

# VerEventos

2021-2022

Proyecto final de grado



FAMILIA INFORMÀTICA I COMUNICACIONS



Juan Raul Nan  
Desarrollo de aplicaciones web  
IES La Vereda  
Tutor: Enrique Tortajada González

## Contenido

1. INTRODUCCIÓN.....	2
1.1. MÓDULOS A LOS QUE IMPLICA.....	2
1.2. BREVE DESCRIPCIÓN DEL PROYECTO.....	3
2. ESTUDIO PREVIO.....	4
2.1. ESTUDIO SOLUCIONES EXISTENTES.....	4
3. PLANIFICACIÓN DE TRABAJO.....	5
4. DISEÑO.....	6
4.1. FRONT-END.....	6
4.2. Back-end.....	17
4.2.1. Patrón MVC (Modelo Vista Controlador).....	17
4.2.2. Symfony.....	18
4.2.3. Doctrine.....	18
4.2.4. Composer.....	19
4.2.5. Twig.....	19
4.2.6. MariaDB.....	19
4.2.7. Partes de código, estructuración de archivos y base de datos.....	19
5. RECURSOS.....	27
6. CONCLUSIONES.....	28
7. ANEXOS COMPLEMENTARIOS.....	29
7.1. Manual de instalación y despliegue de la aplicación.....	29
7.2. Explicación del contenido ubicado en el repositorio GitHub.....	30
8. BIBLIOGRAFÍAS.....	31
LEYENDA DE IMÁGENES.....	32
RESUMEN DEL CONTENIDO.....	33

## 1. INTRODUCCIÓN

### 1.1. MÓDULOS A LOS QUE IMPLICA

Estos son los módulos del curso de Desarrollo de Aplicaciones Web que han sido implicado en el proyecto.

Desarrollo web en entorno servidor (PHP):

- Desarrollar páginas web.
- Utilizar la programación basada en lenguajes de marcas con código embebido, para ello se nos ha facilitado el gestor de plantillas “Twig”.
- Interpretación y manipulación de la información proporcionada por el usuario para la generación de páginas web interactivas, sobre todo por medio de formularios.
- Utilización de diversas técnicas para el acceso y modificación de datos de forma segura. En este proyecto, únicamente he utilizado una forma de acceso a datos, pero conocer otras, me ha servido para elegir la más adecuada y eficaz.

Base de datos:

- Interpretar el diseño lógico de bases de datos, analizando y cumpliendo las especificaciones relativas a su aplicación.
- Seleccionar solo los datos necesarios para la creación de tablas.
- Visualizar las relaciones que existen entre las distintas tablas de una base de datos.
- Generar, extraer, modificar y eliminar datos de forma correcta y eficaz.

Lenguaje de marcas (HTML Y CSS):

- Interpretar lenguajes de marcas reconociendo sus principales características e identificar sus elementos.
- Utilizar lenguajes de marcas para la transmisión de información a través de la web.
- Agilizar la aplicación de estilos a una página web mediante lenguajes de marcas.

Diseño de interfaces web (BOOTSTRAP):

- Planificar la creación de una interfaz web valorando y aplicando especificaciones de diseño.
- Creación de interfaces web homogéneas, fáciles, intuitivos y fáciles de aprender por parte del usuario.
- Preparación e integración de archivos multimedia para la web.

- Conocimiento y manejo del “framework” “Bootstrap” para la gestión de “CSS”

## 1.2. BREVE DESCRIPCIÓN DEL PROYECTO

El objetivo del proyecto consta en desarrollar una página web, combinando distintas tecnologías, para llevar a cabo de manera pragmática la visualización, la recopilación y el manejo de datos. No consta de ningún requisito previo para visualizar el contenido de la web, o casi todo. Para la interacción de los usuarios con los eventos, tanto como para crearlos, como para valorarlos, sí que existe un requisito previo, y es iniciar sesión en la aplicación, habiendo creado previamente una cuenta mediante un correo electrónico.

Es una aplicación que se encarga de mostrar a los usuarios que visitan la web, los eventos publicados por las organizaciones que se han registrado en la aplicación. Su contenido principal se divide en 3 pestañas: inicio, eventos, y organizaciones, que se muestran en todo momento en el menú de navegación.

## 2. ESTUDIO PREVIO

La aplicación pretende cubrir la necesidad de publicitación de los distintos eventos que las empresas quieren organizar, para comunicar al resto de personas, las diversas características de sus acontecimientos, como por ejemplo: de qué se trata dicho acto, la fecha, el lugar, de qué se trata...

Habitualmente, las empresas utilizan las redes sociales para la publicación de sus eventos, pero, en las redes sociales, la información que llega a los receptores es heterogénea, ya que no suelen ser plataformas enfocadas únicamente a la difusión de contenido por parte de las empresas, sino que también comparten el canal con otras personas que no necesariamente están promoviendo sus eventos empresariales.

Por estos motivos, la aplicación está enfocada en hacer que la información divulgada por las empresas registradas en la plataforma llegue a los usuarios de forma directa y homogénea.

### 2.1. ESTUDIO SOLUCIONES EXISTENTES

En las redes sociales más conocidas y utilizadas (Instagram, Twitter, Tiktok), normalmente ofrecen estímulos cortos y muy seguidos, mediante vídeos cortos, publicaciones... Además, que el tipo de información que los usuarios reciben suele ser muy diferenciadas unas de las otras.

Por eso el objetivo de concretizar, compactar, y hacer que el mensaje llegue de forma más directa y clara al receptor al que las empresas se quieren dirigir con la publicitación de sus eventos.

### 3. PLANIFICACIÓN DE TRABAJO

En esta sección explicaré los pasos que he seguido en la creación del proyecto.

Los primeros pasos fueron el diseño de la base de datos, crear las tablas con sus atributos y relacionar las tablas. Las tablas deben estar relacionadas entre sí, ya que se trabaja con MySQL y es un gestor de bases de datos relacionales.

El siguiente proceso, fue la planificación de páginas que debían aparecer en la aplicación, teniendo en cuenta los distintos tipos de usuario y las funcionalidades que cada uno podía realizar.

La siguiente fase fue la de desarrollo, en esta fase, iba creando la lógica de la aplicación mediante PHP a medida que iba diseñando las páginas mediante las plantillas Twig. Cabe destacar que el uso de Bootstrap, un framework CSS de código abierto, favorece la asignación de estilos a las páginas mientras que, las mismas se van programando.

