**DOCUMENTACION PROYECTO**

1. **Mapa de navegación**

**1.1 Inicio de sesión y registro**

La aplicación contará con dos opciones de login, un usuario con el rol user y un administrador con el rol admin.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Si el usuario no tiene una cuenta, puede registrarse, una vez registrado, será dirigido al login, todos los campos están validados

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* 1. **Navegación Admin**

Una vez entramos como admin, nos encontraremos la siguiente pantalla donde podremos navegar por donde queramos

Diagrama

Descripción generada automáticamente

En el apartado de gestión de productos, nos encontraremos con todos los productos que ofrece la web, pudiendo filtrar por categorias

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Dentro de la misma página podremos acceder a la gestión de usuarios, donde podremos buscar los usuarios por el nombre

Una captura de pantalla de una red social

Descripción generada automáticamente

Si pulsamos en el botón modificar, nos llevará a una pantalla con los datos del usuario para poder modificarlos

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* 1. **Navegación User**

Al entrar como usuario nos encontraremos con esta pantalla de navegación, que cuenta con un cuadro que reproduce chistes de Chuck Norris cada 5s para que el usuario esté entretenido

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

El botón situado arriba a la izquierda nos activa y desactiva un modo oscuro

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente con confianza media

Al entrar en nuestras plantas, nos saldrá un listado de plantas, la cabecera se compone de una navegación en la que podremos movernos por distintas páginas, podremos ver los productos y ver las plantas, además de eso, podremos seleccionar nuestra comunidad autónoma y nos mostrará la meteorología de dicha zona en una ventana modal, también incluye un botón para cerrar sesión

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

La aplicación cuenta con paginación funcional

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Las tarjetas de las plantas se componen de la imagen de la planta, el nombre común y el nombre científico, al hacer click en una imagen de la planta, nos saldrá toda la información que existe a cerca de dicha planta.

Imagen que contiene Calendario

Descripción generada automáticamente

También se genera un mapa con las zonas geográficas donde mejor se adapta la planta seleccionada.

Mapa

Descripción generada automáticamente

E usuario puede comprar los productos disponibles en la tienda.

Interfaz de usuario gráfica

Descripción generada automáticamente

La web dispone de una galería de imágenes de nuestros productos

Texto

Descripción generada automáticamente

1. **APIs empleadas**

**Perenual API**

Esta API cuenta con una cantidad inmensa de plantas, en concreto nos da la información de más de 10000 plantas de todo el mundo.

He usado esta API principalmente para listar todas las plantas y posteriormente obtener por medio de varias peticiones información de cada planta, como por ejemplo, la imagen, el nombre común, el nombre científico, la frecuencia de riego, su nivel de toxicidad, las horas de sol, la descripción, etc.

Esta API también tiene una función muy peculiar y es que te genera un Canvas con las zonas climáticas donde mejor se adapta la planta, lo cual es muy útil e interesante para los usuarios.

La API también cuenta con ciertos errores, ya que al ser una gran amplitud de información, es muy complicado que se gestione toda correctamente, pero hace su función principal. Aún así está en pleno desarrollo.

Para acceder a esta API, debes de registrarte en la web de prenual.com, donde se te facilitará una API-KEY gratuita con 100 peticiones diarias, si deseas obtener más peticiones debes de contratar los planes que tienen en dicha web.

**El-tiempo.net**

Esta API es un recurso gratuito ya que pertenece a la agencia de meteorología española, con lo cual no se precisa de ningún registro previo para su uso.

Dicha API es empleada en mi aplicación para poder obtener las características meteorológicas de la provincia que se desee, la cual le proporcionará la temperatura máxima, mínima, un icono en función del estado atmosférico y una breve descripción de dicho estado.

Aunque yo no haya usado todos sus recursos, es una API muy completa, con la que puedes obtener una gran variedad de información muy útil y en tiempo real, como por ejemplo, las provincias, las comunidades, las temperaturas máximas y mínimas, descripción detallada del tiempo atmosférico, etc.

Considero que es una API muy útil para proyectos que necesiten datos atmosféricos en tiempo real ya que es muy actualizada y sus errores son muy escasos.

1. **Funciones**

**Login**

Para comprobar los errores, realizamos un fetch a nuestro archivo PHP, para posteriormente gestionar las distintas respuestas. En función de esas respuestas, el formulario comprobará si el correo existe, si la contraseña es correcta y valorará el rol, dependiendo del Rol nos dirigirá a una vista u a otra. También comprobaremos si la API recibe datos o no.

Texto

Descripción generada automáticamente

**Registro**

En cuanto al registro, es un formulario el cual envía los datos a la API para registrar el usuario, dependiendo de la respuesta de la API, registrará o no al usuario, posteriormente a la petición hay una serie de validaciones que se deben cumplir para que el usuario se pueda registrar.

