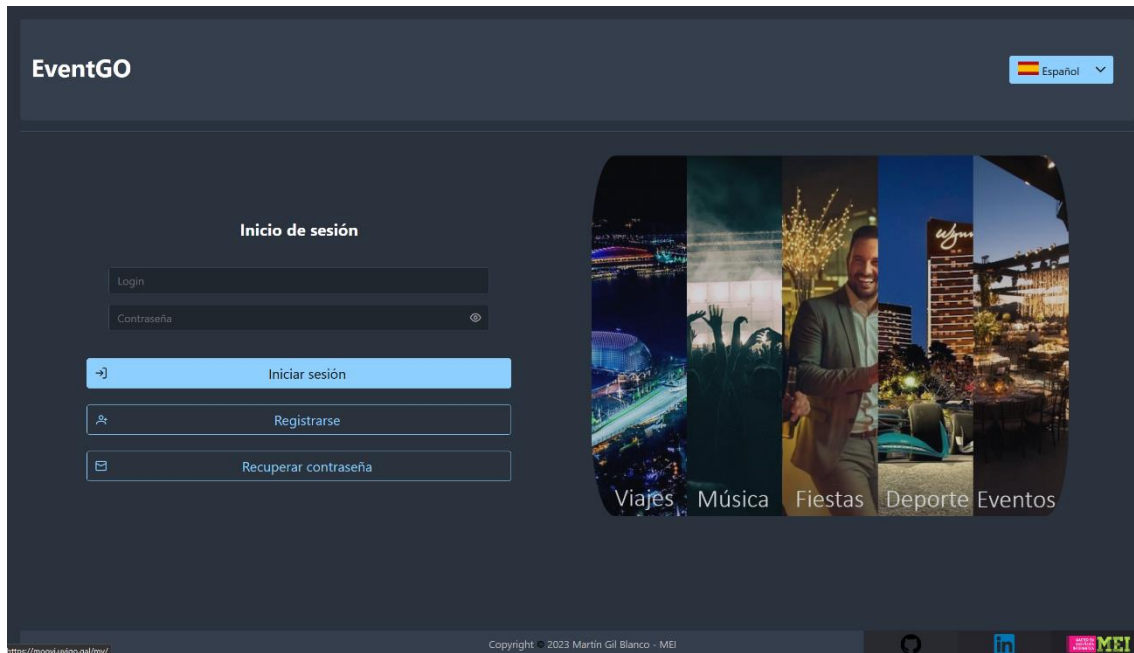


Ilustración 15: Login

Pudiera darse el caso de que existiera algún error a nivel de validación de datos en cliente, si este fuera el caso se le notificaría al usuario con un mensaje como se puede apreciar en la *Ilustración 16: Validación de datos en cliente*.

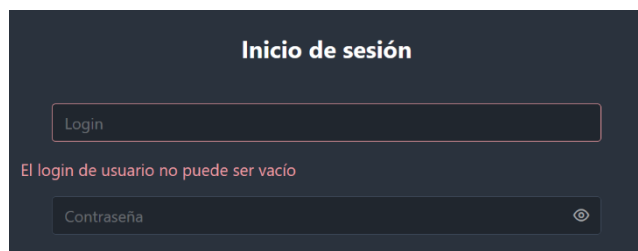


Ilustración 16: Validación de datos en cliente

También es posible que los datos introducidos fueran correctos pero que la acción no se pudiera realizar, dicho de otro modo, el servidor realiza una validación de acciones y en caso de no ser posible realizar la acción se le notifica al usuario por parte del servidor. Esto se puede apreciar en la *Ilustración 17: Validación de acciones del servidor*.

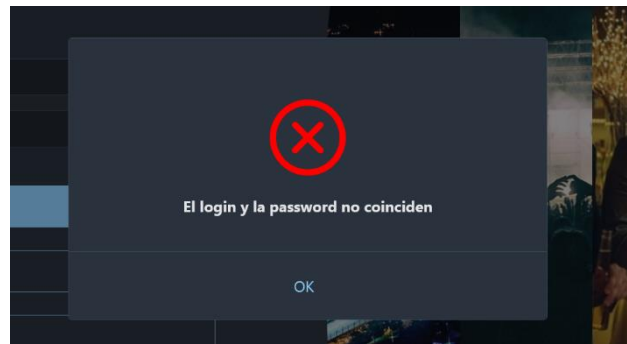
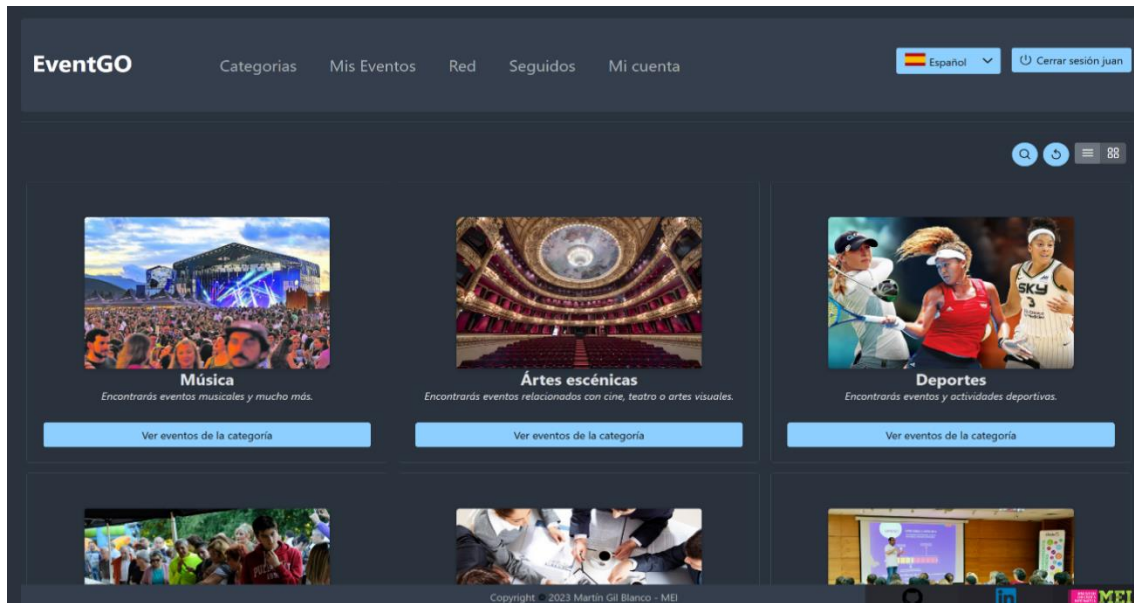


Ilustración 17: Validación de acciones del servidor

Estas comprobaciones de datos se realizarán de forma análoga en cualquier funcionalidad de *EventGO*, por lo que su comportamiento será similar en el resto de casos y no se representará en el resto de la documentación. Si las credenciales de usuario son correctas el usuario accederá a *EventGO*.

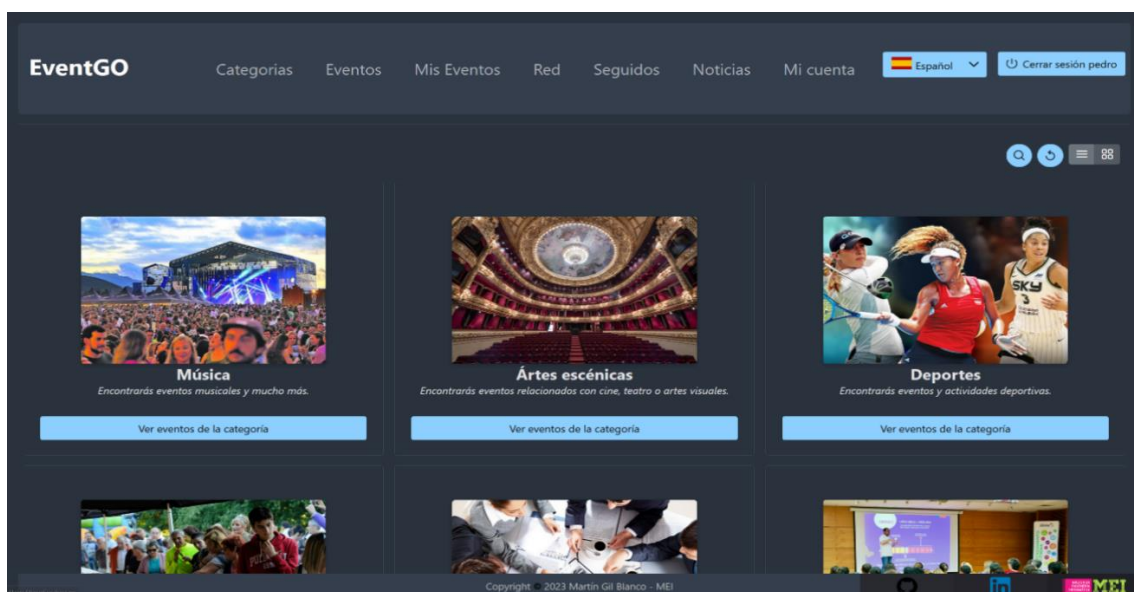
A continuación, se muestran los tres casos de vista en función del rol del usuario logeado, en un primer lugar tenemos la vista de un usuario registrado en el sistema en la *Ilustración 18: Vista de usuario*, la vista de gerente en *Ilustración 19: Vista de gerente* y finalmente, la vista de administración en *Ilustración 20: Vista de administrador*.

En la vista de usuario, dentro de “Categorías” podemos ver el conjunto de categorías del sistema y acceder a estas para ver que eventos existen en cada una. A mayores podemos consultar en “Mis Eventos” los eventos en los que estamos suscritos o hemos solicitado participar. Existe un apartado de “Red” en el cual podemos ver los diferentes gerentes de los que queremos saber las novedades que quieran compartir con su comunidad. Existe otro apartado de “Seguidos” donde tenemos un listado de los gerentes a los que recibir notificaciones por correo y por último un apartado “Mi cuenta” donde podemos consultar los datos personales. A mayores podemos cerrar sesión y volver a la vista de login.



