

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE APLICACIÓN WEB DE PROMOCIÓN DE EMPRESARIOS DE LA COMUNIDAD DE LA ESFOT BACKEND

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

MARÍA JOSÉ CHALÁ SANTACRUZ

DIRECTOR: ING. JUAN PABLO ZALDUMBIDE PROAÑO

DMQ, marzo 2023

CERTIFICACIONES

Yo, María José Chalá Santacruz declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

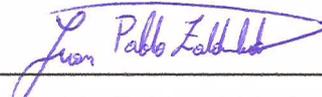


MARÍA JOSÉ CHALÁ SANTACRUZ

maria.chala@epn.edu.ec

mariajochala@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por María José Chalá Santacruz, bajo mi supervisión.



Juan Pablo Zaldumbide Proaño

DIRECTOR

juan.zaldumbide@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



María José Chalá Santacruz

DEDICATORIA

Este trabajo de titulación lo dedico a mis padres, quienes a pesar de todo lograron brindarme siempre una mano amiga para ayudarme a levantar y seguir en esta etapa.

María José Chalá Santacruz

AGRADECIMIENTO

Agradezco a quienes me acompañaron durante mi proceso de formación profesional principalmente a mis padres y hermanos por el apoyo, cariño y sobre todo la paciencia brindada en esta etapa; a cada uno de mis profesores quienes fueron parte crucial en este proceso, ya que, por medio de los conocimientos impartidos fue posible alcanzar los objetivos planteados en tareas y proyectos empleados a lo largo de la carrera y así permitirme escalar en conocimientos y encontrarme culminando este proceso. Finalmente, pero no menos importante, quiero agradecer a mis compañeros con quienes compartimos no solo conocimientos, desveladas y frustraciones; sino también los sueños y anhelos por culminar con éxito la carrera.

María José Chalá Santacruz

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT.....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	1
1.2 Objetivos específicos	1
1.3 Alcance.....	1
1.4 Marco Teórico.....	2
Metodología.....	2
Metodología ágil	3
Backend.....	3
2 METODOLOGÍA.....	4
2.1 Metodología de Desarrollo	4
Roles.....	4
Recopilación de Requerimientos	5
2.2 Diseño de la arquitectura	6
Patrón arquitectónico	7
2.3 Herramientas de desarrollo.....	7
3 RESULTADOS	9
3.1 Iteración 0. Configuración del ambiente de desarrollo	9
3.2 Iteración 1. Autenticación de usuario	12
3.3 Iteración 2. Implementación de CRUD de usuarios, emprendimientos y videoconferencias.....	16
3.4 Iteración 3. Almacenamiento de imágenes en Cloudinary.....	28
3.5 Iteración 4. Despliegue a Heroku y pruebas del sistema	29
4 Conclusiones	36
5 RECOMENDACIONES.....	37
6 REFERENCIAS BIBLIOGRÁFICAS.....	38

7 ANEXOS.....	1
ANEXO I Certificado Turnitin.....	2
ANEXO II Manual técnico.....	3
ANEXO III Manual de usuario	17
ANEXO IV Manual de Instalación.....	18

RESUMEN

Actualmente, la comisión Emprende, quien se encarga de fomentar la creación de emprendimientos dentro de la comunidad ESFOT, a través de su proyecto “Radar comercial ESFOT”, propone dar mayor visibilidad a emprendimientos; sin embargo, esta propuesta no cuenta con un espacio en el cual almacene y presente información a los miembros de la comunidad ESFOT sobre emprendimientos. Por lo mencionado anteriormente, se ha propuesto el desarrollo de una aplicación web con el objetivo de promocionar dichos emprendimientos.

Para poder almacenar y presentar la información de los emprendimientos, en el presente proyecto de Integración curricular se ha desarrollado el backend de la aplicación, el cual se encarga de gestionar la información y procesar las solicitudes provenientes del usuario final a través del componente frontend.

Siguiendo la metodología XP, el proyecto presenta la siguiente estructura: dentro de la primera sección, se describe detalladamente el problema de la aplicación, junto con los objetivos, el alcance y el marco teórico; en la segunda sección, se describe el uso de la metodología seleccionada y las herramientas a utilizar; dentro de tercera sección se muestran los resultados obtenidos durante el desarrollo del componente; finalmente, como parte de la cuarta sección se describen las conclusiones y recomendaciones adquiridas a lo largo del desarrollo.

Como parte de las herramientas utilizadas para el desarrollo del *backend* se ha utilizado el framework Laravel para la codificación y el motor de base de datos MySQL para almacenar la información ingresada en la aplicación.

PALABRAS CLAVE: backend, emprendimientos, Laravel, MySQL.

ABSTRACT

Currently, the Empreende commission, responsible for promoting the creation to new undertaking within ESFOT community, through its project “Radar Comercial Esfot”, proposes to give greater visibility to this undertaking; but this proposal doesn’t have a space to store and publish information to members of the ESFOT community. For the previously, it has been proposed the development if web application to promoting these undertaking.

To store and present the information to undertaking, in the current project of Curricular Integration it has been developed the backend of the application, which is in charge of managing the information and processing the requests coming from user through the frontend component.

Follow the XP methodology, the project has the following structure: in the first section, it describes the application problem to resolved in conjunction with objectives, scope and theoretical framework; in the second section, it describes the use of the selected methodology and the tools to be used; in the next section, the results obtained while it was in progress the component are shown; finally, the conclusions and recommendations obtained during the development of the project are described.

The main tools used for the development of the backend component were: Laravel framework for coding and MySQL Database to store the application information.

KEYWORDS: backend, undertaking, Laravel, MySQL.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Con la finalidad de dar mayor visibilidad a las ideas emprendedoras generadas por quienes conforman la comunidad de la ESFOT y apoyar su economía por medio de la difusión de sus ideas, la comisión Emprende creó el proyecto "Radar Comercial ESFOT" [1] para la publicidad de dichas ideas; en donde, luego de haber recolectado la información necesaria de los emprendimientos y negocios de los estudiantes, se ha generado un catálogo para dar a conocer dichos emprendimientos dentro de la misma comunidad. Sin embargo, el proyecto propuesto por la comisión no cuenta con un espacio para el almacenamiento, administración y respaldo de la información previamente recolectada.

Por tal motivo, se ha planteado el desarrollo del backend para el sistema web de promoción de emprendimientos. Este componente contendrá la lógica del sistema para administrar y acceder a la información registrada y a su vez comunicar al componente frontend con la base de datos por medio de una API REST [2] para que dicha información sea mostrada al usuario final del sistema.

1.1 Objetivo general

Desarrollar el backend para la aplicación web de promoción de emprendimientos de la comunidad de la ESFOT.

1.2 Objetivos específicos

Levantar los requerimientos para el componente backend.

Diseñar el modelo de base de datos tomando en cuenta los requerimientos obtenidos.

Implementar la lógica del modelo de base de datos.

Verificar el funcionamiento del backend a través de diferentes pruebas.

Realizar el despliegue a producción del backend desarrollado.

1.3 Alcance

El presente proyecto propone la implementación de un API REST utilizando el framework Laravel para su codificación. Esta implementación ha permitido que el usuario final del sistema web pueda visualizar, guardar, editar o eliminar la información de los emprendimientos registrados en la base de datos tomando en consideración su rol o tipo de usuario.

Los tipos de usuarios con las respectivas acciones que a realizar dentro del sistema son:

Usuario superadministrador

- Ingresar al sistema
- Administrar los usuarios que pueden ingresar al sistema.
- Administrar la información de los emprendimientos.
- Administrar la información de videoconferencias.

Usuario administrador

- Ingresar al sistema.
- Administrar la información de los emprendimientos.
- Administrar la información de videoconferencias.

Usuario emprendedor

- Visualizar lista de emprendimientos registrados.
- Solicitar el registro de su emprendimiento a través de un formulario.
- Visualizar la lista de conferencias registradas

1.4 Marco Teórico

Metodología

Una metodología es una serie de pasos efectuados para organizar mejor los procesos que deben ser llevados a cabo para cumplir con éxito un proyecto. Su objetivo es definir las tareas a realizar mientras se va desarrollando el proyecto junto con los tiempos en los que serán efectuadas [3].

En lo que respecta a metodología para el desarrollo de software, esta se define como una estructura que ayuda en la organización, monitoreo y optimización de los recursos dentro de los procesos del desarrollo de un sistema. Una metodología de software comprende el análisis para determinar el método más adecuado para utilizarlo durante cada una de las etapas del desarrollo del software [4]. Este tipo de metodologías son esenciales para desarrollar software ya que permiten que las actividades planteadas sean llevadas a cabo a través de estándares de calidad que cumplan con los requerimientos planteados por los usuarios. Una metodología de desarrollo de software pretende la construcción de un sistema documentado y que sea fácil de mantener ya sea a corto, mediano o largo plazo

[3], además de ayudar a identificar los cambios que requieran efectuarse dentro del proceso; por tal motivo es importante elegir una metodología que se adapte a lo que requiera el sistema a desarrollar [5].

Metodología ágil

Las metodologías ágiles fueron creadas para reducir las probabilidades de fracaso que predominaban en los proyectos de desarrollo de software hace algunos años atrás.

Consisten en dividir el proyecto en pequeñas tareas que han sido entregadas en un periodo corto de tiempo. Este método permite que los proyectos sean más adaptables a los cambios debido a que al finalizar cada una de las tareas, éstas son entregadas para efectuar las pruebas correspondientes y; en caso de requerirlo, realizar correcciones, caso contrario, se procede al desarrollo de la siguiente actividad [4].

Otra particularidad en cuanto al uso de las metodologías ágiles es la comunicación constante con los involucrados en el proyecto ayudando en la verificación y cumplimiento de las tareas asignadas [6].

Backend

Dentro del desarrollo web, el término backend hace referencia al funcionamiento lógico que se le proporciona a una aplicación para optimizar las interacciones que realice el usuario dentro de la misma a través del acceso a las bases de datos [7].

El propósito del backend es permitir la relación y comunicación entre el servidor y la aplicación para que funcione adecuadamente [8].

Debido a que la lógica establecida dentro del backend no puede ser visualizada por el usuario final, para que este pueda hacer uso de esta y acceder a la información almacenada, se utiliza en este caso la arquitectura API REST, que a través del estándar HTTP, permite gestionar las peticiones que el usuario realice. En otras palabras, esta arquitectura, es el puente de comunicación entre los componentes backend y frontend los sistemas.

2 METODOLOGÍA

El uso de las metodologías en proyectos de software ha ayudado en la planificación y toma de decisiones a lo largo del desarrollo de estos proyectos, utilizando ciertas técnicas como la separación de tareas para lograr que cada una de las etapas del proyecto sean llevadas a cabo con éxito en un tiempo limitado.

Por lo mencionado anteriormente, el presente componente usa la metodología con un enfoque cualitativo debido a que este tipo de metodologías ayuda en la comprensión del problema a solucionar. En este caso, se emplea para el análisis de los requerimientos realizados por los clientes del sistema, recolectados a través de las reuniones efectuadas [9].

Adicional a ello, se plantea el tipo de trabajo experimental en el componente para verificar el funcionamiento del backend a través de diferentes pruebas en cada una de las tareas completadas.

A continuación, se definió la metodología escogida para el desarrollo del componente.

2.1 Metodología de Desarrollo

La metodología seleccionada para el desarrollo de este componente es la metodología de Programación Extrema o XP debido a que utiliza un modelo de trabajo compartido, en el cual, se trabaja en conjunto con el cliente y el desarrollador, ayudando a que el sistema sea desarrollado cumpliendo con los requeridos establecidos. Otra de las razones por las cuales se ha seleccionado esta metodología es que está basada en prueba y error, lo que permite verificar el funcionamiento de cada uno de los módulos que se han ido implementando al sistema [10].

Roles

En la metodología XP, los roles hacen referencia a los miembros que actúan directamente en el proyecto, los cuales son:

- **Programador**

Es quien se encarga de la implementación de las tareas establecidas en las historias de usuario [10].

- **Cliente**

El cliente es quien expone las funcionalidades que debe cumplir el sistema a través de las historias de usuarios [10].

- **Encargado de seguimiento**

Dentro de la metodología XP, como su nombre mismo lo dice, el encargado del seguimiento es quien supervisa el avance de las tareas correspondientes al proyecto mediante la implementación de pruebas [10].

En la TABLA I TABLA I Asignación de roles para el componente backend, se presenta la asignación de los respectivos roles para el componente.

TABLA I Asignación de roles para el componente backend

Nombre	Rol
Eduardo Farinango	Cliente (Desarrollador backend)
Ing. Juan Pablo Zaldumbide	Encargado de seguimiento
María José Chalá	Programador

Recopilación de Requerimientos

Los requerimientos de un sistema consisten en pequeñas descripciones de las funcionalidades implementadas dentro del sistema. Dichos requerimientos ayudan a establecer las necesidades del cliente [11].

Para esta etapa de levantamiento de requerimientos, se han efectuado reuniones en conjunto con el desarrollador del frontend de la aplicación, quien cumple el rol de cliente para el presente componente. Esta información se la encuentra dentro del ANEXO II Manual técnico.

Historias de Usuario

Dentro de la metodología planteada para el desarrollo del componente se establecen las historias de usuario, las cuales contienen las funcionalidades principales que debe tener el sistema, reemplazando así el documento de requerimientos.

Para el presente componente, se han planteado un total de 11 Historias de usuario. El resumen de lo que contiene cada una de ellas se encuentran en la TABLA II.

TABLA II Resumen de historias de usuario

Número	Nombre
HU01	Inicio de sesión
HU02	Restaurar contraseña
HU03	Actualizar información de usuario
HU04	Almacenar información de usuarios
HU05	Eliminar/Desactivar usuarios
HU06	Almacenar información de emprendimientos
HU07	Eliminar información de emprendimientos
HU08	Actualizar información de emprendimientos
HU09	Almacenar información de videoconferencias
HU10	Visualizar información de videoconferencias
HU11	Eliminar información de videoconferencia

Dentro de la TABLA III, se muestra una de las historias de usuario realizadas, las demás historias se encuentran en el ANEXO II Manual técnico.

