

# DESARROLLO DE SISTEMA WEB PARA LA ADMINISTRACIÓN DE EVENTOS ORGANIZADOS POR INCONCERTO

Pizarra Jhonathan<sup>1</sup>

<sup>1</sup>Escuela Politécnica Nacional, Escuela de Formación de Tecnólogos, Quito, Ecuador

---

**Resumen:** Este trabajo propone el diseño, desarrollo y despliegue de un sistema web, aplicando la metodología SCRUM y el entorno de desarrollo libre de Laravel, permitiendo, de esta manera, administrar eventos, llevar un adecuado control registros, y mayor organización interna, notificaciones de próximos eventos del colectivo a realizarse, apoyando a la fundación en la difusión de música clásica, reconocimiento y actividades de la institución.

Finalmente, toda la información fue registrada y almacenada en una base de datos relacional, la cual es gestionada por el Sistema Gestor de Base de Datos (SGBD) MySQL, debido a su rendimiento y escalabilidad. Mejorando así la organización de estos procesos y proporcionando acceso a la información por medio de la tecnología.

Palabras clave: Sistema Web, Scrum, Laravel, MySQL, Gestión de eventos.

## Development of web system for the management of events organized by Inconcerto

**Abstract:** This work proposes the design, development and deployment of a web system, applying the SCRUM methodology and the free development environment of Laravel, allowing, in this way, to manage events, keep adequate control records, and greater internal organization, notifications of upcoming events of the collective to be held, supporting the foundation in the dissemination of classical music, recognition and activities of the institution.

Finally, all the information was registered and stored in a relational database, which is managed by the MySQL Database Management System (DBMS), due to its performance and scalability. Thus, improving the organization of these processes and providing access to information through technology.

Keywords: Web System, Scrum, Laravel, MySQL, Event Management

---

### 1. INTRODUCCIÓN

El presente informe se centra en el desarrollo de un Sistema Web para la gestión de eventos organizados por la fundación Inconcerto, aplicando la metodología *Scrum*. Logrando de esta manera la automatización de los procesos llevados manualmente, mejorando la organización y experiencia de gestión para la fundación.

Inconcerto es un colectivo interdisciplinario de artistas, sociólogos, comunicadores y gestores culturales que crean espacios y canales para difundir y revalorizar la música clásica en nuestra época y lugar a través de proyectos que resignifican los protocolos que han situado a ésta, como un arte de y para una determinada clase social e intelectual. [1].

A partir de la experiencia de músicos, gestores e investigadores, se ha identificado la concentración de la música clásica en un sector minoritario y un número limitado de espacios, en su mayoría en las principales urbes del país, lo cual ha impedido su constitución como una práctica cultural de los distintos grupos de la sociedad, en particular de poblaciones en condiciones de doble vulnerabilidad.

Adicionalmente, la larga trayectoria que ha tenido desde el año 2012 con actividades artísticas para los distintos públicos, en particular con personas que no han tenido contacto con este tipo de arte, han comprobado el potencial de la música de provocar, despertar y reactivar afectos y reacciones únicas en las audiencias.

Actualmente, el colectivo no cuenta con un sitio web en la cual se pueda publicar actualizaciones sobre los eventos que se organizan a lo largo del año y la logística de todos los eventos organizados son llevados mediante un documento de Excel lo que, en algunas ocasiones, genera conflictos para la correcta organización de los eventos y la comunicación entre los involucrados.

Por lo citado anteriormente, se ha desarrollado un Sistema Web que permite llevar un control de los eventos, artistas invitados, hospedajes, conciertos, notificaciones sobre los próximos eventos a realizarse, entre otros. Mejorando la organización y experiencia de gestión para la fundación.

### 2. METODOLOGÍA

Las metodologías ágiles son procesos de la ingeniería de software que permiten responder adecuadamente a las condiciones propuestas por el cliente, permitiendo immediatez

y flexibilidad [2]. Una comunicación efectiva con el cliente y miembros del equipo de trabajo ayudan en este tipo de proyectos donde los requerimientos son cambiantes.

Las subsecciones siguientes describen la implementación de la metodología *Scrum* en el desarrollo de este proyecto integrador. Posterior a ello, se procede con las siguientes etapas que son la elaboración de Historias de Usuario, *Product Backlog* y *Sprints*.

## 2.1. Metodología de Desarrollo

La metodología por utilizar en el proyecto es *Scrum*, la cual se basa en el trabajo por iteraciones o *Sprints*. *Scrum* es una metodología ágil que tiene por objetivo planificar y controlar proyectos con una gran medida a cambios de última hora, en donde la incertidumbre sea elevada [3]. Garantizando en todo momento la calidad de los sistemas propuestos y el cumplimiento de los requerimientos establecidos.

### 2.1.1. Roles

*Scrum* al ser una metodología ágil basado en roles asegura el control de todos los procesos durante el desarrollo del software. Además, las personas a quienes se les asigna un rol deben comprometerse y cumplir con las necesidades del proyecto propuesto [4]. A continuación, se especifican los roles del proyecto integrador:

**Product Owner:** Es la persona encargada de revisar los resultados y mantener el control de todos los entregables por parte del equipo de desarrollo [4]. Este rol es representado por la productora del programa Inconcerto, quien proporciona todos los requerimientos e información necesaria en los procesos de gestión de los eventos, logrando con esta información determinar perfiles, módulos y funcionalidades para el Sistema Web.

**Scrum Master:** Es el líder del proyecto y su responsabilidad es la de guiar y cumplir las reglas de la metodología ágil *Scrum* [4]. Este rol es representado por el director del proyecto integrador, quien posee los conocimientos necesarios y técnicos para ayudar al equipo de desarrollo a solventar dudas e inconvenientes y lograr el cumplimiento del objetivo planteado.

**Development Team:** El presente rol está representado por el autor y desarrollador del proyecto, quien dispone de todos los conocimientos y habilidades necesarias para transformar las tareas del *Sprint Backlog* en pequeños entregables funcionales que aporten valor a los usuarios finales [4] quien junto al *Scrum Master* toman las decisiones adecuadas para conseguir los objetivos propuestos durante el desarrollo de cada *Sprint* del proyecto.

Para cumplir con el desarrollo del proyecto integrador los roles se han dividido como se describe en la **TABLA I**.

**TABLA I:** Equipo de trabajo y asignación de roles

NOMBRE	ROL
Srta. Carlera Moncayo	<i>Product Owner</i>
Ing. Edwin Salvador, MSc.	<i>Scrum Master</i>
Jhonathan Pizarra	<i>Development Team</i>

### 2.1.2. Artefactos

Son elementos utilizados por *Scrum*, diseñados para garantizar el registro y la transparencia de la información fundamental del proceso de *Scrum* [5]. Permitiendo de esta manera organizar el desarrollo por etapas y que los roles mencionados anteriormente puedan realizar las tareas definidas en cada *Sprint*. A continuación, se detallan los artefactos que se han utilizado:

**Recopilación de Requerimientos:** Para dar inicio a la primera etapa para el desarrollo del software, se define con claridad la lista de requerimientos del cliente permitiendo desarrollar sistemas óptimos y apegados a la necesidad del negocio [6]. Por tal motivo, para cumplir con esta etapa se han mantenido una serie de reuniones con el *Product Owner* e interesados en el proyecto, permitiendo obtener la lista inicial de requerimientos general para el desarrollo del sistema web, la cual se encuentra detallada dentro del Manual Técnico en el apartado recopilación de requerimientos (pág. 2 - 4).

**Historias de Usuario:** Consiste en una descripción general de una función de software descrita desde el punto de vista del usuario final. Tiene como propósito estipular cómo esta función aportará valor al negocio del cliente [7]. Estas se basan en reuniones con los miembros del proyecto entre el cliente y el equipo de desarrollo. A continuación, se detalla los elementos que componen una Historia de Usuario:

- **Identificador (ID):** Es un identificador único alfanumérico.
- **Usuario:** Información del perfil del usuario.
- **Nombre Historia:** Nombre para definir la Historia de Usuario.
- **Prioridad en negocio:** Califica la prioridad del negocio en una escala de alta, medio y bajo.
- **Riesgo de desarrollo:** Califica el grado de riesgo en el desarrollo del *Sprint*.
- **Iteración:** Calificativo numérico de un *Sprint*.
- **Responsable:** Miembro del equipo de desarrollo quien es el encargado de dicha funcionalidad.
- **Descripción:** Información de la Historia de Usuario.
- **Observación:** Información breve que se debe de tomar antes de elaborar dicha funcionalidad.

La **TABLA II** detalla una de las 55 Historias de Usuario que se han obtenido para el desarrollo del sistema web. Por otra parte, se detallan las demás historias de usuario dentro del Manual Técnico en el apartado Historias de Usuario (pág. 5 - 32).

**TABLA II:** Historia de Usuario 1 – Inicio de sesión del usuario Administrador

HISTORIA DE USUARIO	
<b>Identificador (ID):</b> HU001	<b>Usuario:</b> Administrador/Operario
<b>Nombre Historia:</b> Iniciar sesión usuario Administrador	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> Alta
<b>Iteración Asignada:</b> 1	
<b>Responsable (es):</b> Jhonathan Pizarra	
<b>Descripción:</b> Para el acceso a las funcionalidades, el sistema web posee autenticación de perfiles uno de ellos es del administrador quien para identificarse es necesario el correo institucional y contraseña las cuales fueron proporcionadas por el desarrollador.	
<b>Observación:</b> El correo debe ser el institucional del colectivo Inconcerto y la clave es proporcionada por el desarrollador del sistema web.	

