

Daw

Sistema gestión incidencias

4 de Marzo del 2022

Teresa Melero Ligeró



1.Introducción	3
2.Descripción	4
3.Instalación y preparación	5
4. Guía de estilos y prototipado	6
5. Diseño	8
6. Desarrollo	9
7. Pruebas	10
8. Despliegue	11
9. Manual	13
10. Conclusiones	14
11. Índice de tablas e imágenes	15

1.Introducción

La idea inicial de proyecto, nada tiene que ver con la desarrollada finalmente. La primera opción, era crear una tienda online, sin embargo, cuando realizaba mis prácticas de empresa, observe que el ayuntamiento de vejer, al cual mi empresa daba soporte técnico, no disponía de un sistema que gestionará las incidencias de sus equipos y que para comunicar una incidencia lo realizaban a través de correos electrónicos que gestiona un funcionario del ayuntamiento. En ese momento, decidí cambiar la propuesta de proyecto y realizar un pequeño sistema que gestionará las incidencias (sistema ticketing).

Una vez dilucidada la idea a desarrollar, el siguiente paso era decidir qué tecnologías iba a utilizar para el proyecto. Finalmente me decidí por las siguientes tecnologías:

En la parte del cliente:

→ Framework vue.js

- es un **framework progresivo** Javascript para crear interfaces de usuario

- Html
- css
- Javascript
- Ajax
- JQuery

Para el servidor elegí:

- Framework Laravel
 - Es un popular framework de PHP. Permite el desarrollo de aplicaciones web totalmente personalizadas de elevada calidad.
- Php

Tenía cierta idea del funcionamiento de este tipo de sistemas, pero para tener las ideas claras me documenté y así obtener las funcionalidades a destacar, tipo de estructura, usuarios a los que va destinada, las metodologías en las que se basan, etc.

Destaco algunos ejemplos de sistemas de gestión de incidencias:

- ☐ JIRA Service Management
- ☐ Zendesk

- ☐ Inbenta
- ☐ Zoho desk

2.Descripción

El sistema de ticketing o de seguimiento de incidentes consiste en crear, gestionar y hacer un seguimiento de las incidencias y peticiones de los clientes. Este tipo de sistemas es muy utilizado por los departamentos de **SAT** (Servicio Asistencia Técnica).

Se define una incidencia como *“una interrupción no planificada de un servicio, o reducción en la calidad de un servicio”*. El objetivo del proceso de gestión de incidentes en **ITIL** (*conjunto de las mejores prácticas para ayudar a mejorar la prestación de un servicio, en particular un servicio de TI (tecnología de la información).*) es minimizar el impacto negativo de los incidentes mediante la restauración del funcionamiento normal del servicio lo más rápidamente que sea posible.

La gestión de incidencias bajo la metodología ITIL, tiene como objetivo prevenir o restaurar en el menor tiempo posible cualquier interrupción o retraso que afecte a la calidad del servicio (no planificada) y minimizar el impacto de las operaciones.

Cuando un cliente contacta con atención al cliente, el personal que atiende dicha incidencia abre un “ticket” con la información de la incidencia o duda

a tratar. Este ticket permanecerá abierto durante todo el seguimiento de la incidencia hasta que se haya resuelto la incidencia. Este tipo de sistemas se suele usar con bastante frecuencia en empresas de instalaciones y mantenimiento.



¿Qué es un ticket?

Un ticket es donde se encuentra la información de la incidencia que el cliente comunica. En él está la categoría a la que pertenece la incidencia, se realiza para tratar de comprender e identificar las causas por las que se dio una incidencia. También tenemos la severidad, que se clasifica en alta,

media o baja, el título y descripción, para que se describa el problema surgido.

¿Cómo funciona?

A través de una herramienta de ticketing, cada vez que un cliente realiza una solicitud de atención/servicio o relata un problema; se envía un ticket para que una persona de servicio técnico lo atienda.

1. El sistema de tickets convierte las solicitudes de los clientes en documentos digitales que son conocidos como tickets de soporte.
2. Cada uno de los tickets de soporte aparece en la pantalla principal de la herramienta y son catalogados automáticamente como incidente “pendiente”.
3. Los Estados de una incidencia son:

Estados	Descripción
Pendiente	Cuando se crea un ticket se categoriza en este estado. Así permanece hasta que algún agente realice una acción.
Asignado	Significa que el ticket es atendido por el personal de

	soporte y está en proceso de resolución. En este estado es el personal de soporte quien tiene que realizar la próxima acción.
Resuelto	Indica que el personal de soporte ha resuelto el ticket y brindó una solución al cliente.

4. El personal técnico podrá atender la incidencia, si es así, el estado de la incidencia pasará a estado “asignado”. Podrá editar la incidencia si fuera necesario, si el problema al que se enfrenta no puede resolverlo podrá derivarlo al siguiente soporte, para que pueda ser atendida. Si el problema puede ser resuelto podrá marcarla como resuelta y su estado pasará a “resuelto”.
5. Si el cliente que creó el ticket marca como resuelta la incidencia está no volverá a abrirse.
6. El sistema de gestión de incidencias cuenta con un administrador, que es el encargado de crear los proyectos, asignar los proyectos al personal de soporte técnico y de crear incidencias si fuera necesario.

7. El equipo de soporte técnico, tendrá proyectos asignados, podrá atender incidencias que no tenga asignadas y podrá crear incidencias.

¿Cómo se encuentra organizado el sistema?

Roles	Descripción
Administrador	<p>Gestiona el sistema</p> <hr/> <ul style="list-style-type: none">● Crea los proyectos● Asigna proyectos al personal de soporte.● Puede crear incidencias
Técnico de soporte	<ul style="list-style-type: none">● Atiende incidencias● Puede crear incidencias● Puede derivar incidencias a otro soporte
Cliente	<ul style="list-style-type: none">● Informa incidencias

Incidencias

Categorías
Programa malicioso
Pérdida de información personal
intrusión
Problema con hardware

Proyectos

Proyectos	Tipo soporte
proyecto a	<ul style="list-style-type: none">● soporte telefónico● envío técnico
proyecto b	<ul style="list-style-type: none">● soporte telefónico● envío técnico

Para concluir, esta aplicación ha sido creada siguiendo el patrón arquitectónico MVC(modelo vista controlador), que separa una aplicación en tres capas descritas como su acrónimo lo indica. Laravel, así como la

mayoría de frameworks en PHP implementan este patrón de diseño en donde cada capa maneja un aspecto de la aplicación.

Modelo: Hace referencia a la estructura de datos de la aplicación. Los datos pueden ser transferidos desde la base de datos, una clase, un servicio, u otros, directamente a la vista o ser transformados en el controlador para ser actualizados nuevamente al origen.

Vista: Es la representación de la información en una interfaz de usuario. Por lo general en interfaces no estáticas se representan los datos que vienen directamente del modelo o estos son transformados en un proceso intermedio en el controlador. En vistas estáticas por lo general no hace falta que las vistas sean renderizadas con datos enviados del controlador.

Controlador: Es el lugar en donde se implementa la lógica de la aplicación, los procedimientos, algoritmos y rutinas que hacen que funcione el software. Actúa como interfaz entre los componentes de modelo y vista aplicando las transformaciones y lógica necesarias.

3.Instalación y preparación

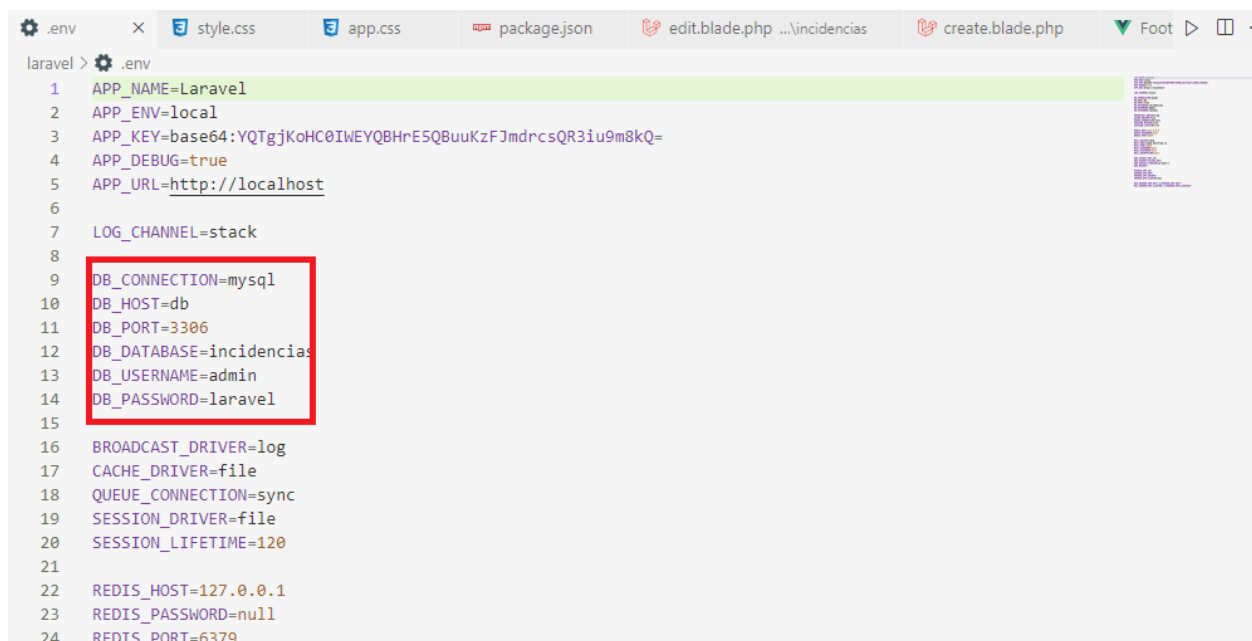
Para que nuestro sistema funcione correctamente, necesitamos de un entorno para desplegarlo, para ello, tenemos docker. Con esta herramienta podremos ejecutar nuestra aplicación en cualquier entorno.

Necesitaremos el archivo docker-compose.yml, donde estarán todas las estructuras necesarias para el sistema.

```
docker-compose.yml
1  version: '3.7'
2
3  services:
4    servidor:
5      image: fjortegan/dwes:laravel
6      stdin_open: true # docker run -i
7      tty: true        # docker run -t
8      ports:
9        - "80:80"
10     volumes:
11       - ./laravel:/var/www/html/
12
13   db:
14     image: mariadb
15
16     ports:
17       - "3306:3306"
18     volumes:
19       - ./db-data:/var/lib/mysql/
20     environment:
21       MYSQL_DATABASE: incidencias
22       MYSQL_USER: admin
23       MYSQL_PASSWORD: laravel
24       MYSQL_ROOT_PASSWORD: pestillo
```

En el framework de laravel, tendremos que configurar algunos parámetros en la base de datos, el nombre que le hemos otorgado, el usuario, contraseña y el servidor.