TABLA III Historia de usuario HU01

HISTORIA DE USUARIO	
Número: HU01	Usuario: Desarrollador frontend (cliente)
Nombre Historia: Inicio de sesión	
Propiedad en Negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable: María José Chalá	
Descripción: El desarrollador del componente backend requiere un módulo para iniciar sesión para que los usuarios puedan ingresar al sistema.	
Observación: El inicio de sesión lo podrán realizar quienes estén registrados dentro de la base de datos a través del correo y una contraseña, los cuales son: <ul style="list-style-type: none"> • Administrador principal (superadmin) para gestión de usuarios • Administrador secundario (admin) para gestión de emprendimientos. 	

2.2 Diseño de la arquitectura

La arquitectura de software hace referencia a la estructuración de los datos dentro de un sistema tomando en cuenta los objetivos y las restricciones presentes en el mismo. La forma en la que se estructuran los datos es la que ha permitido generar la respuesta a las

peticiones generadas por el usuario, por lo tanto, es una pieza clave para la implementación de este componente ya que de está depende su funcionamiento

Dicho esto, se ha establecido el diseño de la arquitectura para el desarrollo de proyecto.

Patrón arquitectónico

El patrón arquitectónico establecido para este proyecto es Modelo Vista Controlador o MVC, en donde el modelo de negocio se encuentra dividido en tres componentes: administración de la información, lógica del negocio e interfaz de usuario.

A continuación, en la Fig. 1, se presenta un esquema del patrón arquitectónico establecido con las respectivas herramientas que se han utilizado.

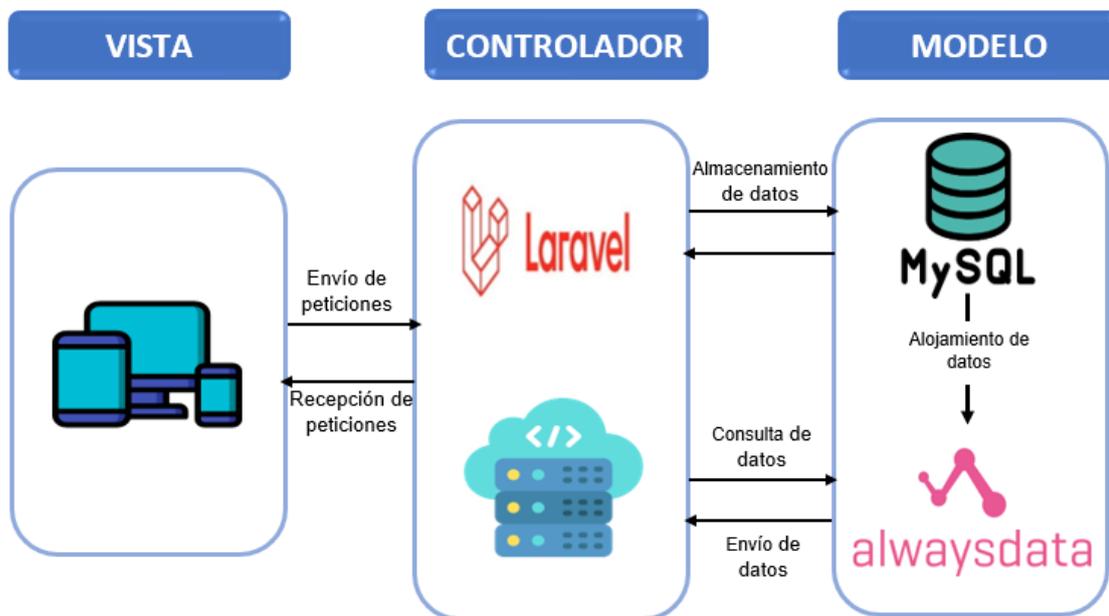


Fig. 1: Patrón arquitectónico de backend

2.3 Herramientas de desarrollo

Dentro de la TABLA IV se han citado las herramientas principales utilizadas para el desarrollo de este componente.

TABLA IV: Herramientas para el desarrollo del componente backend

Herramienta	Justificación
Node.js	La implementación de esta herramienta ha permitido la obtención de paquetes requeridos para el proceso de ejecución [12].
Composer	Composer, al ser un administrador de paquetes para PHP, se vuelve una opción viable para el proyecto debido a que, para el desarrollo del componente, se ha empleado el uso de este lenguaje de programación, por lo tanto, se requieren dependencias y librerías adicionales que esta herramienta nos proporcionará [13].
Laravel	Debido a que se ha planteado el uso del lenguaje de programación PHP para implementar los módulos definidos, el uso de este framework permite facilitar el desarrollo [14].
Thunder Client	Thunder Client es una alternativa para la aplicación de testeo de APIs, Postman. Esta herramienta, es una extensión liviana de Visual Studio Code que se ha utilizado para verificar que las funcionalidades se encuentren bien implementadas según la lógica establecida [15].
GitHub	A través de esta herramienta, se facilita el proceso de almacenamiento y publicación del código para tener un respaldo en caso de existir contratiempos [16].

3 RESULTADOS

Para esta sección, se ha planteado el desarrollo del componente en 5 iteraciones, como se muestran a continuación en la TABLA V.

TABLA V: Iteraciones

Número de iteración	Nombre de la iteración
Iteración 0	Configuración del ambiente de trabajo
Iteración 1	Autenticación de usuarios
Iteración 2	CRUD de usuarios, CRUD de emprendimientos, CRUD de videoconferencias.
Iteración 3	Almacenamiento de imágenes en Cloudinary
Iteración 4	Despliegue y pruebas del sistema

3.1 Iteración 0. Configuración del ambiente de desarrollo

Como parte de la iteración 0, correspondiente a la configuración del ambiente de desarrollo, se ha planteado el desarrollo de las siguientes tareas:

- Creación de diseño de la base de datos.
- Creación de proyecto en Laravel.
- Creación de base de datos en AlwaysData.

Creación de diseño de la base de datos

Para la administración y almacenamiento de datos para este proyecto se ha creado una base de datos MySQL alojada en AlwaysData. En la Fig. 2, se muestra el modelo de esta base de datos conformado por las tablas:

- 'users'. – para almacenar la información correspondiente a los usuarios del sistema.
- 'tipo_usuarios'. – para asignar un rol a cada usuario que sea registrado.
- 'emprendimientos'. – donde se almacena la información correspondiente a los emprendimientos.
- 'videoconferencias '. – para almacenar la información de las videoconferencias, tomando en cuenta el nombre, el enlace donde esta puede ser observada y una pequeña descripción.

- 'images'. - para el almacenamiento del path de las imágenes para las entidades Usuario y Emprendimientos.

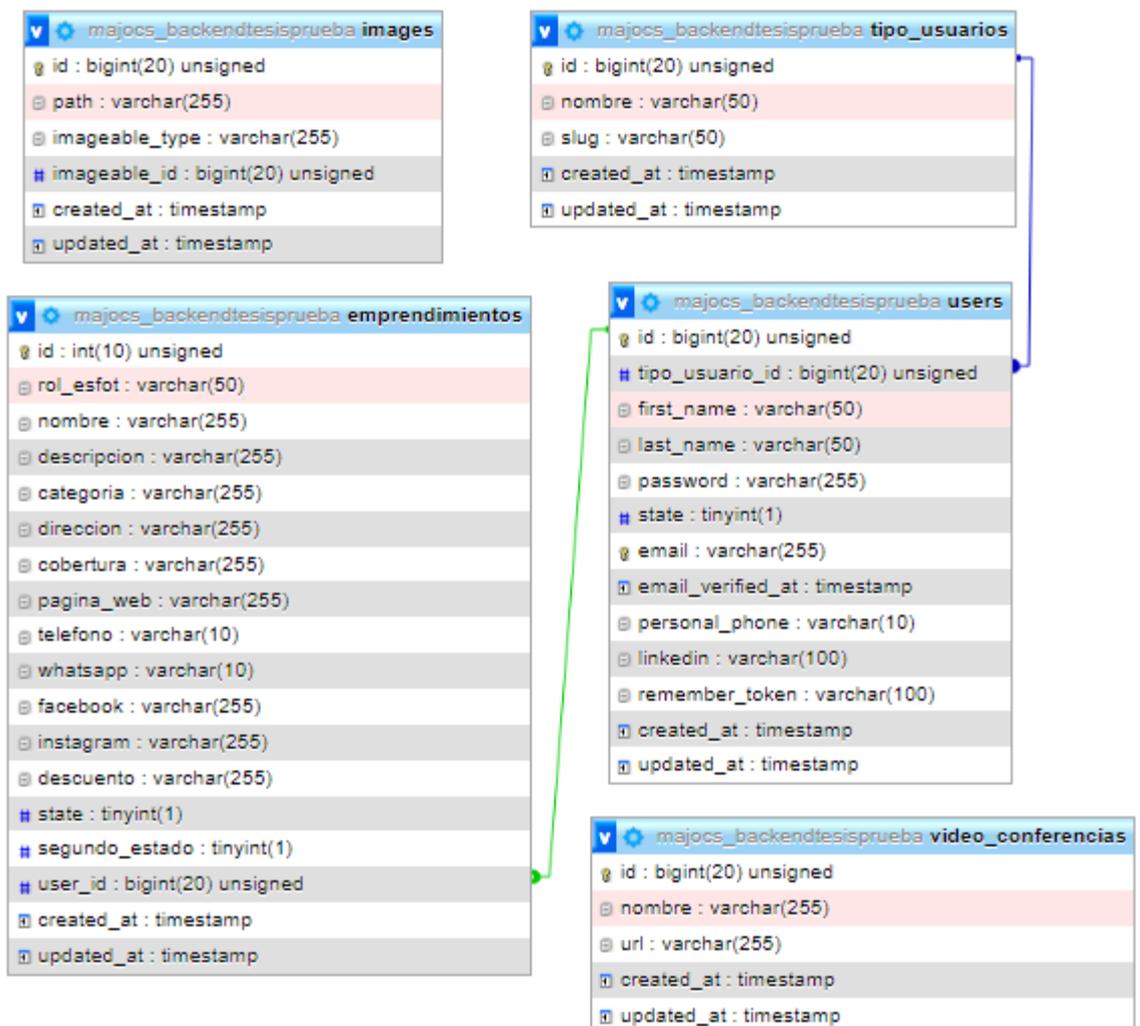


Fig. 2: Modelo de la base de datos

Creación de proyecto en Laravel

Se procede con la creación del proyecto en Laravel a través de la ejecución del comando 'laravel new backend-tesis', siendo "backend-tesis" en nombre del proyecto.

Una vez ejecutado el comando, se ha presentado el proyecto con la estructura mostrada en la Fig. 3.

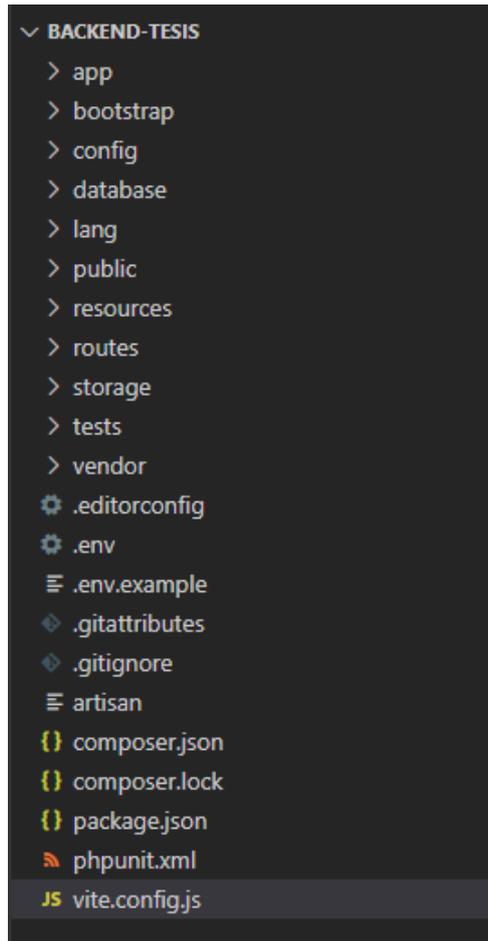


Fig. 3: Estructura de proyecto de Laravel

Creación de base de datos en AlwaysData

Como se ha mencionado anteriormente, la información se almacena en una base de datos alojada en AlwaysData. Esta base de datos contiene las tablas presentadas en la Fig. 4.

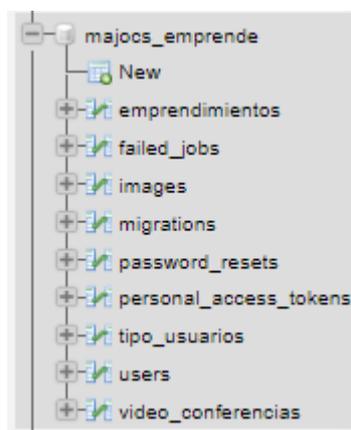


Fig. 4: Creación de base de datos en AlwaysData

3.2 Iteración 1. Autenticación de usuario

Dentro de la siguiente iteración, correspondiente al módulo de autenticación, se implementa el inicio y cierre de sesión para los usuarios junto con otras funcionalidades que puede realizar el usuario registrado en el sistema. Estas funcionalidades son las siguientes:

- Creación de entidad Usuario y Tipo de Usuario.
- Autenticación de usuarios mediante el inicio de sesión.
- Implementación de rutas para módulos creados.

Creación de entidades Usuario y Tipo de usuario

Entidad Usuario

Para la tabla usuario, se han definido los campos correspondientes al nombre, apellido, contraseña, estado que se encuentra el usuario en el sistema, correo electrónico y verificación del mismo; dentro del archivo de migraciones para esta entidad como se observa en la Fig. 5.