**Product Backlog:** Es una lista ordenada de todos los requerimientos que sean necesarios para la elaboración del proyecto [8]. En el *Product Backlog* se lista de forma ordenada los requerimientos funcionales del Sistema Web, permitiendo ordenar estos requerimientos funcionales según la prioridad que se tiene en el negocio y el nivel de complejidad que puede tener durante el desarrollo. Esta lista se encuentra detallada dentro del Manual Técnico en el apartado *Product Backlog* (pág. 33 - 34).

**Sprint Backlog:** El *Sprint Backlog* es una lista de tareas identificadas por el equipo de *Scrum* que se completa durante el *Sprint*. Durante la reunión de planificación del *Sprint*, el equipo selecciona algunas historias de usuario e identifica las tareas necesarias para completarlas [9].

En base a la información obtenida anteriormente y siguiendo la metodología, se procede con el desarrollo de 6 *Sprints*: Configuración del ambiente de desarrollo, Inicio de sesión del usuario, eventos, alimentación, tareas, logística, personal. Finalmente, pruebas de rendimiento y el despliegue del sistema web a producción. Esta lista de actividades por cada *Sprint* se encuentra detallado dentro del Manual Técnico en el apartado *Sprint Backlog* (pág. 35 - 40).

## 2.2. Diseño de interfaces

Una vez determinado los *Sprints* para el desarrollo del proyecto, a continuación, se detalla la herramienta que se ha utilizado para el diseño de las interfaces del sistema web.

### 2.2.1. Herramienta utilizada para el diseño

Para el diseño de las interfaces y cada una de las funcionalidades del sistema web se han realizado prototipos

(*mockups*) utilizando para ello la herramienta Balsamiq mockups, permitiendo la creación de los prototipos del sistema web de forma rápida y sencilla [10]. Tomando en cuenta la experiencia de usuario (UX) como la consistencia, accesibilidad, usabilidad, disponibilidad y simplicidad.

La **Fig. 1** ilustra el prototipo de la interfaz del inicio de sesión del sistema web, mientras que el detalle de los 23 prototipos restantes se encuentra dentro del Manual Técnico en el apartado Diseño de Interfaces (pág. 41 – 52).

**Fig. 1:** Interfaz del inicio de sesión del Sistema Web

## 2.3. Diseño de la arquitectura

Después de haber establecido los *Sprints* para este proyecto conjuntamente con el diseño de las interfaces para el sistema web. A continuación, se describe el patrón de arquitectura que se ha utilizado para el desarrollo

### 2.3.1. Patrón arquitectónico

El sistema web está basado sobre el patrón de arquitectura Modelo, Vista y Controlador (MVC). Consiste en un modelo maduro que ha demostrado su validez en todo tipo de aplicaciones, sobre multitud de lenguajes y plataformas de desarrollo [11]. Permitiendo la clasificación de los componentes de una aplicación en tres grupos (o capas) primordiales: el modelo, la vista, y el controlador, y describe cómo se relacionan entre ellos para mantener una estructura organizada, limpia y acoplada entre las distintas capas [12]. Una vez analizado las ventajas que se tiene al usar un patrón de arquitectura de 3 capas.

La **Fig. 2** ilustra el detalle de la implementación de la arquitectura junto a las herramientas para el desarrollo del sistema web.

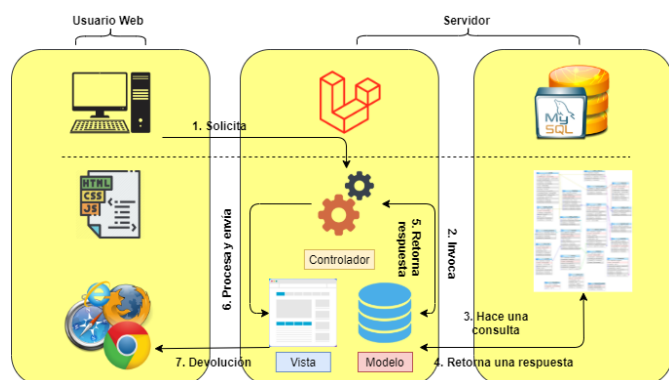


Fig. 2: Arquitectura del Sistema Web

## 2.4. Herramientas de desarrollo

Una vez establecido el patrón arquitectónico y los requerimientos funcionales. A continuación, en las **TABLA III** y **TABLA IV** se detallan las herramientas y librerías utilizadas para desarrollo del Sistema Web.

**TABLA III:** Herramientas utilizadas en el desarrollo del Sistema Web

HERRAMIENTA	JUSTIFICACIÓN
<b>XAMPP</b>	Consiste en un sistema de gestión de base de datos MySQL y de servidor Apache, así como de intérpretes de lenguajes PHP y Perl [13]. Es una herramienta idónea para convertir el computador en un servidor Apache y MySQL, perfecto para el desarrollo del Sistema Web [13].
<i>Composer</i>	Es un gestor de dependencias en proyectos, que nos permite gestionar los paquetes de software en los que se basa nuestro proyecto PHP [14].
<b>JWT</b>	Se define como un mecanismo para propagar de forma segura la información de un usuario, a través de un método de autenticación en servicios API. En dónde el cliente envía sus credenciales de acceso y luego la API le retorna un token que se envía en todas las peticiones que el usuario haga, para que ésta compruebe que tiene acceso a las acciones que se quieren realizar [15].

<b>Laravel</b>	La implementación de este <i>Framework</i> de tipo MVC permite codificar todo el sistema web, haciendo énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad [16].
<b>Material-UI</b>	La implementación de Material-UI permitió la creación del sistema webde forma más rápida, hermosa y accesible [17]. Además de tener un diseño adaptable a los diferentes dispositivos.
<b>JavaScript</b>	Es un lenguaje de secuencias de comandos que permite implementar funciones complejas en las páginas web y crear contenido dinámico en el proyecto [18].
<b>HTML5</b>	La integración de HTML permite en el desarrollo del proyecto distribuir de forma ordenada todo el contenido del Sistema Web en base a etiquetas preestablecidas [19].
<b>CSS</b>	La integración de CSS permite en el desarrollo del proyecto establecer una capa de personalización sobre el sistema web para que sea visualmente agradable para el usuario [20].
<b>MySQL</b>	La integración de MySQL permite trabajar con bases de datos relacionales, almacenar información y administrar los datos de nuestro proyecto a través de las relaciones de las tablas [21].

**TABLA IV:** Librerías utilizadas en el desarrollo del Sistema Web

LIBRERÍA	DESCRIPCIÓN
<b>“react-hook-form”:</b> “^7.2.1”	Librería para usar formularios [22].
<b>“yup”:</b> “^0.32.9”	Librería para validar formularios [23].
<b>“material-ui/icons”:</b> “^4.11.2”	Librería para usar íconos [24].

### 3. RESULTADOS Y DISCUSIÓN

En esta sección, se detallan los resultados obtenidos en cada una de las tareas de los 6 *Sprints* de desarrollo.

#### 3.1. Sprint 0: Configuración del ambiente de desarrollo

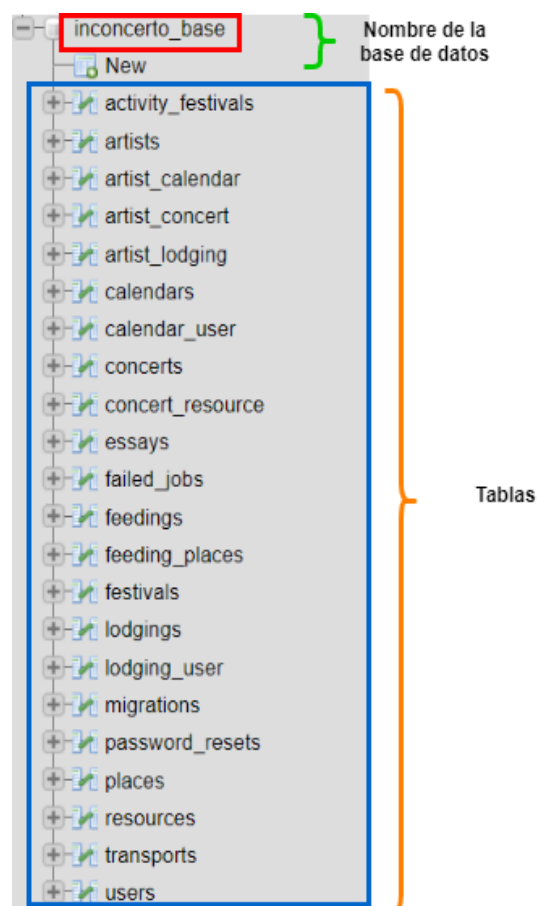
Siguiendo las directrices del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 0* detalla las actividades concernientes a la configuración del ambiente de desarrollo contemplan las siguientes tareas:

- Creación de la Base de Datos.
- Creación del proyecto con Laravel.
- Estructura del proyecto.
- Creación de migraciones, modelos, rutas y controladores.
- Usuarios.
- Requerimientos específicos para el Sistema Web.

##### 3.1.1. Creación de la Base de Datos

Para la creación de la Base de Datos se ha utilizado MySQL, el cual almacena toda la información de usuarios, eventos, recursos, artistas, tareas, notificaciones de los próximos eventos, entre otros.