Para ello, nos iremos a archivo `.env` de laravel y modificaremos lo anteriormente mencionado. En la siguiente imagen puede observarse los cambios añadidos.



```
laravel > .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:YQTgjK0HC0IWEYQBHR5QBuuKzFJmdrcsQR3iu9m8kQ=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=db
11 DB_PORT=3306
12 DB_DATABASE=incidencias
13 DB_USERNAME=admin
14 DB_PASSWORD=laravel
15
16 BROADCAST_DRIVER=log
17 CACHE_DRIVER=file
18 QUEUE_CONNECTION=sync
19 SESSION_DRIVER=file
20 SESSION_LIFETIME=120
21
22 REDIS_HOST=127.0.0.1
23 REDIS_PASSWORD=null
24 REDIS_PORT=6379
```

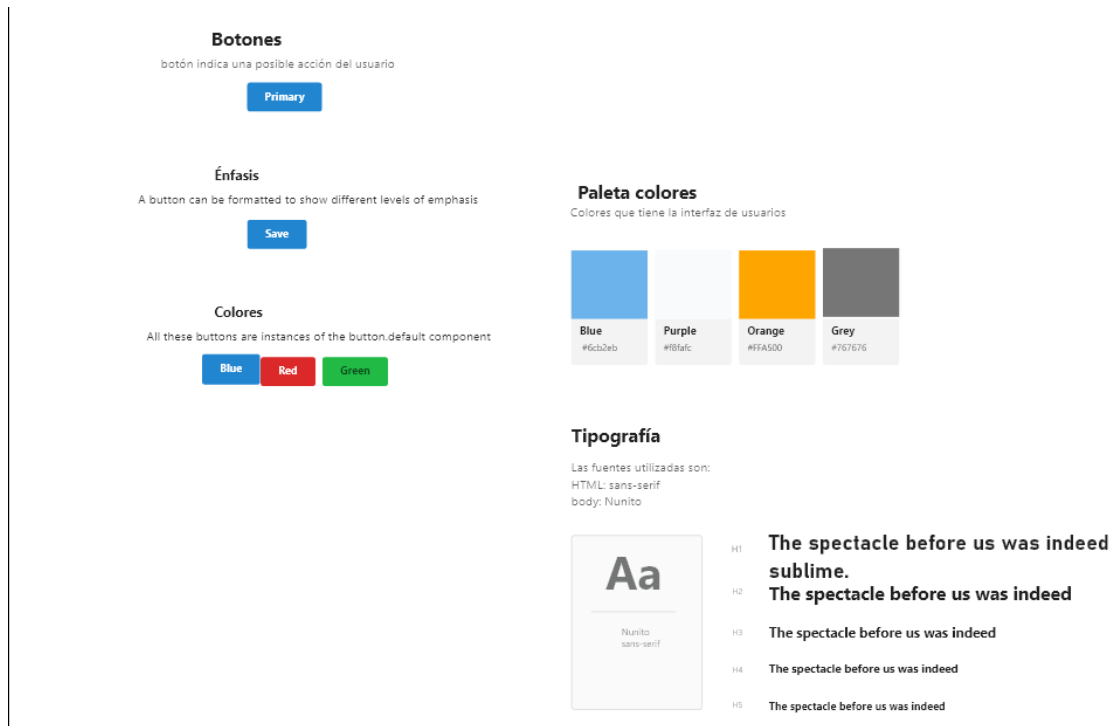
Realizado todo lo anteriormente explicado, nos iremos a nuestra terminal y ejecutaremos la orden: **docker-compose up**, para que se lleven a cabo las instrucciones indicadas en las estructuras de nuestro sistema, y así comenzar a utilizar la aplicación. Así pues, estará listo para poder utilizarse, y por tanto, poder satisfacer las necesidades para la cual fue creado.

También tenemos que hablar de los sistemas de control de versiones, son herramientas de software que ayudan a gestionar los cambios en el código fuente a lo largo del tiempo. Otra ventaja que nos aporta, es tener una copia de nuestro código en caso de que exista algún problema, como ya sea pérdida del proyecto, debido algún fallo técnico o humano.

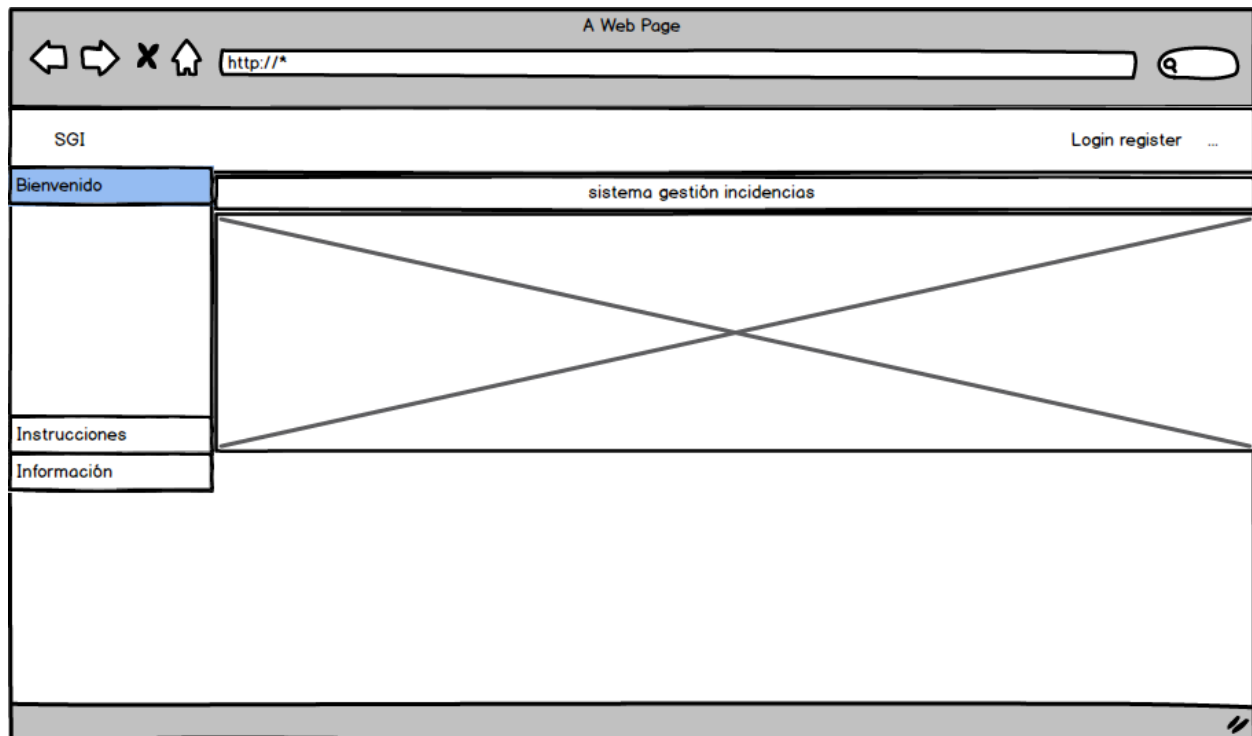
Para ello hemos utilizado github, un servicio en línea al que los desarrolladores que utilizan Git pueden conectarse y cargar o descargar recursos. En github, tenemos nuestro repositorio, donde se van subiendo nuestros cambios, para ello, lo hemos sincronizado desde el editor de código visual studio code, el cuál nos facilita la sincronización y desde él podemos ir subiendo todos los cambios a nuestro repositorio.