La penúltima fase fue la de pruebas, en esta fase hay que tener en cuenta, errores de navegación, como por ejemplo enlaces rotos dentro de la web, o interacciones con el usuario mediante el uso de formularios proporcionados por Symfony para introducir datos, que pueden, o no, ser el tipo de datos que el controlador espera recibir.

La última fase, está relacionada con el front-end, ya que la aplicación es reponsive, esto quiere decir que es una web que está diseñada para responder y ajustarse a los dispositivos móviles, tanto por su aspecto atractivo como por su funcionamiento eficaz para aquellos usuarios que se conectan a la web desde dispositivos móviles.

## 4. DISEÑO

En esta sección se explicarán el diseño de la aplicación tanto front-end y back-end.

### 4.1. FRONT-END

Es el diseño de la aplicación que es visible al usuario, la cual está constituida por las diferentes interfaces con las que el usuario va a interactuar. Para el diseño frontend se ha utilizado Bootstrap, ya que es un framework de diseño frontend responsive, lo que da flexibilidad al desarrollador de hacer la aplicación adaptable para los diferentes dispositivos del mercado. El objetivo del diseño de la aplicación fue la comodidad y uso del usuario, para ello se tuvo en cuenta que las funcionalidades fueran de rápido acceso e intuitivas.

Las principales pantallas de la aplicación son:

Inicio. En esta primera página se muestran las funcionalidades de la aplicación mediante imágenes, de un modo interactivo para el usuario, ya que mediante un carrusel (código de Bootstrap) puede ir haciendo clic en los botones de control para ver la imagen que él mismo desee. En las imágenes aparecen breves textos descriptivos, explicando para que se utiliza la aplicación y justificando el uso de esta.



i. Ventana de inicio de la aplicación VerEventos





ii. Carrusel de Bootstrap imagen 1



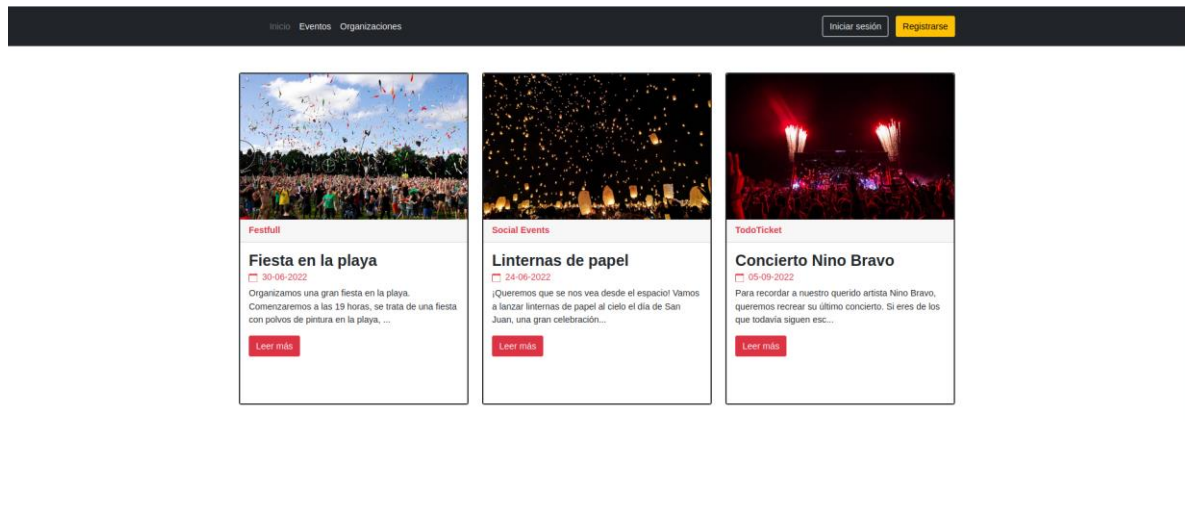
iii. Carrusel de Bootstrap imagen 2





*iv. Carrusel de Bootstrap imagen 3*

Eventos. Esta ventana muestra los datos de todos los eventos que están cargados en la base de datos de la aplicación: la información de los eventos consta de: el nombre de la organización que los crea, el título, la fecha, y una breve descripción.

*v. Venta de Eventos de la aplicación VerEventos*

Los eventos se muestran mediante un elemento “card” (de Bootstrap). Este elemento contiene la información de los eventos y un botón para leer más sobre el evento, cuando se hace clic, la aplicación redirige a una ventana para la visualización individual del evento seleccionado, pero si el usuario no ha iniciado sesión le redirige al formulario para iniciar sesión.

Esto es lo que aparece cuando el usuario está registrado, también es un elemento “card” de Bootstrap. Aquí la descripción del evento sí que está completa, además en la parte inferior aparece un apartado con las valoraciones del evento aportadas por otro usuario.



## FIESTA EN LA PLAYA

30-06-2022

Organizamos una gran fiesta en la playa. Comenzaremos a las 19 horas, se trata de una fiesta con polvos de pintura en la playa, el motivo de la celebración es el FINAL DE EXÁMENES. Se realizará en la playa de Malvarrosa, quedas invitado.

Valoraciones [+](#)

 user5. comment

Evento organizado por **FESTFULL**

vi. Visualización individual de un evento

Cuando el usuario hace clic en el icono con el símbolo +, aparece una entrada de texto para que pueda publicar su comentario.

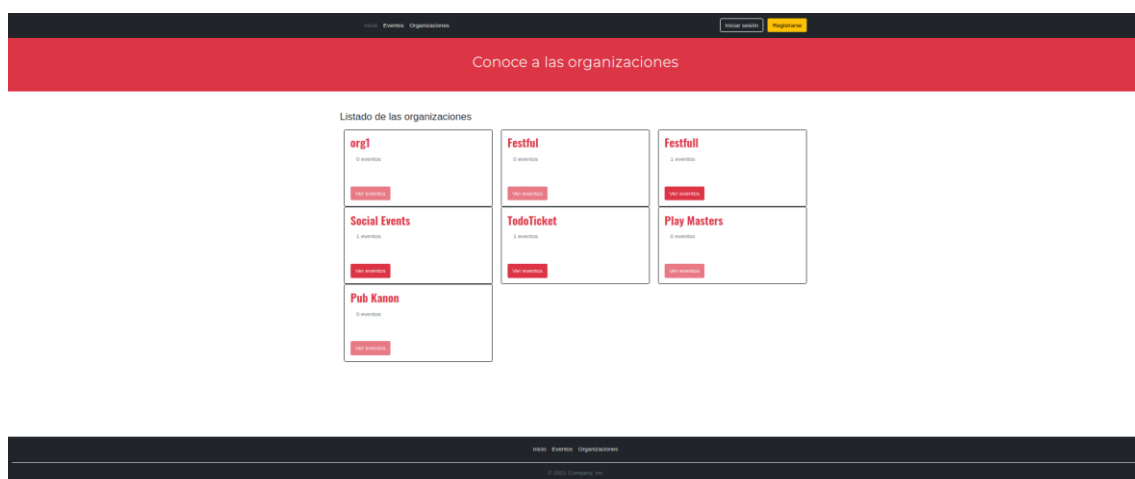


A modal window titled "Tu comentario" with a close button (X) in the top right corner. It contains a large text input field and a blue button labeled "Aceptar" below it.