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('first_name', 50);
        $table->string('last_name', 50);
        $table->string('password');
        $table->boolean('state')->default(true);
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->rememberToken();
        $table->timestamps();
    });
}
```

Fig. 5: Campos de tabla usuarios

Entidad Tipo de usuario

Dentro de la entidad tipo de usuario, se han definido los campos de la Fig. 6, un identificador y nombre del tipo de usuario o rol para cada uno de los usuarios; ya sea de tipo superadministrador o administrador.

```
public function up()
{
    Schema::create('tipo_usuarios', function (Blueprint $table) {
        $table->id();
        $table->string('nombre', 50);
        $table->timestamps();
    });
}
```

Fig. 6: Campos para entidad 'tipo de usuario'

Posterior a ello, se han creado los tipos de usuarios definidos anteriormente para el sistema dentro del archivo TipoUsuarioSeeder.php como se muestra en la Fig. 7.

```
TipoUsuarioSeeder.php X
database > seeders > TipoUsuarioSeeder.php > ...
16     public function run()
17     {
18         $rols = ['superadmin', 'admin'];
19
20         for($i=0 ; $i<2 ; $i++)
21         {
22             Tipo_usuario::create([
23                 'nombre' => $rols[$i]
24             ]);
25         }
26     }
27 }
28
```

Fig. 7: Definición de tipos de usuarios

Como siguiente paso, se han establecido la conexión con la base de datos a través del comando 'php artisan migrate --seed', para que las entidades se registren en la base de datos. El argumento '--seed' que se han empleado en dicho comando, ha permitido que se registren los tipos de usuarios implementados en el paso anterior, además de los datos falsos que se establecieron en los respectivos seeders.

Finalmente, se ha verificado la creación de las tablas correspondientes a las entidades dentro de la base de datos, Fig. 8. Cabe mencionar que algunas tablas que se visualizan en la imagen corresponden a la lógica generada por Laravel.

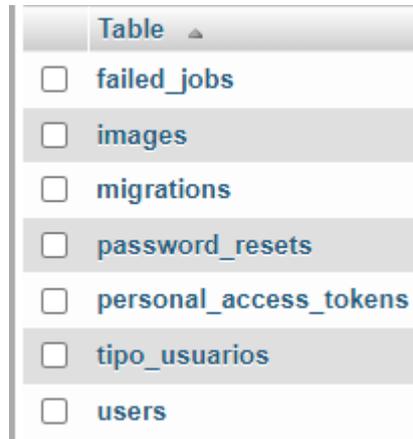


Fig. 8: Visualización de entidades creadas

Autenticación de usuarios mediante el inicio de sesión

Para la autenticación de usuarios se ha procedido con la instalación y publicación de los archivos de configuración de Laravel Sanctum, el cual es un paquete que proporciona un sistema de autenticación para que cada usuario genere varios Access token cuando ingrese al sistema [17]. Luego se ha creado el controlador AuthController.php en donde se ha establecido la lógica para el inicio de sesión como se muestra en la Fig. 9; validando que el usuario ingrese las credenciales correctas, caso contrario no se le ha permitido iniciar sesión.

```

AuthController.php x
app > Http > Controllers > Auth > AuthController.php > PHP Intelephense > AuthController > login
1  <?php
2
3  namespace App\Http\Controllers\Auth;
4
5  use App\Http\Controllers\Controller;
6  use App\Http\Resources\UserResource;
7  use App\Models\User;
8  use Illuminate\Http\Request;
9  use Illuminate\Support\Facades\Hash;
10
11 class AuthController extends Controller
12 {
13     public function login(Request $request)
14     {
15         $request -> validate([
16             'email' => ['required', 'string', 'email'],
17             'password' => ['required', 'string'],
18         ]);
19
20         $user = User::where('email', $request['email'])->first();
21
22         if (!$user || !$user->state ||
23             !Hash::check($request['password'], $user->password))
24         {
25             return $this->sendResponse(message: 'Las credenciales ingresadas
26         }
27
28         if (!$user->tokens->isEmntv())

```

Fig. 9: Módulo de inicio de sesión

Implementación de rutas para módulos creados en la Iteración 1.

Para finalizar la Iteración 1, se han implementado cada una de las rutas para los módulos creados como muestra la Fig. 10 a continuación:

```

Route::post('/login', [AuthController::class, 'login'])->name('login');

// Rutas para modulo de recuperación de contraseña
Route::post('/forgot-password', [PasswordController::class, 'resendLink'])->name('password.resend-link');
Route::get('/reset-password/{token}', [PasswordController::class, 'redirectReset'])->name('password.reset');
Route::post('/reset-password', [PasswordController::class, 'restore'])->name('password.restore');

Route::post('/register', [AuthController::class, 'register'])->name('register');

Route::middleware(['auth:sanctum'])->group(function ()
{
    // Ruta para el cierre de sesión
    Route::post('/logout', [AuthController::class, 'logout'])->name('logout');
    // Ruta para el cambio de contraseña del usuario
    Route::post('/update-password', [PasswordController::class, 'update'])->name('password.update');
});

```

Fig. 10: Implementación de rutas para módulos creados

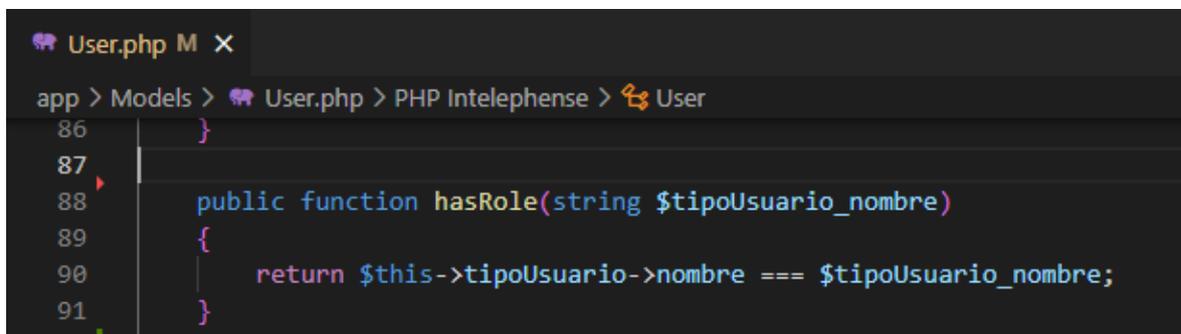
3.3 Iteración 2. Implementación de CRUD de usuarios, emprendimientos y videoconferencias

Como parte de la iteración 2 correspondiente a la implementación de los CRUD, se han definido las siguientes tareas:

- CRUD para gestionar información de usuarios.
- CRUD para gestionar información de emprendimientos.
- CRUD para gestionar información de las videoconferencias.

CRUD para gestionar información de usuarios

Dentro del modelo de usuarios, se ha creado un método como se muestra en la Fig. 11 para acceder al tipo de rol de al que pertenece el usuario registrado.



```
app > Models > User.php > PHP Intelephense > User
86     }
87
88     public function hasRole(string $tipoUsuario_nombre)
89     {
90         return $this->tipoUsuario->nombre === $tipoUsuario_nombre;
91     }
```

Fig. 11: Función para acceder al rol del usuario

Como siguiente paso, se ha creado los middlewares para el manejo de usuarios:

En el primer middleware, Fig. 12, se ha definido la lógica para verificar que el usuario se encuentre con un estado activo.

```
IsUserActive.php M X
app > Http > Middleware > IsUserActive.php > ...
1  <?php
2  namespace App\Http\Middleware;
3  use Closure;
4  use Illuminate\Http\Request;
5
6  class IsUserActive
7  {
8      public function handle(Request $request, Closure $next)
9      {
10         $user = $request->route('user');
11         if (!$user->state)
12         {
13             return abort(403, 'Acción no autorizada');
14         }
15         return $next($request);
16     }
17 }
18 }
```

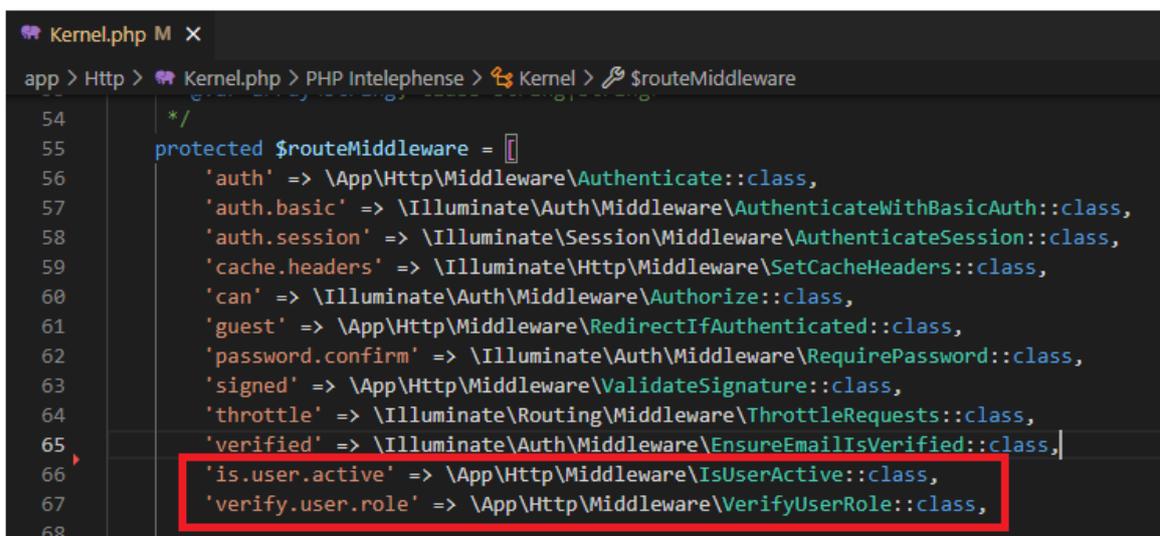
Fig. 12: Función para verificar que el estado del usuario este activo

Para el siguiente middleware, definido en la Fig. 13, se ha implementado la lógica para identificar con que tipo de rol se ha registrado el usuario.

```
VerifyUserRole.php M X
app > Http > Middleware > VerifyUserRole.php > PHP Intelephense > VerifyUserRole > handle
1  <?php
2  namespace App\Http\Middleware;
3  use Closure;
4  use Illuminate\Http\Request;
5
6  class VerifyUserRole
7  {
8      public function handle(Request $request, Closure $next, string $tipoUsuario_nombre)
9      {
10         $user = $request->route('user');
11         if (!$user->hasRole($tipoUsuario_nombre))
12         {
13             return abort(403, 'Acción no autorizada');
14         }
15         return $next($request);
16     }
17 }
18 }
```

Fig. 13: Función para verificar el tipo de rol del usuario

Dentro del archivo Kernel.php, se han registrado los middlewares para poderlos usar en los siguientes pasos, como se observa a través de la Fig. 14.

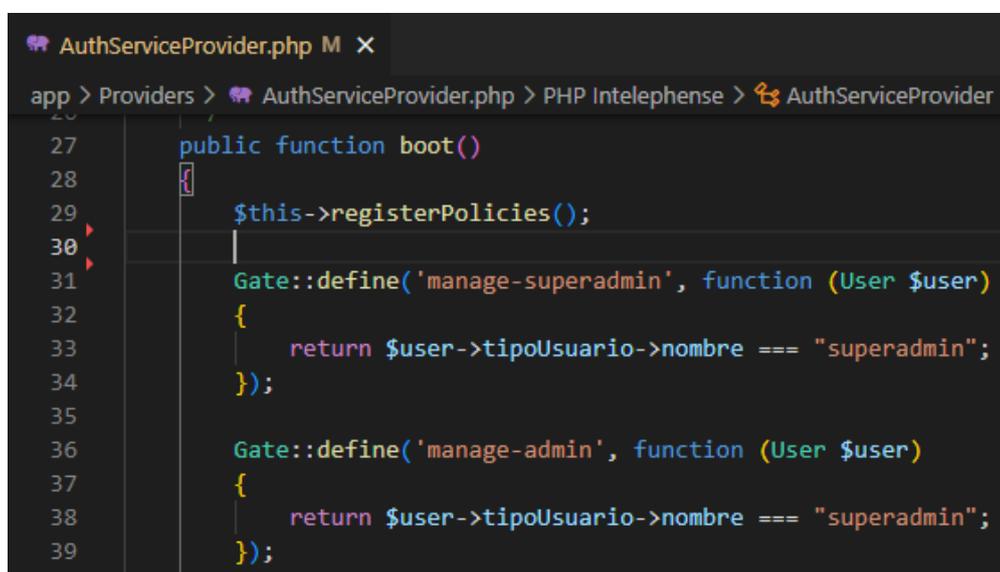


```
Kernel.php M X
app > Http > Kernel.php > PHP Intelephense > Kernel > $routeMiddleware
54 */
55 protected $routeMiddleware = [
56     'auth' => \App\Http\Middleware\Authenticate::class,
57     'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
58     'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
59     'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
60     'can' => \Illuminate\Auth\Middleware\Authorize::class,
61     'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
62     'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
63     'signed' => \App\Http\Middleware\ValidateSignature::class,
64     'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
65     'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
66     'is.user.active' => \App\Http\Middleware\IsUserActive::class,
67     'verify.user.role' => \App\Http\Middleware\VerifyUserRole::class,
68 ]
```

Fig. 14: Publicación de los middlewares implementados

A continuación, dentro del archivo AuthServiceProvider.php, se ha implementado la lógica de los Gates, estos son unos servicios de autenticación que vienen integrados en Laravel [14] que, para este proyecto, ha permitido determinar las acciones que cada tipo de usuario tiene autorizado realizar.

En este caso, se han definido las acciones para el usuario “superadmin” quien, como ya se había comentado anteriormente, se encarga de gestionar los usuarios registrados, como se muestra en la Fig. 15.