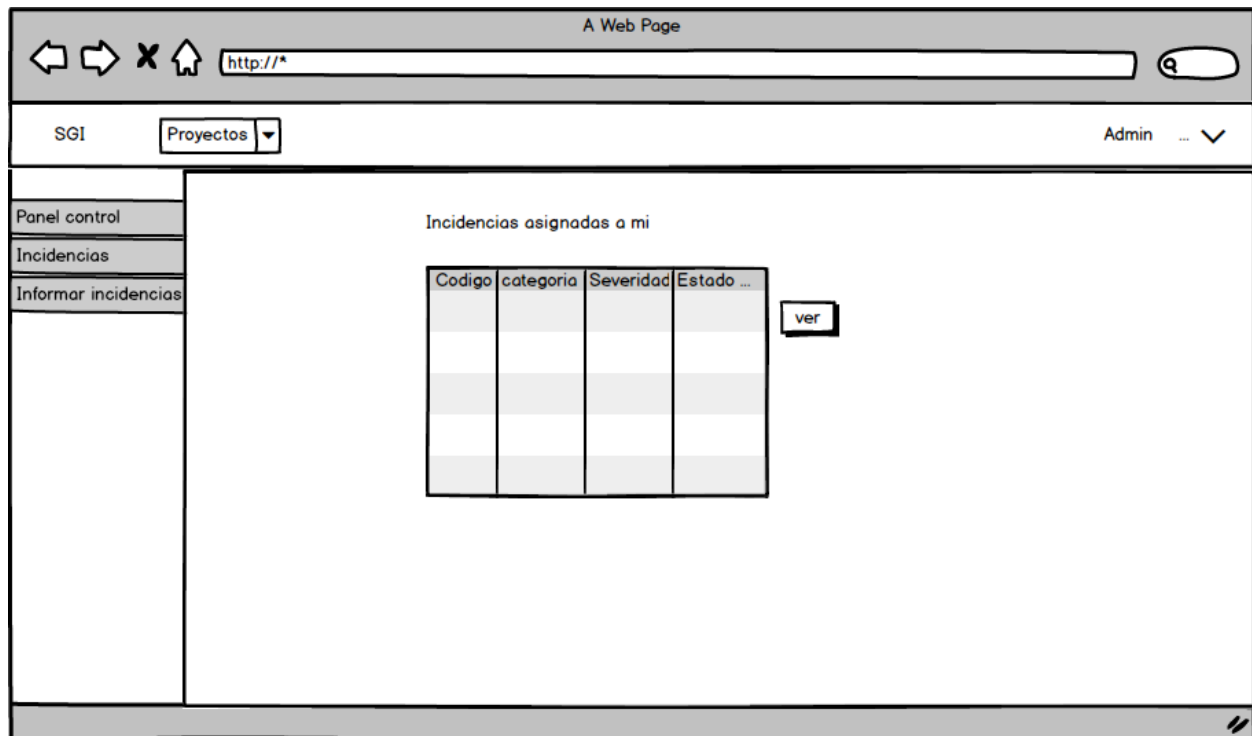
4. Guía de estilos y prototipado

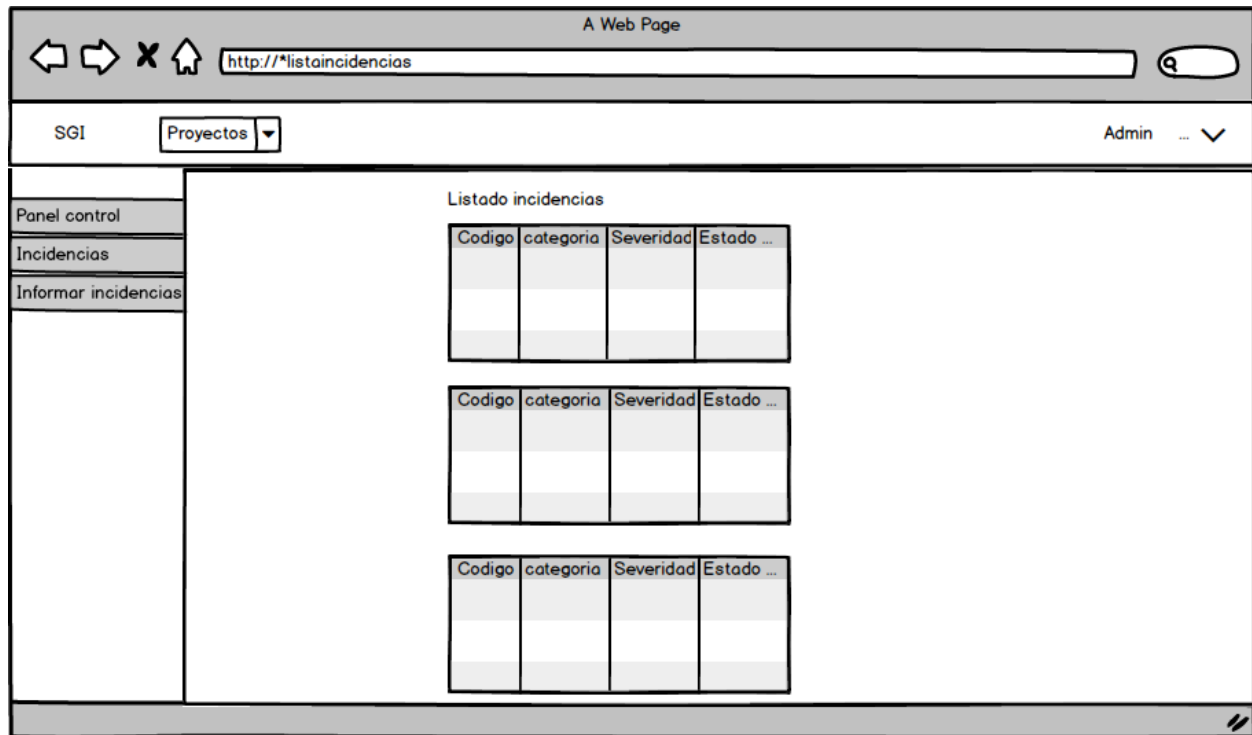
Definiremos nuestra guía de estilos, a continuación se muestra la imagen:



A continuación tenemos los mockups de la aplicación







A Web Page

http://*

SGI Proyectos Admin ▾

Administracion ▾



usuarios

E-mail

Nombre

Contraseña

Button

asignado a	Técnico de soporte	Opciones
tecnico@tecnico.com	trabajador soporte tecnico	 

A Web Page

http://*

SGI Proyectos Admin ▾

editar usuario

Administracion ▾

usuarios

E-mail

Nombre

Contraseña

guardar usuario

seleccionar proyecto ▾ Seleccionar Soporte ▾ Asignar proyecto

proyectos asignados

Proyecto	soporte	Opciones

🗑

A Web Page

http://*

SGI Proyectos Admin ▾

Administracion ▾

Proyectos

Nombre



Descripción

Fecha inicio

Registrar proyecto

FEBRUARY 2022

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Nombre proyecto	Descripción	Fecha inicio	Opciones
Proyecto1	descip	22/2/2022	 

A Web Page

http://*

SGI Proyectos Admin ▾

Administración ▾

Proyectos

Nombre

Descripción

Fecha inicio

Registrar proyecto

FEbruary 2022

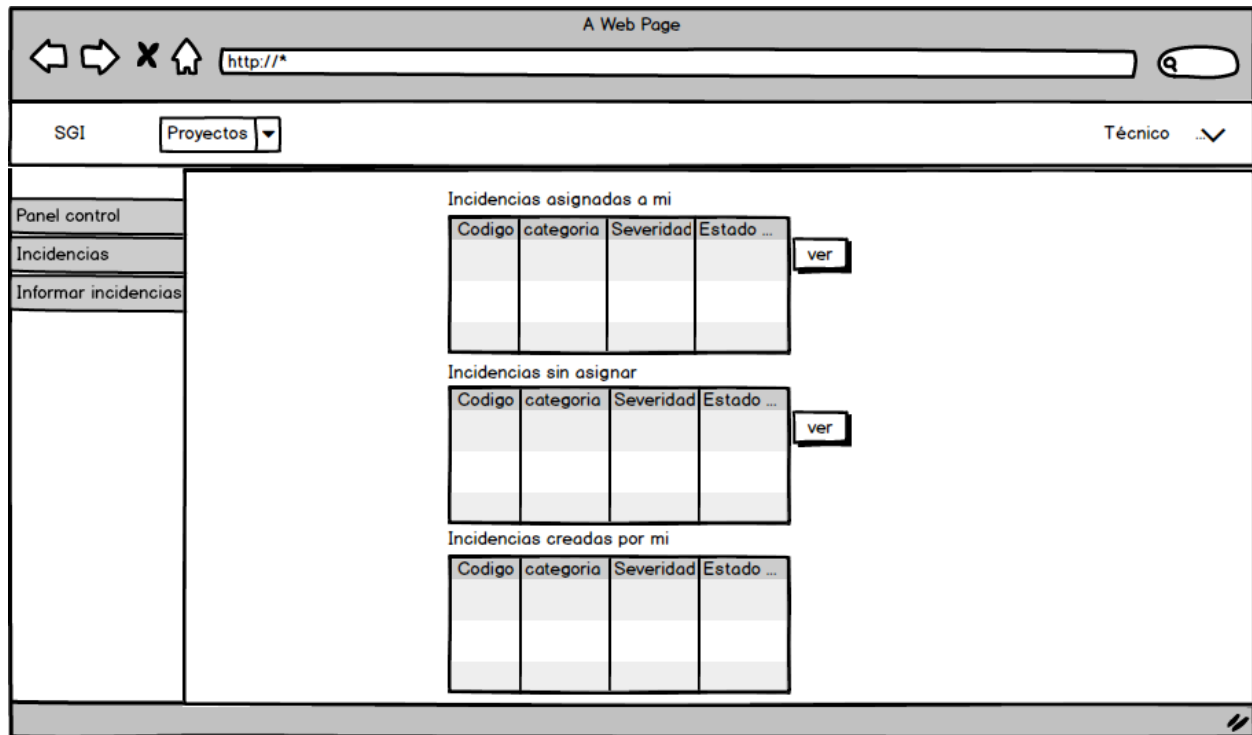
S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