vii. *Elemento modal de Bootstrap para realizar el comentario*

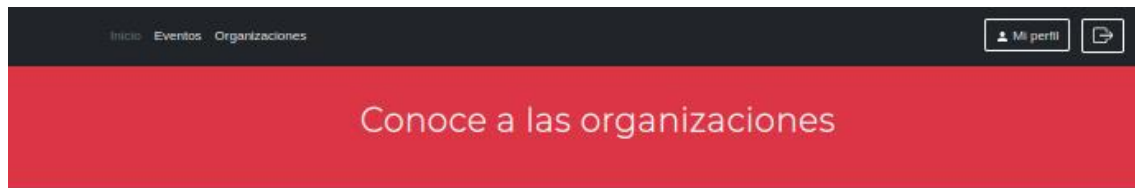
Cuando el usuario envía su comentario aparece inmediatamente en las valoraciones del evento

- Organizaciones. En esta ventana, se encuentra un listado de todas las usuarios registrados con el role "ROLE\_ORG" en la aplicación, contiene el nombre de las organizaciones y el número de eventos que han creado. También se visualiza con el elemento "card" de Bootstrap. Cuando se hace clic en el botón la aplicación redirecciona a una nueva página.



viii. *Venta de Organizaciones de la aplicación VerEventos*

Esta es la página a la que se redirecciona tras accionar el botón. Su función es mostrar un listado de todos los eventos que tiene creados la organización.



#### Listado de eventos



**CONCIERTO NINO BRAVO**

■ 05-09-2022

Para recordar a nuestro querido artista Nino Bravo, queremos recrear su último concierto. Si eres de los que todavía siguen escuchando sus canciones como nosotros, entra en nuestra web y compra tus entradas para el concierto. Sus canciones nos devolverán a aquellos viejos tiempos y crearán un increíble ambiente. Comenzará a las 20:00 en la ciudad de las artes y las ciencias. Entra ya en nuestra web y compra tus entradas, te estamos esperando.

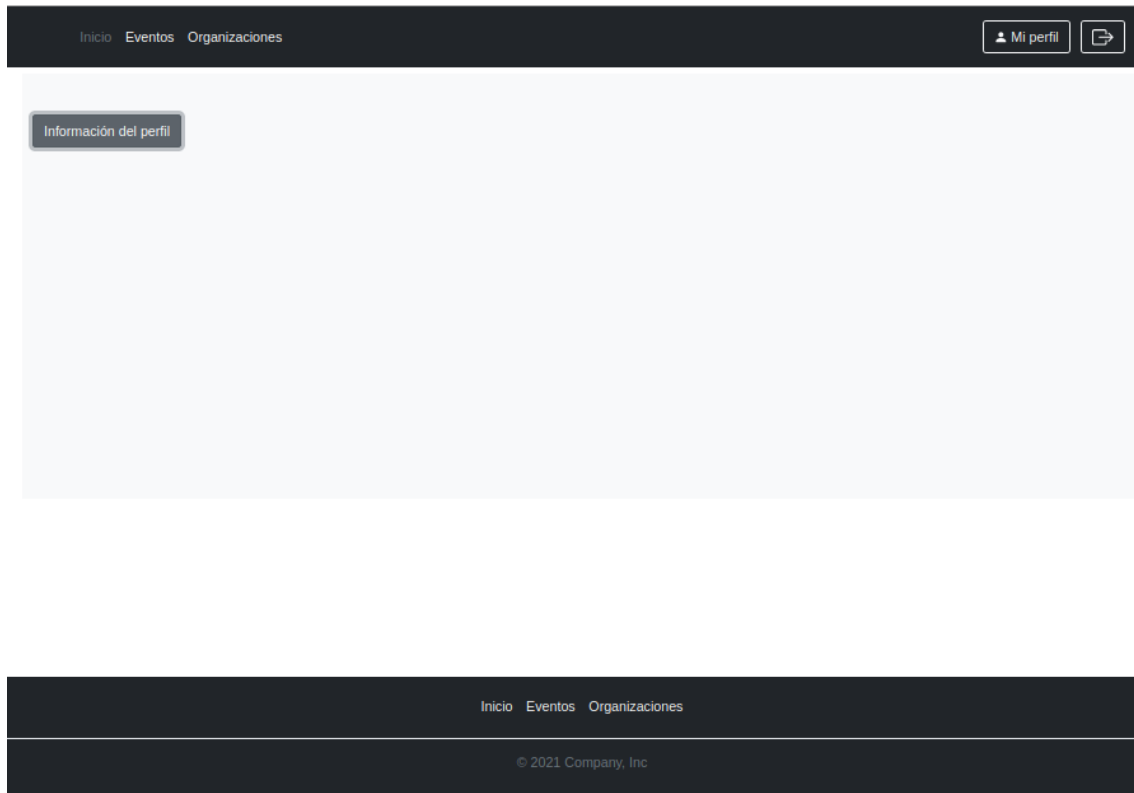
**Valoraciones**

#### ix. *Venta de listado de eventos de la aplicación VerEventos*

- El menú de navegación

Éste es el menú de navegación que aparece cuando un usuario no está registrado. Contiene los elementos Inicio, Eventos y Organizaciones ya descritos anteriormente. En la parte derecha aparecen dos botones, para iniciar sesión y para registrarse.

El botón “mi perfil” redirige a una nueva ventana, que es la siguiente:



x. *Vista de Mi perfil de la aplicación VerEventos*

El único contenido que aparece en la página es un botón, que al ser accionado, mediante la clase “dropdown” de Bootstrap, despliega un “card” que muestra la información del usuario.