La **Fig. 3** ilustra la Base de Datos creada en MySQL esta Base de Datos contiene 22 tablas relacionadas entre sí por medio de las claves primarias<sup>2</sup> (PK) y foráneas<sup>3</sup> (FK) siguiendo los parámetros establecidos por el *Framework* para una fácil integración y comunicación. Finalmente, el diseño completo se encuentra dentro del Manual Técnico en el apartado Diseño de la Base de Datos (pág. 52).



**Fig. 3:** Base de datos

##### 3.1.2. Creación del proyecto con Laravel

Se procede a la creación de un proyecto en Laravel, usando la línea de comandos y el sistema gestor de dependencias *Composer*<sup>4</sup>, el cual permite gestionar las instalaciones de todos los paquetes y recursos necesarios para la creación de un proyecto en Laravel. A continuación, se presenta los comandos que se han utilizado para la creación del proyecto. Mientras que la **Fig. 4**, ilustra la ejecución del proyecto de forma satisfactoria.

```
composer create-project laravel/laravel
PROYECTO_Inconcerto-Back-End
```

```
php artisan serve
```

<sup>2</sup> **Clave Primaria:** Es una llave primaria o identificación de una tabla en la Base de Datos relacional.

<sup>3</sup> **Clave foránea:** Es una llave foránea empleada para las tablas secundarias que necesitan conectarse a la principal en la Base de Datos relacional.

<sup>4</sup> **Composer:** es un administrador de dependencias para la instalación, modificación y eliminación de paquetes [32].



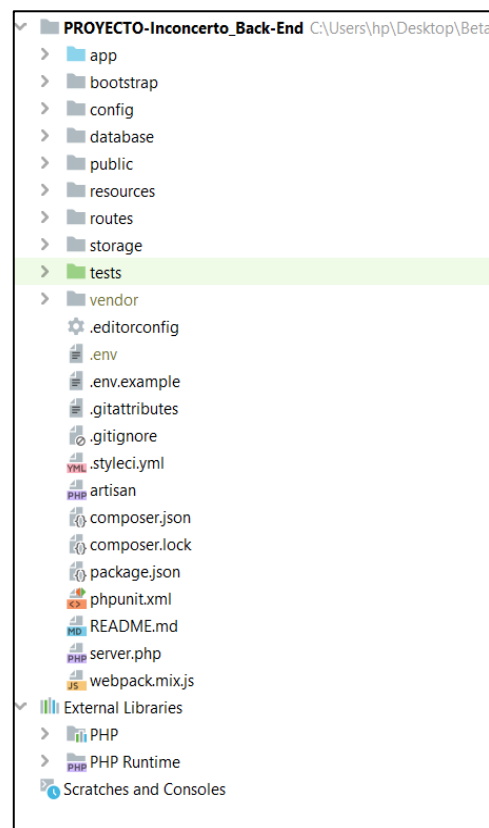


**Fig. 4:** Ejecución del proyecto en Laravel

### 3.1.3. Estructura del proyecto

Para la codificación y estructura de directorios y archivos del sistema web, se ha utilizado *PhpStorm*<sup>5</sup>, tomando en cuenta el patrón arquitectónico de tres capas (MVC) como ilustra la **Fig. 5**. Por último, se detalla el contenido de los directorios y archivos más importantes:

- **Directorio app:** contiene los controladores, modelos, middleware, providers, reglas de validación entre otros.
- **Directorio config:** contiene los archivos de rutas, cors, entre otras configuraciones del proyecto.
- **Directorio database:** contiene las tablas y los seeders creadas por las migraciones.
- **Directorio public:** contiene los archivos y directorios públicos para el usuario final.
- **Directorio resources:** contiene los archivos de la personalización de la respuesta del API.
- **Directorio routes:** contiene el archivo para el direccionamiento de las rutas del proyecto.
- **Archivo .env:** contiene la configuración del proyecto.



**Fig. 5:** Estructura del proyecto

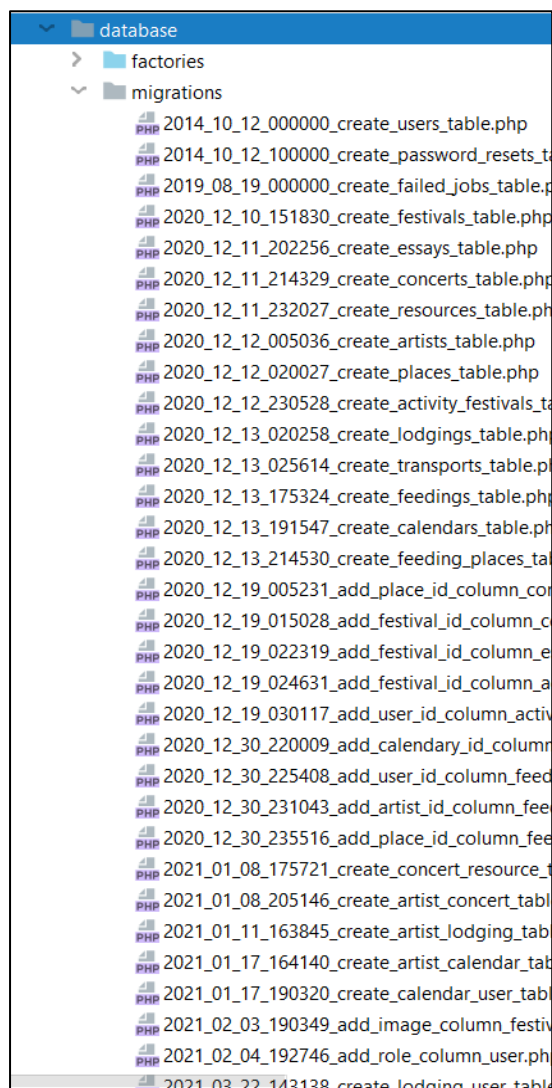
### 3.1.4. Creación de migraciones, modelos, rutas y controladores

#### Migraciones

En Laravel, las migraciones nos permiten crear tablas, modificarlas, eliminarlas y establecer relaciones mediante la consola de comandos [25]. Es por ello que, para obtener las 22 tablas de la Base de Datos se ha utilizado las migraciones de Laravel, el cual permite tener un control más estructurado en los cambios que se realicen en la Base de Datos. A continuación, se presenta el comando que se ha utilizado para la creación de la migración mientras que la **Fig. 6**, ilustra la creación de las migraciones localizadas dentro del directorio *database* subcarpeta *migrations*

```
php artisan make:migration create_users_table
```

<sup>5</sup> *PhpStorm*: editor de texto para HTML, JavaScript y PHP y JavaScript con análisis de código automatizadas [33].



**Fig. 6:** Migraciones creadas

Posterior a la creación de las migraciones, se establece por cada tabla el nombre de la columna, tipo de dato, clave primaria, claves foráneas, timestamps, como ilustra la **Fig. 7**.

```
public function up()
{
    Schema::create( table: 'users', function (Blueprint $table) {
        $table->id();
        $table->string( column: 'name');
        $table->string( column: 'email')->unique();
        $table->timestamp( column: 'email_verified_at')->nullable();
        $table->string( column: 'password');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

**Fig. 7:** Columnas de la tabla users

Por último, el siguiente comando sirve para ejecutar las migraciones automáticamente y con ello crear las tablas en la Base de Datos, como se ilustra en la **Fig. 3**.

*php artisan migrate*

## Modelos

Los modelos son uno de los componentes principales de las aplicaciones desarrolladas bajo el patrón MVC, tienen la responsabilidad de mantener lo que se llama la lógica de negocio, es decir, un conjunto de reglas que deben cumplirse para acceder a los datos, modificarlos, etc. [26]. Para crear un modelo, se debe respetar su convención, el nombre del modelo debe estar en singular y en idioma inglés.

*php artisan make:model User*

## Controladores

Permiten agrupar la lógica peticiones HTTP relacionadas, aquí residen las acciones solicitadas por el cliente que son referenciadas desde las rutas, y de esta forma permiten organizar mejor el código [27]. A continuación, se presenta el comando utilizado para la creación de los controladores en el desarrollo del sistema web.

*php artisan make:controller userController*

## Rutas

Las rutas definen la dirección URL y el método por el cual se puede ingresar a dicha ruta, se encargan de manejar el flujo de solicitudes y respuestas, desde y hacia el cliente (GET, POST, etc.) [28], como se ilustra en la **Fig. 8** un ejemplo de las rutas de la entidad concierto.