```
AuthServiceProvider.php M X
app > Providers > AuthServiceProvider.php > PHP Intelephense > AuthServiceProvider >
27 public function boot()
28 {
29     $this->registerPolicies();
30
31     Gate::define('manage-superadmin', function (User $user)
32     {
33         return $user->tipoUsuario->nombre === "superadmin";
34     });
35
36     Gate::define('manage-admin', function (User $user)
37     {
38         return $user->tipoUsuario->nombre === "superadmin";
39     });
40 }
```

Fig. 15: Implementación de Gates para gestionar usuarios

Posterior a la implementación de los Gates, se ha creado el controlador UserController en donde se han definido los métodos correspondientes al CRUD de usuarios.

A continuación, en la TABLA VI se han detallado cada uno de los métodos implementados dentro de este controlador con sus respectivas funcionalidades.

TABLA VI: Métodos para CRUD de Usuarios

Nombre de método	Funcionalidad
index()	Retorna los datos de todos los usuarios registrados, categorizándolos por su tipo de rol.
store(Request \$request)	Crea y almacena la información de los usuarios dentro de la base de datos una vez llenados los datos de entrada establecidos para esta función.
show(User \$user)	Retorna los datos de un usuario que se especifique.
update(Request \$request, User \$user)	Actualiza los datos de un usuario en específico validando los que los datos de entrada sean los mismos que se establecieron para la creación del usuario.
destroy(User \$user)	Activa o desactiva los datos de un usuario específico a través de su estado para restringir o autorizar las funciones que este pueda realizar dentro del sistema.

Dentro de las Fig. 16, Fig. 17, Fig. 18, Fig. 19 y Fig. 20 se visualiza la implementación de los métodos citados en la TABLA VI: Métodos para CRUD de Usuarios TABLA VI.

```
UserController.php M ●
app > Http > Controllers > Users > UserController.php > PHP Intelephense > UserController > index
29
30 public function index()
31 {
32     $tipoUsuario = Tipo_usuario::where('nombre', $this->tipoUsuario_nombre)
33     ->first();
34     $users = $tipoUsuario->users;
35     return $this->sendResponse([message: "Lista de usuarios generada exitosamente",
36     result: [
37         'users' => UserResource::collection($users),
38     ]]);
39 }
```

Fig. 16: Método index para mostrar usuarios

```
UserController.php M X
app > Http > Controllers > Users > UserController.php > PHP Intelephense > UserController > index
39 // Crear un nuevo usuario
40 public function store(Request $request)
41 {
42     $request -> validate([
43         'first_name' => ['required', 'string', 'min:3', 'max:35'],
44         'last_name' => ['required', 'string', 'min:3', 'max:35'],
45         'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
46         'personal_phone' => ['required', 'numeric', 'digits:10'],
47         'linkedin' => ['required', 'string', 'min:3', 'max:35'],
48     ]);
49     $tipoUsuario = Tipo_usuario::where('nombre', $this->tipoUsuario_nombre)->first();
50     $user = new User($request->all());
51     $temp_password = PasswordHelper::generatePassword();
52     $user->password = Hash::make($temp_password);
53     $tipoUsuario->users()->save($user);
54     if ($this->can_receive_notifications)
55     {
56         $this->sendNotifications($user, $temp_password);
57     }
58     return $this->sendResponse(message: 'Usuario registrado correctamente');
59 }
```

Fig. 17: Método para crear nuevos usuarios

```
// Mostrar la información personal del usuario
public function show(User $user)
{
    return $this->sendResponse(message: 'Perfil de usuario', result: [
        'user' => new ProfileResource($user),
    ]);
}
```

Fig. 18: Método para mostrar usuario en específico

```

// Actualizar el usuario
public function update(Request $request, User $user)
{
    // Validación de los datos de entrada
    $user_data=$request -> validate([
        'first_name' => ['required', 'string', 'min:3', 'max:35'],
        'last_name' => ['required', 'string', 'min:3', 'max:35'],
        'email' => ['required', 'string', 'email', 'max:255',
            Rule::unique('users')->ignore($user),
        ],
        'personal_phone' => ['required', 'numeric', 'digits:10'],
        'linkedin' => ['required', 'string', 'min:3', 'max:35'],
    ]);
    $old_user_email = $user->email;
    $user->fill($user_data);
    $user->save();
    // Mandar la notificación si en el caso del que el correo sea diferente
    if ($this->can_receive_notifications && $old_user_email !== $user->email)
    {
        $temp_password = PasswordHelper::generatePassword();
        $user->password = Hash::make($temp_password);
        $user->save();
        $this->sendNotifications($user, $temp_password);
    }
    return $this->sendResponse(message: 'Usuario actualizado correctamente.');
```

Fig. 19: Método para actualizar información de usuario

```

// Cambiar estado de usuario
public function destroy(User $user)
{
    $user_state = $user->state;
    $message = $user_state ? 'inactivo' : 'activo';
    $user->state = !$user_state;
    $user->save();
    // Invoca el controlador padre para la respuesta json
    return $this->sendResponse(message: "Usuario $message correctamente");
}
```

Fig. 20: Método para activar o desactivar usuario

Para finalizar con el CRUD de usuarios, se han creado las rutas de tipo respectivas, mostradas en la Fig. 21, para proporcionar el acceso al componente frontend y pueda consumir los módulos implementados.

Para definir estas rutas, se ha tomado en consideración los tipos de usuarios creados anteriormente para que, dependiendo de ello, se realicen las acciones establecidas.



```
api.php x
routes > api.php > #Function#c7c5686a > #Function#c8c569fd > #Function#cbc56eb6 > #Function#
26     });
27
28     Route::prefix("superadmin")->group(function ()
29     {
30         Route::controller(SuperAdminController::class)->group(function () {
31             Route::get('/', 'index');
32             Route::post('/create', 'store');
33             Route::get('/{user}', 'show');
34             Route::post('/{user}/update', 'update');
35             Route::get('/{user}/destroy', 'destroy');
36         });
37     });
38
39     Route::prefix("admin")->group(function ()
40     {
41         Route::controller(AdminController::class)->group(function () {
42             Route::get('/', 'index');
43             Route::post('/create', 'store');
44             Route::get('/{user}', 'show');
45             Route::post('/{user}/update', 'update');
46             Route::get('/{user}/destroy', 'destroy');
47         });
48     });
49
```

Fig. 21: Rutas para CRUD de usuarios 'superadmin' y 'admin'

CRUD para gestionar información de emprendimientos

La entidad 'emprendimientos' se han creado a través del comando '*php artisan make:model Emprendimiento -mfs*'. A partir de dicho comando, se han añadido los archivos correspondientes al modelo, migración, factory y seeder; dentro de cada uno de los archivos, se han construido la lógica para el funcionamiento de esta entidad.

En el archivo correspondiente a la migración, se han establecido los campos para la entidad, como se observa en la Fig. 22. Adicional a ello, dentro del mismo archivo, se ha añadido la relación uno a muchos con la entidad de usuarios, Fig. 23 para que cada usuario pueda añadir los emprendimientos que requiera.

```
2022_12_19_073256_create_emprendimientos_table.php M X
database > migrations > 2022_12_19_073256_create_emprendimientos_table.php > #Class#31
14 public function up()
15 {
16     Schema::create('emprendimientos', function (Blueprint $table) {
17         $table->id();
18         $table->string('rol_esfot', 50);
19         $table->string('nombre', 50);
20         $table->string('descripcion', 50);
21         $table->string('categoria', 50)->nullable();
22         $table->string('direccion')->nullable();
23         $table->string('cobertura')->nullable();
24         $table->string('pagina_web')->nullable();
25         $table->string('telefono', 10)->nullable();
26         $table->string('whatsapp', 10)->nullable();
27         $table->string('facebook')->nullable();
28         $table->string('instagram')->nullable();
29         $table->string('porcentaje')->nullable();
30         $table->boolean('state')->default(true);
31     });
}
```

Fig. 22: Migración de entidad 'emprendimientos'

```
2022_12_19_073256_create_emprendimientos_table.php M X
database > migrations > 2022_12_19_073256_create_emprendimientos_table.php >
31
32 // Relación
33 $table->unsignedBigInteger('user_id');
34 $table->foreign('user_id')
35     ->references('id')
36     ->on('users')
37     ->onDelete('cascade')
38     ->onUpdate('cascade');
39
```

Fig. 23: Relación de uno a muchos entre entidad 'emprendimiento' y 'usuarios'

En la TABLA VII se muestran los métodos implementados para el funcionamiento del CRUD de emprendimientos.

TABLA VII: Métodos para CRUD de emprendimientos

Nombre de método	Funcionalidad
index()	Retorna la información de todos los emprendimientos almacenados
store(Request \$request)	Crea y almacena los datos de los emprendimientos, ingresados por parte del usuario final.
show(User \$user)	Retorna la información de un emprendimiento en específico
update(Request \$request, User \$user)	Actualiza la información de un emprendimiento en específico.
destroy(User \$user)	Cambia la información del emprendimiento a un estado 'inactivo' en caso de que el emprendimiento ya no exista.

A continuación, en las Fig. 24, Fig. 25, Fig. 26 y Fig. 27, se muestra la implementación de los métodos *index*, *store*, *show*, *update* y *destroy* respectivamente.

```
//Visualizar la lista de emprendimientos
public function index()
{
    $user = Auth::user();
    $emprendimientos = Emprendimiento::all();
    return $this->sendResponse(message: 'Lista de emprendimientos creada exitosamente',
        result: [
            'emprendimientos' => EmprendimientoResource::collection($emprendimientos)
        ]);
}
```

Fig. 24: Método para visualizar emprendimientos creados

```

public function store(Request $request)
{
    $request -> validate([
        'rol_esfot' => ['required', 'string', 'min:1', 'max:50'],
        'nombre' => ['required', 'string', 'min:4', 'max:255'],
        'descripcion' => ['required', 'string', 'min:4', 'max:255'],
        'categoria' => ['string', 'min:4', 'max:255'],
        'direccion' => ['string', 'min:4', 'max:255'],
        'cobertura' => ['string', 'min:4', 'max:255'],
        'pagina_web' => ['string', 'min:4', 'max:255'],
        'telefono' => ['numeric', 'digits:10'],
        'whatsapp' => ['string', 'min:4', 'max:50'],
        'facebook' => ['string', 'min:4', 'max:255'],
        'instagram' => ['string', 'min:4', 'max:255'],
        'descuento' => ['numeric', 'min:4', 'max:255'],
        'image' => ['nullable', 'image', 'mimes:jpg,png,jpeg', 'max:512'],
    ]);
    $emprendimiento_data = $request->all();
    $emprendimiento = new Emprendimiento($emprendimiento_data);
    $emprendimiento->save();
    if ($request->has('image'))
    {
        $uploadedFileUrl = Cloudinary::upload($request->file('image')
        ->getRealPath(),['folder'=>'emprendimientos']
        ->getSecurePath());

        $emprendimiento->attachImage($uploadedFileUrl);
    }
    return $this->sendResponse(message: 'Datos de emprendimiento almacenados exitosamente');
}

```

Fig. 25: Método para crear nuevos emprendimientos

```

public function update(Request $request, Emprendimiento $emprendimiento)
{
    $request -> validate([
        'rol_esfot' => ['string', 'min:10', 'max:14'],
        'nombre' => ['string', 'min:4', 'max:50'],
        'descripcion' => ['string', 'min:4', 'max:100'],
        'categoria' => ['string', 'min:5', 'max:20'],
        'direccion' => ['string', 'min:5', 'max:100'],
        'cobertura' => ['string', 'min:5', 'max:30'],
        'pagina_web' => ['min:4', 'max:50'],
        'telefono' => ['numeric', 'digits:10'],
        'whatsapp' => ['string', 'min:4', 'max:50'],
        'facebook' => ['string', 'min:4', 'max:50'],
        'instagram' => ['string', 'min:4', 'max:50'],
        'descuento' => ['numeric', 'digits:2'],
        'image' => ['nullable', 'image', 'mimes:jpg,png,jpeg', 'max:512'],
    ]);
    $emprendimiento_data = $request->all();
    $emprendimiento->fill($emprendimiento_data)->save();

    if ($request->has('image'))
    {
        $uploadedFileUrl = Cloudinary::upload($request->file('image')
        ->getRealPath(),['folder'=>'emprendimientos']->getSecurePath());
        $emprendimiento->attachImage($uploadedFileUrl);
    }
    return $this->sendResponse(message: 'Datos de emprendimiento actualizados exitosamente');
}

```

Fig. 26: Método para actualizar información de emprendimientos

```

public function destroy(Emprendimiento $emprendimiento)
{
    $emprendimiento_state = $emprendimiento->state;
    $message = $emprendimiento_state ? 'desactivado' : 'activado';
    $emprendimiento->state = !$emprendimiento_state;
    $emprendimiento->save();
    return $this->sendResponse(message: "Emprendimiento $message correctamente");
}

```

Fig. 27: Método para activar o desactivar información de emprendimientos

Posterior a ello, en la Fig. 28, se han definido las rutas correspondientes al CRUD de emprendimientos.

```

Route::prefix('emprendimiento')->group(function () {
    Route::controller(EmprendimientoController::class)->group(function ()
    {
        Route::get('/{emprendimiento}', 'show');
        Route::post('/{emprendimiento}/update', 'update');
        Route::get('/{emprendimiento}/destroy', 'destroy');
        Route::get('/{emprendimiento}/estado', 'estado');
    });
});

```

Fig. 28: Rutas para CRUD de emprendimientos

CRUD para gestionar información de videoconferencias

En la Fig. 29, se muestra el archivo de migraciones para la entidad de videoconferencias en donde se han definido los campos requeridos para la misma, como lo son el nombre y el enlace del video, seguido del argumento timestamps para almacenar la fecha y hora en la que ha sido creada o actualizada la información de forma automática [18].