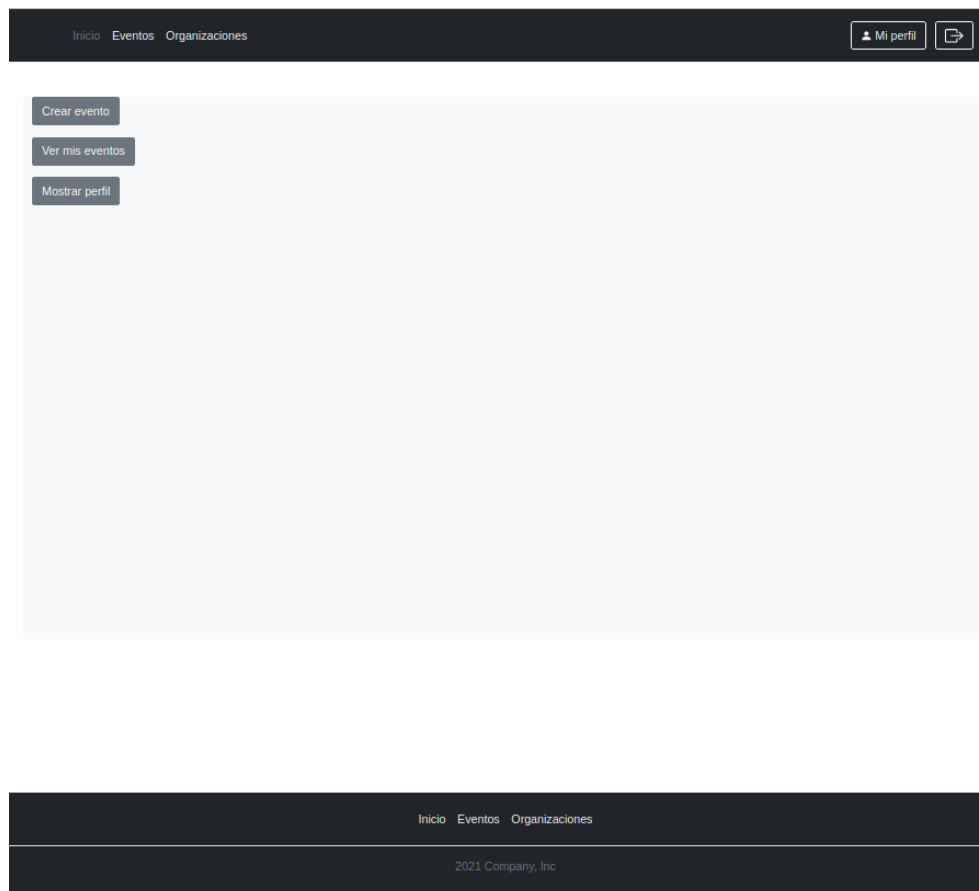


*xi. Botón que despliega una card con información sobre usuario*

La “card” muestra un botón para eliminar cuenta, que elimina toda la información respecto al usuario.

En el caso de que el usuario que hay registrado sea de tipo organización, el botón mi perfil redirige a una venta diferente al caso anterior, que es la siguiente:





xii. *Editar perfil para el rol organización*

- El primer botón *Crear evento redirecciona a una ventana nueva para la creación de un nuevo evento:*

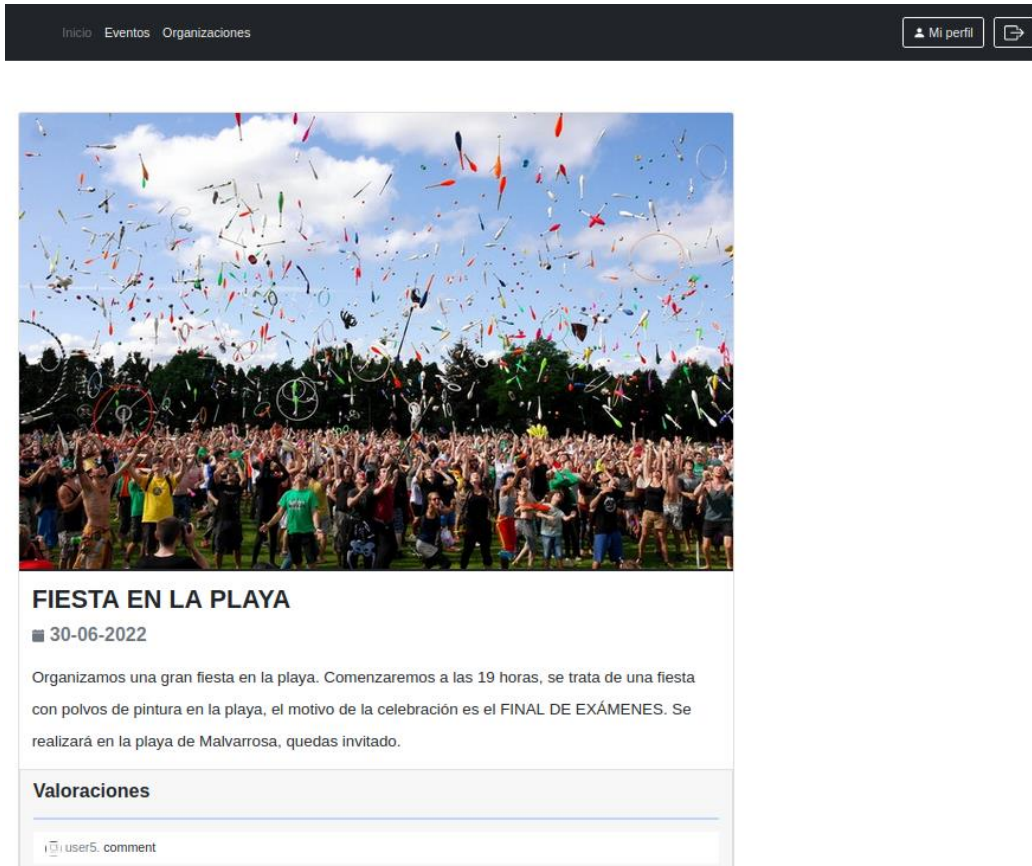
The screenshot shows a web form for creating an event. It includes a header with navigation links and a user profile button. The form itself has fields for an event image (with an 'Examinar...' button and a message 'No se ha seleccionado ningún archivo.'), a 'Title' text input, a 'Description' text area, and a 'Date' section with dropdown menus for month (Jan), day (1), and year (2017). An 'Aceptar' button is at the bottom.

xiii.

*Ventana crear evento, para el rol organización*

Esta ventana muestra un formulario el cual tiene por función crear un nuevo evento para la organización, los datos a insertar son: la imagen del evento, el título, una descripción y una fecha.