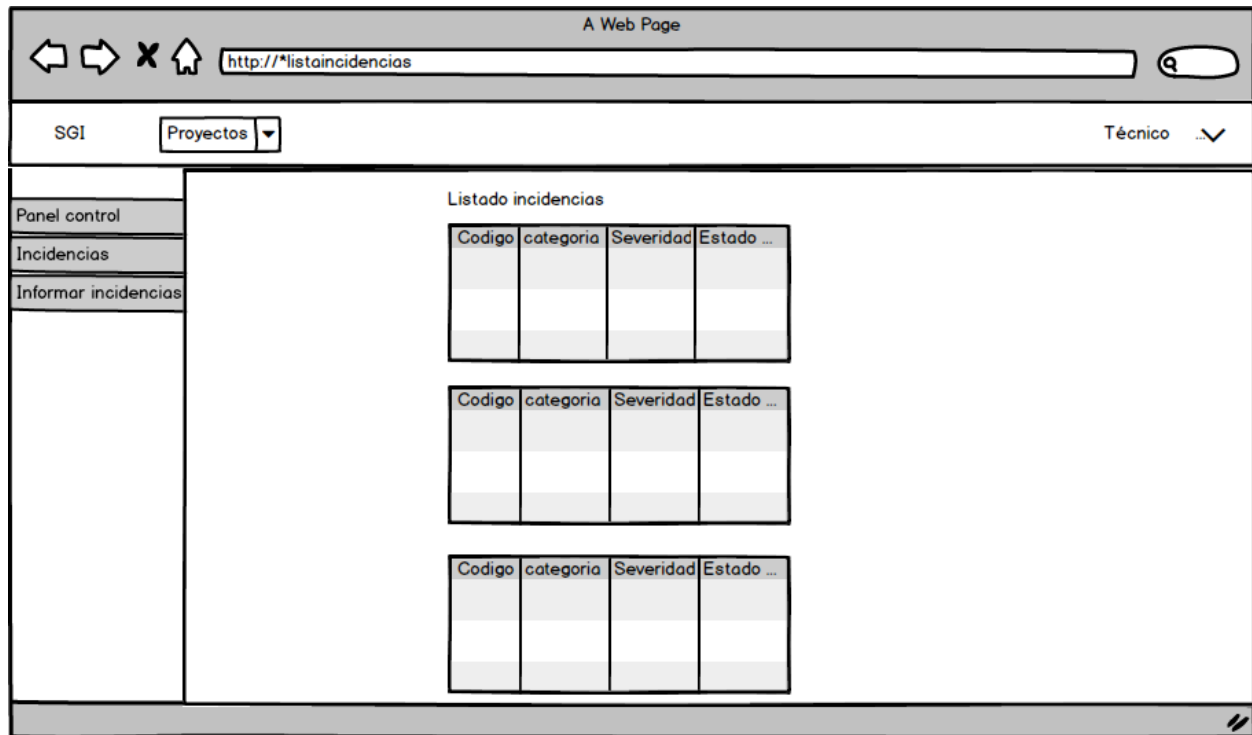
categorias Añadir

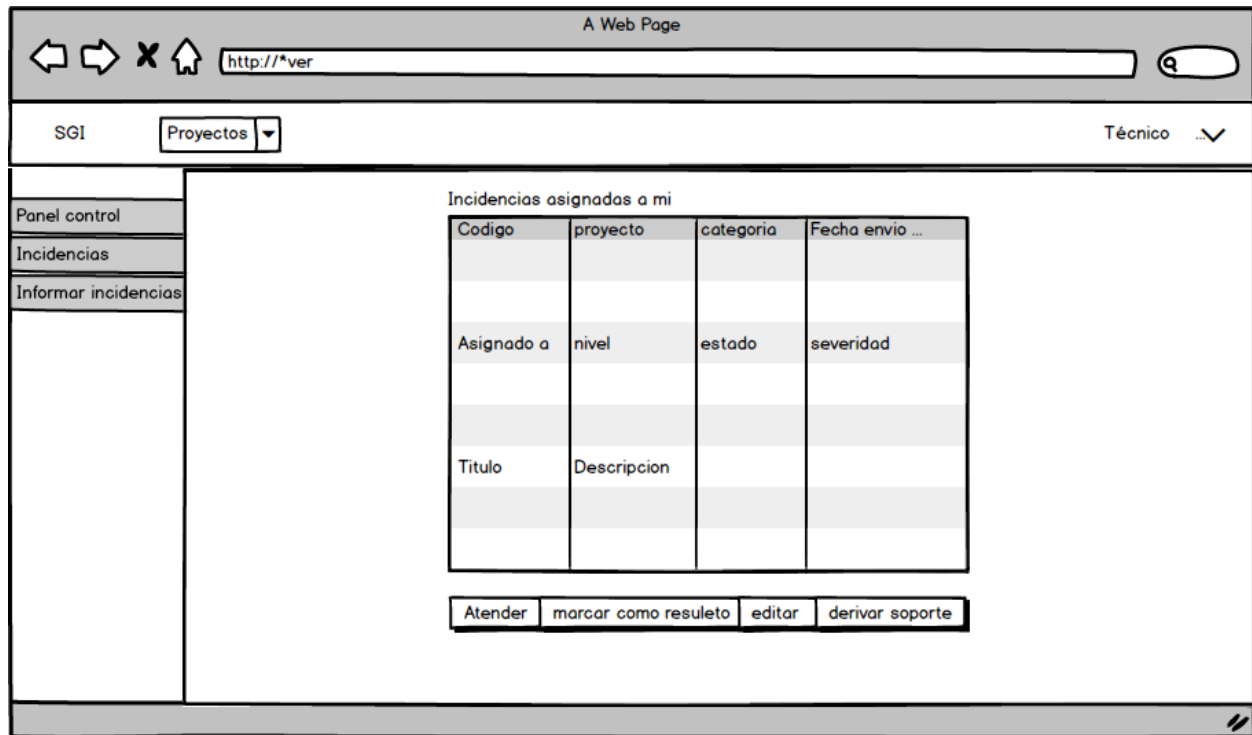
Nombre Opciones

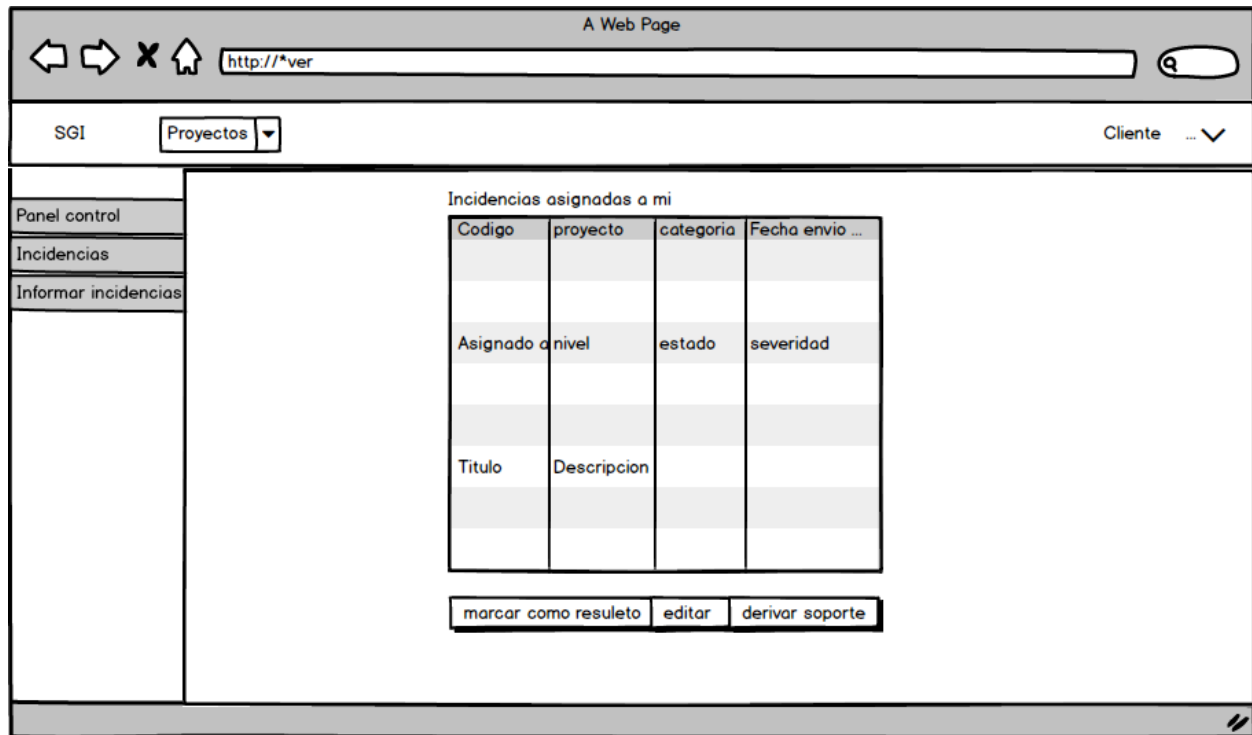
Soporte Añadir

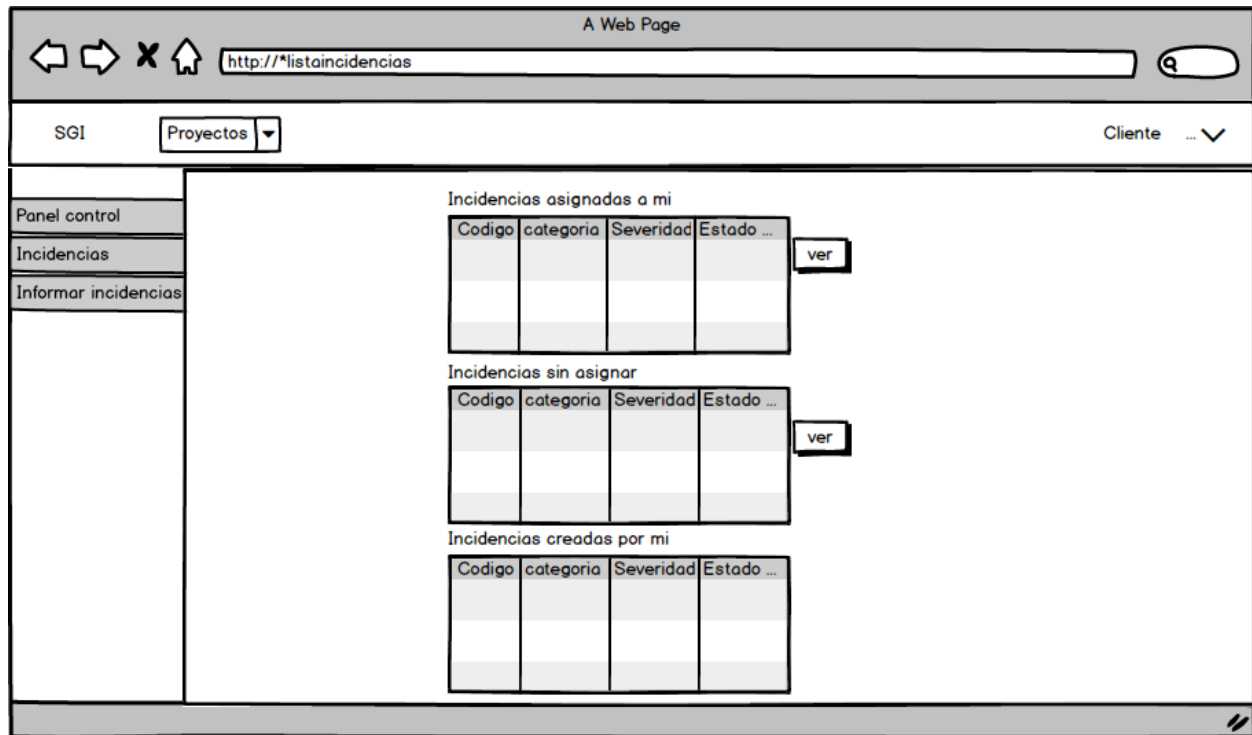
Nombre Opciones











5. Diseño

En este apartado se describirán los pasos dados, para diseñar esta aplicación. En primer lugar, basándonos en la ingeniería del software, hemos recabado los requisitos funcionales y no funcionales que tendrá nuestro sistema.

Los requerimientos funcionales de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones.

Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el sistema y otros requerimientos de negocio, cumplimiento, seguridad u otra índole.

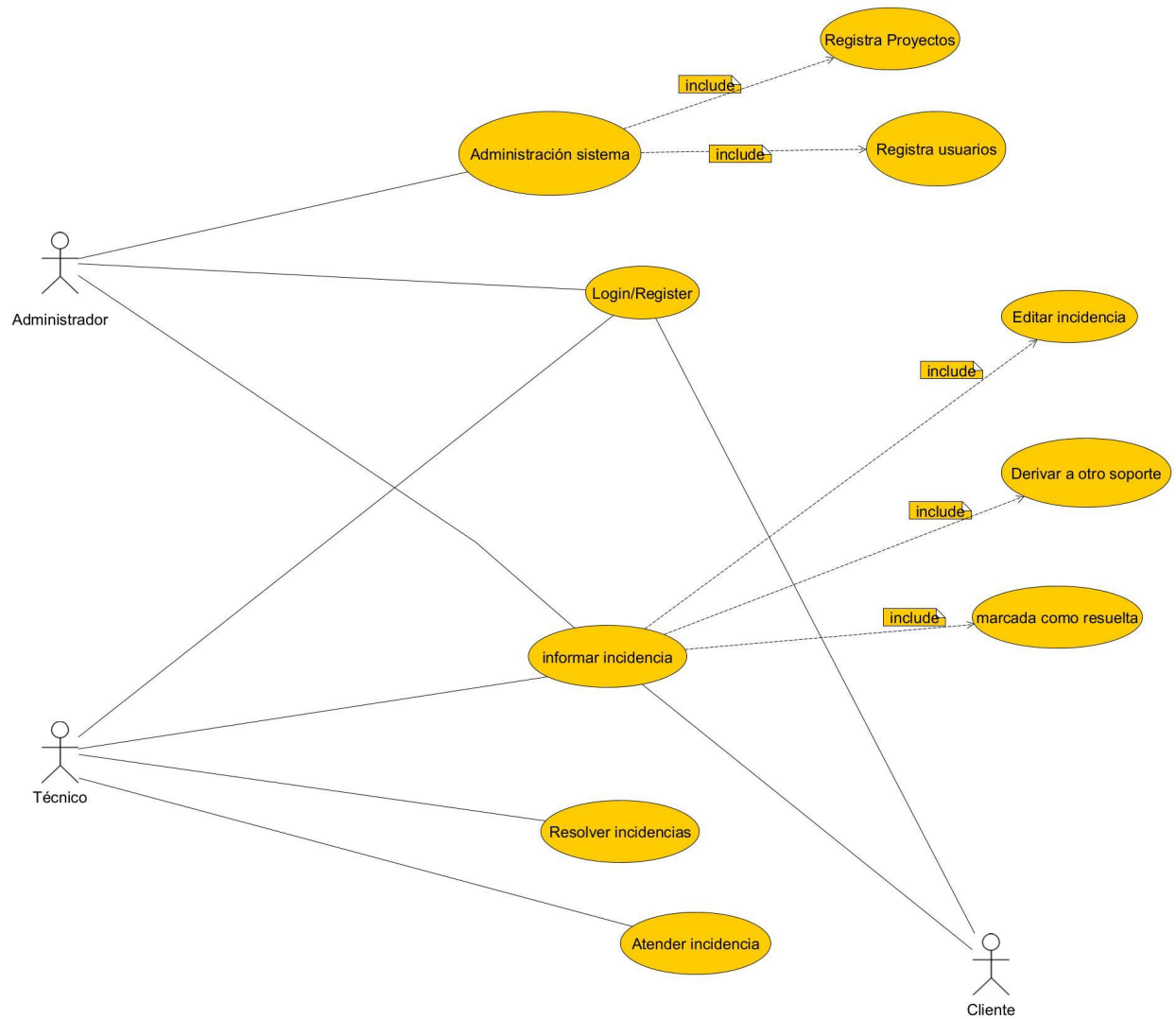
Los requisitos no funcionales se originan en la necesidad del usuario, debido a restricciones presupuestarias, políticas organizacionales, la necesidad de interoperabilidad con otros sistemas de software o hardware, o factores externos tales como regulaciones de seguridad, políticas de privacidad, entre otros.