```

public function up()
{
    Schema::create('video_conferencias', function (Blueprint $table) {
        $table->id();
        $table->string('nombre');
        $table->string('url');
        $table->timestamps();
    });
}

```

Fig. 29: Definición de campos para tabla videoconferencia

Dentro de la TABLA VIII se han definido los métodos implementados para el funcionamiento del CRUD de videoconferencias con sus respectivos detalles de la funcionalidad.

TABLA VIII: Métodos para CRUD de videoconferencias

Nombre de método	Funcionalidad
index()	El método 'index' permite mostrar la información almacenada de las videoconferencias.
store(Request \$request)	Con el método 'store', se almacena la información de las videoconferencias.
show(User \$user)	El método 'show' permite mostrar los datos en específico de la videoconferencia que se seleccione.
update(Request \$request, User \$user)	El método 'update' permite actualizar la información de las videoconferencias.

A continuación, en las Fig. 30, Fig. 31, Fig. 32 y Fig. 33, se muestra la implementación de los métodos *index*, *store*, *show* y *update* respectivamente.

```
public function index()
{
    $videoconferencias = VideoConferencia::all();

    return $this->sendResponse(message: 'Lista de videoconferencias',
        result: [
            'video_conferencias' => VideoConferenciaResource::collection($videoconferencias),
        ]);
}
```

Fig. 30: Método index para CRUD de videoconferencias

```

public function store(Request $request)
{
    $request -> validate([
        'nombre' => ['required', 'string', 'min:4', 'max:100'] ,
        'url' => ['required', 'string'],
    ]);
    $videoconferencias_data = $request->all();
    $videoconferencias = new VideoConferencia($videoconferencias_data);
    $videoconferencias->save();

    return $this->sendResponse(message: 'Datos de videoconferencia almacenados exitosamente');
}

```

Fig. 31: Método store para CRUD de videoconferencias

```

public function show(VideoConferencia $videoconferencia)
{
    return $this->sendResponse(message: 'Detalle de la videoconferencia', result: [
        'video_conferencias' => new VideoConferenciaResource($videoconferencia),
    ]);
}

```

Fig. 32: Método show para CRUD de videoconferencias

```

public function update(Request $request, VideoConferencia $videoconferencia)
{
    $request -> validate([
        'nombre' => ['required', 'string', 'min:4', 'max:100'] ,
        'url' => ['required', 'string'],
    ]);
    $videoconferencia_data = $request->all();
    $videoconferencia->fill($videoconferencia_data)->save();

    return $this->sendResponse(message: 'Actualización exitosa');
}

```

Fig. 33: Método update para CRUD de videoconferencias

3.4 Iteración 3. Almacenamiento de imágenes en Cloudinary

Como parte de la iteración 3, se ha definido la implementación de la herramienta Cloudinary dentro del sistema para el almacenamiento de las imágenes.

Para esta implementación se ha instalado el paquete “yoelpc4/laravel-cloudinary” utilizando el administrador de paquetes composer [13] a través del comando que se muestra en la Fig. 34.

```
> composer require yoelpc4/laravel-cloudinary
Info from https://repo.packagist.org: #StandWithUkraine
Using version ^3.0 for yoelpc4/laravel-cloudinary
./composer.json has been updated
Running composer update yoelpc4/laravel-cloudinary
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating optimized autoload files
```

Fig. 34: Instalación de paquete de Cloudinary

Dentro de los controladores que necesitan de almacenamiento de imágenes, se ha implementado la lógica de la Fig. 35. En esta 'condición if' se verifica si dentro del formulario se ha cargado una imagen; si la condición se cumple, los datos de la imagen se almacenan dentro Cloudinary, caso contrario se ignora el proceso.

```
if ($request->has('image'))
{
    $uploadedFileUrl = Cloudinary::upload($request->file('image')
->getRealPath(), ['folder'=>'emprendimientos']
->getSecurePath());

    $emprendimiento->attachImage($uploadedFileUrl);
}
```

Fig. 35: Lógica para almacenamiento de imágenes

Una vez implementada la lógica, se han generado las variables de entorno y se han agregado al archivo .env como se observa en la Fig. 36.

```
CLOUDINARY_URL=cloudinary: [redacted]
CLOUDINARY_API_KEY=1926368 [redacted]
CLOUDINARY_API_SECRET=Sh9G [redacted]
CLOUDINARY_CLOUD_NAME=dm0qsdpr8
CLOUDINARY_SECURE=true
```

Fig. 36: Variables de entorno para implementación de Cloudinary.

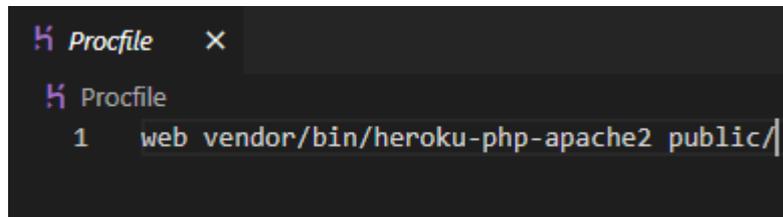
3.5 Iteración 4. Despliegue a Heroku y pruebas del sistema

Como parte de las tareas asignadas para la iteración 4, se tienen:

- Despliegue del componente hacia la plataforma Heroku.
- Implementación de pruebas para verificar el funcionamiento del sistema:
 - Pruebas de integración.
 - Pruebas de carga.
 - Pruebas de estrés.

Despliegue

Para el despliegue hacia la plataforma Heroku [19], se han añadido el archivo Procfile [20]. En este archivo se han establecido al directorio “public/” como el directorio principal para que, una vez que se ejecute la aplicación, esta busque los archivos de tipo .php y los inicializa primero [21].



```

Procfile
1 web vendor/bin/heroku-php-apache2 public/

```

Fig. 37: Creación de archivo Procfile

Luego se han añadido las variables de entorno del archivo .env en las configuraciones de Heroku como se muestra a continuación en la Fig. 38

Config Vars		Hide Config Vars
APP_FRONTEND_URL	https://front-end-emprende.vercel.app/con	✎ ✕
APP_KEY	base64:CTBe1SubPrxS481mpyUwkpodpGicmL01He	✎ ✕
CLOUDINARY_API_KEY	192636836795493	✎ ✕
CLOUDINARY_API_SECRET	Sh9GHe-GvdJ85UZf9hAIhbQDhQw	✎ ✕
CLOUDINARY_CLOUD_NAME	dm0qsdpr8	✎ ✕
CLOUDINARY_SECURE	true	✎ ✕
CLOUDINARY_URL	cloudinary://192636836795493:Sh9GHe-GvdJ8	✎ ✕
DB_CONNECTION	mysql	✎ ✕
DB_DATABASE	majocs_backendtesisprueba	✎ ✕

Fig. 38: Estableciendo variables de entorno.

Pruebas del sistema

Pruebas de integración

Para la implementación de las pruebas de integración, se han añadido la colección que contiene cada uno de los endpoints creados dentro de la herramienta Postman, como se muestra en la Fig. 39.



Fig. 39: Colección de Postman

Dentro del apartado de “Test” en Postman se añadió un Snippet, el cual hace referencia a un pequeño fragmento de código para verificar el funcionamiento de una API [22]. Postman ofrece varios Snippets para poder ejecutarlos sin complicaciones. En este caso se han seleccionado el que se muestra en la Fig. 40 para verificar que el estado de la petición de tipo “POST” para el inicio de sesión sea correcto, retornando un 200, 201 o 202.



Fig. 40: Implementación de prueba de integración en Postman

En este contexto, se han verificado los demás métodos implementados en cada una de las peticiones principales. A continuación, en la Fig. 41, se presenta el resultado de la petición de tipo “GET” para obtener la información de todos los usuarios superadministradores registrados, las demás pruebas se encuentran en el ANEXO II Manual técnico.

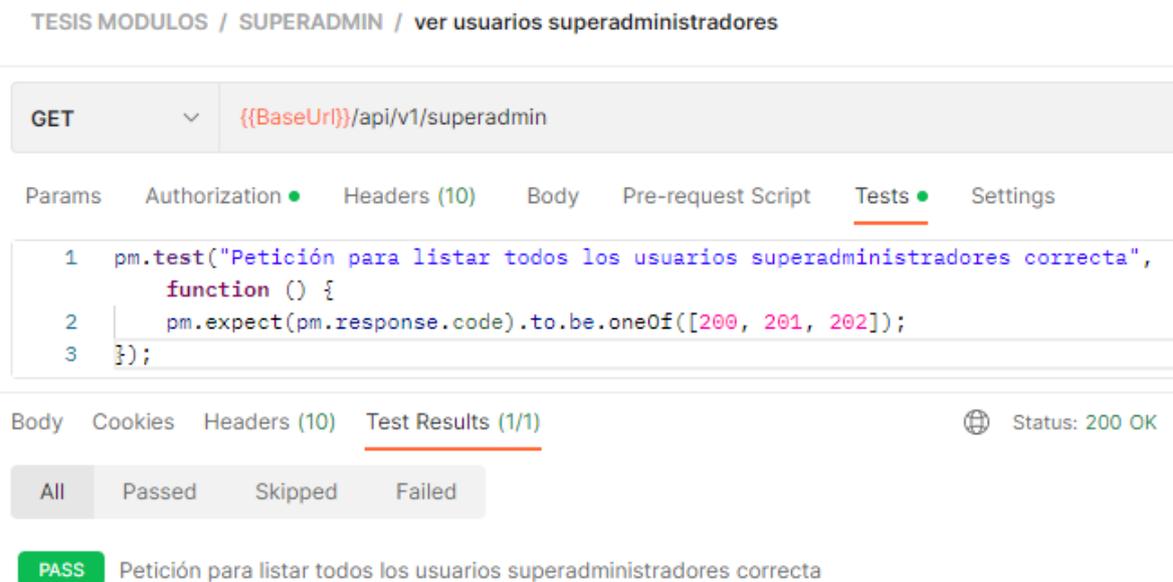


Fig. 41: Prueba unitaria para la petición de listar usuarios administradores

En la Fig. 42 presentada a continuación, se presentan los resultados de la prueba de integración en caso de que esta retorne otro código de respuesta; en donde, al recibir un

código de respuesta diferente a 200, 201 o 202, la prueba ha retornado el error correspondiente.

TESIS MODULOS / SUPERADMIN / ver usuario superadministrador especifico Save

GET `{{BaseUrl}}/api/v1/superadmin/90`

Params Authorization Headers (10) Body Pre-request Script Tests Settings