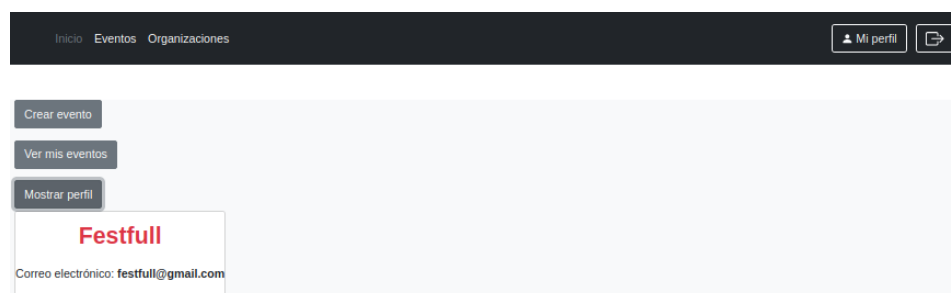
El siguiente botón que muestra la página es *ver mis eventos*, el cual redirecciona a una ventana nueva.



xiv. *Página Ver mis eventos*

La función es mostrar los eventos que tiene creados la propia organización y los comentarios del resto de usuarios.

- Por último, el botón mostrar perfil



xv. *Botón mostrar perfil*

## 4.2. Back-end

El back-end es el lado del servidor que procesa las solicitudes del cliente recibidas desde el front-end, lo que el back-end realmente espera son los datos recolectados del usuario desde el front-end, para procesarlos y luego devolver una respuesta al front-end para generar una salida entendible para el usuario. El back-end de la aplicación se ha implementado con tecnología PHP.

### 4.2.1. Patrón MVC (Modelo Vista Controlador)

#### El Modelo

El modelo es la capa donde se trabaja con los datos, por tanto, contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán los correspondientes selects, updates, inserts, etc. Es responsable de acceder a la capa de almacenamiento de datos.

#### Vista

Contiene el código de la aplicación que va a producir la visualización de las interfaces de usuario, es, el código que nos permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida.

Las vistas requerirán los datos a los modelos y ellas se generará la salida, tal como nuestra aplicación requiera.

- Recibir datos del modelo y mostrarlos al usuario.

#### Controlador

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación.

En realidad, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es

manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.

### 4.2.2. Symfony

La arquitectura de la aplicación está creada con Symfony → *Symfony es un framework PHP de tipo full-stack construido con varios componentes independientes creados por el proyecto Symfony. Es un proyecto de código abierto.*

#### información

Plantea las aplicaciones web de una determinada manera a la que el desarrollador se tiene que acoplar.

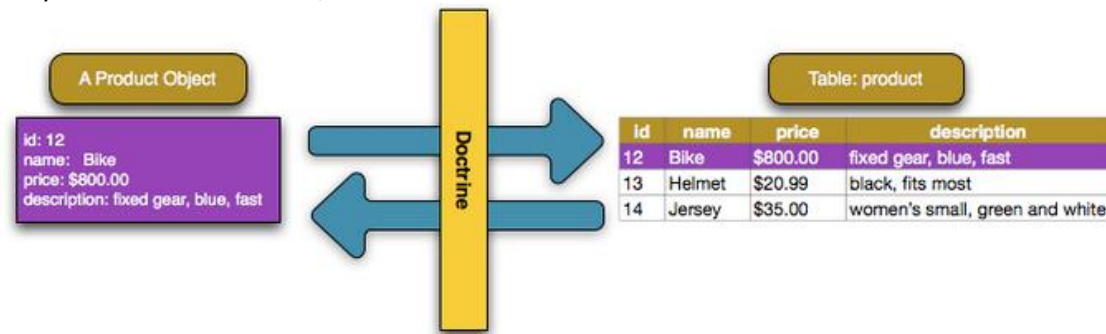
Incluye componentes para muchas tareas habituales, como formularios o seguridad. Estos componentes son librerías independientes de Symfony y pueden usarse en otros proyectos.

Symfony tiene varias ventajas: permite ahorrar tiempo, el sitio web es fácil de mantener, ofrece seguridad mejorada, ya que este marco está respaldado por actualizaciones para la mejoría de este aspecto.

También tiene algunos aspectos negativos a tener en cuenta como, por ejemplo: tiene un rendimiento bastante bajo, especialmente en aplicaciones de alta carga en tiempo real, además para programas en Symfony, no basta con tener conocimiento sobre los lenguajes de programación que se van a utilizar, ya que entender el funcionamiento del marco requiere tiempo.

### 4.2.3. Doctrine

Es el encargado del modelo, Doctrine es el ORM utilizado por Symfony por defecto. ORM son las siglas de Object-Relational Mapping, es decir, el mapeo relacional de objetos. Esto significa que va a trasladar los datos de una base de datos relacional, como puede ser MySQL en este caso, a un sistema de clases y de objetos, donde las clases serían las tablas y los registros pasarían a ser lo equivalente a objetos.



xvi. *Imagen extraída de la web de Symfony*

#### 4.2.4. Composer

La instalación de Symfony se realiza mediante Composer, es un gestor de dependencias para PHP, muy simple de operar.

Es una aplicación PHP que ayuda a administrar las librerías desarrolladas por terceros que vas a incorporar a tu proyecto.

#### 4.2.5. Twig

Twig es un motor de plantillas, se utiliza con el objetivo de separar al máximo el código PHP de la estructura HTML de la página, de forma que toda la lógica de la aplicación queda fuera de la vista.

#### 4.2.6. MariaDB

El servidor MariaDB es una base de datos relacional de código abierto, permite editar, guardar y extraer información de una base de datos,

#### 4.2.7. Partes de código, estructuración de archivos y base de datos

En esta sección se va a describir la forma en la que Symfony organiza la jerarquía de sus directorios, ya que en este framework la ubicación de los ficheros es muy importante.

También se van a mostrar y explicar partes del código importantes.

Juan Raul Nan



Estos son los directorios principales de Symfony:

config/ → guarda archivos de configuración, para los bundles instalados mediante Composer.

templates/ → este directorio solamente se crea cuando se instala Twig. Guarda las plantillas utilizadas en la vista, y sus archivos contienen código HTML y scripts.

public/ → aquí es donde se guardan los archivos web

### Controladores en profundidad

Son el elemento principal de Symfony, contienen métodos que reciben peticiones del cliente, las procesan y generan una salida o una redirección.