```
//CONCIERTO
//GET
Route::get( uri: '/concerts', [ConcertController::class, 'index']);
//GET by ID
Route::get( uri: '/concerts/{concert}', [ConcertController::class, 'show']);
//POST
Route::post( uri: '/concerts', [ConcertController::class, 'store']);
//PUT
Route::put( uri: '/concerts/{concert}', [ConcertController::class, 'update']);
//DELETE
Route::delete( uri: '/concerts/{concert}', [ConcertController::class, 'delete']);
```

**Fig. 8:** Rutas de una entidad

### 3.1.5. Usuarios

En la **Fig. 9** ilustra los usuarios que pueden interactuar con el sistema web, cada uno de ellos tiene un rol y permisos asignados a los diferentes módulos una vez que se realice la autenticación en el sistema web.

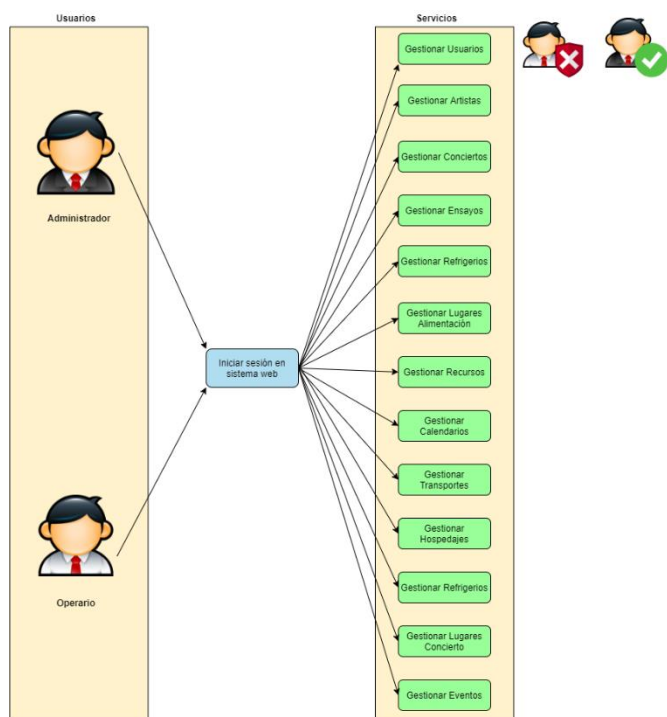


Fig. 9: Perfiles, usuarios y módulos del Sistema Web

### 3.1.6. Requerimientos específicos del Sistema Web

#### Autenticación de usuarios

Todos los usuarios tienen la posibilidad de acceder al sistema web con sus credenciales que son email y contraseña.

El sistema web verifica que el usuario que intenta ingresar esté registrado, lo que implica que, si un usuario no está registrado no podrá gestionar sobre el sistema.

#### Gestión de roles

El usuario con perfil administrador es el encargado de gestionar los roles y permisos para cada usuario en el sistema web. También puede gestionar los demás módulos del sistema web.

#### Gestión de usuarios

El usuario con perfil administrador es quien debe de gestionar los usuarios del sistema web, en este caso si un usuario ingresa por primera vez al sistema web su rol por defecto es operario.

#### Gestión de eventos, alimentación, tareas, logística y personal

Tanto el usuario con perfil administrador como el usuario con perfil operario son encargados de registrar, modificar y eliminar eventos, alimentación, tareas y logística. Además, el usuario administrador tiene la posibilidad de gestionar usuarios.

#### Envío de notificaciones de eventos

Tanto el usuario con perfil administrador como el usuario con perfil operario tienen la posibilidad de enviar notificaciones de los nuevos eventos al crear un nuevo festival.

### 3.2. Sprint 1: Inicio de sesión del usuario administrador, usuarios, eventos, conciertos, ensayos, artistas, calendarios y notificaciones.

Siguiendo el lineamiento del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 1* contempla las siguientes tareas:

- Visualización de la página inicio del sistema web.
- Inicio de sesión del usuario administrador.
- Listado, registro, actualización y eliminación de usuarios.
- Listado, registro, actualización y eliminación de eventos.
- Listado, registro, actualización y eliminación de conciertos.
- Listado, registro, actualización y eliminación de ensayos.
- Listado, registro, actualización y eliminación de artistas.
- Listado, registro, actualización y eliminación de calendarios.
- Notificaciones sobre próximos mantenimientos.

#### 3.2.1. Visualización de la página inicio del Sistema Web

La Fig. 10 ilustra la página de inicio del sistema web la cual contiene el nombre del proyecto, información, un video tutorial y en la parte inferior se encuentran los enlaces a redes sociales. Mientras que, en el Manual de Usuario se puede apreciar detalladamente las demás secciones de la página web.

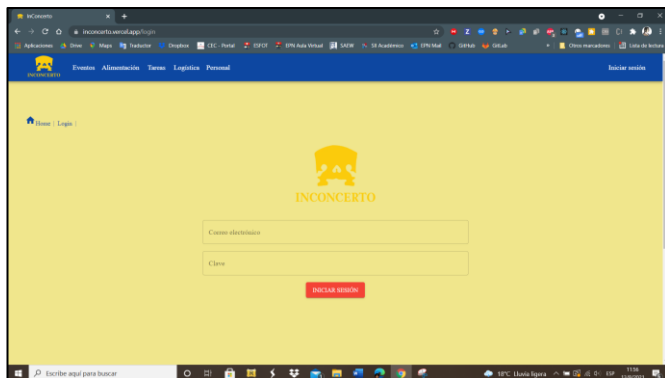


Fig. 10: Página de inicio del Sistema Web



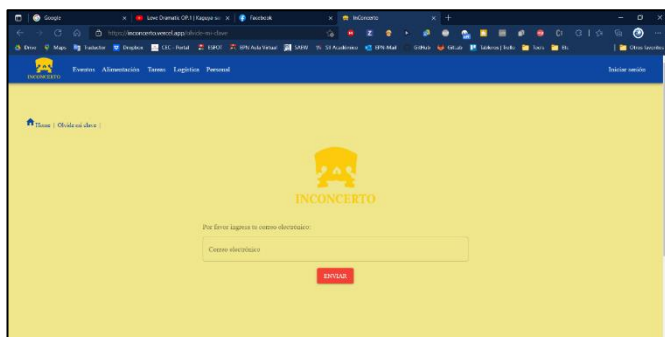
### 3.2.2. Inicio de sesión del usuario administrador

La **Fig. 11** ilustra el formulario y la validación de campos para el ingreso al sistema web. Posterior a ello, el sistema valida las credenciales del usuario en la Base de Datos, si estas son válidas, lo redirecciona a la página home. Por último, el procedimiento para el inicio de sesión y restablecimiento de contraseñas es el mismo para todos los perfiles



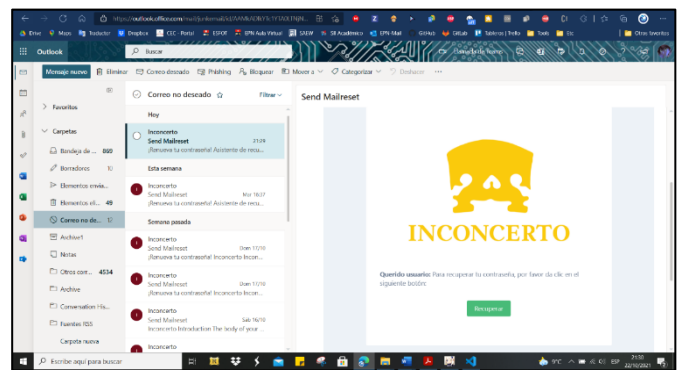
**Fig. 11:** Inicio de sesión para los usuarios registrados en el Sistema Web

En el caso de que el usuario administrador u operario olvide su contraseña, existe la opción de restablecer contraseña. La **Fig. 12** ilustra el formulario de solicitud de correo a donde se envía un enlace de restablecimiento de contraseña.

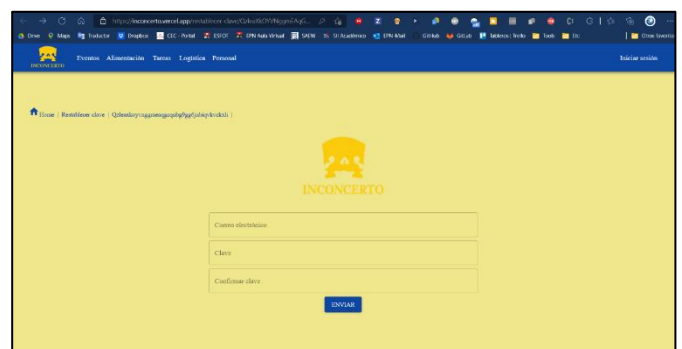


**Fig. 12:** Formulario de solicitud de correo

Finalmente, la **Fig. 13** ilustra la bandeja de entrada del usuario que consta de un enlace para acceder al formulario de restablecimiento de contraseña; donde cambia sus credenciales de acceso para poder ingresar al sistema web, como lo indica la **Fig. 14**



**Fig. 13:** Bandeja de entrada del usuario



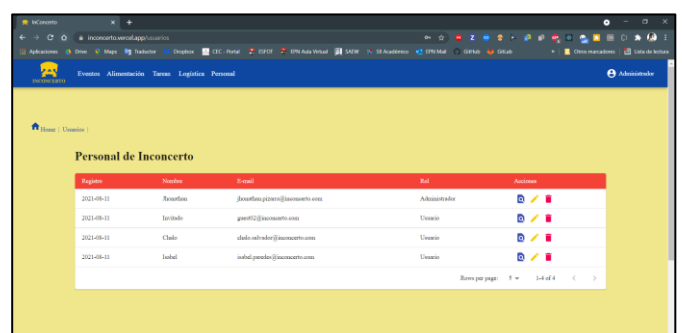
**Fig. 14:** Formulario de restablecimiento de contraseña

### 3.2.3. Listado, registro, actualización y eliminación de usuarios

El procedimiento para registrar, modificar y eliminar usuarios del sistema web lo puede realizar el usuario administrador únicamente.

La **Fig. 15** ilustra el listado de usuarios registrados en el sistema web, el cual presenta información como: nombre, correo electrónico y opciones que se pueden realizar cuando sobre el usuario.

Por último, los usuarios que son registrados en el Sistema Web deben de tener un correo organizacional, si un usuario es identificado con un correo diferente es eliminado del sistema web manualmente por el administrador.

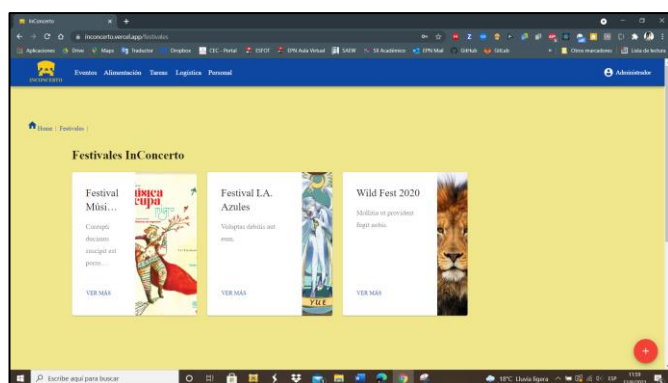


**Fig. 15:** Listado de usuarios

### 3.2.4. Listado, registro, actualización y eliminación de eventos

El procedimiento para registrar, modificar y eliminar eventos (llámense también festivales) lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 16** ilustra el listado de eventos registrados en el sistema web, el cual presenta información como: nombre, descripción, imagen y las opciones que se pueden realizar sobre el evento. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

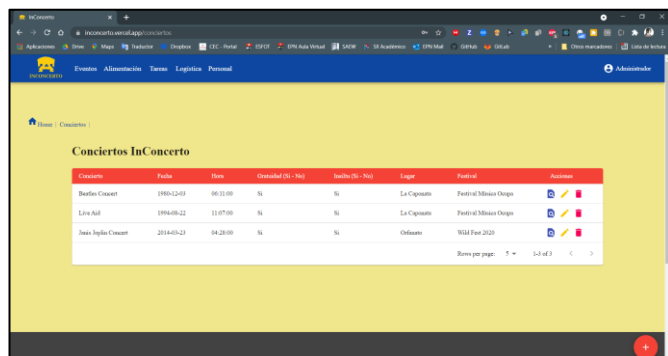


**Fig. 16:** Listado de eventos

### 3.2.5. Listado, registro, actualización y eliminación de conciertos

El procedimiento para registrar, modificar y eliminar conciertos lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 17** ilustra el listado de los conciertos registradas en el sistema web, el cual presenta información como: fecha de concierto, hora, nombre, gratuidad, modalidad y las opciones que se pueden realizar sobre cada concierto. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