```
1 pm.test("Petición para visualizar datos de usuario superadministrador", function ()
2   pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
3 });
```

Test scripts are written and are run after the request is received. Learn more [about tests scripts](#)

Snippets

Body Cookies Headers (10) Test Results (0/1) Status: 404 Not Found Time: 3.62 s Size: 15.83 KB

All Passed Skipped Failed

FAIL Petición para visualizar datos de usuario superadministrador | AssertionError: expected 404 to be one of [200, 201, 202]

Fig. 42: Error recibido luego de que el código de respuesta no sea el esperado

Pruebas de carga

A través de la herramienta JMeter, se ha evaluado el módulo referente al CRUD de emprendimientos, en donde, se midió el tiempo de respuesta para 36 usuarios concurrentes en un periodo de 1 segundos como se estableció en la Fig. 43.

Thread Group

Name:

Comments:

Action to be taken after a Sampler error

Continue Start Next Thread Loop Stop Thread Stop Test

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count: Infinite

Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Fig. 43: Definición de argumentos para prueba de carga

Los resultados obtenidos, tomando en cuenta los parámetros antes mencionados, se muestran en la Fig. 44. En donde, se observa que no se han presentado ningún error al momento de enviar cada una de las peticiones y sus tiempos de ejecución no son altos.

View Results in Table

Name: RESULTADOS PARA ENDPOINT DE VISUALIZACIÓN DE EMPRENDIMIENTOS

Comments:

Write results to file / Read from file

Filename: Log/Display Only: Errors Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Co
1	05:14:34.016	Módulo de em...	ver emprendi...	3720	✓	8996	156	3720	
2	05:14:33.861	Módulo de em...	ver emprendi...	3879	✓	8996	156	3879	
3	05:14:33.990	Módulo de em...	ver emprendi...	3755	✓	8996	156	3754	
4	05:14:33.932	Módulo de em...	ver emprendi...	3814	✓	8996	156	3811	
5	05:14:34.045	Módulo de em...	ver emprendi...	7006	✓	8996	156	7005	
6	05:14:34.072	Módulo de em...	ver emprendi...	6996	✓	8996	156	6995	
7	05:14:33.903	Módulo de em...	ver emprendi...	7170	✓	8996	156	7170	
8	05:14:33.959	Módulo de em...	ver emprendi...	7115	✓	8996	156	7115	
9	05:14:34.100	Módulo de em...	ver emprendi...	10279	✓	8996	156	10279	
10	05:14:34.182	Módulo de em...	ver emprendi...	10205	✓	8996	156	10205	
11	05:14:34.130	Módulo de em...	ver emprendi...	10258	✓	8996	156	10258	
12	05:14:34.153	Módulo de em...	ver emprendi...	10239	✓	8996	156	10239	
13	05:14:34.263	Módulo de em...	ver emprendi...	13448	✓	8996	156	13446	
14	05:14:34.207	Módulo de em...	ver emprendi...	13505	✓	8996	156	13504	
15	05:14:34.292	Módulo de em...	ver emprendi...	13430	✓	8996	156	13428	
16	05:14:34.246	Módulo de em...	ver emprendi...	13489	✓	8996	156	13487	
17	05:14:34.319	Módulo de em...	ver emprendi...	16699	✓	8996	156	16699	
18	05:14:34.372	Módulo de em...	ver emprendi...	16646	✓	8996	156	16646	
19	05:14:34.347	Módulo de em...	ver emprendi...	16681	✓	8996	156	16681	
20	05:14:34.400	Módulo de em...	ver emprendi...	16646	✓	8996	156	16646	

Scroll automatically? Child samples? No of Samples 36 Latest Sample 29565 Average 16680 Deviation 8304

Fig. 44: Resultados de ejecución de pruebas de carga

Dentro del ANEXO II Manual técnico se presentan las demás pruebas de carga implementadas en los módulos de emprendimiento.

Pruebas de estrés

Para el caso de las pruebas de estrés, se han utilizado el mismo módulo que en el caso de la prueba anterior para verificar cuantos usuarios en un mismo periodo de tiempo pueden hacer uso del sistema.

En la Fig. 45 se muestra que la aplicación únicamente ha logrado realizar 36 peticiones, a partir de la petición 37 el servidor ya no se encontraba disponible.

View Results in Table

Name: RESULTADOS PARA ENDPOINT DE VISUALIZACIÓN DE EMPRENDIMIENTOS

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency
26	05:24:45.845	Módulo de em...	ver emprendi...	23430	✓	8994	156	23430
27	05:24:45.803	Módulo de em...	ver emprendi...	23574	✓	8994	156	23574
28	05:24:45.866	Módulo de em...	ver emprendi...	23536	✓	8994	156	23536
29	05:24:45.887	Módulo de em...	ver emprendi...	26640	✓	8994	156	26637
30	05:24:45.929	Módulo de em...	ver emprendi...	26688	✓	8994	156	26684
31	05:24:45.915	Módulo de em...	ver emprendi...	26775	✓	8994	156	26770
32	05:24:45.952	Módulo de em...	ver emprendi...	26760	✓	8994	156	26760
33	05:24:45.973	Módulo de em...	ver emprendi...	29860	✓	8994	156	29860
34	05:24:45.995	Módulo de em...	ver emprendi...	29933	✓	8994	156	29933
35	05:24:46.039	Módulo de em...	ver emprendi...	29960	✓	8994	156	29960
36	05:24:46.016	Módulo de em...	ver emprendi...	30005	✓	8994	156	30005
37	05:24:46.063	Módulo de em...	ver emprendi...	30367	✗	715	156	30366
38	05:24:46.102	Módulo de em...	ver emprendi...	30371	✗	715	156	30371
39	05:24:46.082	Módulo de em...	ver emprendi...	30413	✗	715	156	30413

Fig. 45: Resultados de ejecución de pruebas de estrés

En la Fig. 46, correspondiente al árbol de resultados obtenidos una vez finalizada la prueba, se observa que la respuesta obtenida del servidor en la última petición fue un error 503. Este error hace referencia a que el servidor ha recibido demasiadas peticiones durante un periodo de tiempo corto y ya no se encuentra disponible, por lo que no puede manejar la solicitud enviada.

Thread Name: Módulo de emprendimientos 1-120
 Sample Start: 2023-01-26 19:55:38 COT
 Load time: 30355
 Connect Time: 268
 Latency: 30355
 Size in bytes: 715
 Sent bytes: 187
 Headers size in bytes: 209
 Body size in bytes: 506
 Sample Count: 1
 Error Count: 1
 Data type ("text"|"bin"|""): text
Response code: 503
 Response message: Service Unavailable

HTTPSampleResult fields:
 ContentType: text/html; charset=utf-8
 DataEncoding: utf-8

Fig. 46: Error recibido luego de ejecución de pruebas de estrés

En el ANEXO II Manual técnico se han añadido las demás pruebas de estrés ejecutadas dentro del módulo de emprendimientos.

4 CONCLUSIONES

Dentro de este apartado, se describen las conclusiones obtenidas durante la ejecución de este proyecto.

- A través de reuniones con el cliente de este componente se ha logrado establecer una lista de requerimientos a llevar a cabo.
- Al establecer el levantamiento de requerimientos como fase inicial para este proyecto se ha logrado identificar las herramientas y metodologías adecuadas para cumplir con cada uno de estos de forma organizada y en un tiempo específico.
- La implementación de la arquitectura Modelo-Vista-Controlador (MVC) ha contribuido con la organización en el desarrollo del backend ya que se pudo dividir la lógica del negocio dependiendo del componente al que pertenecía cada ejecución.
- La metodología aplicada ha permitido dividir las secciones del proyecto y ordenarlas según los procesos que se requerían dentro de los tiempos establecidos.
- La ejecución de las pruebas establecidas para el backend, en este caso las pruebas de integración, pruebas de carga y estrés, han permitido verificar el funcionamiento de cada uno de los módulos implementados.
- Es importante tomar en consideración el límite de peticiones que soporta el backend en un determinado periodo de tiempo ya que de esto depende la capacidad de respuesta y la operatividad del sistema.

5 RECOMENDACIONES

Dentro de este apartado, se describen las recomendaciones recopiladas durante el desarrollo de este proyecto.

- Para el almacenamiento de las imágenes, tanto de perfiles de usuarios como logos de emprendimientos y demás, se ha utilizado la herramienta Cloudinary a través de credenciales “de prueba”. Por lo tanto, es recomendable cambiar dichas credenciales a unas propias de la comisión para tener acceso a la información registrada allí.
- Se recomienda el uso de credenciales propias de la comisión para el envío de correos en el caso de la creación de usuarios, ya que, aunque no se almacena información importante dentro del correo registrado actualmente como en el caso anterior, estas modificaciones pueden cambiar con el paso del tiempo.
- Se recomienda la implementación de un módulo para controlar el acceso de los datos de los emprendimientos creados desde fuera del sistema.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Comisión Emprende, «Lanzamiento de Radar Comercial ESFOT,» 16 Julio 2020. [En línea]. Available: https://www.facebook.com/watch/live/?ref=watch_permalink&v=3176977972389612.
- [2] M. Coppola, «¿Qué es una API REST?,» 23 Noviembre 2022. [En línea]. Available: <https://blog.hubspot.es/website/que-es-api-rest>. [Último acceso: 28 Noviembre 2022].
- [3] ULADECH CATÓLICA, «Universidad Católica Los Ángeles Chimbote,» 19 Diciembre 2017. [En línea]. Available: <https://www.uladech.edu.pe/images/stories/universidad/documentos/2018/metodologia-desarrollo-software-v001.pdf>. [Último acceso: 25 Noviembre 2022].
- [4] J. R. J. L. O. G. G. Holguer Romero, «Metodología de la Investigación,» 12 Agosto 2022. [En línea]. Available: https://acvenisproh.com/libros/index.php/Libros_categoria_Academico/article/view/22/29. [Último acceso: 1 Diciembre 2022].
- [5] V. Corona, J. Gutiérrez, L. Hernández y C. Rivas, «Metodologías actuales de desarrollo de software,» *Revista Tecnología e Innovación*, vol. II, nº 5 980-986, Diciembre 2015.
- [6] K.-D. Schewe y B. Thalheim, *Design and Development of Web Information Systems*, Primera ed., Berlin, 2019.
- [7] F. Machuca, «Crehana,» 18 Mayo 2022. [En línea]. Available: <https://www.crehana.com/blog/transformacion-digital/que-es-el-backend-y-como-usarlo/>. [Último acceso: 26 Noviembre 2022].
- [8] S. Pérez, J. Quispe, F. Mullicundo y D. Lamas, «Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd,» 16 Abril 2021. [En línea]. Available: <http://sedici.unlp.edu.ar/bitstream/handle/10915/120476/Ponencia.pdf-PDFA.pdf?sequence=1&isAllowed=y>.
- [9] E. Schenke y M. Pérez, «Un abordaje teórico de la investigación cualitativa como enfoque metodológico,» vol. 12, nº 30, pp. 227-233.
- [10] S. Meléndez, E. Gaitan y N. Pérez, 28 Enero 2016. [En línea]. Available: <https://repositorio.unan.edu.ni/1365/1/62161.pdf>. [Último acceso: 29 Noviembre 2022].
- [11] T. L. A. V. y R. F. Peláez Valencia, «INGE CUC,» 11 Noviembre 2019. [En línea]. Available: <https://revistascientificas.cuc.edu.co/ingecuc/article/view/2058>. [Último acceso: 20 Diciembre 2022].
- [12] Node.js, «NodeJs,» [En línea]. Available: <https://nodejs.org/es/>.

- [13] Composer, «Composer,» [En línea]. Available: <https://getcomposer.org/doc/00-intro.md>. [Último acceso: 28 Noviembre 2022].
- [14] Laravel, «Laravel,» [En línea]. Available: <https://laravel.com/docs/9.x/authorization>. [Último acceso: 06 Diciembre 2022].
- [15] R. Vadhineni, «Thunder Client,» [En línea]. Available: <https://www.thunderclient.com/>. [Último acceso: 29 Noviembre 2022].
- [16] GitHub, «GitHub,» [En línea]. Available: <https://docs.github.com/es>.
- [17] Laravel, «Laravel,» 29 Marzo 2020. [En línea]. Available: <https://laravel.com/docs/9.x/sanctum>. [Último acceso: 18 Diciembre 2022].
- [18] IBM, «IBM Integration Bus,» 24 Agosto 2022. [En línea]. Available: <https://www.ibm.com/docs/es/integration-bus/10.0?topic=types-esql-timestamp-data-type>. [Último acceso: 28 Diciembre 2022].
- [19] Heroku, «Heroku Dev Center,» 2022. [En línea]. Available: <https://devcenter.heroku.com/categories/reference>. [Último acceso: 04 Enero 2023].
- [20] Heroku, «The Procfile,» 16 Febrero 2022. [En línea]. Available: <https://devcenter.heroku.com/articles/procfile>. [Último acceso: 7 Enero 2023].
- [21] Heroku, «Customizing Web Server and Runtime Settings for PHP,» 10 Febrero 2020. [En línea]. Available: <https://devcenter.heroku.com/articles/custom-php-settings#setting-the-document-root>. [Último acceso: 09 Enero 2023].
- [22] D. Bhattacharjee, «What are Snippets in Postman?,» 25 Junio 2021. [En línea]. Available: <https://www.tutorialspoint.com/what-are-snippets-in-postman>. [Último acceso: 08 Febrero 2023].
- [23] Y. Chandra, «GitHub,» 17 Agosto 2022. [En línea]. Available: <https://github.com/yoelpc4/laravel-cloudinary>. [Último acceso: 30 Diciembre 2022].

7 ANEXOS

Dentro de la sección detallada a continuación, se han implementado los siguientes módulos:

- ANEXO I Certificado Turnitin.
- ANEXO II Manual técnico.
- ANEXO III Manual de usuario.
- ANEXO IV Manual de Instalación.



**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 1 de marzo de 2023

De mi consideración:

Yo, Juan Pablo Zaldumbide Proaño, en calidad de Director del Trabajo de Integración Curricular titulado **DESARROLLO DE UN BACKEND** asociado al proyecto denominado **DESARROLLO DE APLICACIÓN WEB DE PROMOCIÓN DE EMPRENDIMIENTOS DE LA COMUNIDAD DE LA ESFOT** elaborado por la estudiante **MARIA JOSÉ CHALÁ SANTACRUZ** de la carrera en Tecnología Superior en Desarrollo de Software, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 9%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,

Juan Pablo Zaldumbide Proaño
Profesor ocasional a tiempo completo
ESFOT

ANEXO II MANUAL TÉCNICO

Recopilación de requerimientos

En la TABLA IX mostrada a continuación, se presenta los requerimientos que se han obtenido durante las reuniones con el cliente.

TABLA IX Recopilación de requerimientos

RECOPIACIÓN DE REQUERIMIENTOS	
ID-RR	ENUNCIADO DEL ITEM
RR01	Tanto el usuario superadministrador y administrador necesitan un módulo de inicio de sesión para acceder al sistema.
RR02	Como usuarios superadministrador y administrador necesitan un módulo para: <ul style="list-style-type: none">• Restaurar la contraseña en caso de requerirlo.
RR03	Como usuarios superadministrador y administrador necesitan un módulo para: <ul style="list-style-type: none">• Actualizar la información de su perfil de usuario
RR04	Como usuario superadministrador requiere un módulo para: <ul style="list-style-type: none">• Almacenar información de nuevos usuarios dentro del sistema.
RR05	Como usuario superadministrador necesita de un método que le permita: <ul style="list-style-type: none">• Eliminar / Desactivar usuarios
RR06	Como usuario administrador necesita un método que le permita: <ul style="list-style-type: none">• Almacenar información de emprendimientos.
RR07	Como usuario administrador necesita un método que le permita: <ul style="list-style-type: none">• Eliminar información de emprendimientos en caso de requerirlo.
RR08	Como usuario administrador necesita un método que le permita: <ul style="list-style-type: none">• Actualizar información de emprendimientos en caso de requerirlo.
RR09	Se requiere un método para: <ul style="list-style-type: none">• Almacenar información de las videoconferencias
RR010	Se requiere un módulo para: <ul style="list-style-type: none">• Visualizar la información almacenada de las videoconferencias.

RR011	Se requiere un método para: <ul style="list-style-type: none"> • Eliminar la información de las videoconferencias.
--------------	---

Historias de usuario

En las tablas a continuación, desde la TABLA X hasta la TABLA XIX, se muestran las historias de usuarios del componente backend.

TABLA X Historia de usuario HU02

HISTORIA DE USUARIO	
Número: HU02	Usuario: Desarrollador frontend (cliente)
Nombre Historia: Restaurar contraseña	
Propiedad en Negocio: Medio	Riesgo en desarrollo: Medio
Iteración asignada: 1	
Responsable: María José Chalá	
Descripción: Como desarrollador frontend requiero un módulo para que el usuario final del sistema web pueda cambiar su contraseña en caso de olvidarla.	
Observación: El enlace para que el usuario pueda cambiar su contraseña, debe llegar al correo de dicho usuario	

TABLA XI Historia de usuario HU03

HISTORIA DE USUARIO	
Número: HU03	Usuario: Desarrollador frontend (cliente)
Nombre Historia: Actualizar información de usuario	
Propiedad en Negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable: María José Chalá	
Descripción: Como desarrollador frontend requiero que el usuario pueda actualizar la información de su perfil.	
Observación: Cada usuario podrá actualizar su información en caso de requerirlo.	

TABLA XII Historia de usuario HU04

HISTORIA DE USUARIO	
Número: HU04	Usuario: Desarrollador frontend
Nombre Historia: Almacenar información de usuarios	
Propiedad en Negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable: María José Chalá	
Descripción: Como desarrollador frontend requiero un módulo de registro de usuarios para agregar nuevos usuarios al sistema.	
Observación: Solo quienes se encuentren registrados con el tipo de usuario "superadmin" podrán almacenar información de nuevos usuarios dentro del sistema.	

TABLA XIII Historia de usuario HU05

HISTORIA DE USUARIO	
Número: HU05	Usuario: Desarrollador frontend (cliente)
Nombre Historia: Eliminar / Desactivar usuarios	
Propiedad en Negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 1	
Responsable: María José Chalá	
Descripción: Como desarrollador frontend requiero un módulo para eliminar cuentas de los usuarios que ya no pertenezcan al sistema.	
Observación: Solo quienes se encuentren registrados con el tipo de usuario "superadmin" podrán almacenar información de nuevos usuarios dentro del sistema.	

TABLA XIV Historia de usuario HU06

HISTORIA DE USUARIO	
Número: HU06	Usuario: Desarrollador frontend (cliente)
Nombre Historia: Almacenar información de emprendimientos	
Propiedad en Negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 2	
Responsable: María José Chalá	
Descripción: Como desarrollador frontend requiero un módulo para almacenar la información de los emprendimientos.	

Observación: Para el almacenamiento de emprendimientos es importante tomar en cuenta que se requiere un campo adicional para almacenamiento de la imagen.

TABLA XV Historia de usuario HU07

HISTORIA DE USUARIO	
Número: HU07	Usuario: Desarrollador frontend (cliente)
Nombre Historia: Eliminar información de emprendimientos	
Propiedad en Negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 2	
Responsable: María José Chalá	
Descripción: Como desarrollador frontend requiero que el usuario final del sistema pueda eliminar la información de los emprendimientos registrados en caso de requerirlo.	

TABLA XVI Historia de usuario HU08

HISTORIA DE USUARIO	
Número: HU08	Usuario: Desarrollador frontend (cliente)
Nombre Historia: Actualizar información de emprendimientos	
Propiedad en Negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 2	
Responsable: María José Chalá	
Descripción: Como desarrollador frontend requiero que el usuario final del sistema pueda eliminar la información de los emprendimientos registrados en caso de requerirlo.	
Observación: Para actualizar la información de los emprendimientos se deben tomar en consideración los valores de entrada requeridos.	

TABLA XVII Historia de usuario HU09

HISTORIA DE USUARIO	
Número: HU09	Usuario: Desarrollador frontend (cliente)
Nombre Historia: Almacenar información de videoconferencias	
Propiedad en Negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 2	
Responsable: María José Chalá	

Descripción: Los usuarios ‘superadministradores’ y ‘administradores’ requieren que la aplicación almacene la información de las videoconferencias que se vayan realizando.

Observación: Este módulo se encuentra disponible tanto para los superadministradores como los administradores por lo tanto ambos usuarios podrán almacenar la información.

TABLA XVIII Historia de usuario HU10

HISTORIA DE USUARIO	
Número: HU10	Usuario: Desarrollador frontend (cliente)
Nombre Historia: Publicar información de videoconferencias	
Propiedad en Negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 2	
Responsable: María José Chalá	
Descripción: Se requiere que los usuarios puedan visualizar la información de las videoconferencias en la aplicación.	
Observación: La información de las videoconferencias debe ser visible tanto para usuarios registrados en el sistema como para visitantes.	

TABLA XIX Historia de usuario HU11

HISTORIA DE USUARIO	
Número: HU11	Usuario: Desarrollador frontend (cliente)
Nombre Historia: Eliminar información de videoconferencias	
Propiedad en Negocio: Media	Riesgo en desarrollo: Media
Iteración asignada: 2	
Responsable: María José Chalá	
Descripción: Los usuarios ‘superadministradores’ y ‘administradores’ requieren eliminar información de las videoconferencias que ya no se encuentren disponibles	
Observación: Este módulo se encuentra disponible tanto para los superadministradores como los administradores por lo tanto ambos usuarios podrán eliminar la información.	

Pruebas de integración

A continuación, desde la Fig. 47 hasta la Fig. 57 se presentan las pruebas de integración faltantes:

TESIS MODULOS / SUPERADMIN / ver usuario superadministrador específico

GET `{{BaseUrl}}/api/v1/superadmin/21`

Params Authorization ● Headers (10) Body Pre-request Script Tests ● Settings