Por ejemplo:

Así es como se renderiza una plantilla desde el controlador, el método empleado, contiene paso de parámetros, que se verá mas a continuación.

```
return $this->render('main/index.html.twig', [ 'role'=> $tipoDeUsuario,
]);
```

xvii. *Imagen extraída del controlador de la aplicación*

Así se utiliza la redirección

```
return $this->redirectToRoute("app_login");
```

xviii. *Imagen extraída del controlador de la aplicación*

Los controladores, contienen métodos a los que se accede mediante rutas, y son definidos mediante connotaciones específicas de Symfony:

```

/**
 * @Route("/logout",name="app_logout")
 */
public function loguot() {
    $tipoDeUsuario='no';
    return $this->render('main/index.html.twig', [ 'role'=> $tipoDeUsuario,
        ]);
}

```

xix. *Método para cerrar sesión, situado en el controlador*

Este método en concreto es el de “logout” (cerrar sesión). Aunque para hacer realizar esta funcionalidad es necesario hacer algunos cambios.

Se trata de modificar el archivo *security.yaml* que está situado en la carpeta *config/packages*

```

firewalls:
    dev:
        pattern: ^/(_(profiler|wdt)|css|images|js)/
        security: false
    main:
        lazy: true
        provider: app_user_provider
        form_login:
            login_path: app_login
            check_path: app_login
        logout:
            path: app_logout

```

Para la creación de un formulario para el inicio de sesión, también hay que modificar el archivo *security.yaml*, y se añade lo siguiente:

```

main:
    lazy: true
    provider: app_user_provider
    form_login:
        login_path: app_login
        check_path: app_login

```

xx. *Imagen extraída del archivo security.yaml del proyecto*

Esto indica que la función del controlador que se encarga de comprobar la validación de las credenciales está definida por el nombre ‘app\_login’.

La función utiliza *AuthenticationUtils* para realizar la comprobación en la base de datos.

```
/**
 * @Route("/login", name="app_login")
 */
public function login(AuthenticationUtils $authenticationUtils): Response {
    $error=$authenticationUtils->getLastAuthenticationError();
    $lastUsername=$authenticationUtils->getLastUsername();

    return $this->render('login/login.html.twig', [ 'controller_name' => 'MainController'
        'error'=> $error,
        'last_username'=> $lastUsername,
        'role' => 'no'
    ]);
}
```

xxi. *Imagen de la función de inicio de sesión*

Previo al inicio de sesión se debe realizar el registro, y en esta aplicación existen 2 tipos de registro, la diferencia está en el rol con el que interactuarán los usuarios con la aplicación. Existen 2:

- “ROLE\_USER”: este tipo de rol representa al usuario visitante de la página, que podrá interactuar con los eventos publicados por el siguiente rol.
- “ROLE\_ORG”: este rol, es el que representa a los usuarios de tipo organización. Estos disponen de la funcionalidad de crear eventos y publicarlos en la página web

También se puede navegar por la página sin un inicio de sesión previo, pero algunas funcionalidades están capadas.

Siguiendo el orden cronológico en el que se ha desarrollado esta aplicación, este método ha sido el primero en incorporar formularios (registro).

Los formularios se pueden asociar a una entidad, esto facilita mucho la inserción de datos, ya que se trabaja con objetos.

Mediante Symfony el uso de formularios se agiliza bastante. Pero antes de explicar como se realiza, hay que crear las entidades.

Una entidad, es la representación de una tabla de la base de datos, con sus atributos que lo diferencian. Para crear una entidad se ejecuta el siguiente comando:

Juan Raul Nan

```
php bin/console make:entity
```

Después de ejecutarlo, pedirá que definas sus atributos (nombre y tipo). Al finalizar con la definición de la entidad se ejecutan estos 2 comandos para insertarlos en la base de datos

```
php bin/console make:migration
```

```
php bin/console doctrine:migration:migrate
```

Para crear la entidad usuario, la única diferencia es que, en vez de ejecutar el primer comando, se ejecuta el siguiente:

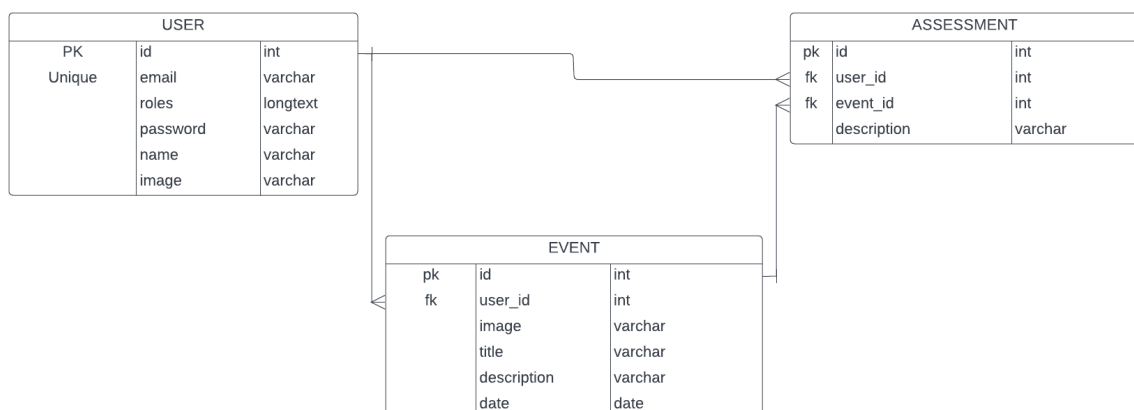
```
php bin/console make:user
```

Cabe destacar la importancia del método `getUser()`. se utiliza para devolver el objeto de tipo Usuario actual. Este método es muy necesario si se desea obtener información sobre el usuario que ha iniciado sesión y está navegando por la página.

## Base de datos

Se utiliza, Xampp es un paquete de software libre, que trata de un sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP.

Este proyecto contiene una base de datos muy sencilla, únicamente con 3 tablas, que se relacionan entre ellas.



xxii. *Modelo entidad relación de la app*

Juan Raul Nan

La tabla USER representa al usuario de la aplicación, es la tabla principal ya que también forma parte en las otras tablas como clave ajena, ya que un usuario tiene muchos ASSESSMENT (valoraciones), y también cabe la posibilidad de que tenga muchos EVENT (eventos). A su vez un evento tiene muchos comentarios.