*Ilustración 18: Vista de usuario*

En la vista de gerente tenemos a mayores los apartados de “Mis Eventos” desde la cual los gerentes pueden llevar a cabo una gestión de sus eventos y el apartado de “Noticias” donde pueden crear nuevos avisos a su comunidad de seguidores.



*Ilustración 19: Vista de gerente*

Por último, tenemos el caso del administrador, el cual tiene los apartados de “Gestión de categorías”, donde puede gestionar las categorías y los eventos que contienen. Además, en “Gestión de usuarios” puede realizar acciones de gestión de los usuarios del sistema.

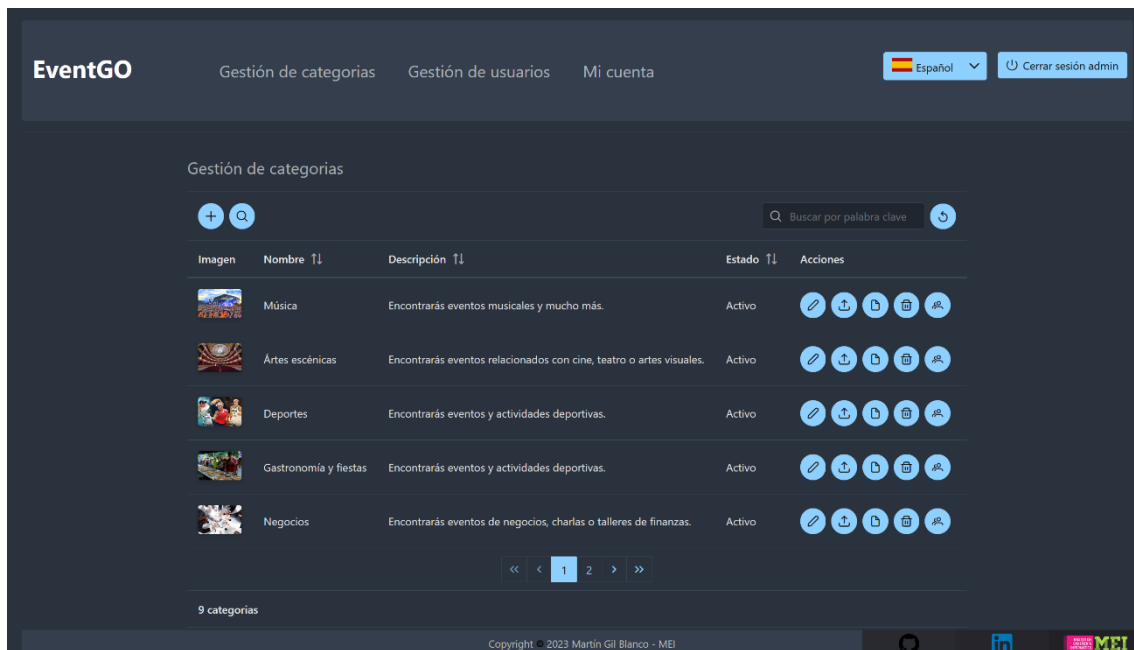


Ilustración 20: Vista de administrador

#### 11.2.1.2 Ver eventos sin hacer login

Los usuarios no registrados en el sistema pueden ver el sistema de forma limitada, si clican sobre las imágenes de eventos en la página de inicio acceden a las categorías del sistema y pueden consultar los eventos que existen en cada categoría, pero no pueden llevar a cabo más acciones sin estar registrados. Esto se puede ver reflejado en la *Ilustración 21: Ver eventos*.

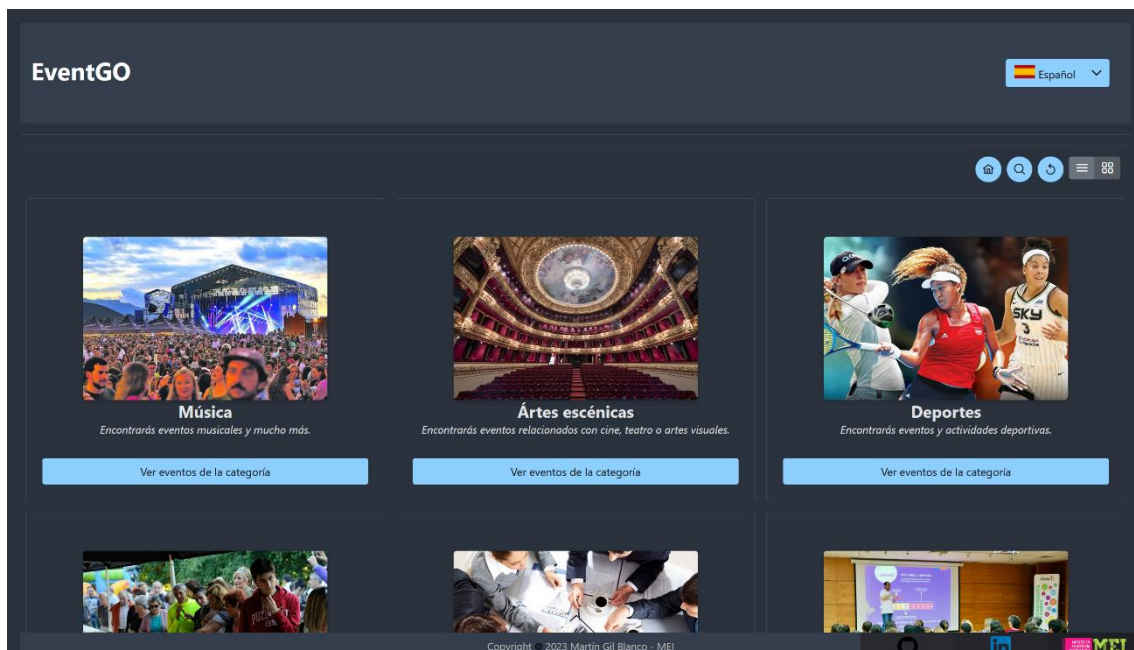
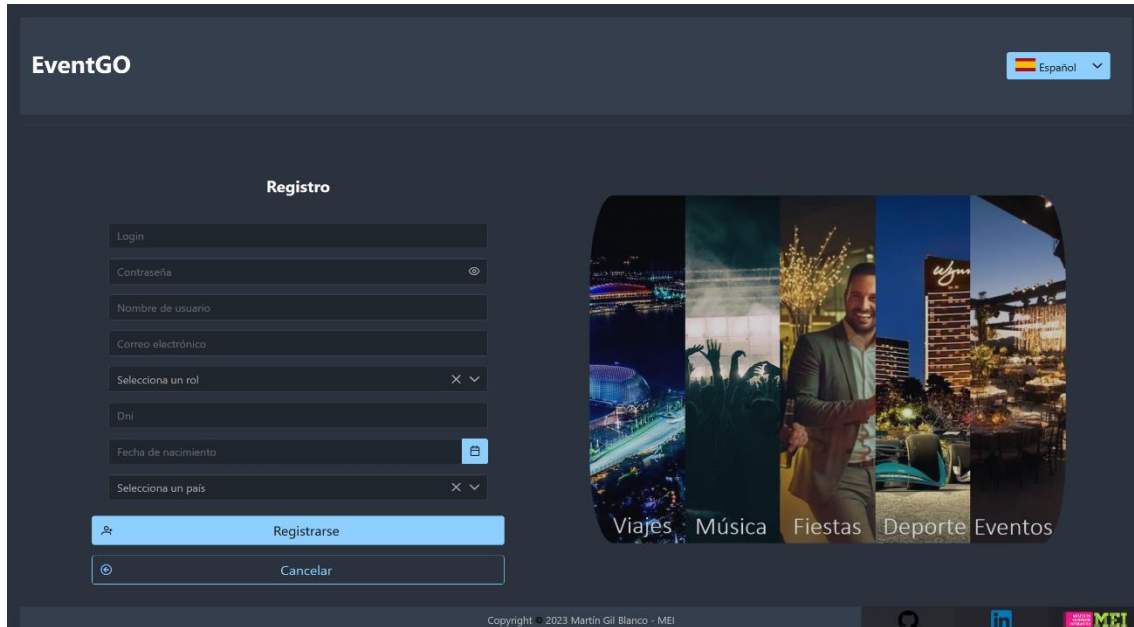


Ilustración 21: Ver eventos

### 11.2.1.3 Registro

Si un usuario no dispone de una cuenta en *EventGO* tiene la opción de registrarse en el sistema rellorando el formulario de registro para posteriormente acceder a la aplicación, tal y como podemos apreciar en la *Ilustración 22: Registro*.