A continuación mostramos los requisitos obtenidos y el diagrama de casos de uso.

Requisitos Funcionales	Requisitos no Funcionales
La aplicación guardará los datos de los usuarios: administrador, técnico soporte y cliente. Como su nombre, email, imagen...	Los datos de la aplicación serán modificados por las personas autorizadas como por ejemplo el administrador
La aplicación creará, guardará y editará los distintos proyectos las categorías a la que pertenecen	Los datos registrados serán guardados en el gestor de base de datos, donde en un futuro se podrán realizar consultas.
La aplicación creará, guardará y editará los datos de los usuarios(técnicos)	La aplicación resuelve las incidencias.
La aplicación asignará a cada usuario(técnicos) con un proyecto	La aplicación deriva las incidencias a otro técnico de soporte, si el primero no puede resolverlo.
La aplicación creará, guardará, editará a través de los formularios de incidencias la información indicada en ellos.	

La aplicación mostrará una lista con las incidencias	
------------------------------------------------------	--

Diagrama uso de casos



6. Desarrollo

Secuencia de desarrollo aplicación web:

1. Tormenta de ideas, para la elección de nuestro proyecto.
2. Una vez que ya tenemos la idea, el siguiente paso será obtener los requisitos, que necesidades puede satisfacer nuestro sistema y qué problemas puede solucionar.
3. Definición de las funcionalidades de nuestro sistema.
4. El siguiente paso, es elegir que tipo de tecnología que vamos a utilizar:
 - Codificación del lado del cliente:
 - ❖ HTML
 - ❖ Css
 - ❖ Javascript
 - ❖ Ajax
 - ❖ JQuery
 - ❖ Librería bootstrap css
 - Codificación del lado del servidor:

❖ PHP

- Framework Cliente:

❖ Vue.js

- Framework Servidor

❖ Laravel

En algunos momentos nos hemos encontrado con dificultades tanto en la programación del backend como en el front-end.

Detallamos alguno de los inconvenientes en la parte del backend:

- ☐ Con el framework laravel, aunque ya tenía algunas nociones de cómo funciona, me he topado con algunas dificultades como el funcionamiento de los controladores, estructura y organización del framework.
- ☐ Con el lenguaje de php, que en ocasiones he tenido que buscar información sobre alguna sintaxis.

En el front-end también ha existido sus dificultades:

- ☐ Con vue.js, ya que no lo había utilizado antes, he tenido que documentarme un poco para familiarizarme con la sintaxis, y su uso.

Aunque la mayor dificultad a la que me he enfrentado es la combinación del servidor y el cliente. Cómo pintar los datos obtenidos del backend para que vue.js la represente.

Para el control de versiones y revisión de código, he utilizado github, donde he ido subiendo a mi repositorio las distintas versiones del proyecto, así me aseguraba todo esta a buen recaudo, o por si en algún momento realizaba modificaciones en el proyecto y luego estas no funcionaban volvía a una versión anterior.

7. Pruebas

En este apartado describimos las pruebas realizadas al código para comprobar si todo funcionaba satisfactoriamente, además así verificamos partes del código y saber si se están obteniendo datos, o guardando en la base de datos correctamente.

Laravel proporciona una función para depurar se llama Dump and Die(**dd()**). Es una función auxiliar para volcar el contenido de una variable al navegador. También detiene la ejecución del script.

Veamos un ejemplo práctico de la salida que produce

Para imprimir los datos del usuario usando dd():

```
$users = User::all(); dd($users);
```

En la siguiente imagen se puede ver el resultado

```
Illuminate\Database\Eloquent\Collection {#1028 ▼  
  #items: array:3 [▼  
    0 => App\Models\User {#1027 ▶}  
    1 => App\Models\User {#621 ▶}  
    2 => App\Models\User {#989 ▶}  
  ]  
}
```

A lo largo del código fui colocando esta función para comprobar si se obtenían datos, o por el contrario las funciones no retornaban el resultado esperado.

También he utilizado la función incorporada de php `var_dump()` función que muestra información estructurada sobre variables / expresiones, incluido su tipo de datos y el valor de la variable. La matriz y el objeto se exploran de forma recursiva con valores para mostrar su estructura.

Por ejemplo:

```
$users = User::find(1);  
  
var_dump($users);
```

Nos muestra el objeto con todos los datos.

Estas funciones me han ayudado a poder resolver dificultades encontradas, saber si las funciones y variables tomaban bien los datos y solucionar código erróneo.

8. Despliegue

La tecnología utilizada para desplegar el proyecto es docker.

Docker es una plataforma de software que permite crear, probar e implementar aplicaciones rápidamente.

Para ello definimos nuestro archivo, docker-compose.yml para laravel, donde tenemos definidos la versión, todos nuestros servicios y la base de datos. Se muestra un ejemplo del archivo en la siguiente imagen:

The image shows a code editor with a file named 'docker-compose.yml'. The code is as follows:

```
1 version: '3.7'
2
3 services:
4   servidor:
5     image: fjortegan/dwes:laravel
6     stdin_open: true # docker run -i
7     tty: true        # docker run -t
8     ports:
9       - "80:80"
10    volumes:
11      - ./laravel:/var/www/html/
12
13 db:
14   image: mariadb
15
16   ports:
17     - "3306:3306"
18
19   volumes:
```

Iniciaremos nuestro archivo, ejecutando el comando: **docker-compose up**, para iniciar la pila de la aplicación. Mostramos un ejemplo en la imagen:

```
PS C:\Users\34667\Desktop\Nueva carpeta (4)\ProyectoFinalDaw_GestionIncidencias> docker-compose up
[+] Running 3/3
- Container proyectofinaldaw_gestionincidencias_phpmyadmin_1 Recreated 1.9s
- Container proyectofinaldaw_gestionincidencias_servidor_1 Re... 1.9s
- Container proyectofinaldaw_gestionincidencias_db_1 Recreate... 1.9s
Attaching to proyectofinaldaw_gestionincidencias-db-1, proyectofinaldaw_gestionincidencias-phpmyadmin-1, proyectofinaldaw_gestionincidencias-servidor-1
proyectofinaldaw_gestionincidencias-db-1 | 2022-02-18 16:22:36+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.6.5+maria~focal started.
[ ok ctofinaldaw_gestionincidencias-servidor-1 | [....] Starting nginx: nginx.
```

Podemos comprobar si está en ejecución con el comando: ***docker ps***.

Si tenemos que realizar alguna operación sobre el sistema ejecutaremos el comando: ***docker exec -it (nombre:proyecto_servidor_1) /bin/bash***

```
PS C:\Users\34667\Documents\GitHub\proyectoDaw\ProyectoFinalDaw_GestionIncidencias> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
2dda92830570   mariadb       "docker-entrypoint.s..." 6 minutes ago  Up 6 minutes  0.0.0.0:3306->3306/tcp              proyectofina
ldaw_gestionincidencias-db-1
72e99eeca6f6   fjortegan/dwes:laravel  "/docker-entrypoint...." 6 minutes ago  Up 6 minutes  0.0.0.0:80->80/tcp                  proyectofina
ldaw_gestionincidencias-servidor-1
4df5a5aefdd1   phpmyadmin    "/docker-entrypoint...." 6 minutes ago  Up 6 minutes  0.0.0.0:8080->80/tcp                proyectofina
ldaw_gestionincidencias-phpmyadmin-1
PS C:\Users\34667\Documents\GitHub\proyectoDaw\ProyectoFinalDaw_GestionIncidencias> docker exec -it proyectofinaldaw_gestionincidencias-servidor-1 /bin/bash
root@72e99eeca6f6:/#
```

Con estos comandos tendremos en ejecución nuestro proyecto.

9. Manual

1. *Introducción*

La aplicación sistema de gestión de incidencias está pensada y diseñada para crear, gestionar y hacer seguimiento de las incidencias y peticiones de los clientes.

Existen 3 roles: Administrador, soporte técnico y cliente.

2. *Entrada al sistema*

Para acceder a la aplicación, el usuario debe hacer uso de sus credenciales de acceso (usuario y contraseña).

- **IMPORTANTE:**

- En caso el usuario , no figuren como datos de alta en nuestra base de datos, podrán solicitar dichas altas desde la página principal (portal de entrada), de nuestra aplicación.

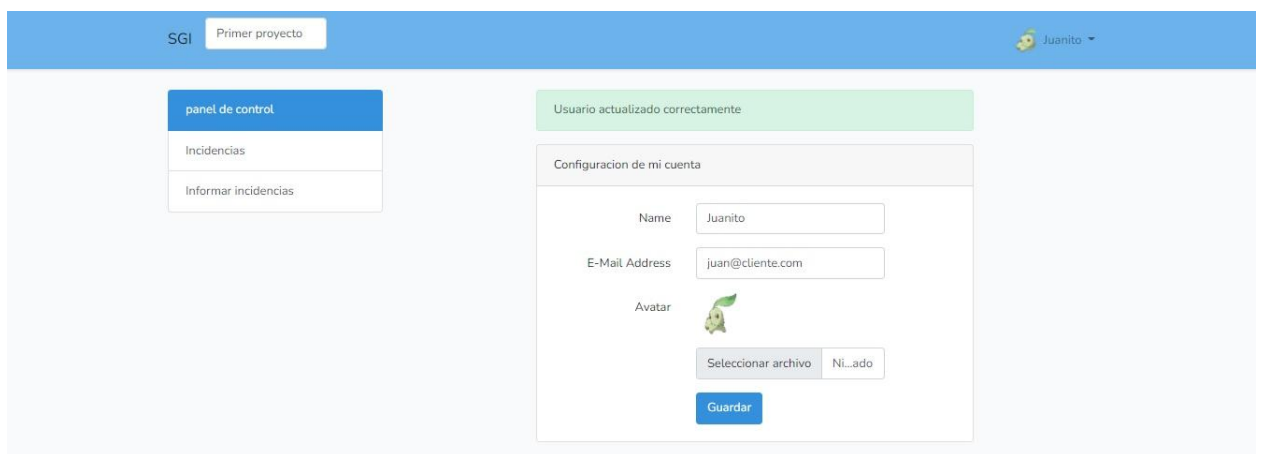
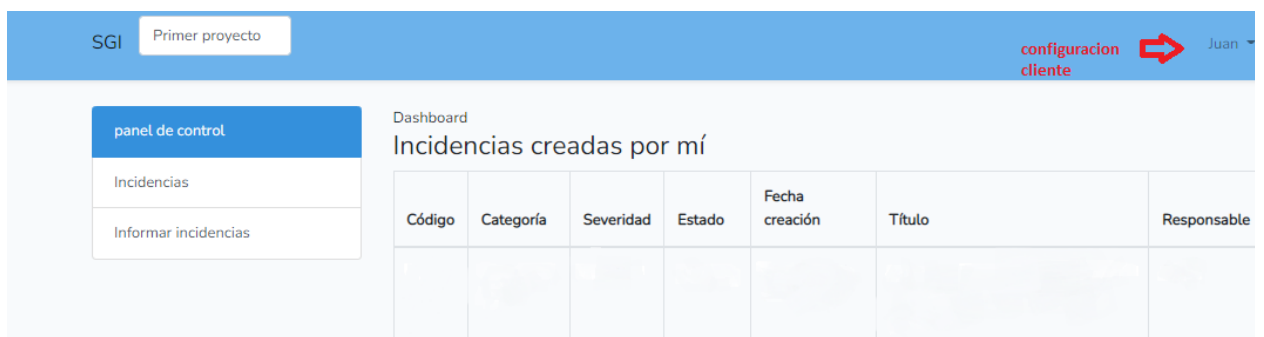
2.1 Portal de entrada

Para acceder a la aplicación:

puede acceder con sus credenciales y/o acceder a los formularios de “login o register”.