```
1 pm.test("Petición para visualizar datos de usuario superadministrador correcta",
2   function () {
3     pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
4   });
```

Body Cookies Headers (10) Test Results (1/1) 🌐 Status: 200 OK 1

All Passed Skipped Failed

PASS Petición para visualizar datos de usuario superadministrador correcta

Fig. 47: Prueba de integración en petición para visualizar datos de usuarios

TESIS MODULOS / SUPERADMIN / crear usuario superadministrador

POST `{{BaseUrl}}/api/v1/superadmin/create`

Params Authorization ● Headers (12) Body ● Pre-request Script Tests ● Settings

```
1 pm.test("Petición para almacenar datos de usuario superadministrador", function (
2   pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
3 });
```

Body Cookies Headers (10) Test Results (1/1) 🌐 Status: 200 OK

All Passed Skipped Failed

PASS Petición para almacenar datos de usuario superadministrador

Fig. 48: Prueba de integración en petición para almacenar datos de usuarios

The screenshot displays a REST client interface for a POST request. The URL is `{{BaseUrl}}/api/v1/superadmin/21/update`. The 'Tests' tab is active, showing a single test case with the following code:

```
1 pm.test("Petición para actualizar datos de usuario superadministrador", function ()
2   pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
3 });
```

Below the code, the 'Test Results (1/1)' section shows a 'PASS' status for the test 'Petición para actualizar datos de usuario superadministrador'. The overall status is '200 OK'.

Fig. 49: Prueba de integración en petición para actualizar datos de usuarios

The screenshot displays a REST client interface for a GET request. The URL is `{{BaseUrl}}/api/v1/superadmin/22/destroy`. The 'Tests' tab is active, showing a single test case with the following code:

```
1 pm.test("Petición para eliminar datos de usuario superadministrador", function ()
2   pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
3 });
```

Below the code, the 'Test Results (1/1)' section shows a 'PASS' status for the test 'Petición para eliminar datos de usuario superadministrador'. The overall status is '200 OK'.

Fig. 50: Prueba de integración en petición para eliminar datos de usuarios

GET ▼ `{{BaseUrl}}/api/v1/emprendimiento`

Params Authorization ● Headers (10) Body Pre-request Script **Tests ●** Settings

```
1 pm.test("Petición para listar todos los emprendimientos", function () {
2   pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
3 });
```

Body Cookies Headers (10) **Test Results (1/1)** 🌐 Status: 200 OK

All Passed Skipped Failed

PASS Petición para listar todos los emprendimientos

Fig. 51: Prueba de integración en petición para visualizar el listado de emprendimientos

GET ▼ `{{BaseUrl}}/api/v1/emprendimiento/7`

Params Authorization ● Headers (10) Body Pre-request Script **Tests ●** Settings

```
1 pm.test("Petición para visualizar información de emprendimiento", function () {
2   pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
3 });
```

Body Cookies Headers (10) **Test Results (1/1)** 🌐 Status: 200 OK

All Passed Skipped Failed

PASS Petición para visualizar información de emprendimiento

Fig. 52: Prueba de integración en petición para visualizar información a detalle de emprendimiento en específico

The screenshot displays a REST client interface for a POST request to the endpoint `{{BaseUrl}}/api/v1/emprendimiento/create`. The 'Tests' tab is active, showing a single test case: `pm.test("Petición para crear emprendimientos", function () { pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]); });`. The test results section shows a green 'PASS' status for the test 'Petición para crear emprendimientos'. The overall status is '200 OK'.

Fig. 53: Prueba de integración en petición para crear emprendimientos

The screenshot displays a REST client interface for a POST request to the endpoint `{{BaseUrl}}/api/v1/emprendimiento/3/update`. The 'Tests' tab is active, showing a single test case: `pm.test("Petición para actualizar emprendimientos", function () { pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]); });`. The test results section shows a green 'PASS' status for the test 'Petición para actualizar emprendimientos'. The overall status is '200 OK'.

Fig. 54: Prueba de integración en petición para actualizar información de emprendimientos

GET ▼ `{{BaseUrl}}/api/v1/emprendimiento/1/destroy`

Params Authorization ● Headers (10) Body Pre-request Script Tests ● Settings

```

1 pm.test("Petición para eliminar emprendimientos", function () {
2     pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
3 });
    
```

Body Cookies Headers (10) Test Results (1/1) 🌐 Status: 200 OK

All Passed Skipped Failed

PASS Petición para eliminar emprendimientos

Fig. 55: Prueba de integración en petición para eliminar información de emprendimientos

GET ▼ `{{BaseUrl}}/api/v1/videoconferencia`

Params Authorization ● Headers (10) Body Pre-request Script Tests ● Settings

```

1 pm.test("Petición para listar todos las videoconferencias", function () {
2     pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
3 });
    
```

Body Cookies Headers (10) Test Results (1/1) 🌐 Status: 200 OK

All Passed Skipped Failed

PASS Petición para listar todos las videoconferencias

Fig. 56: Prueba de integración en petición para enlistar información de videoconferencias

TESIS MODULOS / VIDEOCONFERENCIAS / ver videoconferencia en específico

GET `{{BaseUrl}}/api/v1/videoconferencia/6`

Params Authorization Headers (10) Body Pre-request Script Tests Settings