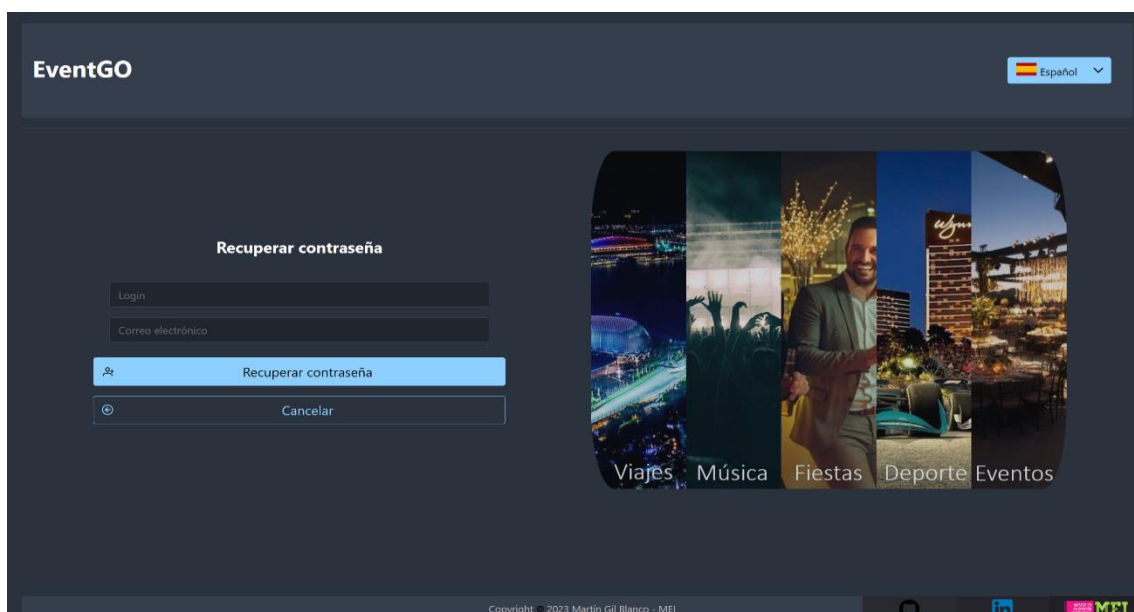


The screenshot shows the 'Registro' (Registration) page of the EventGO application. The page has a dark theme. At the top left is the 'EventGO' logo, and at the top right is a language selector set to 'Español'. The main content area is titled 'Registro' and contains a form with the following fields: 'Login', 'Contraseña' (password), 'Nombre de usuario', 'Correo electrónico', 'Selecciona un rol' (role selection), 'Dni', 'Fecha de nacimiento' (birth date), and 'Selecciona un país' (country selection). Below the form are two buttons: 'Registrarse' (Register) and 'Cancelar' (Cancel). To the right of the form is a horizontal carousel of five images with labels: 'Viajes' (Travel), 'Música' (Music), 'Fiestas' (Parties), 'Deporte' (Sports), and 'Eventos' (Events). The bottom of the page features a copyright notice 'Copyright © 2023 Martín Gil Blanco - MEI' and social media icons for LinkedIn and MEI.

Ilustración 22: Registro

### 11.2.1.4 Recuperar contraseña

Si un usuario que dispone de una cuenta en el sistema ha olvidado su contraseña puede solicitar otra contraseña nueva con la que acceder a su perfil. El usuario recibirá un correo con la nueva contraseña y desde el apartado de “Mi cuenta” podrá realizar el cambio. Para ello deberá cubrir el formulario de la *Ilustración 23: Recuperar contraseña*, y recibirá un correo electrónico similar al de la *Ilustración 24: Correo de recuperación de contraseña*.



The screenshot shows the 'Recuperar contraseña' (Recover password) page of the EventGO application. The page has a dark theme. At the top left is the 'EventGO' logo, and at the top right is a language selector set to 'Español'. The main content area is titled 'Recuperar contraseña' and contains a form with two fields: 'Login' and 'Correo electrónico'. Below the form are two buttons: 'Recuperar contraseña' (Recover password) and 'Cancelar' (Cancel). To the right of the form is a horizontal carousel of five images with labels: 'Viajes' (Travel), 'Música' (Music), 'Fiestas' (Parties), 'Deporte' (Sports), and 'Eventos' (Events). The bottom of the page features a copyright notice 'Copyright © 2023 Martín Gil Blanco - MEI' and social media icons for LinkedIn and MEI.

Ilustración 23: Recuperar contraseña

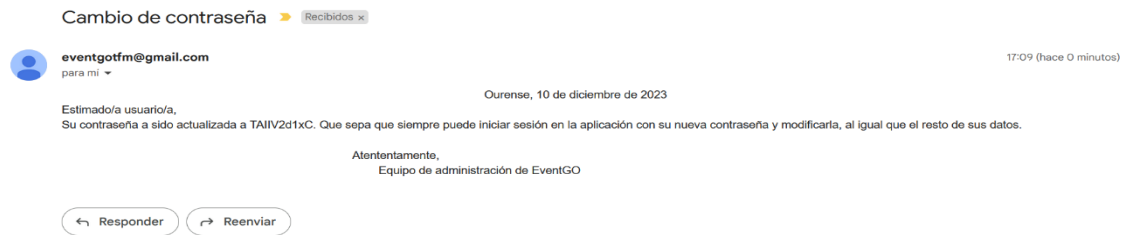


Ilustración 24: Correo de recuperación de contraseña

### 11.2.2 Gestión de usuarios

En los siguientes puntos se recogen las acciones de gestión que un administrador puede realizar sobre los usuarios. La vista de gestión de usuarios, como se puede apreciar en la *Ilustración 25: Gestión de usuarios*, recoge una tabla paginada de usuarios del sistema.

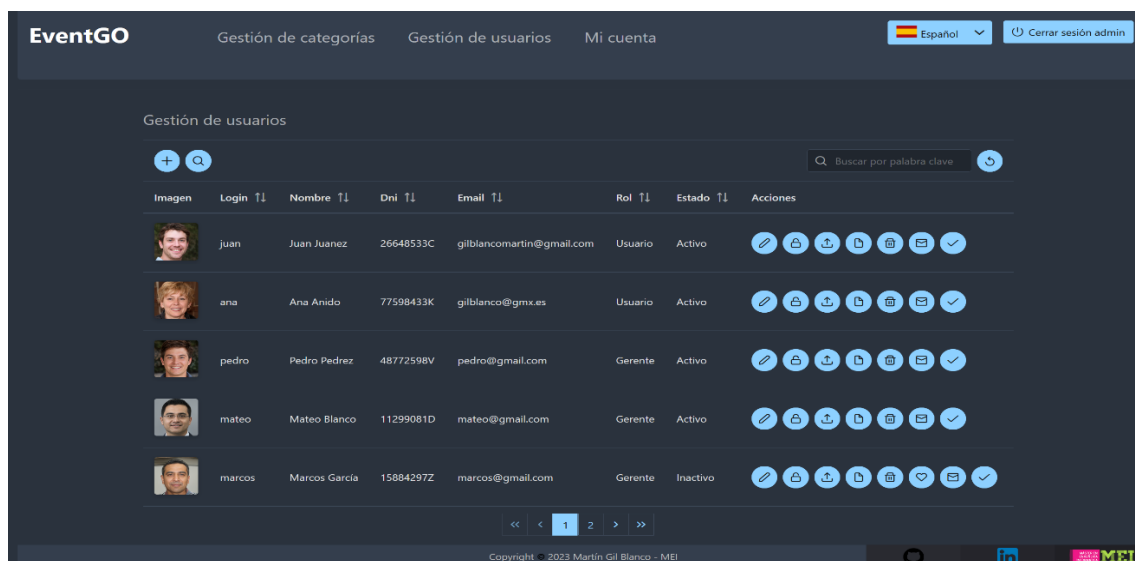


Ilustración 25: Gestión de usuarios

De forma análoga al resto de tablas de gestión, el usuario podrá:

- Crear elementos nuevos en el botón con el símbolo “+”.
- Buscar elementos en el sistema con el botón de símbolo “lupa”.
- Buscar elementos por una palabra clave.
- Recargar los elementos de la página si se ha aplicado algún filtro de búsqueda.

Podrá realizar acciones concretas sobre un elemento concreto. Estas se recogen en la siguiente imagen, si bien sobre casos concretos se pueden realizar más acciones.

- Editar los datos de un elemento en el botón del lápiz.
- Ver los datos del elemento en el botón con el símbolo del archivo.
- Eliminar el elemento en el botón con el símbolo de la papelera.
- Reactivar el elemento en el botón con el símbolo del corazón.

Todas estas acciones se realizan mediante ventanas modales que hacen más que hacen la más visual la presentación de los datos. En otros puntos de la aplicación existen otros botones diferentes, pero estos son los botones básicos de cualquier gestión en *EventGO*.



### 11.2.2.1 Crear usuario

Si nos fijamos en la siguiente *Ilustración 26: Crear usuario* podemos ver como obtenemos una ventana modal una vez clicado el botón con símbolo “+” donde insertar los datos de un nuevo usuario. Si la inserción es correcta, una vez pulsado el botón de “Crear” obtendremos una notificación de que el proceso se ha realizado correctamente similar a la *Ilustración 27: Acción realizada correctamente*.

Vas a crear un elemento en el sistema

Login

Contraseña

Nombre de usuario

Correo electrónico

Selecciona un rol

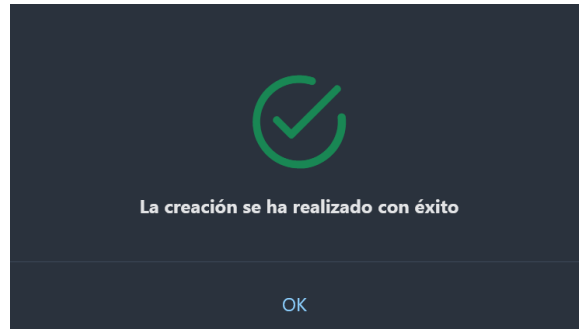
Dni

Fecha de nacimiento

Selecciona un país

+ Crear

*Ilustración 26: Crear usuario*



*Ilustración 27: Acción realizada correctamente.*

### 11.2.2.2 Buscar usuario

Si volvemos a observar *Ilustración 25: Gestión de usuarios*, el segundo botón de la lupa permitirá hacer una búsqueda entre los usuarios del sistema. En la *Ilustración 28: Buscar usuario*, vamos a realizar la búsqueda del usuario Juan en el sistema, una vez realizada la búsqueda, podemos pulsar en el botón de “recarga” para eliminar el filtro y volver a visualizar todos los usuarios. El resultado de la búsqueda lo podemos ver en la *Ilustración 29: Resultado de la búsqueda*.