Para acceder

Como representación de los detalles para tener en cuenta para realizar un borrado en la base de datos, está la siguiente función:

```

/**
 * @Route("/deleteUser/{user}", name="app_deleteUser")
 */
public function deleteUser(int $user, EntityManagerInterface $em, AuthenticationUtils $authenticationUtils) {
    $usuarioObj = $em->getRepository(User::class)->findBy(array('id'=>$user));

    $susComentarios = $em->getRepository(Assessment::class)->findBy(array('user'=>$usuarioObj));

    //ELIMINA TODOS LOS COMENTARIOS DEL USUARIO
    foreach($susComentarios as $comentario) {
        try {
            $em->remove($comentario);
            $em->flush();
        } catch (\Exception $e) {
            echo $e->getMessage();
        }
    }

    //ELIMINA LA CUENTA
    try {
        foreach($usuarioObj as $usuario) {
            $em->remove($usuario);
            $em->flush();
        }

        return $this->redirectToRoute("app_login");
    } catch (\Exception $e) {
        echo $e->getMessage();
    }

    $error=$authenticationUtils->getLastAuthenticationError();
    $lastUsername=$authenticationUtils->getLastUsername();

    return $this->render('login/login.html.twig', [ 'controller_name'=> 'MainController',
        'error'=> $error,
        'last_username'=> $lastUsername,
        'role' => 'no'
    ]);
}

```

xxiii. *Imagen del método para borrar*

Esta función sirve para eliminar un usuario de 'ROLE\_USER'.

A la hora de obtener objetos de una tabla, existen diferentes métodos que se pueden emplear, en este caso, se utiliza el método `findBy()`, localiza todos los objetos que cumplan los criterios de búsqueda pasados como parámetro. Mediante este método se obtiene el usuario, y después los comentarios de el mismo usuario, para primero borrar los comentarios, y luego el propio usuario. Es decir, primero se elimina los comentarios para que `ASSESSMENT.user_id` no se quede a nulo, ya que esto daría error al ser un clave ajena.

Después de borrar las claves ajenas, finalmente se borra el usuario mediante el método `remove()`

El gestor de plantillas Twig permite crear estructuras de control en sus plantillas como. También facilita la herencia de plantillas, que ha resultado de gran utilidad en este proyecto, ya que todas las pestañas comparten el mismo menú de navegación y el mismo "footer".



```

10 | {{ include ('header.html.twig', {}) }}
11 | <div class="container mt-5">
12 |
13 |     {% if eventos %}
14 |     <div class="d-flex justify-content-center">
15 |     {% for evento in eventos %}
16 |     <div class="card border border-dark" style="width: 50%;">
17 |         <div id="contenedor">
18 |             
19 |         </div>
20 |         <div class="card-body">
21 |             <p class="card-title fw-bold fs-2 text-uppercase">{{evento.title}}</p>
22 |             <h5 class="card-subtitle mb-2 text-primary"><svg xmlns="http://www.w3.org
23 |                 fill="currentColor" class="bi bi-calendar" viewBox="0 0 16 16">
24 |                 <path
25 |                     d="M3.5 0a.5.5 0 0 1 .5.5V1h8V.5a.5.5 0 0 1 1 0V1h1a2 2 0 0 1
26 |                 </svg> {{ evento.date|date('d-m-Y')}}</h5>
27 |             <hr class="border border-primary mt-0">
28 |             <p class="card-text lh-lg fs-5 mt-0">{{evento.description}}</p>
29 |         </div>

```

xxiv. Captura extraída de la plantilla *visualizarEvento.html.twig*

En la primera línea que aparece en esta imagen (nº 10), se aprecia el modo en el cual funciona la herencia para implementar “header.html.twig”, que es el menú de navegación.

En la línea número 13 se observa la creación de una estructura condicional, que, a su vez, dentro contiene un recorrido de array.

En la línea 18 se observa como se define la raíz de la etiqueta <img> mediante el atributo image de un objeto llamado evento.

## 5. RECURSOS

En esta sección se van a explicar los requisitos que han sido únicamente necesarios para la creación del proyecto.

Como recurso hardware, se ha empleado un ordenador personal capaz de soportar una máquina virtual con 2GB de RAM con conexión a internet.

Las herramientas software empleadas han sido una distribución de Linux basada en Ubuntu y Debian (Linux Mint 20.2 Cinnamon).

Framework Symfony → <https://symfony.com/>

Xampp → <https://www.apachefriends.org/es/index.html>

Visual Studio Code, como herramienta de desarrollo → <https://code.visualstudio.com/>

Navegador Firefox → <https://www.mozilla.org/es-MX/firefox/new/>

Virtual Box → <https://www.virtualbox.org/>

Doctrine ORM → <https://www.doctrine-project.org/>

## 6. CONCLUSIONES

Finalizando el proyecto se han llegado a las siguientes conclusiones:

- Se presenta una interfaz sencilla y clara, para presentar la información con un nivel adecuado de densidad.
- Los conocimientos adquiridos a lo largo de los últimos dos años me han permitido obtener unos conocimientos técnicos para plantear, evaluar y realizar el desarrollo de esta aplicación de manera exitosa.
- El proyecto puede mejorar en muchos aspectos, pero por la falta de antelación en cuanto al planteamiento y al comienzo de desarrollo, me considero satisfecho con el resultado de la aplicación.
- Realmente no me he dado cuenta de todos los conocimientos que he adquirido a lo largo del curso, hasta que he tenido que redactarlos

En cuanto ha la consecución de objetivos, no estoy satisfecho al 100%, ya que no he podido aplicar mis conocimientos adquiridos en el módulo de Desarrollo de Aplicaciones en entorno cliente. Esto es debido a 2 motivos: el principal motivo, es la facilidad que ofrece el framework Symfony, ya que este marco realiza casi por su propia cuenta la comunicación entre front-end y back-end.

Por lo tanto, la necesidad de comunicar las dos capas entre sí, queda cubierta por esta herramienta.

El segundo problema que he encontrado para intentar incorporar Javascript a la aplicación, ha sido el desconocimiento técnico, de cómo funcionan las peticiones Ajax cuando se combinan con las rutas de los controladores que ofrece Symfony.

Tampoco he podido incorporar otras funcionalidades que tenía pensadas como por ejemplo una verificación de correo mediante alguna API.