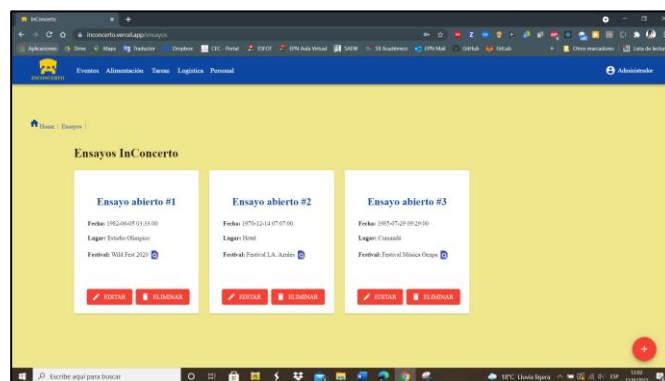


**Fig. 17:** Listado de conciertos

### 3.2.6. Listado, registro, actualización y eliminación de ensayos

El procedimiento para registrar, modificar y eliminar ensayos lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 18** ilustra el listado de los ensayos registrados en el sistema web, el cual presenta información como: fecha del ensayo, nombre, lugar, festival para el que se ensaya y las opciones que se pueden realizar sobre el ensayo. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

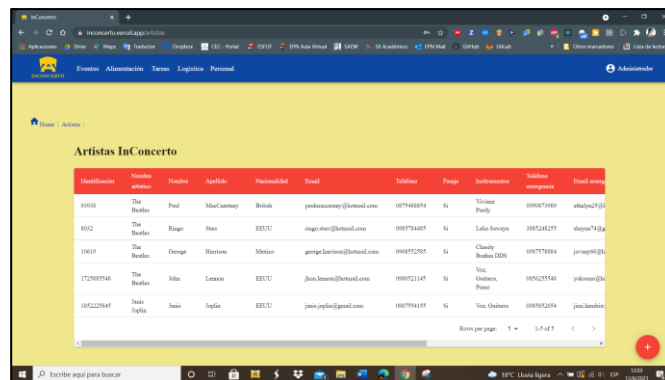


**Fig. 18:** Listado de ensayos

### 3.2.7. Listado, registro, actualización y eliminación de artistas

El procedimiento para registrar, modificar y eliminar artistas lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 19** ilustra el listado de artistas, el cual presenta información como: cédula, nombre artístico, nombre del artista, apellido, nacionalidad, mail, teléfono, pasaje, instrumentos que toca, teléfono de emergencia, mail de emergencia, grupo alimenticio, observación y las acciones que se pueden realizar sobre el artista. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

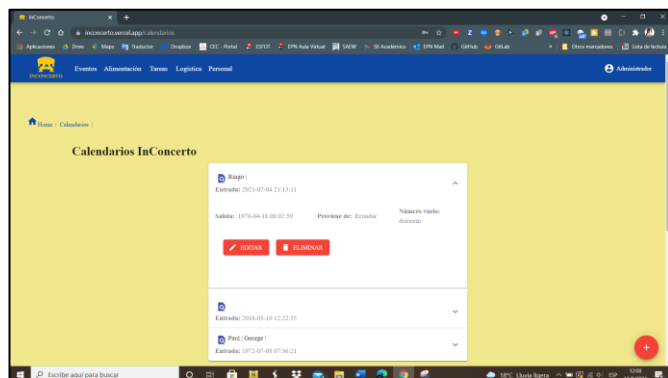


**Fig. 19:** Listado de artistas

### 3.2.8. Listado, registro, actualización y eliminación de calendarios

El procedimiento para registrar, modificar y eliminar calendarios lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 20** ilustra el listado calendarios, el cual presenta información como: llegada del artista, salida del artista, país del que viene, número de vuelo, y las acciones que se pueden realizar sobre el calendario. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

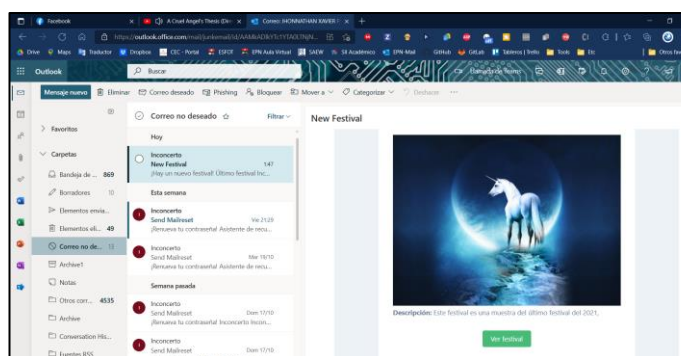


**Fig. 20:** Listado de calendarios

### 3.2.9. Notificaciones sobre próximos eventos

El procedimiento para enviar notificaciones lo puede realizar tanto el usuario administrador como el usuario operario.

La **Fig. 21** ilustra una notificación recibida automáticamente luego de crear un nuevo festival, donde constan: nombre del nuevo evento, descripción e imagen. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.



**Fig. 21:** Notificación del próximo evento

## 3.3. Sprint 2: Transportes, hospedajes, lugares de concierto, tareas.

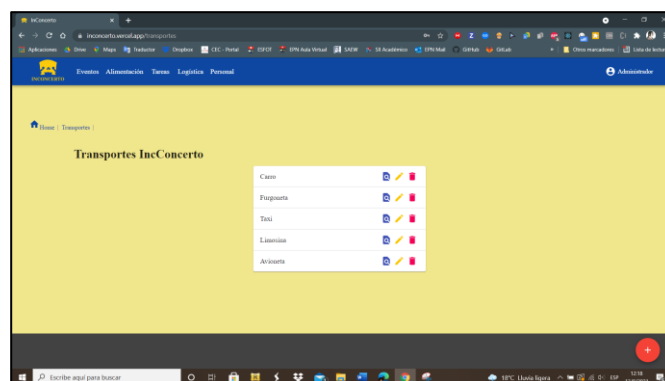
Siguiendo las directrices del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 2* se contemplan las siguientes tareas:

- Listado, registro, actualización y eliminación de transportes.
- Listado, registro, actualización y eliminación de hospedajes.
- Listado, registro, actualización y eliminación de lugares de concierto.
- Listado, registro, actualización y eliminación de tareas.

### 3.3.1. Listado, registro, actualización y eliminación de transportes

El procedimiento para registrar, modificar y eliminar calendarios lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 22** ilustra el listado de transportes, el cual presenta información como: tipo, capacidad, disponibilidad, matrícula, y las acciones que se pueden realizar sobre el transporte. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.

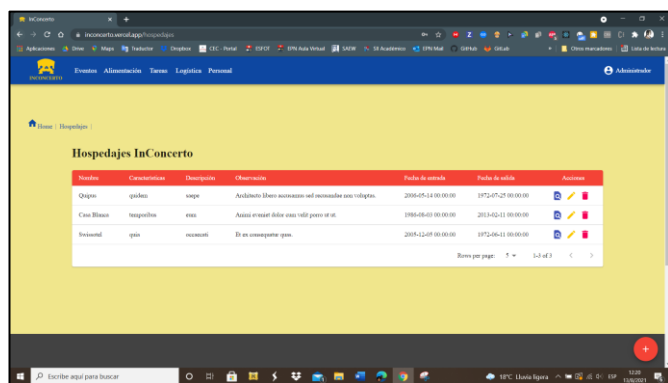


**Fig. 22:** Listado de transportes

### 3.3.2. Listado, registro, actualización y eliminación de hospedajes

El procedimiento para registrar, modificar y eliminar hospedajes lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 23** ilustra el listado de hospedajes, el cual presenta información como: tipo, nombre, descripción, observación, entradas, salidas, y las acciones que se pueden realizar sobre el hospedaje. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.



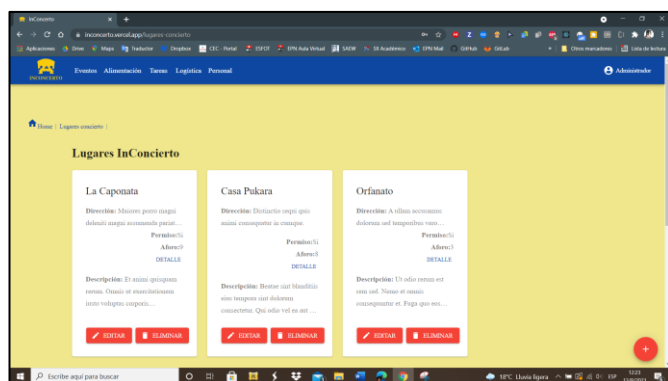
Nombre	Categorías	Descripción	Observación	Fecha de entrada	Fecha de salida	Acciones
Quipo	quipo	supe	Acción: libro accesorio del momento con volapés.	2000-07-24 00:00:00	1972-07-27 00:00:00	[iconos]
Casa Blanca	temples	en	Aviso: en el dolor con volapés en el.	1990-08-03 00:00:00	2013-02-11 00:00:00	[iconos]
Islandia	qto	recuent	Es en conaparte qto.	2005-12-07 00:00:00	1972-06-11 00:00:00	[iconos]

Fig. 23: Listado de hospedajes

### 3.3.3. Listado, registro, actualización y eliminación de lugares de concierto

El procedimiento para registrar, modificar y eliminar lugares de concierto lo pueden realizar tanto el usuario administrador, como el usuario operario.

La Fig. 24 ilustra el listado de lugares de concierto, el cual presenta información como: nombre, dirección, disponibilidad, aforo, y las acciones que se pueden realizar sobre el lugar. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.



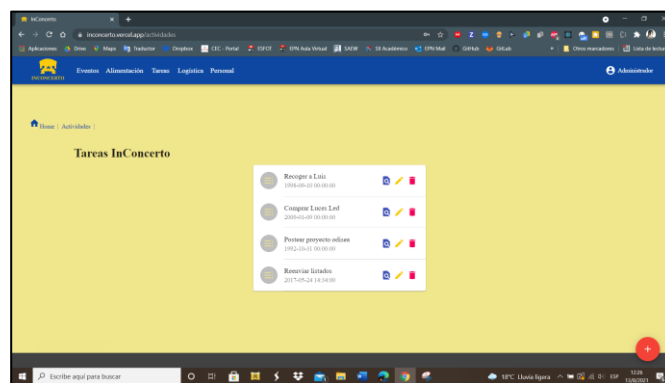
Nombre	Dirección	Disponibilidad	Aforo	Acciones
La Capemata	Dirección: Muestra para supe. Admisión: supe conaparte supe.	Disponibilidad: [iconos]	Aforo: [iconos]	[iconos]
Casa Pikara	Dirección: Direcciona supe qto. supe conaparte supe.	Disponibilidad: [iconos]	Aforo: [iconos]	[iconos]
Orfanato	Dirección: A supe conaparte. Admisión: supe conaparte supe.	Disponibilidad: [iconos]	Aforo: [iconos]	[iconos]

Fig. 24: Listado de lugares de concierto

### 3.3.4. Listado, registro, actualización y eliminación de tareas

El procedimiento para registrar, modificar y eliminar tareas lo pueden realizar tanto el usuario administrador, como el usuario operario.

La Fig. 25 ilustra el listado de tareas, el cual presenta información como: nombre, fecha, descripción, observación, festival al que pertenece, encargado, y las acciones que se pueden realizar sobre la tarea. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.



Nombre	Fecha	Descripción	Observación	Festival	Encargado	Acciones
Registrar a Lili	2000-08-03 00:00:00					[iconos]
Registrar Lili	2000-08-03 00:00:00					[iconos]
Registrar Lili	2000-08-03 00:00:00					[iconos]
Registrar Lili	2000-08-03 00:00:00					[iconos]
Registrar Lili	2000-08-03 00:00:00					[iconos]

Fig. 25: Listado de tareas

## 3.4. Sprint 3: Recursos

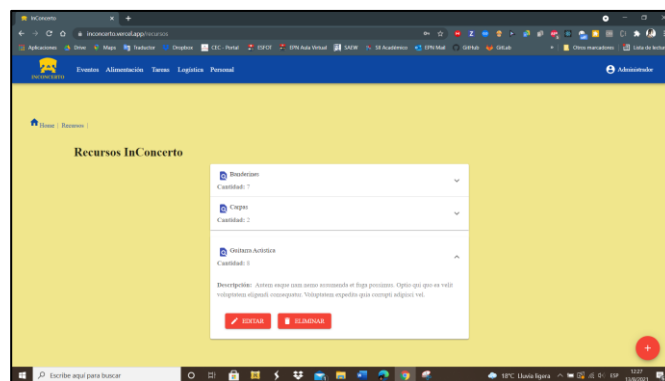
Siguiendo las directrices del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 3* se contemplan las siguientes tareas:

- Listado, registro, actualización y eliminación de recursos.

### 3.4.1. Listado, registro, actualización y eliminación de recursos

El procedimiento para registrar, modificar y eliminar recursos lo pueden realizar tanto el usuario administrador, como el usuario operario.

La Fig. 26 ilustra el listado de recursos, el cual presenta información como: nombre, cantidad, descripción, y las acciones que se pueden realizar sobre el recurso. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.



Nombre	Cantidad	Descripción	Acciones
Descripción	Cantidad: 7		[iconos]
Categoría	Cantidad: 2		[iconos]
Orfanato	Cantidad: 1		[iconos]

Fig. 26: Lista de recursos

## 3.5. Sprint 4: Cuadros alimenticios, lugares de alimentación

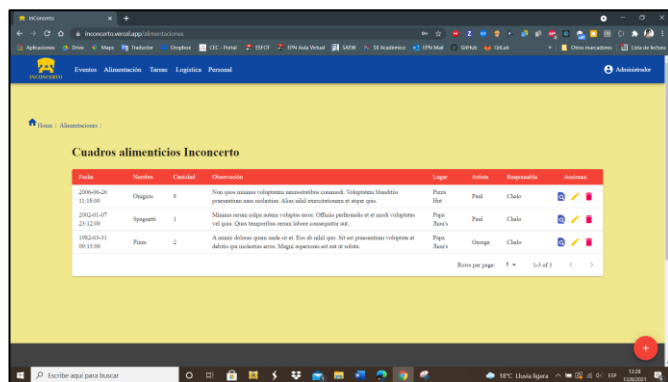
Siguiendo las directrices del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 4* se contemplan las siguientes tareas:

- Listado, registro, actualización y eliminación de cuadros alimenticios.
- Listado, registro, actualización y eliminación de lugares de alimentación.

### 3.5.1. Listado, registro, actualización y eliminación de cuadros alimenticios

El procedimiento para registrar, modificar y eliminar cuadros alimenticios lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 27** ilustra el listado de tareas, el cual presenta información como: nombre, fecha, observación, cantidad, artista al que pertenece, usuario responsable, lugar y las acciones que se pueden realizar sobre el cuadro alimenticio. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.



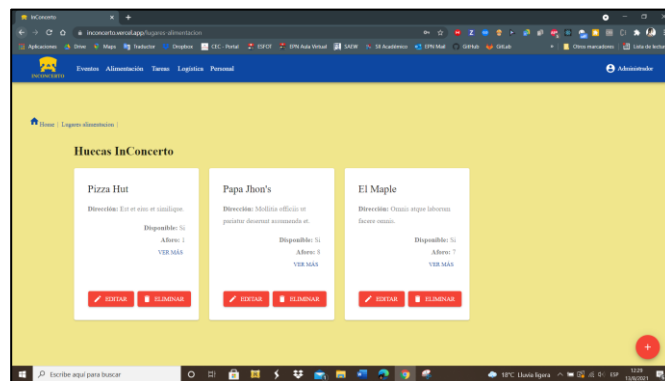
Fecha	Nombre	Cantidad	Observación	Lugar	Artista	Responsable	Acciones
2024-05-20 11:18:00	Chagrin	8	Non-que minime voluptatem accusantium molestiae. Voluptatem blanditiis praesentium aeneanest.	Papa Ihon's	Paul	Chado	[Iconos de acciones]
2024-05-27 23:12:00	Syngenti	1	Mauris venen volupis aenean volupis aenean. Officia perferendo et et aenean voluptatem vel quis. Quis tempusque aenean libero accipietur eae.	Papa Ihon's	Paul	Chado	[Iconos de acciones]
1992-05-31 09:13:00	Papa	2	Aenean dolore quam nulla et. Eius et rati quis. Sit ut praesentibus voluptatem et debitis qui suscipit aenean. Nuncia sequitur et ut ut et soluta.	Papa Ihon's	George	Chado	[Iconos de acciones]

**Fig. 27:** Lista de cuadros alimenticios

### 3.5.2. Listado, registro, actualización y eliminación de lugares alimenticios

El procedimiento para registrar, modificar y eliminar lugares alimenticios lo pueden realizar tanto el usuario administrador, como el usuario operario.

La **Fig. 28** ilustra el listado de tareas, el cual presenta información como: nombre, dirección, aforo y las acciones que se pueden realizar sobre los lugares alimenticios. Mientras que el procedimiento, interfaces y validaciones correspondientes a este módulo se detalla en el Manual de Usuario.



**Fig. 28:** Lista de lugares alimenticios

## 3.6. Sprint 5: Inicio de sesión del usuario operario

Siguiendo las directrices del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 5* se contemplan las siguientes tareas:

- Inicio de sesión del usuario operario

### 3.6.1. Inicio de sesión del usuario operario

La **Fig. 11** ilustra el formulario y la validación de campos para el ingreso al sistema web. Posterior a ello, el sistema valida las credenciales del usuario en la Base de Datos, si estas son válidas, lo redirecciona a la página home. Por último, el procedimiento para el inicio de sesión y restablecimiento de contraseñas es el mismo para todos los perfiles

## 3.7. Sprint 6: Pruebas y despliegue del Sistema Web

Siguiendo el lineamiento del Manual Técnico en el apartado *Sprint Backlog*. El *Sprint 6* se contemplan las siguientes tareas

- Pruebas de carga.
- Pruebas de compatibilidad.
- Pruebas de funcionalidad.
- Despliegue del sistema web.

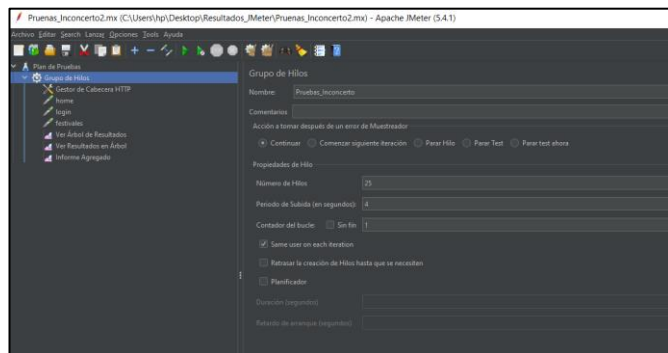
### 3.7.1. Pruebas de carga

Las pruebas de rendimiento sirven para determinar la rapidez de un sistema mientras realiza una tarea en determinadas condiciones de trabajo [29]. El tipo de pruebas de rendimiento más sencillo, son las pruebas de carga, estas permiten observar el comportamiento de una aplicación bajo una cantidad esperada de peticiones. Esta carga puede interpretarse como el número estimado de usuarios concurrentes al sistema web, que realizan una cantidad determinada de transacciones, durante el tiempo que dura la carga. Esta prueba recopila y presenta los tiempos de respuesta de todas las transacciones realizadas a la aplicación [29].

Existen numerosas herramientas, tanto open source como privativas, para realizar las pruebas de rendimiento: NeoLoad,



LoadRunner, LoadUI, WebLOAD, etc [29]. Una de las más populares es JMeter (open source), que nos permite conocer el comportamiento funcional del sistema web frente a una gran cantidad de peticiones y de esta manera determinar la estabilidad del mismo. La **Fig. 29** ilustra la configuración y la creación de un grupo de hilos de prueba llamado: “Pruebas\_Inconcerto”.



**Fig. 29:** Configuración del grupo de hilos en JMeter

Donde:

- **Número de Hilos:** es la cantidad de usuarios que van a realizar una petición.
- **Periodo de subida:** es la representación en segundos su representación es en segundos del intervalo de tiempo que le toma en llegar al máximo hilos.
- **Contador del Bucle:** indica la cantidad de peticiones que se envían por cada hilo.

Posterior a ello se procede a realizar la primera prueba con una carga de 100 peticiones. Mientras que los resultados de las dos pruebas restantes se detallan en el apartado Pruebas de funcionalidad del Manual Técnico (pág. 53).

### Primera prueba (100 Peticiones)

A continuación, la **Fig. 30** ilustra el resultado que se ha obtenido de la primera prueba, en donde:

- El porcentaje de error es 0% lo que significa que todas las peticiones se ejecutan exitosamente con un tiempo promedio de 303 milisegundos (0.303 segundos) aproximadamente.
- La ruta Home es la que más tiempo le tomado al servidor en responder las solicitudes con un tiempo de 1048 milisegundos (1.048 segundos) aproximadamente.
- La ruta página de inicio de sesión del sistema, es la más rápida en responder las solicitudes con un tiempo de 127 milisegundos (0.127 segundos) aproximadamente.