Gestión de usuarios

Buscar elemento en el sistema

juan

Nombre de usuario

Correo electrónico

Selecciona un rol

Dni

Fecha de nacimiento

Selecciona un país

Activo/Inactivo

Buscar

Imagen Login Nombre Dni

juan Juan Juanez 2664853

ana Ana Anido 7759843

pedro Pedro Pedrez 4877259

mateo Mateo Blanco 1129908

Acciones

Buscar por palabra clave

*Ilustración 28: Buscar usuario*

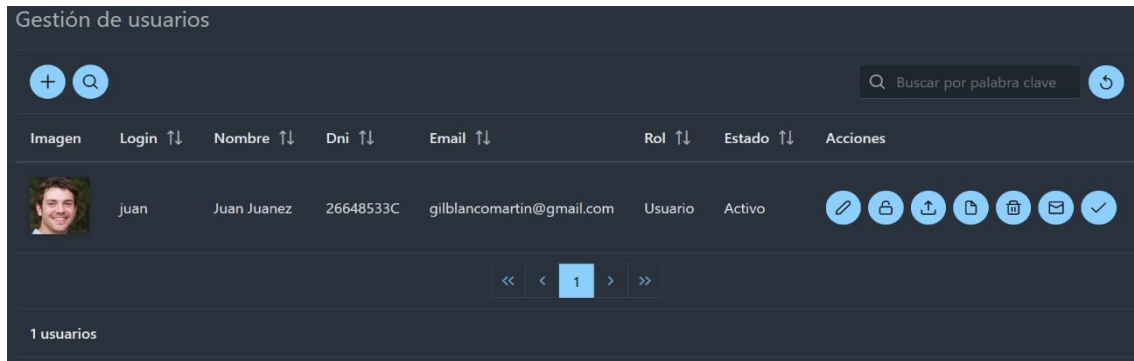


Ilustración 29: Resultado de la búsqueda

### 11.2.2.3 Editar usuario

Si volvemos a observar *Ilustración 25: Gestión de usuarios*, existe una columna de acciones donde se recogen las acciones concretas a realizar sobre un elemento del sistema. Si nos fijamos la primera opción de todas nos permite editar los datos del usuario, tal y como se muestra en la *Ilustración 30: Editar usuario*.

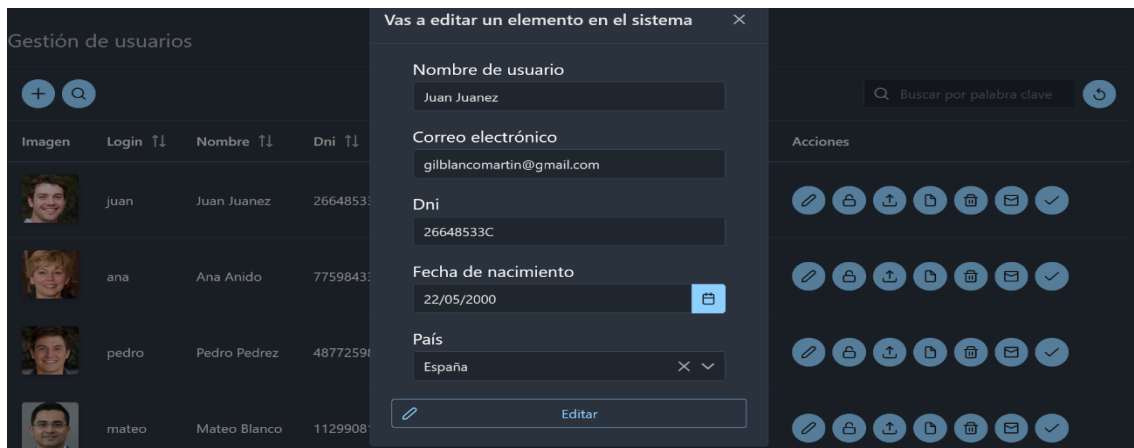


Ilustración 30: Editar usuario

### 11.2.2.4 Ver en detalle usuario

Si volvemos a observar *Ilustración 25: Gestión de usuarios*, existe una columna de acciones donde se recogen las acciones concretas a realizar sobre un elemento del sistema. Si nos fijamos la cuarta opción de todas nos permite ver los datos del usuario, tal y como se muestra en la *Ilustración 31: Ver en detalle usuario*.

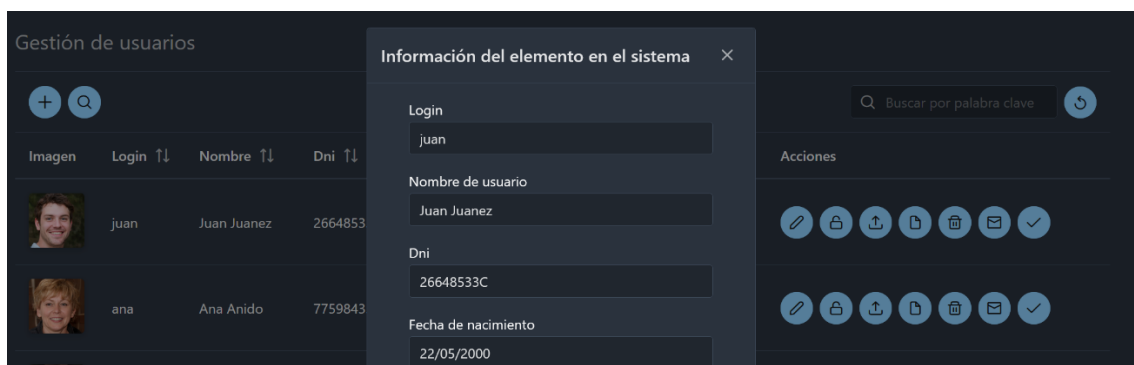


Ilustración 31: Ver en detalle usuario

### 11.2.2.5 Eliminar usuario

Si volvemos a observar *Ilustración 25: Gestión de usuarios*, existe una columna de acciones donde se recogen las acciones concretas a realizar sobre un elemento del sistema. Si nos fijamos la quinta opción de todas nos permite eliminar un usuario, tal y como se muestra en la *Ilustración 32: Eliminar usuario*.

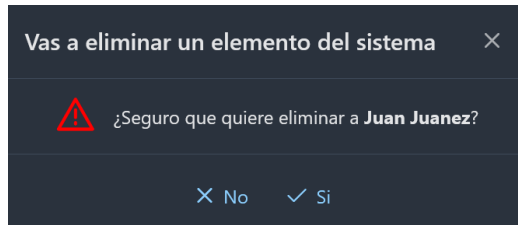


Ilustración 32: Eliminar usuario

Puede darse el caso de que el elemento en cuestión no se pueda eliminar una vez aceptada la eliminación, lo cual se le notificará al usuario. También puede darse el caso de que el usuario no se elimine de forma física, pasando a un estado inactivo como se puede ver en la *Ilustración 33: Usuario eliminado de forma lógica*, y se puede reactivar.










Imagen	Login ↑↓	Nombre ↑↓	Dni ↑↓	Email ↑↓	Rol ↑↓	Estado ↑↓	Acciones
	juan	Juan Juanes	26648533C	gilblancomartin@gmail.com	Usuario	Inactivo	       

Ilustración 33: Usuario eliminado de forma lógica

### 11.2.2.6 Reactivar usuario

Si volvemos a observar *Ilustración 25: Gestión de usuarios*, existe una columna de acciones donde se recogen las acciones concretas a realizar sobre un elemento del sistema. Si nos fijamos la sexta opción de todas nos permite reactivar un usuario, tal y como se muestra en la *Ilustración 34: Reactivar usuario*.

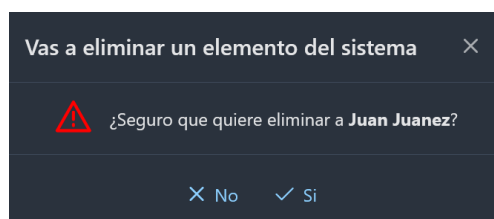


Ilustración 34: Reactivar usuario

Si confirmamos la reactivación el usuario volverá al estado activo, tal y como podemos ver en la *Ilustración 35: Usuario reactivado*, en a columna estado.










Imagen	Login ↑↓	Nombre ↑↓	Dni ↑↓	Email ↑↓	Rol ↑↓	Estado ↑↓	Acciones
	juan	Juan Juanes	26648533C	gilblancomartin@gmail.com	Usuario	Activo	       

Ilustración 35: Usuario reactivado



### 11.2.2.7 Otras opciones

Existen otras opciones específicas para la gestión de usuarios, estas son las siguientes. Si nos fijamos el tercer botón de acciones va a permitir subir una nueva fotografía para el usuario en nuestro sistema. De este modo podemos apreciar en la *Ilustración 36: Subir imagen de usuario*. El resultado final lo podemos apreciar en *Ilustración 37: Nueva imagen de usuario* E ilustración 38: *Nueva imagen de usuario desde Mi Cuenta*.