❖ *Role usuario*

- si el usuario estaba registrado en el sistema accede a su panel




Puede configurar sus datos y subir su avatar.

También encontrará un formulario de informar incidencias, como se aprecia en la imagen.

The screenshot shows a web application interface with a blue header bar. On the left, there is a 'panel de control' (control panel) with two buttons: 'Incidencias' and 'Informar incidencias'. The 'Informar incidencias' button is highlighted. In the top right of the header, there is a dropdown menu labeled 'Primer proyecto' with a red arrow pointing to it and a text annotation: 'selector de proyectos, para poder crear incidencia en cada proyecto'. The main content area is titled 'Registrar incidencia' and contains four input fields: 'Categoria' (with 'General' selected), 'Severidad' (with 'Menor' selected), 'Titulo', and 'Descripcion'. A blue button labeled 'Registrar incidencias' is at the bottom of the form.

en el cuál podrá informar de las incidencias. En la parte superior del panel tiene un selector donde puede escoger entre los distintos proyectos, para informar donde se ha encontrado una incidencia.

- si el usuario no está dado de alta, podrá registrarse en el formulario de registro:



Name

E-Mail Address

Password

Confirm Password

[Register](#)

2.2 Funcionamiento del sistema

El cliente una vez creada la incidencia tendrá en su panel la siguiente vista:

SGI Primer proyecto configuración cliente Juan

panel de control

- Incidentes
- Informar incidencias

Dashboard

Incidentes creadas por mí

Código	Categoría	Severidad	Estado	Fecha creación	Título	Responsable
1	Categoría A2	Normal	Resuelto	2022-02-21 20:39:36	Lo que ocurre es que se encontró un problema en la página y esta se cerró.	Rocio
7	General	Menor	Resuelto	2022-02-21 21:31:58	t	Rocio

donde aparecerán las incidencias creadas por él.

- si clicas sobre el código de su incidencia aparecerá la siguiente ventana:

SGI Primer proyecto Juanito

panel de control

- Incidentes
- Informar incidencias

Dashboard

Código	Proyecto	Categoría	Fecha de envío
1	Primer proyecto	Categoría A2	2022-02-21 20:39:36

Asignada a	Nivel	Estado	Severidad
Rocio	Envío de técnico	Asignado	Normal

Título	Primera incidencia
Descripción	Lo que ocurre es que se encontró un problema en la página y esta se cerró.
Adjuntos	No se han adjuntado archivos

[Marcar como resuelto](#)
[Editar incidencia](#)
[Derivar al siguiente soporte](#)

en la cual tenemos, tres botones, marcar incidencia, editar y derivar el siguiente soporte.

- Si la incidencia es resuelta por un técnico de soporte y satisfactoria para el cliente, éste la marcará como resuelta y la incidencia se dará por concluida.
- El usuario podrá editar la incidencia, por si quiere incluir alguna nueva información.

The screenshot shows a web application interface with a blue header bar. On the left, there's a sidebar with a 'panel de control' (control panel) containing 'Incidencias' (Incidents) and 'Informar incidencias' (Report incidents). The main area is titled 'SGI Primer proyecto' and features a user profile 'Juanito' with a dropdown arrow. The form includes fields for 'Categoría' (Category) set to 'Categoría A2', 'Severidad' (Severity) set to 'Normal', 'Título' (Title) set to 'Primera incidencia', and 'Descripción' (Description) with the text 'Lo que ocurre es que se encontró un problema en la página y esta se cerró.' A 'Guardar cambios' (Save changes) button is at the bottom.

panel de control	Categoría
Incidencias	Categoría A2
Informar incidencias	

Severidad
Normal

Título
Primera incidencia

Descripción
Lo que ocurre es que se encontró un problema en la página y esta se cerró.

Guardar cambios

- Si la incidencia no es resuelta satisfactoriamente podrá derivar a otro técnico de soporte.

❖ *Role Administrador*

- El usuario administrador tendrá el siguiente panel,

The screenshot displays a web application interface for an administrator. The browser window shows the URL 'localhost' and several open tabs. The application has a blue header with the text 'SGI' and 'Primer proyecto'. Below the header, there is a sidebar on the left with a 'panel de control' section containing 'Incidencias' and 'Informar incidencias', and an 'Administracion' section containing 'usuarios' and 'proyectos'. The main content area is titled 'Dashboard' and 'Incidencias creadas por mí'. It contains a table with the following data:

Código	Categoría	Severidad	Estado	Fecha creación	Título	Responsable
8	General	Alta	Resuelto	2022-02-22 15:28:18	prueba	Rocio

Below the table, there is a blue section with the text 'LINKS' and four columns of links labeled 'Link 1' through 'Link 4'. The footer of the application shows a search bar and a date '24/02/2022'.

- En la parte central se encuentra una tabla donde se muestran las incidencias creadas por él.
- Al pinchar sobre alguna de las incidencias se mostrará la siguiente pantalla:

The screenshot shows a web application interface. At the top, there's a blue header with 'SGI' and 'Segundo proyecto' on the left, and a user profile 'tere' on the right. Below the header, on the left, is a sidebar with a 'panel de control' button and a list of options: 'Incidencias', 'Informar incidencias', 'Administración', 'usuarios', and 'proyectos'. The main content area is titled 'Dashboard' and contains a table with incident data. The table has columns for 'Código', 'Proyecto', 'Categoría', and 'Fecha de envío'. The first row shows '4', 'Segundo proyecto', 'Categoría A1', and '2022-02-21 20:52:30'. Below this, there's another table with columns 'Asignada a', 'Nivel', 'Estado', and 'Severidad'. The second row shows 'Sin asignar', 'Atención por teléfono', 'Pendiente', and 'Menor'. At the bottom of the dashboard, there are three buttons: 'Marcar como resuelto' (blue), 'Editar incidencia' (green), and 'Derivar al siguiente soporte' (red).

Código	Proyecto	Categoría	Fecha de envío
4	Segundo proyecto	Categoría A1	2022-02-21 20:52:30

Asignada a	Nivel	Estado	Severidad
Sin asignar	Atención por teléfono	Pendiente	Menor

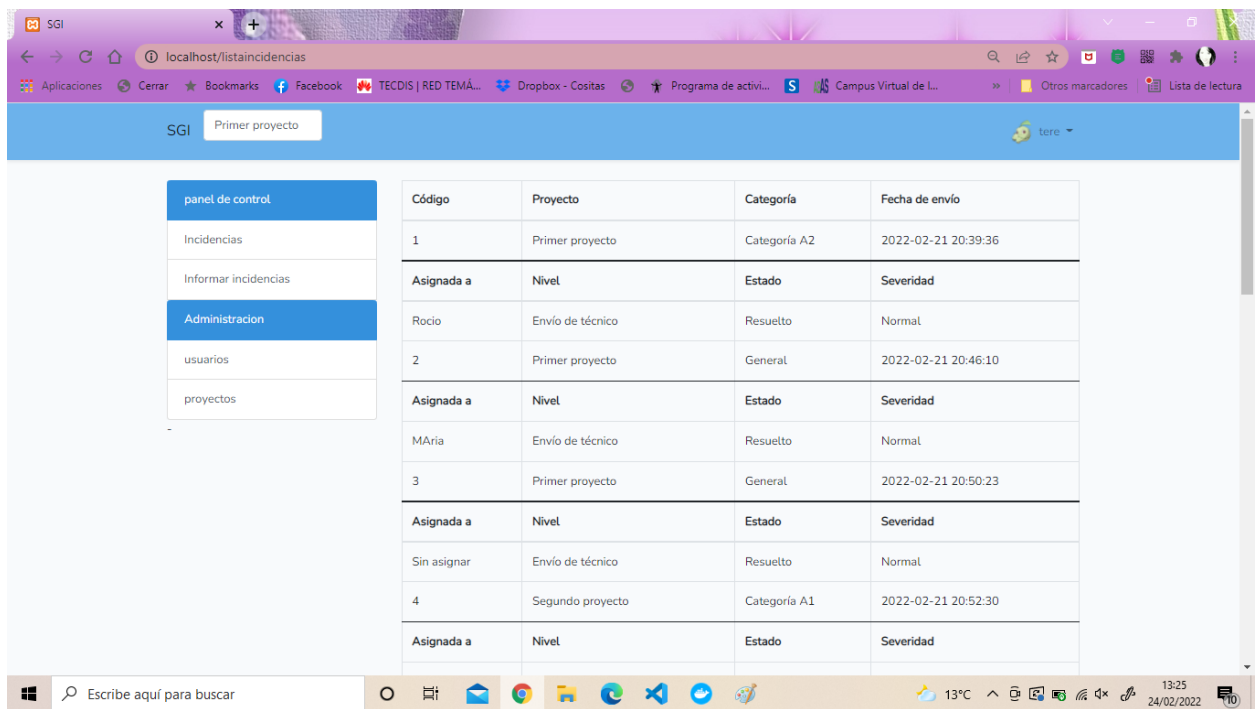
Título	b
Descripción	b
Adjuntos	No se han adjuntado archivos

[Marcar como resuelto](#)
[Editar incidencia](#)
[Derivar al siguiente soporte](#)

en la cual tenemos, tres botones, marcar incidencia, editar y derivar el siguiente soporte.

- Si la incidencia es resuelta satisfactoriamente por uno de los técnicos, la marcará como resuelta. Podrá editar la incidencia y si la incidencia no puede ser resuelta la derivará al siguiente nivel.