Informe Agregado									
Nombre: Informe Agregado									
Comentarios									
Escribir todos los datos a Archivo									
Nombre de archivo						Navegar...	Log/Mostrar sólo:	Escribir en:	
Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Min	Máx	% Error
home	33	628	604	656	697	1048	546	1048	0,00%
login	33	141	143	156	159	162	127	162	0,00%
festivales	33	141	135	158	159	182	128	182	0,00%
Total	99	303	148	619	645	991	127	1048	0,00%

**Fig. 30:** Resultados de la prueba #1 con 100 peticiones simultaneas

Con los resultados obtenidos de las pruebas de carga se determina que el tiempo de respuesta a las peticiones son adecuadas, obteniendo tiempos de respuestas aceptables para su uso

### 3.7.2. Pruebas de compatibilidad

Las pruebas de compatibilidad son las pruebas que se realizan en un software o aplicación determinada y que comprueban que el desarrollo es compatible con todos los navegadores de internet y todos los sistemas operativos del mercado [30]. Permitiendo verificar el correcto comportamiento de las interfaces y con ello cumplir con los requerimientos del usuario.

Esta prueba se realiza en tres de los navegadores web más utilizados por los usuarios en la actualidad como se presenta en la **TABLA V**. Mientras que los resultados de las dos pruebas restantes se detallan en el apartado de Pruebas de compatibilidad dentro del Manual Técnico (pág. 54 – 56).

**TABLA V:** Versiones de navegadores

NOMBRE	VERSIÓN	OBSERVACIÓN
Microsoft Edge	83.0.478.37	Totalmente funcional
Google Chrome	83.0.4103.61	Totalmente funcional
Mozilla Firefox	76.0.1	Totalmente funcional

### 3.7.3. Prueba de funcionalidad

Las pruebas funcionales se realizan para comprobar la usabilidad, funcionalidad y otras características críticas para el negocio. Las pruebas funcionales tienen por objetivo garantizar que las funcionalidades del software se comporten según lo esperado sin ningún problema [31]. A continuación, la **Tabla VI** detalla un ejemplo de la prueba de funcionalidad que se ha realizado Mientras los resultados de las 55 pruebas restantes se detallan en el apartado Pruebas de funcionalidad en el Manual Técnico (pág. 56 – 91).

**Tabla VI:** Prueba de funcionalidad - Inicio de sesión del administrador

PRUEBA DE FUNCIONALIDAD	
Identificador (ID): PA001	Identificador historia de usuario: HU001
<b>Nombre prueba de funcionalidad:</b> Inicio de sesión del Administrador	
<b>Descripción:</b> El usuario administrador debe ingresar al Sistema Web.	
<b>Pasos de ejecución:</b> Ingresar el correo y contraseña (proporcionado por el autor del mismo). Presionar el botón "Ingresar".	
<b>Resultado deseado:</b> El usuario administrador, inicia sesión con su respectivo perfil.	
<b>Evaluación de la prueba:</b> Resultado exitoso. El Sistema Web permite el ingreso del usuario administrador. Aprobación del cliente 100%.	

### 3.7.4. Despliegue del Sistema Web

En la **Fig. 31** se ilustra el despliegue del sistema web. El proceso completo de configuración y despliegue se detallan en el Manual de Instalación (pág. 2).

**Fig. 31:** Despliegue del Sistema Web

Finalmente, la URL para acceder al Sistema Web es:

<https://inconcerto.vercel.app>

Muestra de ello, la jefa de producción, ha otorgado un certificado, el mismo ratifica el cumplimiento de todos los requerimientos y funcionalidades del sistema web solicitados al comienzo del proyecto. El certificado se adjunta en el Manual Técnico (pág. 92).

## 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- Al momento de desarrollar un proyecto siempre es importante tener una idea clara de lo que se pretende realizar, marcando objetivos claros, herramientas necesarias, metodologías y técnicas que promuevan las buenas prácticas y organización. De esa manera se concluye que una adecuada recopilación de los requerimientos satisfacer las expectativas y necesidades tanto de los clientes como de los desarrolladores.
- La metodología ágil *Scrum*, nos brinda reglas, artefactos y funciones fáciles de entender, permitiendo eliminar las ambigüedades durante el desarrollo. Además, la demarcación de funciones y tiempos aseguran que haya transparencia y colaboración durante todo el ciclo de desarrollo.
- La arquitectura MVC implementada en el Sistema Web hace de este un sistema escalable y fácil de mantener, permitiendo implementar nuevos módulos sin afectar a los otros componentes de nuestra aplicación ni hacer configuraciones adicionales.
- La utilización del Sistema Gestor de Base de Datos (SGBD) MySQL, permite almacenar los datos del colectivo y al ser un SGBD relacional ayuda al rendimiento, escalabilidad e integración con las diferentes herramientas utilizadas para el desarrollo del proyecto.
- La utilización del *Framework* Laravel nos ayuda a mantener una estética de código ordenado y al ser un framework de tipo MVC permite que el desarrollo del Sistema Web funcione como una herramienta tecnológica, para cumplir los objetivos propuestos.
- Para comprobar el correcto funcionamiento del Sistema Web se han realizado pruebas de carga, compatibilidad y funcionalidad, las cuales permiten identificar problemas, medir su eficiencia y comprobar su eficacia.

### 4.2 Recomendaciones

- Una vez puesto en producción el Sistema Web se recomienda que los administradores establezcan las frecuencias de mantenimiento con el objetivo de tener un sistema actualizado, altamente disponible y siempre dispuesto a atender los requerimientos del cliente.
- En el caso de que se requieran mejoras en el Sistema Web se recomienda realizar un previo análisis de compatibilidad de las librerías utilizadas y de ser

necesario, actualizarlas ya que puede ocasionar bugs en el sistema.

- Se recomienda realizar backups de la Base de Datos cuando se requiera hacer algún cambio y que los mismos se envíen al líder del proyecto para su almacenamiento y custodia.
- Se recomienda que el administrador y encargados del laboratorio establezcan políticas de seguridad para el uso del Sistema Web.
- Se recomienda trazar un plan de contingencia en caso de imprevistos que no permitan realizar algún objetivo trazado durante el desarrollo del proyecto.

## REFERENCIAS

- [1] S. Gangotena y R. Becerra, «MusicaOcupa,» Carla Moncayo, 06 2017. [En línea]. Available: <https://www.musicaocupa.com/quienessomos>. [Último acceso: 15 01 2020].