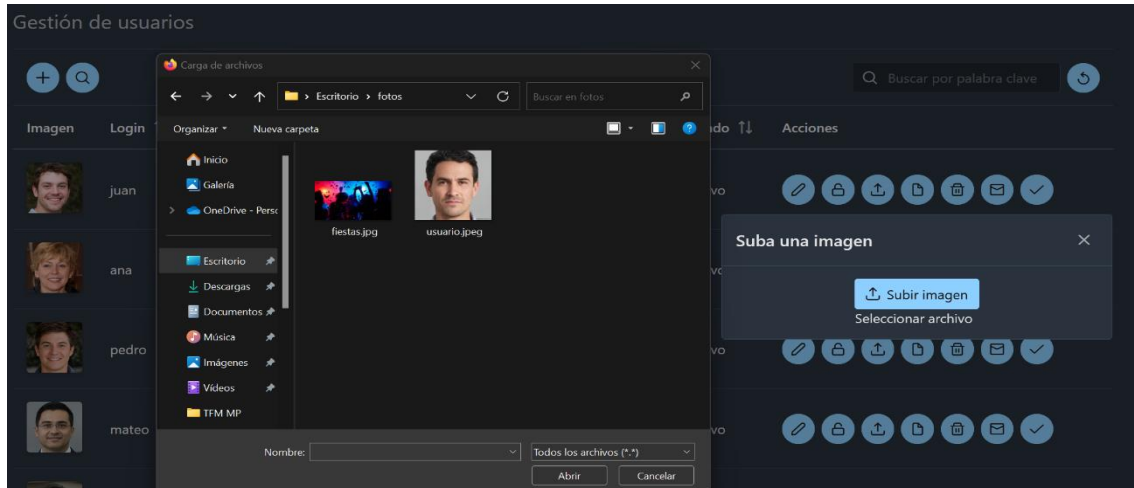


Ilustración 36: Subir imagen de usuario

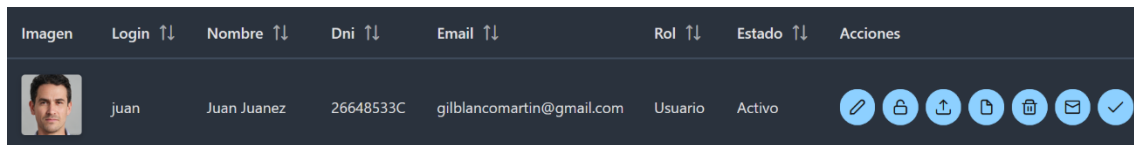


Ilustración 37: Nueva imagen de usuario

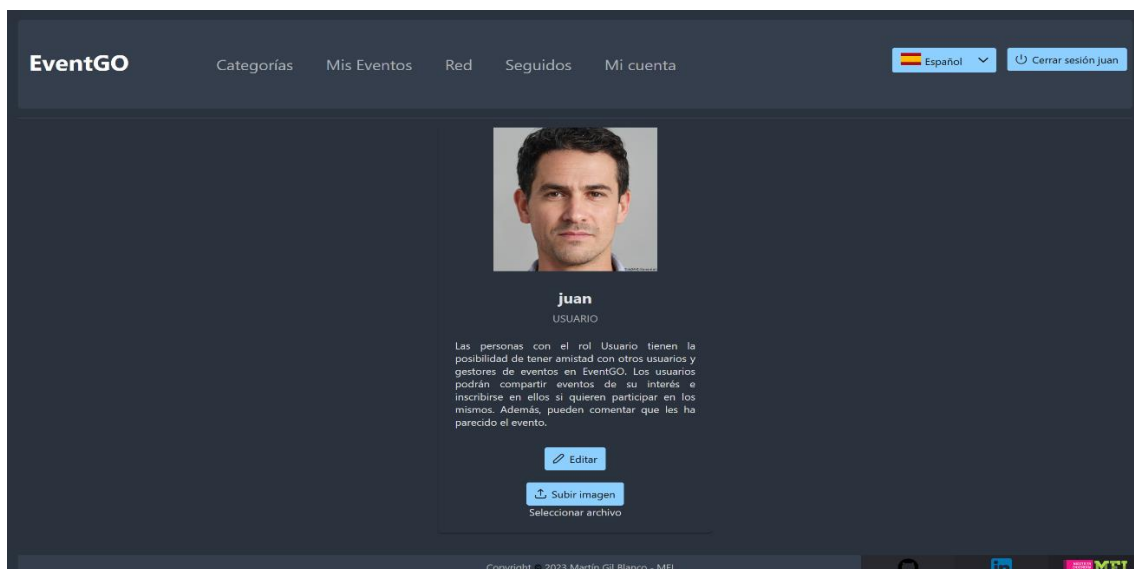


Ilustración 38: Nueva imagen de usuario desde Mi Cuenta

A mayores, existen otras dos opciones que corresponden a los dos últimos botones de cada elemento. Estos dos botones van a permitir ver las solicitudes y suscripciones de cada usuario respectivamente. Más adelante hablaremos de estos dos aspectos.

### 11.2.3 Gestión de categorías

La gestión de las categorías dentro de *EventGO* es responsabilidad del administrador del sistema. Tal y como se puede apreciar en la *Ilustración 39: Gestión de categorías*, la gestión de las categorías consta de las acciones comunes a cualquier gestión de las vistas anteriormente. A mayores, el último botón de cada tupla permite acceder a la gestión de los eventos en esa categoría.




Imagen	Nombre ↑↓	Descripción ↑↓	Estado ↑↓	Acciones
	Música	Encontrarás eventos musicales y mucho más.	Activo	
	Artes escénicas	Encontrarás eventos relacionados con cine, teatro o artes visuales.	Activo	
	Deportes	Encontrarás eventos y actividades deportivas.	Activo	
	Gastronomía y fiestas	Encontrarás eventos y actividades deportivas.	Activo	
	Negocios	Encontrarás eventos de negocios, charlas o talleres de finanzas.	Activo	

9 categorías

Ilustración 39: Gestión de categorías

### 11.2.4 Gestión de eventos

La gestión de los eventos dentro de *EventGO* es responsabilidad tanto del administrador del sistema, el cual podrá actuar sobre cualquier evento y de los gerentes, quienes podrán actuar sobre los eventos que hayan creado. Tal y como se puede apreciar en la *Ilustración 40: Gestión de eventos como administrador*, la gestión de los eventos consta de las acciones comunes a cualquier gestión de las vistas anteriormente. Es conveniente indicar que los dos últimos botones de acciones permiten ver un documento de interés del evento (ya que es posible almacenar PDFs de interés para los usuarios), así como ver los comentarios de los usuarios tal y como se puede apreciar en la *Ilustración 41: Comentarios de un evento*.



Imagen y enlace	Nombre ↑↓	Tipo de asistencia ↑↓	Tipo de asistencia ↑↓	Fecha de inicio ↑↓	Dirección ↑↓	Estado ↑↓	Acciones
	Concierto C.Tangana	Abierto	Público	25/11/2024	Av. de Concha Espina, 1, 28036 Madrid	Activo	
	Concierto Beret	Abierto	Público	25/12/2024	Av. de Rodolfo García, 1, 24167 Madrid	Activo	

2 eventos

Ilustración 40: Gestión de eventos como administrador



Ilustración 41: Comentarios de un evento

Los usuarios del sistema podrán ver los comentarios y el nivel de valoraciones que hay para un determinado evento en cuestión. En el caso de la imagen anterior, podemos apreciar que solo existe un comentario en el sistema y que la valoración es de cinco estrellas. Las valoraciones de los usuarios también se ven reflejadas en un gráfico tal y como podemos apreciar en la imagen anterior.

A mayores los usuarios gerentes pueden gestionar los eventos que han sido creados por ellos. Como podemos apreciar en la *Ilustración 42: Gestión de eventos como gerente*, existen otras dos opciones para la gestión de las solicitudes y suscripciones de usuarios al evento (más adelante se tratará esta cuestión).

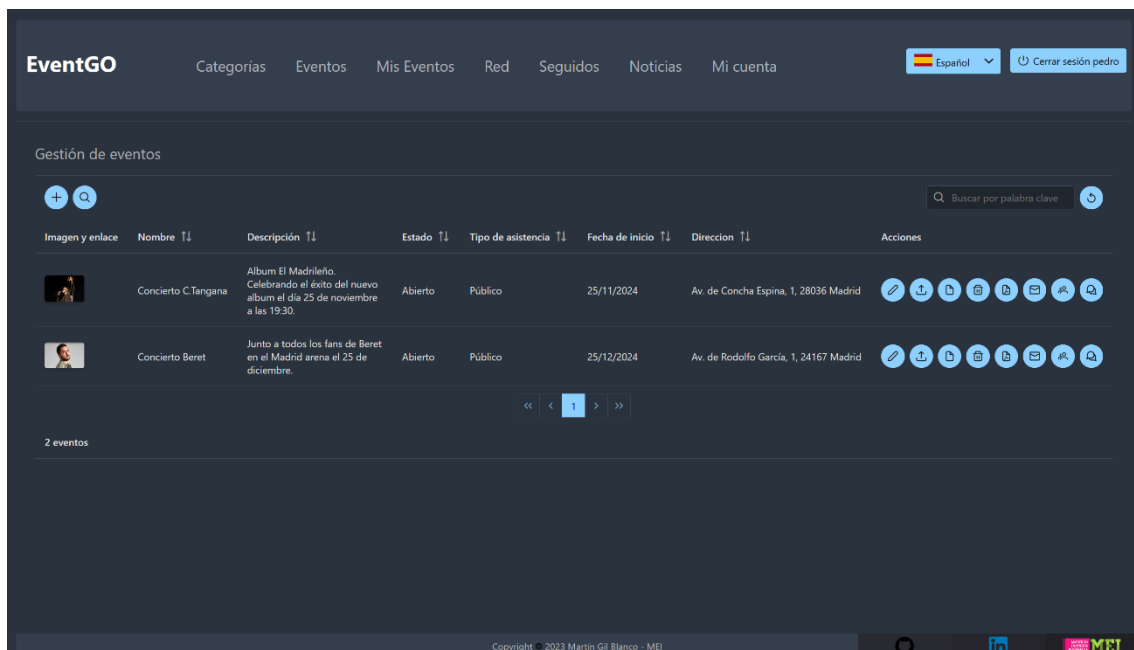


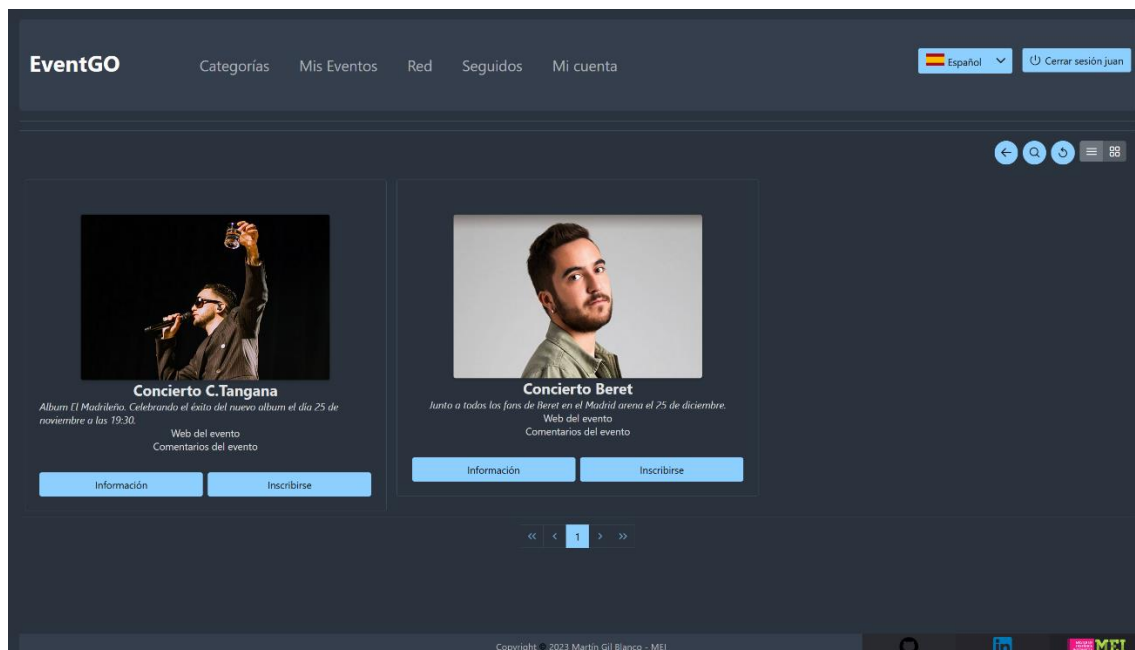
Ilustración 42: Gestión de eventos como gerente