- En la parte izquierda de la pantalla , se encuentra un menú donde tenemos: incidencias(muestra listado de todas las incidencias del sistema)



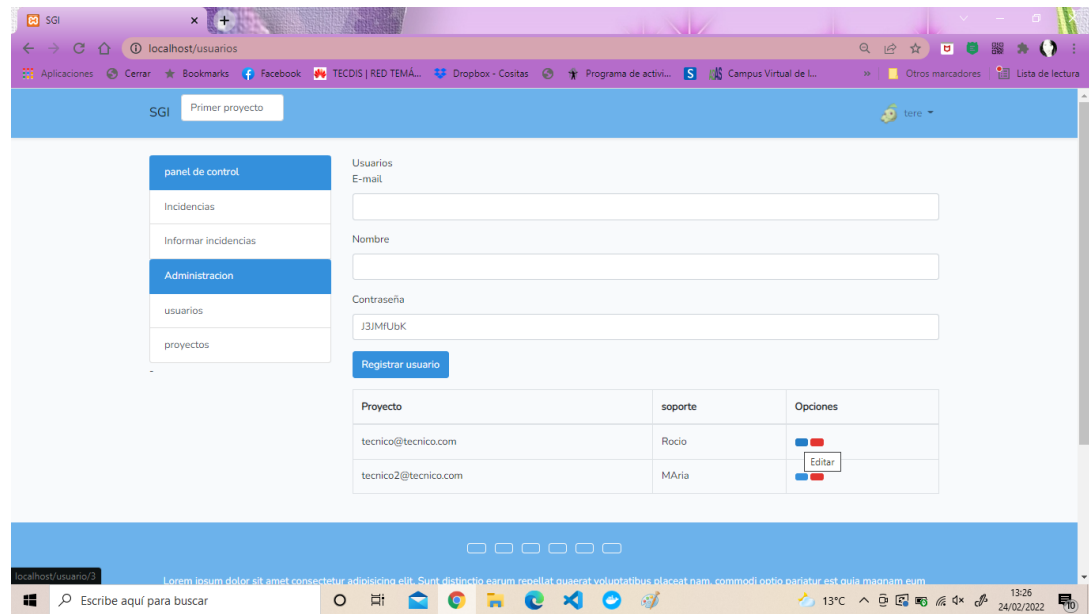
Código	Proyecto	Categoría	Fecha de envío
1	Primer proyecto	Categoría A2	2022-02-21 20:39:36
Asignada a	Nivel	Estado	Severidad
Rocio	Envío de técnico	Resuelto	Normal
2	Primer proyecto	General	2022-02-21 20:46:10
Asignada a	Nivel	Estado	Severidad
MÁria	Envío de técnico	Resuelto	Normal
3	Primer proyecto	General	2022-02-21 20:50:23
Asignada a	Nivel	Estado	Severidad
Sin asignar	Envío de técnico	Resuelto	Normal
4	Segundo proyecto	Categoría A1	2022-02-21 20:52:30
Asignada a	Nivel	Estado	Severidad

a continuación, informar incidencia, donde se encuentra el formulario para informar de algún problema.

The screenshot shows a web browser window with the address bar displaying 'localhost/regar'. The page has a blue header with the text 'SGI' and a search bar containing 'Primer proyecto'. On the left, there is a sidebar menu with the following items: 'panel de control', 'Incidencias', 'Informar incidencias', 'Administracion' (highlighted), 'usuarios', and 'proyectos'. The main content area is titled 'Registrar incidencia' and contains the following form fields: 'Categoria' (with a dropdown menu showing 'General'), 'Severidad' (with a dropdown menu showing 'Menor'), 'Titulo' (a text input field), and 'Descripcion' (a larger text input field). Below these fields is a blue button labeled 'Registrar incidencias'. At the bottom of the page, there is a blue footer with a placeholder text: 'Lorem ipsum dolor sit amet consectetur adipiscing elit. Suspendisse quisque egestas. Suspendisse quisque egestas. Suspendisse quisque egestas. Suspendisse quisque egestas. Suspendisse quisque egestas.' The Windows taskbar at the bottom shows the search bar with the text 'Escribe aquí para buscar', several application icons, and the system clock displaying '13:26' and '24/02/2022'.

- El administrador es el encargado de dar de alta a los técnicos de soporte y de asignarles a los proyectos.

En la siguiente imagen se puede ver el formulario para dar de alta a los técnicos.



y en la parte inferior se ven los usuarios asignados a sus proyectos.

- El administrador podrá editar los datos de los usuarios

SGI Primer proyecto

panel de control

- Incidentes
- Informar incidencias
- Administracion
- usuarios
- proyectos

Editar usuario

E-mail
tecnico@tecnico.com

Nombre
Rocio

Contraseña *Ingresar solo si se desea modificar*

Guardar usuario

Seleccione proyecto Seleccione soporte Asignar proyecto

Proyectos asignados

Proyecto	Soporte	Opciones
Primer proyecto	Atención por teléfono	<input type="checkbox"/>
Segundo proyecto	Consulta especializada	<input type="checkbox"/> Dar de baja
proyecto3	soporte telefonico	<input type="checkbox"/>

- En la pantalla de proyectos, el administrador podrá crear nuevos proyectos y editarlos.

SGI Primer proyecto

Proyectos

Nombre

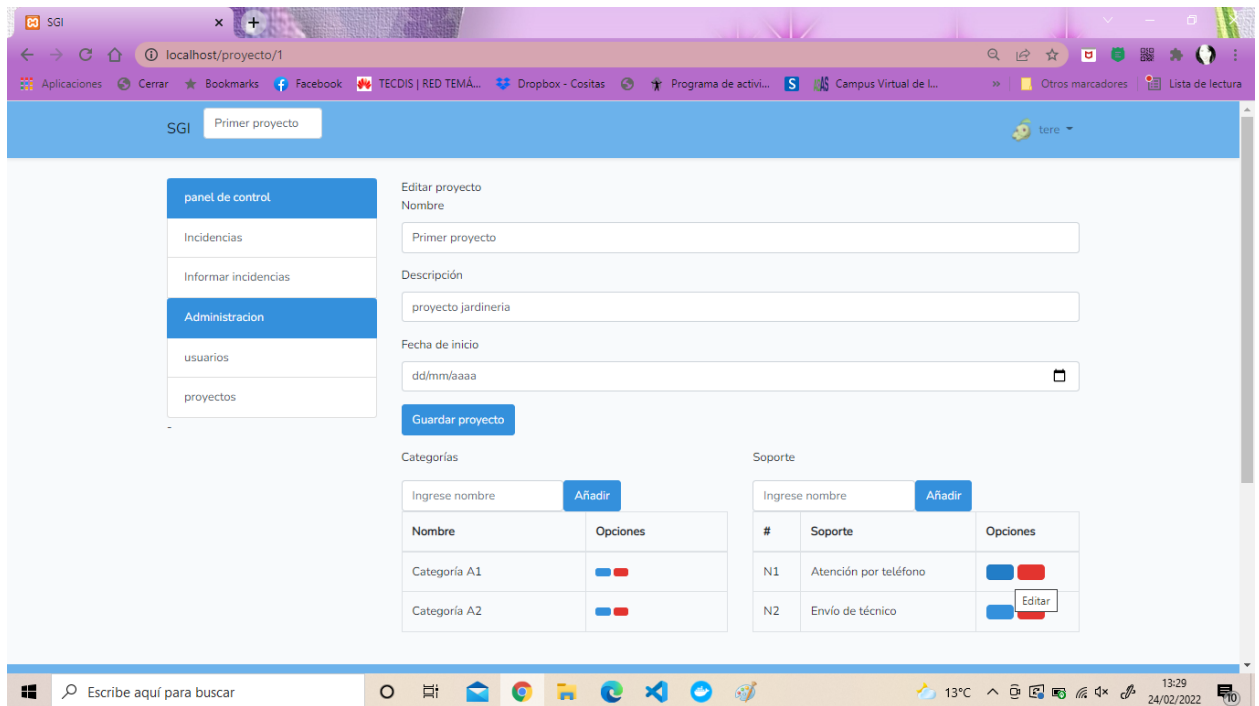
Descripcion

Fecha inicio

dd/mm/aaaa

Registrar proyecto

Nombre Proyecto	Descripcion	Fecha inicio	Opciones
Primer proyecto	proyecto jardineria	no hay fecha	■ ■
Segundo proyecto	proyecto ayuntamiento	no hay fecha	■ ■
Tercer proyecto	proyecto web	no hay fecha	■
proyecto3	desarrollo web	2022-02-22	■ ■
proyecto3	desarrollo web	2022-02-22	■



❖ Rol Técnico de soporte

- El usuario logeado como técnico, tendrá en siguiente panel:

donde se mostrará, las incidencias creadas por él, las que tiene asignadas y las que aún no están asignadas, como puede observarse en la siguiente imagen:

SGI Primer proyecto Rocio

panel de control

Incidentes

Informar incidencias

Dashboard

Incidentes asignados a mí

Código	Categoría	Severidad	Estado	Fecha creación	Título	
1	Categoría A2	Normal	Asignado	2022-02-21 20:39:36	Lo que ocurre es que se encontró un problema en la página y esta se cerró.	Atender
7	General	Menor	Resuelto	2022-02-21 21:31:58	t	Atender
8	General	Alta	Resuelto	2022-02-22 15:28:18	prueba	Atender
11	General	Normal	Resuelto	2022-02-22 21:22:24	tt	Atender

Incidentes sin asignar

Código	Categoría	Severidad	Estado	Fecha creación	Título	Opción
12	General	Alta	Pendiente	2022-02-24 20:27:09	prueba	Atender

Incidentes creadas por mí

Código	Categoría	Severidad	Estado	Fecha creación	Título	Responsable
3	General	Normal	Pendiente	2022-02-21 20:50:23	a	Sin asignar

- El técnico podrá atender incidencias creadas por el mismo o atender las que aún no tienen a nadie asignado.

SGI Primer proyecto Rocio

panel de control

Incidentes

Informar incidencias

Dashboard

Código	Proyecto	Categoría	Fecha de envío
12	Primer proyecto	General	2022-02-24 20:27:09

Asignada a	Nivel	Estado	Severidad
Sin asignar	Atención por teléfono	Pendiente	Alta

Título	prueba
Descripción	prueba
Adjuntos	No se han adjuntado archivos

[Atender incidencia](#)
[Marcar como resuelto](#)
[Editar incidencia](#)
[Derivar al siguiente soporte](#)

- Así pues, podrá atender la incidencia y su estado pasará a asignado.

la información entonces se mostrará en la tabla asignado a mi.

Finalmente, marcará la opción de marcada como resulta.

También tiene la opción de editar la incidencia y de derivarla a otro soporte si no puede resolver dicha incidencia.

- Al igual que el resto de usuarios tiene la opción de crear incidencias nuevas.

10. Conclusiones

Después de haber concluido el proyecto, la idea original y el resultado tienen similitud, aunque siempre se puede mejorar. El sistema cumple con los requisitos y realiza las funciones para las que fue programado.