Texto

Descripción generada automáticamente

**Modificar Usuarios**

En este apartado he usado varias funciones realizando promesas, una primera función para obtener el formulario con los datos de un usuario en concreto.

Para la primera función lo que realizo es obtener el id concreto del usuario del que quiero ver los datos para posteriormente modificarlos

Texto

Descripción generada automáticamente

Cargo el formulario si recibo datos de la API

Texto

Descripción generada automáticamente

Vacio el contenedor del formulario para que no se sobreescriban los datos y llamo a la siguiente funcion donde valido los campos. Y luego controlo los posibles errores.

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Función asíncrona para validar errores, obtengo los campos y trabajo con sus valores

Texto

Descripción generada automáticamente

Si los todo está correcto, habilito el botón, de lo contrario, quedará deshabilitado

Texto

Descripción generada automáticamente

En la siguiente función recibo los datos del formulario mencionado anteriormente y envío los nuevos datos. Gestionando posibles errores y añadiendo carga dinámica.

Texto

Descripción generada automáticamente

En este apartado manejo los datos de los inputs.

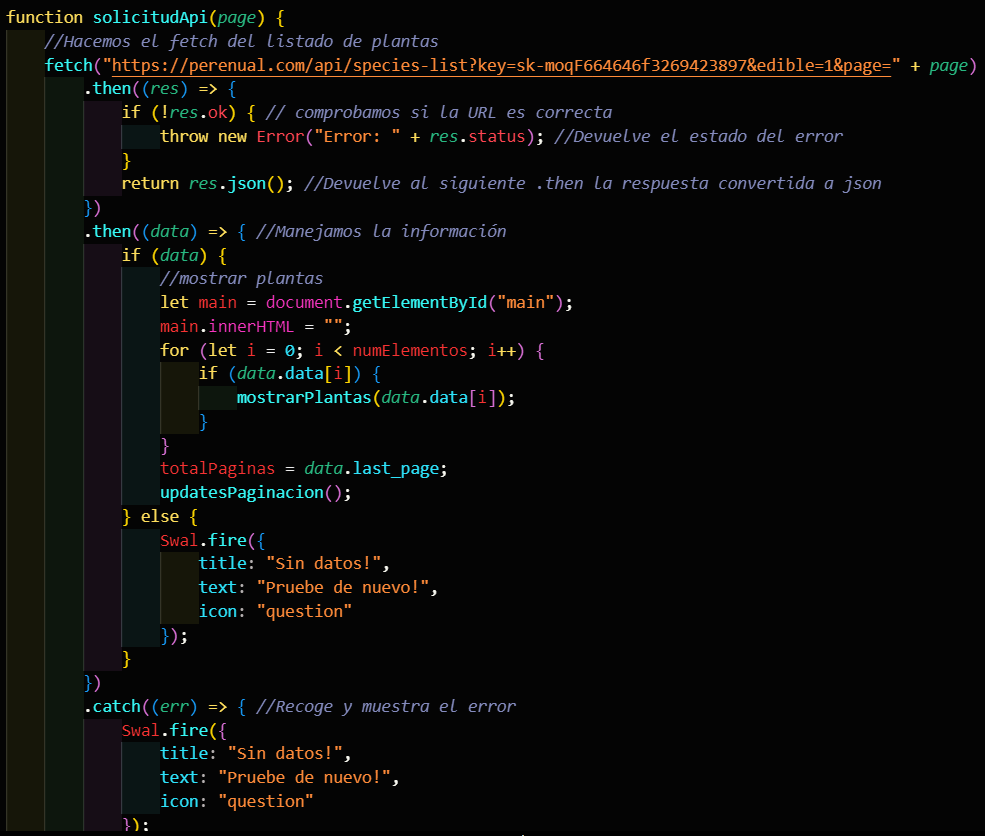
Texto

Descripción generada automáticamente

**Plantas**

En este apartado empleo varias funciones, como la de recibir las plantas, mostrar las plantas y realizar una paginación funcional.

Comenzamos recibiendo los datos que queremos de la API e iterando sobre la función mostrarPlantas, que recibe por parámetro el número de plantas que queremos que nos muestre por página, gestionando los posibles errores que se puedan dar de forma dinámica.



La siguiente función se encarga de la funcionalidad de la paginación.

Texto

Descripción generada automáticamente

También definimos la funcionalidad de los botones.

Texto

Descripción generada automáticamente

Con la siguiente función accedemos a los datos que queremos de la API y construimos nuestra tarjeta con la planta dinámicamente.

Texto

Descripción generada automáticamente

También obtenemos el ID de cada planta para mostrarla individualmente, pasando el ID por localStorage y controlamos el error de la API en el que ponemos una imagen por defecto si la API no nos la proporciona.