### 11.2.4 Gestión de solicitudes y suscripciones

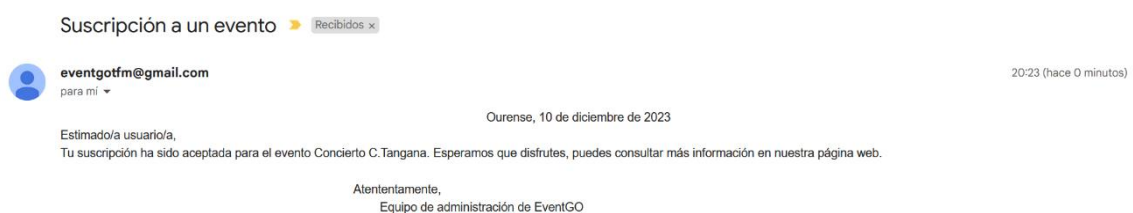
Los usuarios podrán navegar por los eventos que existen dentro de las categorías, podrán visitar la web del evento y los comentarios y valoraciones si existen. Si un usuario decide inscribirse a un evento se contemplan las siguientes casuísticas.

Un usuario solo podrá inscribirse en un evento si este evento está en estado *abierto*, no está borrado de forma lógica y su fecha de celebración es superior a la fecha actual. Dentro de *EventGO* se distinguen eventos *Privados* y *Público*. Un evento privado es aquel en el cual un usuario solicita inscribirse en el evento, creándose una solicitud para participar en este. Una vez creada la solicitud es el gerente (o administrador) quien debe aceptar o rechazar la solicitud, si se acepta la solicitud se crea una suscripción al evento y se notifica por correo electrónico al usuario. Los eventos públicos no necesitan una aceptación por parte del gerente, este proceso se realiza automáticamente, si bien el gerente o el administrador pueden eliminar una suscripción de un evento, al igual que el usuario podrá eliminar sus solicitudes y suscripciones.

En la siguiente *Ilustración 43: Inscripción en evento público*, podemos ver los eventos que existen dentro de la categoría de *Música*, en la cual se encuentran eventos musicales. El primer evento es *Concierto C.Tangana*, el cual es un evento público, si el usuario se inscribe en el evento se crea una suscripción a este y le llega una notificación a su correo como puede apreciar en la *Ilustración 44: Correo de suscripción a evento*.



*Ilustración 43: Inscripción en evento público*



*Ilustración 44: Correo de suscripción a evento*

Como usuario, si se dirige al apartado de “Mis Eventos” podrá acceder a sus suscripciones y solicitudes, tal y como puede apreciar en la *Ilustración 45: Vista de Mis Eventos*. Si accede al apartado de suscripciones podrá ver que existe una suscripción realizada en este momento para el evento, esto se aprecia en la *Ilustración 46: Suscripción a evento*.



Ilustración 45: Vista de Mis Eventos



Ilustración 46: Suscripción a evento

Como se puede apreciar, se ha creado la suscripción sobre el evento y en el apartado de “Acciones” tenemos dos nuevos botones que nos van a permitir crear un nuevo comentario del evento y generar un PDF sobre nuestros datos y el evento al que estamos suscritos (este PDF se generará en el idioma seleccionado en la aplicación y se incluye en el apartado de Anexo). A continuación, podemos crear un comentario como se puede apreciar en la *Ilustración 47: Crear comentario* y ver este nuevo comentario en el evento en cuestión, esto se puede observar en la *Ilustración 48: Comentarios de un evento 2*.

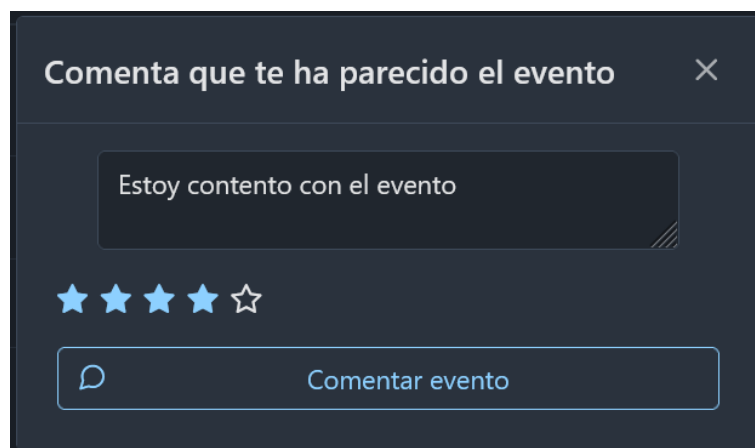


Ilustración 47: Crear comentario



Ilustración 48: Comentarios de un evento 2

Por otra parte, tenemos el caso de los eventos privados, cuyas solicitudes necesitan una aprobación por parte del gerente del evento. A continuación, vamos a inscribirnos a un evento privado como se muestra en la *Ilustración 49: Inscripción en evento privado*. Vamos a inscribirnos al evento Romeo y Julieta y se va a generar una solicitud del evento.

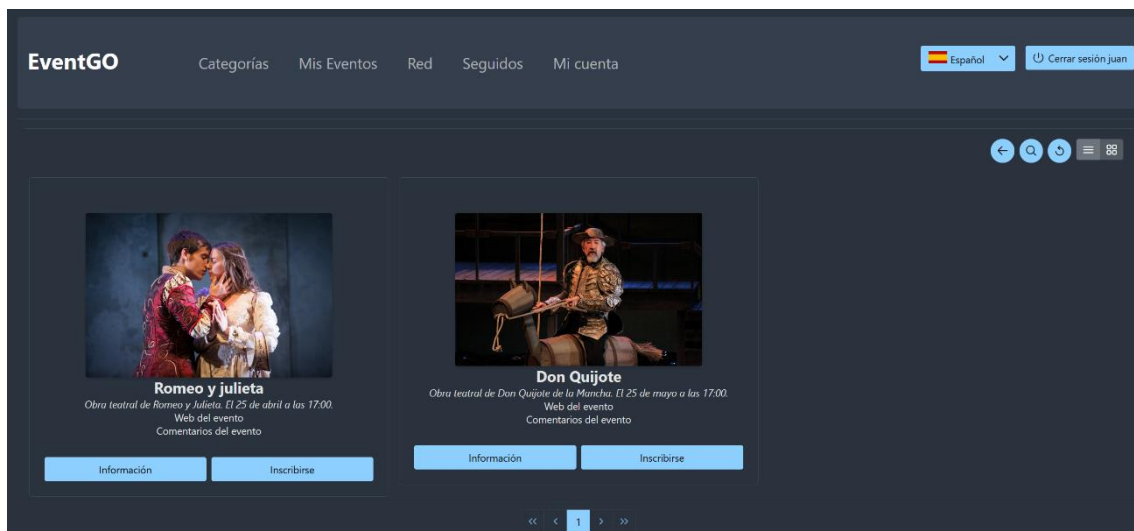


Ilustración 49: Inscripción en evento privado

Como podemos apreciar en la *Ilustración 50: Solicitud a evento privado*, si nos dirigimos al apartado de solicitudes vemos que se ha generado una nueva solicitud a un evento en el sistema.






Gestión de solicitudes						
 		<input type="text" value="Buscar por palabra clave"/> 				
Imagen y enlace	Evento ↑↓	Fecha de la solicitud ↑↓	Fecha de inicio ↑↓	Dirección ↑↓	Email ↑↓	Teléfono ↑↓ Acciones
	Romeo y Julieta	10/12/2023	25/4/2024	Teatro del Mar, 1, 28036 Barcelona	romeoJulieta@gmail.com	698726119 

Ilustración 50: Solicitud a evento privado

A continuación, el gerente tendrá que aprobar nuestra solicitud para que este pase a ser una suscripción, esto se puede apreciar en la *Ilustración 51: Gestión de eventos del gerente* y en la *Ilustración 52: Aprobación de solicitud*.









EventGO							
Categorías	Eventos	Mis Eventos	Red	Seguidos	Noticias	Mi cuenta	<span>🇪🇸 Español</span> <span>Cerrar sesión mateo</span>
Gestión de eventos							
 		<input type="text" value="Buscar por palabra clave"/> 					
Imagen y enlace	Nombre ↑↓	Descripción ↑↓	Estado ↑↓	Tipo de asistencia ↑↓	Fecha de inicio ↑↓	Dirección ↑↓	Acciones
	Romeo y Julieta	Obra teatral de Romeo y Julieta. El 25 de abril a las 17:00.	Abierto	Privado	25/4/2024	Teatro del Mar, 1, 28036 Barcelona	     
	Don Quijote	Obra teatral de Don Quijote de la Mancha. El 25 de mayo a las 17:00.	Abierto	Privado	25/5/2024	Palau Sant Jordi, 1, 28194 Barcelona	     
<span>&lt;&lt; &lt; 1 &gt; &gt;&gt;</span>							
2 eventos							

Ilustración 51: Gestión de eventos del gerente

Podemos apreciar como en la imagen anterior el gerente de los eventos puede acceder en los tres últimos botones de la columna acción a la gestión de solicitudes, gestión de suscripciones y a los comentarios del evento. Si accedemos a la gestión de solicitudes se podrá aprobar la solicitud y se notificará por correo electrónico de ello al usuario. Esto se puede apreciar en las *Ilustración 53: Correo de suscripción a evento 2*.