Aunque admite mejoras, como las que a continuación se enumeran:

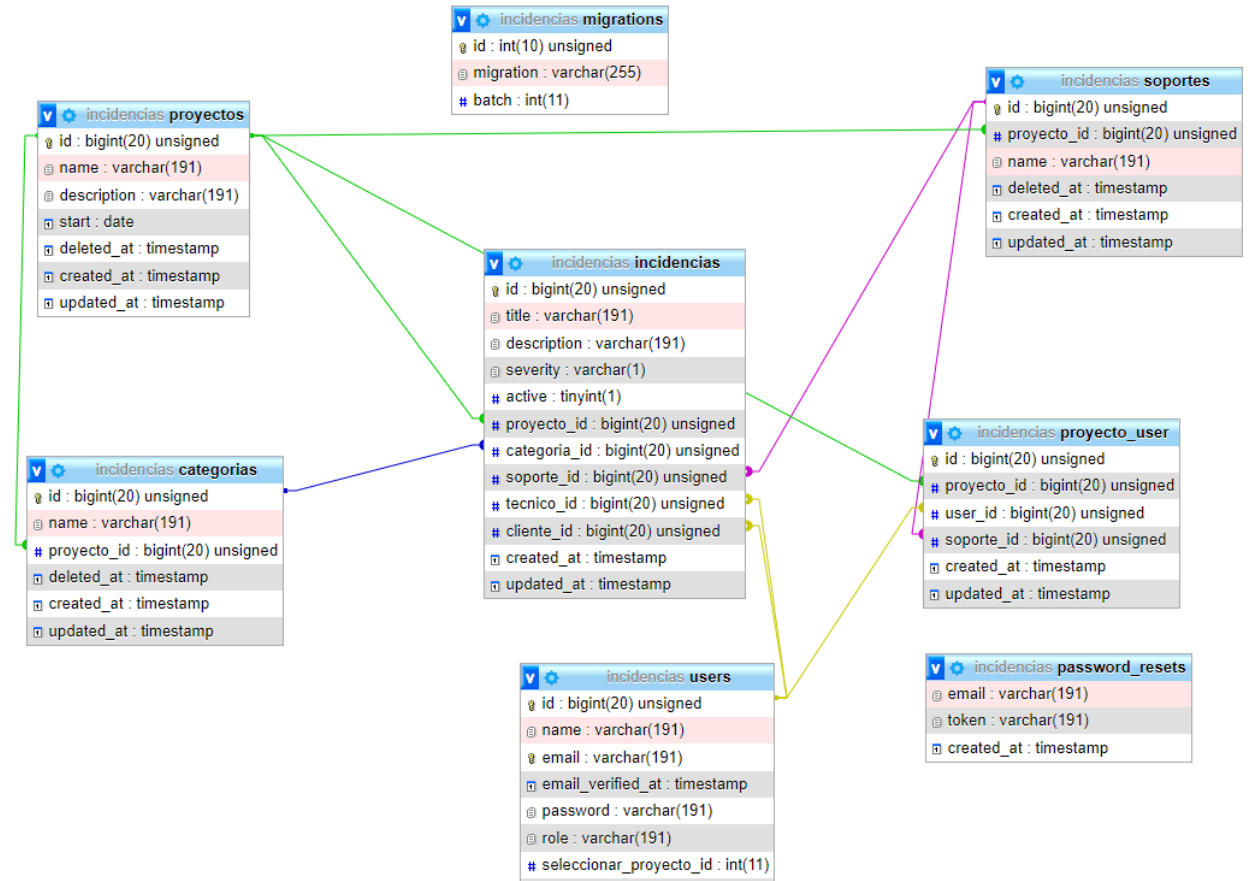
- Interfaz del usuario, la estética de la aplicación puede enriquecerse como por ejemplo: añadiendo imágenes dinámicas, gráficos...
- Otro aspecto mejorable de la aplicación es la accesibilidad web, para que puedan acceder a ellas un número mayor de usuarios. Se podría implantar el uso de letras de gran tamaño, diseños adaptativos, textos predictivos, asistentes a la navegación, etc
- Con respecto a la interacción de clientes y personal de soporte, enriquecería a la aplicación un sistema de comunicación más dinámico, como un chat, donde las pequeñas consultas o dudas se podrían resolver rápidamente.
- Para que al equipo de soporte le resulte más fácil reconocer alguna incidencia, implementar subida de archivos e imágenes para que el

técnico de soporte disponga de esa información cuando el cliente la suba.

- Para concluir, añadir la funcionalidad de emisión de informes en formato pdf o excel, por parte del equipo de soporte, para que así el cliente pueda descargarlo y ver entre otras aportaciones soluciones a su problema, por si vuelve a surgir un problema similar pueda ser capaz de resolverlo por sí mismo.

11. Índice de tablas e imágenes

☐ Base de datos de la aplicación



incidencias users
id : bigint(20) unsigned
name : varchar(191)
email : varchar(191)
email_verified_at : timestamp
password : varchar(191)
role : varchar(191)
seleccionar_proyecto_id : int(11)
image : varchar(191)
deleted_at : timestamp
remember_token : varchar(100)
created_at : timestamp
updated_at : timestamp

□ Imágenes de fragmentos de códigos de la aplicación

● Fragmento controlador Homecontrolador

```

54
55 // Subir la imagen
56 $image_path = $request->file('image_path');
57 if($image_path){
58     // Asignar nombre unico con el timestamp actual como prefijo
59     $image_path_name = time() . $image_path->getClientOriginalName();
60     // Guardar en la carpeta (storage/app/user)
61     Storage::disk('users')->put($image_path_name, File::get($image_path));
62     // Seteo el nombre de la imagen en el objeto
63     $user->image = $image_path_name;
64 }
65
66 $user->update();
67
68 return redirect()->route('config')
69     ->with(['message' => 'Usuario actualizado correctamente']);
70 }
71
72 public function getImage($filename){
73     $file = Storage::disk('users')->get($filename);
74     return response($file, 200);
75 }
76
77 public function index()
  
```

● Fragmento controlador incidencias

```

14 $this->middleware('auth');
15 }
16 public function listincident(){
17     $incidencias = App\Incidencia::all();
18     return view('incidencias.list')->with(compact('incidencias'));
19 }
20 public function show($id){
21     $incidencia = App\Incidencia::findOrFail($id);
22     $mensajes = $incidencia->messages;
23     return view('incidencias.show')->with(compact('incidencia', 'mensajes'));
24 }
25
26 public function create(){
27     //hacemos consulta para obtener categoria del proyecto que queremos
28     $categorias = App\Categoria::where('proyecto_id', auth()->user()->seleccionar_proyecto_id)->get();
29     return view('incidencias.create')->with(compact('categorias'));
30 }
31 public function store(Request $request){
32     $this->validate($request, [
33         'categoria_id' => 'sometimes|exists:categorias,id',
34         'severity' => 'required|in:M,N,A',
35         'title' => 'required|string|max:255',
36         'description' => 'required|string|max:255'
37     ]);
  
```

● Fragmento componente Vue.js formulario proyecto

```

EXPLOADOR ... ProyectoComponent.vue X edit.blade.php ...\proyectos JS edit.js index.blade.php ...\proyectos JS app.js
EDITORES ABIERTOS
X ProyectoComponent.vue laravel/resourc...
edit.blade.php laravel/resources/views/ad...
JS edit.js laravel/public/js/admin/proyectos
index.blade.php laravel/resources/views/...
JS app.js laravel/resources/js
HomeController.php laravel/app/Http/Cont... 2
ProyectosTableSeeder.php laravel/databas...
CategoriasTableSeeder.php laravel/databas...
SoporteTableSeeder.php laravel/databas...
ProyectosUserTableSeeder.php laravel/...
PROYECTOFINALDAW_GESTIONI...
  ExampleComponent.vue
  FooterComponent.vue
  ListaadmiComponent.vue
  ListaComponent.vue
  ProyectoComponent.vue
JS app.js
bootstrap.js
lang
sass

laravel > resources > js > components > ProyectoComponent.vue > {} "ProyectoComponent.vue" > script > default > methods > agre

1 <template>
2 <div>
3 <form @submit.prevent="agregar">
4 <!-- {{ csrf_field() }} -->
5
6 <div class="form-group">
7 <label for="name">Nombre</label>
8 <input v-model="proyecto.name" type="text" class="form-control">
9 </div>
10 <div class="form-group">
11 <label for="description">Descripcion</label>
12 <input v-model="proyecto.description" type="text" class="form-control" >
13 </div>
14 <div class="form-group">
15 <label for="date">Fecha inicio</label>
16 <input type="date" v-model="proyecto.start" :min="min" :max="max" class="form-control" >
17 <!-- <p>Value: '{{ old('start', date('y-m-sd')) }}'</p> -->
18 </div>
19
20 <div class="form-group">
21 <button class="btn btn-primary" type="submit">Registrar proyecto</button>
22 </div>
23 </form>
24 </div>
25 </template>

```

```

EXPLOADOR ... ProyectoComponent.vue X edit.blade.php ...\proyectos JS edit.js index.blade.php ...\proyectos JS app.js
EDITORES ABIERTOS
X ProyectoComponent.vue laravel/resourc...
edit.blade.php laravel/resources/views/ad...
JS edit.js laravel/public/js/admin/proyectos
index.blade.php laravel/resources/views/...
JS app.js laravel/resources/js
HomeController.php laravel/app/Http/Cont... 2
ProyectosTableSeeder.php laravel/databas...
CategoriasTableSeeder.php laravel/databas...
SoporteTableSeeder.php laravel/databas...
ProyectosUserTableSeeder.php laravel/...
PROYECTOFINALDAW_GESTIONI...
  ExampleComponent.vue
  FooterComponent.vue
  ListaadmiComponent.vue
  ListaComponent.vue
  ProyectoComponent.vue
JS app.js
bootstrap.js
lang
sass
views

laravel > resources > js > components > ProyectoComponent.vue > {} "ProyectoComponent.vue" > script > default > methods > agre

27 </template>
28
29 <script>
30 export default {
31   data() {
32     const now = new Date()
33     const today = new Date(now.getFullYear(), now.getMonth(), now.getDate())
34     // 15th two months prior
35     const minDate = new Date(today)
36     minDate.setMonth(minDate.getMonth() - 2)
37     minDate.setDate(15)
38     // 15th in two months
39     const maxDate = new Date(today)
40     maxDate.setMonth(maxDate.getMonth() + 2)
41     maxDate.setDate(15)
42     return {
43       csrf: document.querySelector('meta[name="csrf-token"]').getAttribute('content'),
44       proyectos: [],
45       proyecto: {name: '', description: '', start: ''},
46       start: '',
47       min: minDate,
48       max: maxDate
49     }
50   },
51 }

```

12. Bibliografía y Referencias

- ☐ Libro Laravel, aprende a crear aplicaciones web desde cero.
 - Autor: José Lopez Quijado
 - Editorial : RC libros
- ☐ Libro Laravel, Curso práctico avanzado.
 - Autor: José Lopez Quijado
 - Editorial : RC libros
- ☐ Libro APRENDER PHP, MYSQL Y JAVASCRIPT
 - Autor: Robin Nixon
 - Editorial: Macondo
- ☐ <https://laravel.com/docs/5.8>
- ☐ <https://vuejs.org/>
- ☐ <https://www.php.net/manual/es/index.php>