```

1 pm.test("Petición para visualizar información de videoconferencia", function () {
2   pm.expect(pm.response.code).to.be.oneOf([200, 201, 202]);
3 });

```

Body Cookies Headers (10) Test Results (1/1) Status: 200 OK

All Passed Skipped Failed

PASS Petición para visualizar información de videoconferencia

Fig. 57: Prueba de integración en petición para ver a detalle la información un de emprendimiento

Pruebas de carga

A continuación, desde la Fig. 58 hasta la Fig. 61 se muestra la ejecución de las pruebas de carga para los módulos faltantes del endpoint de emprendimientos:

Samp...	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	S
1	05:34:09.764	Módulo de emprendimientos 2-1	ver emprendimiento en específico	1913	✓	856	
2	05:34:09.856	Módulo de emprendimientos 2-4	ver emprendimiento en específico	1839	✓	856	
3	05:34:09.886	Módulo de emprendimientos 2-5	ver emprendimiento en específico	1815	✓	856	
4	05:34:09.941	Módulo de emprendimientos 2-7	ver emprendimiento en específico	3187	✓	856	
5	05:34:09.791	Módulo de emprendimientos 2-2	ver emprendimiento en específico	3340	✓	856	
6	05:34:09.913	Módulo de emprendimientos 2-6	ver emprendimiento en específico	3218	✓	856	
7	05:34:09.829	Módulo de emprendimientos 2-3	ver emprendimiento en específico	3857	✓	856	
8	05:34:09.997	Módulo de emprendimientos 2-9	ver emprendimiento en específico	4391	✓	856	
9	05:34:10.053	Módulo de emprendimientos 2-11	ver emprendimiento en específico	4341	✓	856	
10	05:34:09.969	Módulo de emprendimientos 2-8	ver emprendimiento en específico	4428	✓	856	
11	05:34:10.026	Módulo de emprendimientos 2-10	ver emprendimiento en específico	5099	✓	856	
12	05:34:10.081	Módulo de emprendimientos 2-12	ver emprendimiento en específico	5740	✓	856	
13	05:34:10.108	Módulo de emprendimientos 2-13	ver emprendimiento en específico	5713	✓	856	
14	05:34:10.136	Módulo de emprendimientos 2-14	ver emprendimiento en específico	5699	✓	856	
15	05:34:10.164	Módulo de emprendimientos 2-15	ver emprendimiento en específico	6225	✓	855	
16	05:34:10.220	Módulo de emprendimientos 2-17	ver emprendimiento en específico	7018	✓	855	
17	05:34:10.193	Módulo de emprendimientos 2-16	ver emprendimiento en específico	7048	✓	855	
18	05:34:10.248	Módulo de emprendimientos 2-18	ver emprendimiento en específico	7008	✓	855	
19	05:34:10.275	Módulo de emprendimientos 2-19	ver emprendimiento en específico	7539	✓	855	
20	05:34:10.304	Módulo de emprendimientos 2-20	ver emprendimiento en específico	8199	✓	855	

Fig. 58: Resultados de prueba de carga en endpoint para visualizar la información de emprendimiento en específico

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latenc...	Conn...
131	05:40:13.075	Módulo de emprendimientos 2-131	crear emprendimiento	678	✓	20065	639	383	282
132	05:40:13.112	Módulo de emprendimientos 2-132	crear emprendimiento	721	✓	20065	639	401	293
133	05:40:13.143	Módulo de emprendimientos 2-133	crear emprendimiento	717	✓	20065	639	415	303
134	05:40:13.210	Módulo de emprendimientos 2-135	crear emprendimiento	699	✓	20065	639	405	289
135	05:40:13.276	Módulo de emprendimientos 2-137	crear emprendimiento	695	✓	20065	639	399	284
136	05:40:13.244	Módulo de emprendimientos 2-136	crear emprendimiento	701	✓	20065	639	417	302
137	05:40:13.177	Módulo de emprendimientos 2-134	crear emprendimiento	739	✓	20065	639	410	295
138	05:40:13.308	Módulo de emprendimientos 2-138	crear emprendimiento	697	✓	20065	639	405	291
139	05:40:13.340	Módulo de emprendimientos 2-139	crear emprendimiento	685	✓	20065	639	377	277
140	05:40:13.375	Módulo de emprendimientos 2-140	crear emprendimiento	711	✓	20065	639	394	279
141	05:40:13.440	Módulo de emprendimientos 2-142	crear emprendimiento	662	✓	20065	639	381	277
142	05:40:13.409	Módulo de emprendimientos 2-141	crear emprendimiento	732	✓	20065	639	419	306
143	05:40:13.490	Módulo de emprendimientos 2-144	crear emprendimiento	675	✓	20065	639	391	279
144	05:40:13.473	Módulo de emprendimientos 2-143	crear emprendimiento	707	✓	20065	639	422	302
145	05:40:13.524	Módulo de emprendimientos 2-145	crear emprendimiento	713	✓	20065	639	395	283
146	05:40:13.573	Módulo de emprendimientos 2-146	crear emprendimiento	701	✓	20065	639	390	291
147	05:40:13.605	Módulo de emprendimientos 2-147	crear emprendimiento	722	✓	20065	639	396	290
148	05:40:13.637	Módulo de emprendimientos 2-148	crear emprendimiento	705	✓	20065	639	414	302
149	05:40:13.671	Módulo de emprendimientos 2-149	crear emprendimiento	704	✓	20065	639	421	308
150	05:40:13.705	Módulo de emprendimientos 2-150	crear emprendimiento	691	✓	20065	639	387	285

Scroll automatically?
 Child samples?
No of Samples 150
Latest Sample 691
Average 726
Deviation 50

Fig. 59: Resultados de prueba de carga en módulo para crear emprendimientos

Sample #	Start Time	Thread Name	Label	Sample Ti...	Status
40	05:43:49.721	Módulo de emprendimientos 2-34	actualizar emprendimiento	14722	✓
41	05:43:49.819	Módulo de emprendimientos 2-45	actualizar emprendimiento	15846	✓
42	05:43:49.759	Módulo de emprendimientos 2-39	actualizar emprendimiento	15906	✓
43	05:43:49.768	Módulo de emprendimientos 2-40	actualizar emprendimiento	15905	✓
44	05:43:49.839	Módulo de emprendimientos 2-47	actualizar emprendimiento	16022	✓
45	05:43:49.778	Módulo de emprendimientos 2-41	actualizar emprendimiento	17314	✓
46	05:43:49.832	Módulo de emprendimientos 2-46	actualizar emprendimiento	17267	✓
47	05:43:49.798	Módulo de emprendimientos 2-43	actualizar emprendimiento	17324	✓
48	05:43:49.863	Módulo de emprendimientos 2-49	actualizar emprendimiento	17418	✓
49	05:43:49.849	Módulo de emprendimientos 2-48	actualizar emprendimiento	18670	✓
50	05:43:49.912	Módulo de emprendimientos 2-54	actualizar emprendimiento	18610	✓
51	05:43:49.879	Módulo de emprendimientos 2-51	actualizar emprendimiento	18660	✓
52	05:43:49.889	Módulo de emprendimientos 2-52	actualizar emprendimiento	18827	✓
53	05:43:49.867	Módulo de emprendimientos 2-50	actualizar emprendimiento	20080	✓
54	05:43:49.920	Módulo de emprendimientos 2-55	actualizar emprendimiento	20027	✓
55	05:43:49.930	Módulo de emprendimientos 2-56	actualizar emprendimiento	20043	✓
56	05:43:49.940	Módulo de emprendimientos 2-57	actualizar emprendimiento	20204	✓
57	05:43:49.950	Módulo de emprendimientos 2-58	actualizar emprendimiento	21425	✓
58	05:43:49.993	Módulo de emprendimientos 2-62	actualizar emprendimiento	21391	✓
59	05:43:49.959	Módulo de emprendimientos 2-59	actualizar emprendimiento	21596	✓

Fig. 60: Resultados de prueba de carga en módulo para actualizar información de emprendimientos

Sample #	Start Time	Thread Name	Label	Sample Ti...	Status
17	05:47:09.112	Módulo de emprendimientos 2-18	eliminar emprendimiento	7172	✓
18	05:47:08.972	Módulo de emprendimientos 2-13	eliminar emprendimiento	7333	✓
19	05:47:09.196	Módulo de emprendimientos 2-21	eliminar emprendimiento	7119	✓
20	05:47:09.141	Módulo de emprendimientos 2-19	eliminar emprendimiento	7177	✓
21	05:47:09.168	Módulo de emprendimientos 2-20	eliminar emprendimiento	8543	✓
22	05:47:09.252	Módulo de emprendimientos 2-23	eliminar emprendimiento	8480	✓
23	05:47:09.280	Módulo de emprendimientos 2-24	eliminar emprendimiento	8467	✓
24	05:47:09.223	Módulo de emprendimientos 2-22	eliminar emprendimiento	8525	✓
25	05:47:09.308	Módulo de emprendimientos 2-25	eliminar emprendimiento	9832	✓
26	05:47:09.335	Módulo de emprendimientos 2-26	eliminar emprendimiento	9822	✓
27	05:47:09.362	Módulo de emprendimientos 2-27	eliminar emprendimiento	9801	✓
28	05:47:09.420	Módulo de emprendimientos 2-29	eliminar emprendimiento	9751	✓
29	05:47:09.391	Módulo de emprendimientos 2-28	eliminar emprendimiento	11185	✓
30	05:47:09.475	Módulo de emprendimientos 2-31	eliminar emprendimiento	11109	✓
31	05:47:09.447	Módulo de emprendimientos 2-30	eliminar emprendimiento	11139	✓
32	05:47:09.502	Módulo de emprendimientos 2-32	eliminar emprendimiento	11088	✓
33	05:47:09.531	Módulo de emprendimientos 2-33	eliminar emprendimiento	12459	✓
34	05:47:09.558	Módulo de emprendimientos 2-34	eliminar emprendimiento	12452	✓
35	05:47:09.587	Módulo de emprendimientos 2-35	eliminar emprendimiento	12425	✓
36	05:47:09.614	Módulo de emprendimientos 2-36	eliminar emprendimiento	12410	✓

Fig. 61: Resultados de prueba de carga en módulo para eliminar información de emprendimientos

Pruebas de estrés

A continuación, desde la Fig. 62 hasta la Fig. 64 se muestra la ejecución de las pruebas de estrés para los módulos faltantes del endpoint de emprendimientos:

Samp...	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Se
17	05:34:10.193	Módulo de emprendimientos 2-16	ver emprendimiento en específico	7048	✓	855	
18	05:34:10.248	Módulo de emprendimientos 2-18	ver emprendimiento en específico	7008	✓	855	
19	05:34:10.275	Módulo de emprendimientos 2-19	ver emprendimiento en específico	7539	✓	855	
20	05:34:10.304	Módulo de emprendimientos 2-20	ver emprendimiento en específico	8199	✓	855	
21	05:34:10.331	Módulo de emprendimientos 2-21	ver emprendimiento en específico	8181	✓	855	
22	05:34:10.388	Módulo de emprendimientos 2-23	ver emprendimiento en específico	8129	✓	855	
23	05:34:10.360	Módulo de emprendimientos 2-22	ver emprendimiento en específico	8903	✓	855	
24	05:34:10.472	Módulo de emprendimientos 2-26	ver emprendimiento en específico	9142	✗	420	
25	05:34:10.444	Módulo de emprendimientos 2-25	ver emprendimiento en específico	9187	✗	420	
26	05:34:10.416	Módulo de emprendimientos 2-24	ver emprendimiento en específico	9511	✓	855	
27	05:34:10.499	Módulo de emprendimientos 2-27	ver emprendimiento en específico	9709	✗	420	
28	05:34:10.527	Módulo de emprendimientos 2-28	ver emprendimiento en específico	10197	✗	420	
29	05:34:10.555	Módulo de emprendimientos 2-29	ver emprendimiento en específico	10177	✗	420	
30	05:34:10.584	Módulo de emprendimientos 2-30	ver emprendimiento en específico	10455	✗	420	
31	05:34:10.637	Módulo de emprendimientos 2-32	ver emprendimiento en específico	10513	✗	420	
32	05:34:10.666	Módulo de emprendimientos 2-33	ver emprendimiento en específico	11173	✗	420	
33	05:34:10.612	Módulo de emprendimientos 2-31	ver emprendimiento en específico	11232	✗	420	
34	05:34:10.721	Módulo de emprendimientos 2-35	ver emprendimiento en específico	11377	✗	420	
35	05:34:10.693	Módulo de emprendimientos 2-34	ver emprendimiento en específico	11450	✗	420	
36	05:34:10.752	Módulo de emprendimientos 2-36	ver emprendimiento en específico	12510	✓	856	

Fig. 62: Resultados de prueba de estrés en módulo para visualizar información específica de emprendimientos

Sample #	Start Time	Thread Name	Label	Sample Ti...	Status
60	05:43:49.899	Módulo de emprendimientos 2-53	actualizar emprendimiento	21756	✓
61	05:43:50.030	Módulo de emprendimientos 2-66	actualizar emprendimiento	22633	✗
62	05:43:50.009	Módulo de emprendimientos 2-64	actualizar emprendimiento	22749	✗
63	05:43:50.000	Módulo de emprendimientos 2-63	actualizar emprendimiento	22790	✓
64	05:43:50.022	Módulo de emprendimientos 2-65	actualizar emprendimiento	22793	✓
65	05:43:50.050	Módulo de emprendimientos 2-68	actualizar emprendimiento	23726	✗
66	05:43:49.986	Módulo de emprendimientos 2-61	actualizar emprendimiento	23884	✗
67	05:43:49.969	Módulo de emprendimientos 2-60	actualizar emprendimiento	23950	✗
68	05:43:50.069	Módulo de emprendimientos 2-70	actualizar emprendimiento	23866	✗
69	05:43:50.091	Módulo de emprendimientos 2-72	actualizar emprendimiento	24795	✗
70	05:43:50.040	Módulo de emprendimientos 2-67	actualizar emprendimiento	24937	✗
71	05:43:50.080	Módulo de emprendimientos 2-71	actualizar emprendimiento	24968	✗
72	05:43:50.057	Módulo de emprendimientos 2-69	actualizar emprendimiento	24995	✗
73	05:43:50.100	Módulo de emprendimientos 2-73	actualizar emprendimiento	25993	✗
74	05:43:50.111	Módulo de emprendimientos 2-74	actualizar emprendimiento	25982	✗
75	05:43:50.130	Módulo de emprendimientos 2-76	actualizar emprendimiento	26063	✗
76	05:43:50.117	Módulo de emprendimientos 2-75	actualizar emprendimiento	26076	✗
77	05:43:50.141	Módulo de emprendimientos 2-77	actualizar emprendimiento	27078	✗
78	05:43:50.199	Módulo de emprendimientos 2-83	actualizar emprendimiento	27171	✗
79	05:43:50.180	Módulo de emprendimientos 2-81	actualizar emprendimiento	27190	✗

Scroll automatically? Child samples? No of Samples 500 Latest Sample 35620 Average 28831

Fig. 63: Resultados de prueba de estrés en módulo para actualizar información de emprendimientos

Sample #	Start Time	Thread Name	Label	Sample Ti...	Status
45	05:53:01.027	Módulo de emprendimientos 2-46	eliminar emprendimiento	17003	✓
46	05:53:01.064	Módulo de emprendimientos 2-48	eliminar emprendimiento	16989	✓
47	05:53:01.042	Módulo de emprendimientos 2-47	eliminar emprendimiento	17057	✓
48	05:53:01.075	Módulo de emprendimientos 2-49	eliminar emprendimiento	17061	✓
49	05:53:01.091	Módulo de emprendimientos 2-50	eliminar emprendimiento	18374	✓
50	05:53:01.107	Módulo de emprendimientos 2-51	eliminar emprendimiento	18374	✓
51	05:53:01.139	Módulo de emprendimientos 2-53	eliminar emprendimiento	18383	✓
52	05:53:01.123	Módulo de emprendimientos 2-52	eliminar emprendimiento	18440	✓
53	05:53:01.171	Módulo de emprendimientos 2-55	eliminar emprendimiento	19720	✓
54	05:53:01.156	Módulo de emprendimientos 2-54	eliminar emprendimiento	19757	✓
55	05:53:01.188	Módulo de emprendimientos 2-56	eliminar emprendimiento	19758	✓
56	05:53:01.204	Módulo de emprendimientos 2-57	eliminar emprendimiento	19771	✓
57	05:53:01.235	Módulo de emprendimientos 2-59	eliminar emprendimiento	21076	✓
58	05:53:01.220	Módulo de emprendimientos 2-58	eliminar emprendimiento	21115	✓
59	05:53:01.251	Módulo de emprendimientos 2-60	eliminar emprendimiento	21118	✓
60	05:53:01.271	Módulo de emprendimientos 2-61	eliminar emprendimiento	21137	✓
61	05:53:01.300	Módulo de emprendimientos 2-63	eliminar emprendimiento	22120	✗
62	05:53:01.316	Módulo de emprendimientos 2-64	eliminar emprendimiento	22129	✗
63	05:53:01.284	Módulo de emprendimientos 2-62	eliminar emprendimiento	22193	✗
64	05:53:00.643	Módulo de emprendimientos 2-22	eliminar emprendimiento	22872	✗

Fig. 64: Resultados de prueba de estrés en módulo para eliminar información de emprendimientos

ANEXO III Manual de usuario

A continuación, como parte de este apartado se presenta el manual de usuario en donde se detallan los perfiles de usuario que maneja la aplicación, los módulos principales que se implementaron dentro de este componente y la funcionalidad de cada uno de ellos.

<https://github.com/Maria-Jose-11/backend-tesis>

ANEXO IV MANUAL DE INSTALACIÓN

Como parte del manual de instalación, se adjunta el enlace para acceder al backend desplegado:

<https://backend-emprende.herokuapp.com/comision-emprende>

Adicional a ello, se adjunta el enlace para acceder al código fuente del componente backend, mismo que se encuentra dentro de un repositorio en GitHub:

<https://github.com/Maria-Jose-11/backend-tesis>