Gestión de solicitudes					
 		<input type="text" value="Buscar por palabra clave"/> 			
Imagen	Nombre de usuario ↑↓	Fecha de la solicitud ↑↓	Fecha de inicio ↑↓	Email ↑↓	Acciones
	Juan Juanes	10/12/2023	25/4/2024	gilblancomartin@gmail.com	 
<span>&lt;&lt; &lt; 1 &gt; &gt;&gt;</span>					
1 solicitudes					

Ilustración 52: Aprobación de solicitud

 eventgotfm@gmail.com  
para mí ▼

Ourense, 10 de diciembre de 2023

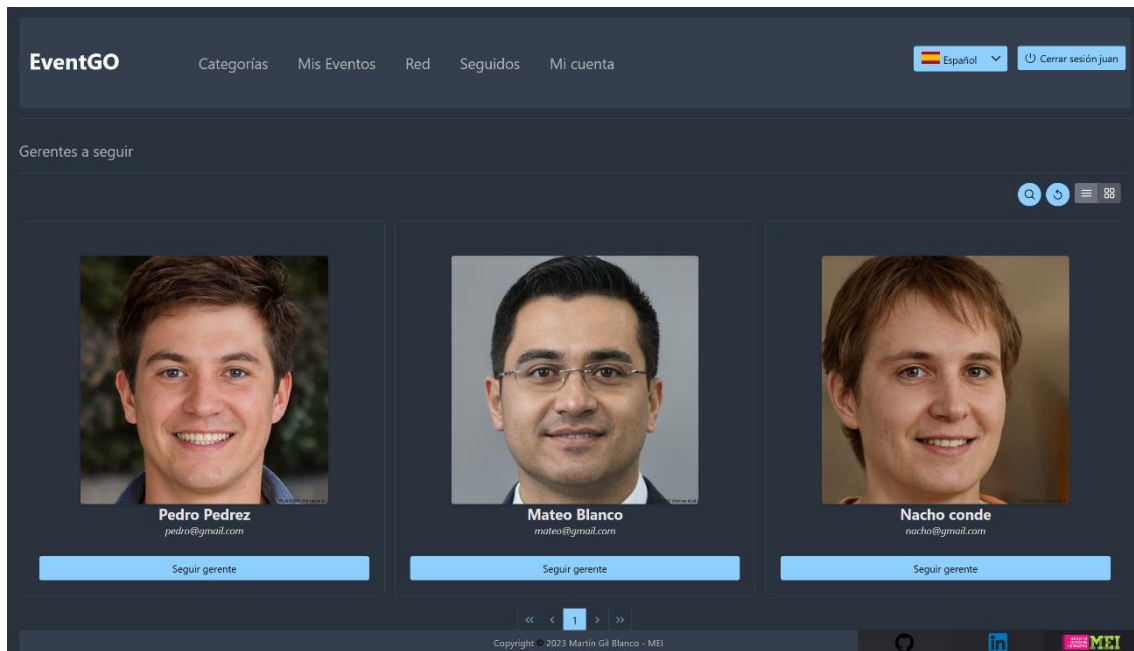
Estimado/a usuario/a,  
Tu suscripción ha sido aceptada para el evento Romeo y Julieta. Esperamos que disfrutes, puedes consultar más información en nuestra página web.

Ilustración 53: Correo de suscripción a evento 2

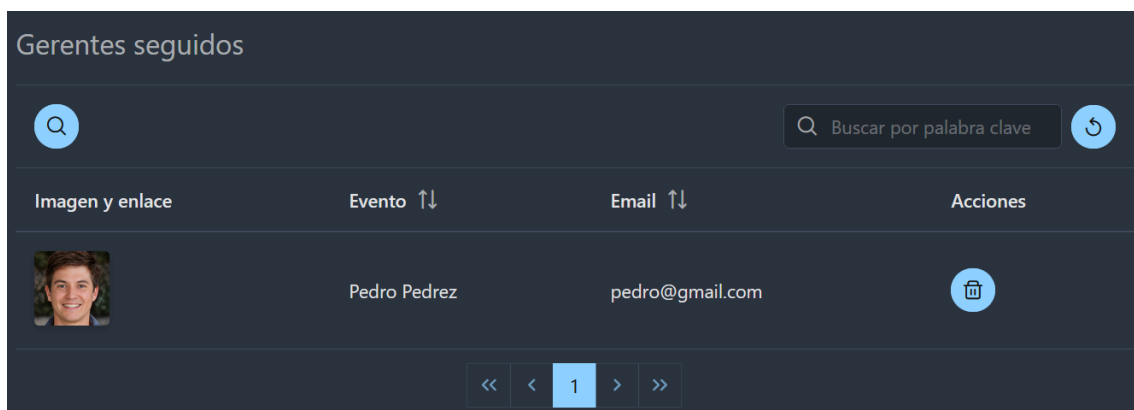


### 11.2.5 Gestión de amistades y noticias

Los usuarios y los gerentes pueden seguir a otros gerentes de su interés para estar al tanto de novedades que estos quieran compartir, de este modo, cualquier cuestión que sea interesante será notificada por correo electrónico al usuario seguidor. En la *Ilustración 54: Gerentes a seguir*, un usuario puede ver que gerentes existen en el sistema de los cuales está interesado en recibir notificaciones, una vez se sigue al gerente, este aparecerá en el espacio de “Seguidos”, desde el cual se podrán gestionar las amistades establecidas, tal y como se ve reflejado en la *Ilustración 55: Gerentes seguidos*.



*Ilustración 54: Gerentes a seguir*

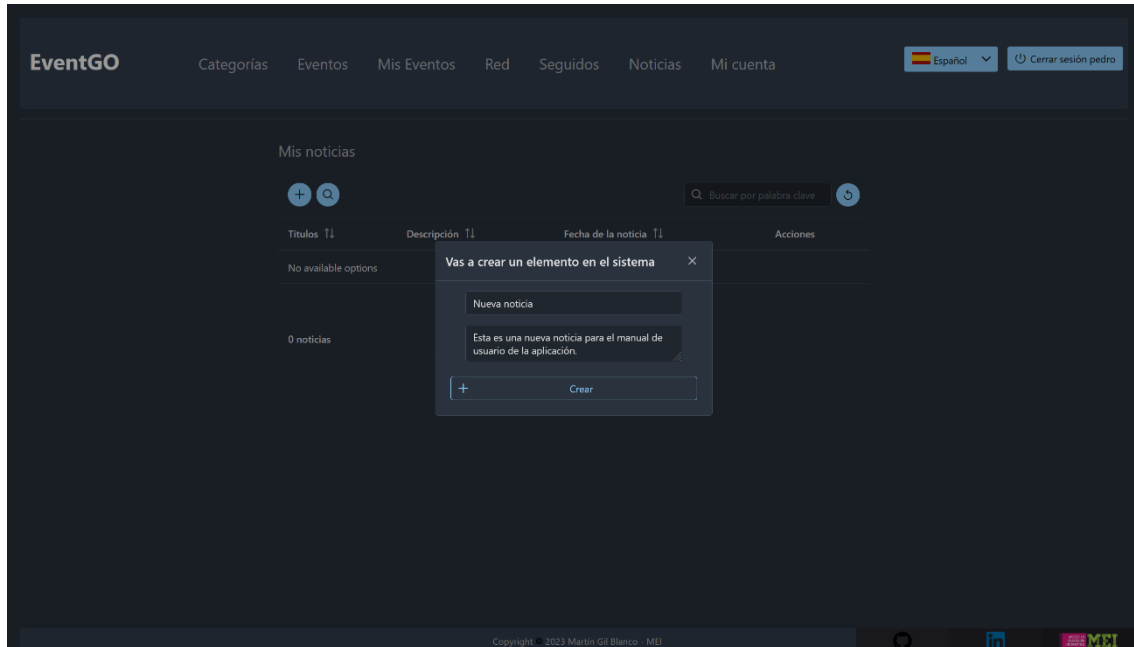


*Ilustración 55: Gerentes seguidos*

En estos momentos, el usuario Juan está siguiendo al gerente Pedro, por lo que este recibirá notificaciones por correo electrónico.

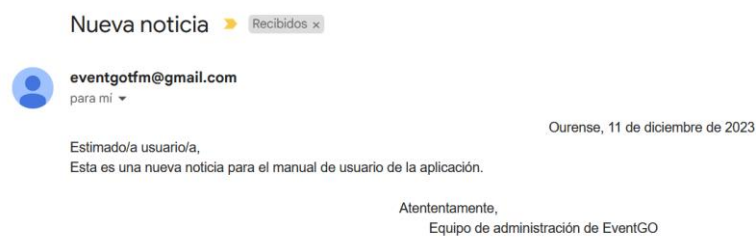


En la siguiente *Ilustración 56: Gestión de noticias*, el gerente puede crear nuevas noticias para compartir con los usuarios que le siguen. Vamos a crear una noticia y como vamos a poder apreciar en las *Ilustración 57: Correo de noticia 1* y en la *Ilustración 58: Correo de noticia 2*, los usuarios reciben una notificación de la noticia en cuestión.



*Ilustración 56: Gestión de noticias*

Dos usuarios diferentes seguían a Pedro para poder estar recibiendo sus notificaciones, al publicarse la noticia, ambos los usuarios han recibido el aviso a sus respectivos correos electrónicos.



*Ilustración 57: Correo de noticia 1*



*Ilustración 58: Correo de noticia 2*

## 12. Principales aportaciones

Este proyecto ha permitido obtener una aplicación web con las principales funciones relacionadas con la gestión de eventos para múltiples usuarios dentro del sistema.