## 7. ANEXOS COMPLEMENTARIOS

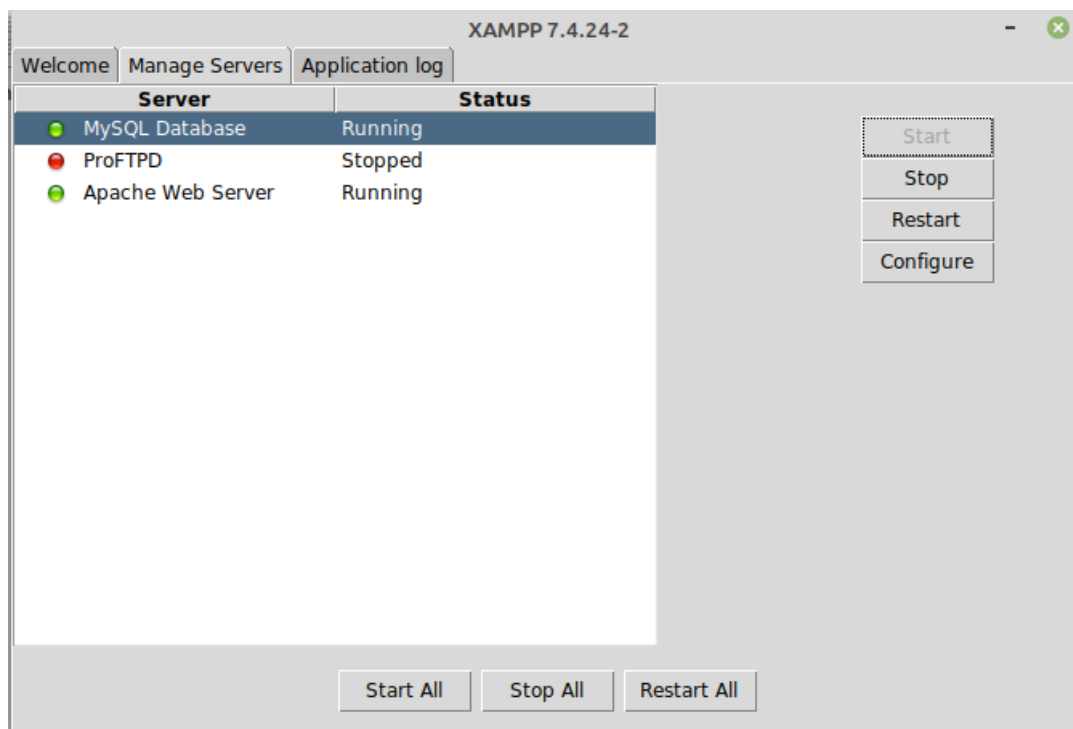
Este anexo explica cómo se debe instalar el proyecto y desplegarlo.

### 7.1. Manual de instalación y despliegue de la aplicación

Lo primero será crear una máquina virtual con un sistema operativo Linux Mint, con al menos 2GB de RAM (son preferibles 3GB). En esta máquina virtual necesitaremos tener instalado Xampp, la versión utilizada para la creación de la app fue la 7.4.24-2.

Teniendo también Symfony y Doctrine instalado y después de descargar el proyecto de Github, los pasos a seguir son los siguientes:

#### 1. Arrancar los servidores MySQL y Apache



xxv. Interfaz del Xampp

2. Ubicar la localización del proyecto en la máquina y arrancador el servidor de Symfony. La salida que devolverá el mensaje contiene la ruta que hay que escribir en la URL del navegador Firefox para usar la aplicación.

```
raul@raul-VirtualBox:/opt/lampp/htdocs/TFG$ symfony server:start
```

```
[OK] Web server listening  
The Web server is using PHP CLI 7.4.3  
http://127.0.0.1:8000
```

## 7.2. Explicación del contenido ubicado en el repositorio GitHub

El repositorio GitHub utilizado para el control de versiones, contiene todas las carpetas y archivos creados por Symfony, y los creados por mí para la generación de la aplicación, también contiene material utilizado para la presentación del proyecto (documento PowerPoint), y esta misma memoria del proyecto.

<https://github.com/RaulNan10/TFG>

## 8. BIBLIOGRAFÍAS

En esta sección se muestran las bibliografías que he empleado para consultar información

W3schools y StackOverflow para la solución de algunos problemas y errores

<https://www.w3schools.com/>

<https://stackoverflow.com/>

La página de Symfony, mayoritariamente para ver los tipos de campos que un formulario puede tener y los atributos que se pueden definir,

<https://symfony.com/>

Bootstrap, para clases CSS e iconos

<https://getbootstrap.com/>

Wikipedia para la definición de palabras

<https://www.wikipedia.org/>

Documentos proporcionados por IES La Vereda



## LEYENDA DE IMÁGENES

- i. *Ventana de inicio de la aplicación VerEventos*
- ii. *Carrusel de Bootstrap imagen 1*
- iii. *Carrusel de Bootstrap imagen 2*
- iv. *Carrusel de Bootstrap imagen 3*
- v. *Venta de Eventos de la aplicación VerEventos*
- vi. *Visualización individual de un evento*
- vii. *Elemento modal de Bootstrap para realizar el comentario*
- viii. *Venta de Organizaciones de la aplicación VerEventos*
- ix. *Venta de listado de eventos de la aplicación VerEventos*
- x. *Venta de Mi perfil de la aplicación VerEventos*
- xi. *Botón que despliega una card con información sobre usuario*
- xii. *Editar perfil para el rol organización*
- xiii. *Ventana crear evento, para el rol organización*
- xiv. *Página Ver mis eventos*
- xv. *Botón mostrar perfil*
- xvi. *Imagen extraída de la web de Symfony*
- xvii. *Imagen extraída del controlador de la aplicación*
- xviii. *Imagen extraída del controlador de la aplicación*
- xix. *Método para cerrar sesión, situado en el controlador*
- xx. *Imagen extraída del archivo security.yaml del proyecto*
- xxi. *Imagen de la función de inicio de sesión*
- xxii. *Modelo entidad relación de la app*
- xxiii. *Imagen del método para borrar*
- xxiv. *Captura extraída de la plantilla visualizarEvento.html.twig*
- xxv. *Interfaz del Xampp*

## RESUMEN DEL CONTENIDO

Esta memoria tiene la intención de explicar tanto a una persona poco habituada a las páginas web como a una persona con conocimientos de programación, la forma en la que se navega por la web y cómo se ha creado el proyecto desde un principio.

Describe las tecnologías implementadas para la creación de la base de datos, la estructuración de carpetas dentro del proyecto, el desarrollo de este y la visualización de la página. También contiene explicaciones sobre las herramientas de trabajo empleadas y el entorno de desarrollo utilizado para toda la enumeración anterior.

Pretende mostrar el caso de uso de esta aplicación para una empresa, describiendo y analizando el mercado y sus competidores.