- [2] C. Z. Mariño, «Los tres roles de la metodología SCRUM,» [En línea]. Available: <https://xn--zoraidaceballosdemario-4ec.info/scrums/scrums-roles-y-responsabilidades-del-scrum-team/>. [Último acceso: 16 Junio 2021].
- [3] P. C. Alejandro Pérez, «METODOLOGÍA SCRUM,» Sinnaps, 2019. [En línea]. Available: <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum>. [Último acceso: 07 02 2020].
- [4] Schwaber, K., & Sutherland, J., «The Scrum guide,» noviembre 2017. [En línea]. Available: Ken Schwaber and Jeff Sutherland, 2017.
- [5] V. S.A., «Artefactos Scrum ¿Qué son y para qué sirven?,» Grupo IBM; IBM España S.A., 2021. [En línea]. Available: <https://www.viewnext.com/artefactos-scrum/>. [Último acceso: 17 06 2021].
- [6] proyectosagiles.org, «Lista de objetivos / requisitos priorizada (Product Backlog),» [En línea]. Available: Lista de objetivos / requisitos priorizada (Product Backlog). [Último acceso: 14 octubre 2020].
- [7] M. REHKOPF, «Historias de usuario con ejemplos y plantilla,» Atlassian, 2021. [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories>. [Último acceso: 17 06 2021].
- [8] K. Kaizenia, «¿Qué son los artefactos Scrum?,» Grupo Oyarde S.A, 2021. [En línea]. Available: <https://kzi.mx/que-son-los-artefactos-de-scrum/>.
- [9] M. Cohn, «Sprint Backlog,» Mountain Goat Software, 2021. [En línea]. Available: <https://www.mountaingoatsoftware.com/agile/scrums/scrums-tools/sprint-backlog>. [Último acceso: 17 06 2021].
- [10] P. SANTAMARÍA, «Balsamiq mockup, una muy buen herramienta para esbozar tus futuras apps,» Applesfera, 25 11 2013. [En línea]. Available: <https://www.applesfera.com/aplicaciones-os-x-1/balsamiq-mockup-una-muy-buen-herramienta-para-esbozar-tus-futuras-apps>. [Último acceso: 17 06 2021].
- [11] U. D. ALICANTE, «Servicio de Informática ASP.NET MVC Framework,» Universidad de Alicante, [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 18 06 2021].
- [12] J. M. Aguilar, «¿Qué es el patrón MVC en programación y por qué es útil?,» Campus MVP, 15 10 2019. [En línea]. Available: <https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>. [Último acceso: 18 06 2021].
- [13] J. G. Gimeno, «Xampp. Herramienta indispensable para desarrollo web,» Código Binario, 12 06 2017. [En línea]. Available: <https://www.codigo-binario.es/xampp-herramienta-para-dev-web/>. [Último acceso: 18 06 2021].
- [14] M. Alvarez, «Composer, gestor de dependencias para PHP,» desarrolloweb.com, 24 01 2020. [En línea]. Available: <https://desarrolloweb.com/articulos/composer-gestor-dependencias-para-php.html>. [Último acceso: 18 06 2021].
- [15] J. L. G. Folgado, «Autenticación con JWT en Laravel,» NIGMACODE, 20201. [En línea]. Available: <https://www.nigmacode.com/laravel/jwt-en-laravel/>. [Último acceso: 18 06 2021].
- [16] C. S.L., «Laravel,» desarrolloweb.com, [En línea]. Available: <https://desarrolloweb.com/home/laravel>. [Último acceso: 18 06 2021].
- [17] S. Silbermann, «Material-UI,» DoiT International, 2021. [En línea]. Available: <https://material-ui.com/es/>. [Último acceso: 18 06 2021].
- [18] M. contributors, «¿Qué es JavaScript?,» MDN Web Docs, [En línea]. Available: [https://developer.mozilla.org/es/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript). [Último acceso: 18 06 2021].
- [19] OpenWebinars, «Qué es HTML5,» 20 enero 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-html5/>.
- [20] A. Robledano, «Qué es CSS y para qué sirve,» OpenWebinars 2021. [En línea]. Available: <https://openwebinars.net/blog/que-es-css/>. [Último acceso: 18 06 2021].
- [21] NeoAttack, «MySQL,» 2021. [En línea]. Available: <https://neoattack.com/neowiki/mysql/>. [Último acceso: 18 06 2021].
- [22] E. Aristizábal, «Formularios de usuarios dinámicos con React Hooks,» Kushki, 16 11 2020. [En línea]. Available: <https://www.kushki.com/en/blog/como-creamos-nuestros-formularios-de-usuario-en-kushki>. [Último acceso: 18 06 2021].
- [23] A. Tinwala, «Formularios en React: React Hook Forms con Material UI y YUP,» DEV Community, 06 09 2020. [En línea]. Available: <https://dev.to/ammartinwala52/forms-in-react-react-hook-forms-with-material-ui-and-yup-4gfb#sub2>. [Último acceso: 18 06 2021].
- [24] ICHI.PRO, «Guía completa para usar iconos de diseño de materiales en React,» ICHI.PRO, 2020. [En línea]. Available: <https://ichi.pro/es/guia-completa-para-usar-iconos-de-diseno-de-materiales-en-react-233576377954331>. [Último acceso: 23 06 2021].
- [25] M. Canales Zapata, «INTRODUCCIÓN A LAS MIGRACIONES CON LARAVEL FRAMEWORK,» ANEXSOFT, 2021. [En línea]. Available: <https://anexsoft.com/introduccion-a-las-migraciones-con-laravel-framework>. [Último acceso: 22 06 2021].
- [26] C. R. Ruso, «Introducción a modelos en Laravel,» desarrolloweb.com, 14 09 2015. [En línea]. Available: <https://desarrolloweb.com/articulos/introduccion-modelos-laravel.html>. [Último acceso: 22 06 2021].
- [27] D. Palacios, «Controladores en Laravel,» Styde, 27 10 2017. [En línea]. Available: <https://styde.net/controladores-en-laravel>. [Último acceso: 22 06 2021].

- [28] LaravelTIP.com, «¿Qué son los Routes en Laravel?», 01 02 2019 [En línea]. Available: <https://www.laraveltip.com/que-son-los-routes-en-laravel/>. [Último acceso: 22 06 2021].
- [29] J. F. Sánchez, «Pruebas de rendimiento con JMeter.», 02 07 2018 [En línea]. Available: <https://www.sdos.es/blog/pruebas-de-rendimiento-con-jmeter-ejemplos-basicos>. [Último acceso: 23 06 2021].
- [30] F. Acero, «PRUEBAS DE COMPATIBILIDAD Y USABILIDAD», Prezi, 01 11 2013. [En línea]. Available <https://prezi.com/e164alfch6wx/pruebas-de-compatibilidad-y-usabilidad/>. [Último acceso: 23 06 2021].
- [31] G. Lee, «Tipos de pruebas de software: diferencias y ejemplos», loadview, 16 10 2020. [En línea]. Available <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>. [Último acceso: 23 06 2021].
- [32] Yair, «Qué es Composer y cómo usarlo», Styde, 23 12 2019. [En línea]. Available: <https://styde.net/que-es-composer-y-como-usarlo/>. [Último acceso: 23 06 2021].
- [33] Darkcrizt, «PhpStorm, un excelente IDE para PHP multiplataforma», LinuxAdictos, [En línea]. Available <https://www.linuxadictos.com/phpstorm-un-excelente-ide-para-php-multiplataforma.html>. [Último acceso: 22 06 2021].