Se ha conseguido implementar una aplicación que concentra todas las funcionalidades contempladas en la planificación y que permite la incorporación de nuevos desarrollos futuros. *Objetivo 1.*

Los usuarios que hagan uso de la aplicación podrán realizar cualquier acción en función de su rol de forma sencilla, amigable e intuitiva dentro de la web. *Objetivo 2.*

Se ha desarrollado una aplicación con autenticación JWT que contempla la existencia de distintos roles de usuarios con funcionalidades diferentes en función de este. *Objetivo 3.*

Se dispone dentro de la aplicación de un conjunto de categorías de eventos donde agrupar los eventos por parte de los gerentes en función de la naturaleza del evento en cuestión. *Objetivo 4.*

Los gerentes podrán publicar nuevos eventos dentro de estas categorías para que otros usuarios interesados puedan inscribirse en ellos. *Objetivo 5.*

Dentro de la aplicación es posible visualizar sin estar autenticado en el sistema las categorías como eventos existentes, esto hace que la aplicación sea más abierta al público ya que pueden realizar ciertas acciones sin estar registrados. *Objetivo 6.*

En función de si el evento es público (cualquier persona se suscribe automáticamente) o si es privado (el gerente tiene que permitir la solicitud del usuario) los usuarios podrán inscribirse en estos. *Objetivo 7.*

Un usuario interesado en recibir avisos de un gerente podrá localizarlo dentro del sistema y seguirlo para estar más informado. *Objetivo 8.*

Los gerentes podrán compartir novedades con sus seguidores en forma de noticias. Estas noticias serán notificadas a los usuarios de la red del gerente por correo electrónico. *Objetivo 9.*

Un usuario que está suscrito a un evento podrá generar un PDF con los datos del evento al que está suscrito y sus datos personales, pudiendo servir como documento identificativo de su suscripción. *Objetivo 10.*

Los usuarios suscritos a un evento pueden publicar comentarios sobre su nivel de agrado. Estos comentarios serán vistos por otros usuarios y por el gerente, sirviendo de indicativo sobre su grado de satisfacción. Además, se generará un gráfico en base a las valoraciones de los usuarios dando una mejor representación de su nivel de satisfacción. *Objetivo 11.*

La información que se crea en el sistema puede ser localizada, modificada o eliminada por los usuarios con los permisos adecuados para ello o por el administrador del sistema. *Objetivo 12.*

## 13. Conclusiones

A continuación, se recogen las conclusiones obtenidas de la realización de este proyecto. Estas conclusiones se encuentran divididas en un apartado de conclusiones personales, donde se habla de las percepciones a nivel individual de carácter subjetivo y técnicas, donde se abordan aspectos más específicos del proyecto.

### 13.1 Conclusiones técnicas

Este proyecto ha supuesto un esfuerzo considerable a pesar del cual estoy contento, ya que se han buscado soluciones a múltiples dificultades complejas y ha servido para comprender conceptos no suficientemente bien interiorizados o que se desconocían.

En términos de tecnologías, ha pretendido abarcar un espectro lo más amplio posible, permitido trabajar con múltiples herramientas avanzadas en el desarrollo de aplicaciones modernas. Se pueden citar, el uso de JPA, fundamental para la gestión de la capa de persistencia, permitiendo una mejor interacción entre aplicación y base de datos. Por otra parte, en el ámbito de la seguridad y autenticación, la implementación JSON Web Tokens, también conocido por sus siglas JWT, ha dado al proyecto una capa adicional de seguridad. El uso de JWT mejora la protección de los recursos y permite una gestión eficiente de las sesiones en el sistema por parte de los usuarios.

A mayores, la realización de este proyecto ha permitido la utilización de múltiples tecnologías como Java, Spring, JavaScript y bibliotecas como React o PrimeReact. Herramientas que, si bien se habían utilizado con anterioridad, no se contaba con un alto grado de dominio que este proyecto ha permitido adquirir.

El proyecto tiene un enfoque global en todas las etapas del desarrollo de una aplicación web, perfeccionando mi comprensión sobre Frontend, Backend, ApiRest, entre muchos otros conceptos, comprendiendo las interacciones entre ellos y gestión de su funcionamiento.

### 13.2 Conclusiones personales

Este proyecto ha sido desarrollado con la finalidad de crear una web que pudiera ser empleada en un entorno real. Con este objetivo se ha llevado a cabo una especificación de requisitos completa y establecimiento de términos necesarios del diseño de la aplicación y funcionalidades.

Con el proceso llevado a cabo he podido profundizar en cómo sería una interacción real con los stakeholders, así como el proceso de análisis de las necesidades de estos. Por otra parte, este proyecto me ha permitido mejorar en el establecimiento de límites en cuanto al alcance del proyecto. Además, he profundizado mis conocimientos de sobre el manejo de Frameworks como Spring o librerías como React y ampliado mis conocimientos en desarrollo y diseño web.

## 14. Vías de trabajo futuro

En relación con vías de trabajo futuro podemos hablar de la posibilidad de desplegar la aplicación en un hosting, ya que existen múltiples plataformas como Azure, AWS o Google Cloud que ofrecen diferentes planes para ello. Además, sería lógico la adopción de un certificado SSL/TLS que permitiera el tráfico HTTPS.

En este proyecto se hace uso de Apache Tomcat embebido que proporciona por defecto el proyecto Spring Boot, este es un contenedor de servlets y sería interesante valorar si el despliegue de la aplicación en lugar de un Tomcat embebido en otro tipo de Servidor de aplicaciones Java/Jakarta EE como WildFly/JBoss o Payara, ya que estos disponen de mejores capacidades de escalado y balanceo de carga.

A nivel de funcionalidad dentro de *EventGO*, sería interesante que los gestores pudieran crear categorías, siendo estos responsables de la gestión de las mismas y de las categorías que estudiaran en niveles inferiores. Esto haría la aplicación más completa y contemplaría nuevas casuísticas como la necesidad de tener un árbol de categorías, lo cual permitiría clasificar de una forma mucho más detallada los eventos.

## 15. Referencias

- Camacho, J. D. (2020). Desarrollo en Cascada (Waterfall) vs Desarrollo Agile-SCRUM. *Northware*. Obtenido de <https://www.northware.mx/wp-content/uploads/2021/04/desarrollo-cascada-waterfall-vs-desarrollo-agile-scrum.pdf>
- Diseño básico de páginas web en HTML*. (2009). Málaga: Vértice. Obtenido de <https://books.google.es/books?hl=es&lr=&id=Q4VL8Tiy7gcC&oi=fnd&pg=PP2&dq=HTML&ots=Vl32BZbrrh&sig=kdpEM051WcxORZoerBxW9w1OtUw#v=onepage&q=HTML&f=false>
- Durango, A. (2015). *Diseño web con CSS*. Obtenido de <https://books.google.es/books?hl=es&lr=&id=bCXsCgAAQBAJ&oi=fnd&pg=PA11&dq=CSS&ots=mKto0PTsW8&sig=ssHTk7UepOqhN5tjiUx02a3bHUE#v=onepage&q=CSS&f=false>
- Eckel, B. (2010). *Thinking in Java*. Obtenido de <http://ftp-developpez.com/bruce-eckel/java/tij3/tij3.pdf>
- Escofet, C. M. (2002). *El lenguaje SQL*. UOC. Obtenido de <https://pdfcursos.com/pdf/0018-lenguaje-sql.pdf>
- Fernando Pech-May, M. A.-R.-D. (s.f.). Desarrollo de Aplicaciones web con JPA, EJB, JSF y PrimeFaces. Obtenido de [https://d1wqtxts1xzle7.cloudfront.net/45290069/Desarrollo\\_de\\_Aplicaciones\\_web\\_con\\_JPA\\_EJB\\_JSF\\_y\\_PrimeFaces-libre.pdf?1462230083=&response-content-disposition=inline%3B+filename%3DDesarrollo\\_de\\_Aplicaciones\\_web\\_con\\_JPA\\_EJB.pdf&Expires=1702166415&Signature=](https://d1wqtxts1xzle7.cloudfront.net/45290069/Desarrollo_de_Aplicaciones_web_con_JPA_EJB_JSF_y_PrimeFaces-libre.pdf?1462230083=&response-content-disposition=inline%3B+filename%3DDesarrollo_de_Aplicaciones_web_con_JPA_EJB.pdf&Expires=1702166415&Signature=)
- Gonzales, A., Lizama, O., Kindley, G., & Jeria Morales, J. (2016). Redes de computadores Arquitectura Cliente-Servidor. *Universidad Técnica Federico Santa María*. Obtenido de <http://profesores.elo.utfsm.cl/~agv/elo322/1s16/projects/reports/Proyecto%20Cliente%20-%20Servidor.pdf>
- Mark Fisher, M. B. (2018). *Spring Integration Reference Manual*. Obtenido de <https://docs.spring.io/spring-integration/docs/5.1.1.RELEASE/reference/pdf/spring-integration-reference.pdf>
- React, D. (2016). *Javi Jimenez Villar*. Leanpub. Obtenido de [https://d1wqtxts1xzle7.cloudfront.net/60098162/Descubre\\_React\\_\\_2da\\_Edicion\\_-\\_Javi\\_Jimenez-FREELIBROS20190723-17083-qn5tuy-libre.pdf?1563948300=&response-content=](https://d1wqtxts1xzle7.cloudfront.net/60098162/Descubre_React__2da_Edicion_-_Javi_Jimenez-FREELIBROS20190723-17083-qn5tuy-libre.pdf?1563948300=&response-content=)

disposition=inline%3B+filename%3DDescubre\_React.pdf&Expires=170216653  
0&Signature=VostU9rO9-SyW

## 16. Anexo

### Datos del evento

**EventGO**

- Usuario inscrito: juan
- Nombre de la categoría: Música
- Nombre del evento: Concierto C.Tangana
- Plazas del evento: 30000
- Número de inscritos: 0
- Teléfono de contacto: 698726119
- Email de contacto: cTanganaConcierto@gmail.com
- Tipo de asistencia: PUBLICO
- Descripción del evento:

Album El Madrileño. Celebrando el éxito del nuevo album el día 25 de noviembre a las 19:30.



---

Copyright 2023 Martín Gil Blanco - MEI

*Ilustración 59: Acreditación de una suscripción en un evento*