Texto

Descripción generada automáticamente

Por último llamamos a la función que solicita a la API para saber en que página nos situamos y realizamos un loader que desaparece cuando la página carga.

Texto

Descripción generada automáticamente

**Planta**

En este apartado mostraremos cada planta individualmente.

Comenzamos obteniendo el ID del localStorage el cual emplearemos para saber que planta queremos ver, realizando un fetch a la API para obtener los datos.

Texto

Descripción generada automáticamente

Obtenemos los datos y volvemos a validar la imagen de la planta

Texto

Descripción generada automáticamente

Creamos la card con iconos y la información que hemos seleccionado de la API

Texto

Descripción generada automáticamente

Cargamos un Canvas con la zona geográfica que mejor se adapta a la planta, la cual recibimos la de API.

Texto

Descripción generada automáticamente

**Modal tiempo**

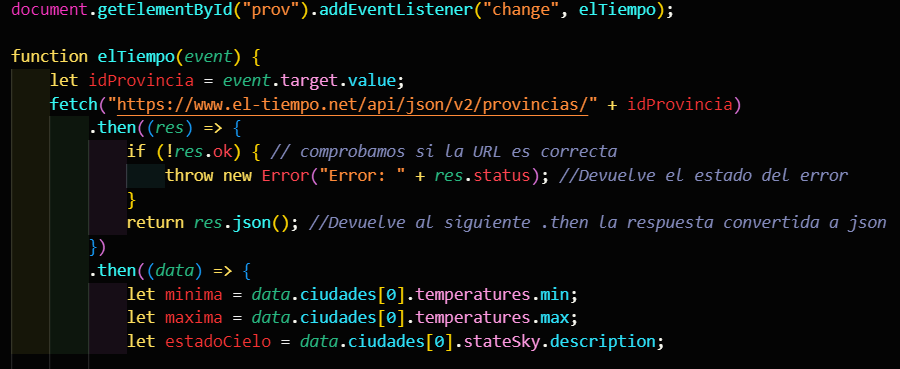
En este apartado empleo un modal para mostrar datos que recojo de la API de AEMET, para mostrar la temperatura máxima, mínima y otros datos de interés.

En esta parte generamos el modal y le damos funcionalidad.

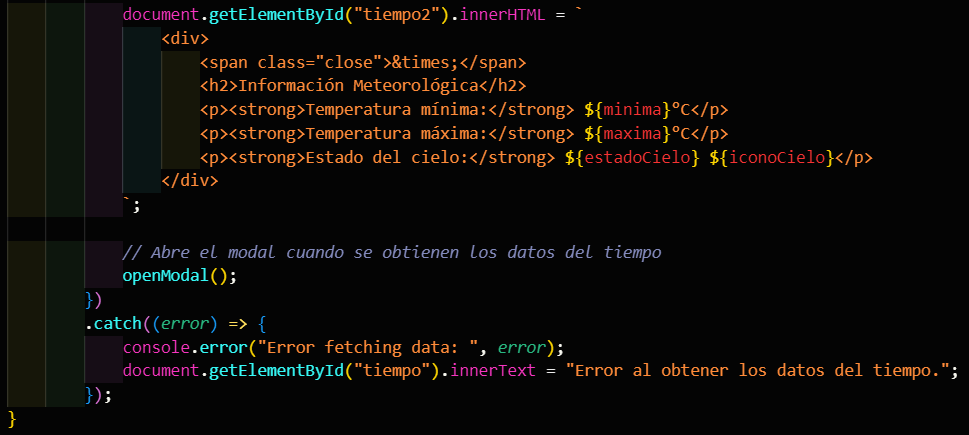
Texto

Descripción generada automáticamente

Posteriormente realizamos un fetch a la API de AEMET para obtener los datos



Mostramos el contenido del modal y gestionamos los posibles errores de la API.



**DarkMode**

En este apartado gestionamos el modo oscuro para el usuario, el cual podrá activar y desactivar a su gusto, por medio de una cookie.

Para ello, creamos una funciones que reciben por parámetro el nombre, el valor y el tiempo de duración de la cookie, para poder crearla y gestionarla a nuestro gusto.

Texto

Descripción generada automáticamente

Le proporcionamos características al fondo de la vista y posteriormente aplicamos una animación en el switch que crea y elimina la cookie

Texto

Descripción generada automáticamente

**Chistes Chuck Norris**

Realizamos una función para que el usuario no se entretenga a la hora de decidir a que apartado ir. La API genera chistes de Chuck Norris aleatorios cada 5 segundos.

Para ello realizamos un fetch a la API para obtener los chistes y controlamos los posibles errores, añadiendo la función setInterval para que se generen de forma indefinida.

Texto

Descripción generada automáticamente