# Práctica de Python, Django y REST

Para afianzar los conocimientos de la asignatura, se deberá realizar una práctica que constará de tres ejercicios, uno de ellos opcional.

# Práctica: Plataforma de Blogging

Se desea crear una plataforma de blogging llamada Wordplease en la cual los usuarios puedan registrarse para crear su blog personal.

## Sitio Web

A continuación se detallan las características deseadas:

- En la página principal, deberán aparecer los últimos posts publicados por los usuarios.
- En la URL /blogs/, se deberá mostrar un listado de los blogs de los usuarios que hay en la plataforma.
- El blog personal de cada usuario, se cargará en la URL /blogs/<nombre\_de\_usuario>/ donde aparecerán todos los posts del usuario ordenados de más actual a más antiguo (los últimos posts primero).
- En la URL /blogs/<nombre\_de\_usuario/<post\_id> se deberá poder ver el detalle de un post.
- Un post estará compuesto de: título, texto a modo de introducción, cuerpo del post, URL de imagen o vídeo destacado (opcional), fecha y hora de publicación (para poder publicar un post en el futuro), categorías en las que se publican (un post puede publicarse en una o varias categorías). Las categorías deben poder ser gestionadas desde el administrador.
- Tanto en la página principal como en el blog personal de cada usuario, se deberán listar los posts con el mismo diseño/layout. Para cada post deberá aparecer el título, la imagen destacada (si tiene) y el resumen.
- En la URL /new-post deberá mostrarse un formulario para crear un nuevo post. Para acceder a esta URL se deberá estar autenticado. En formulario para crear el post deberá identificar al usuario autenticado para publicar el POST en el blog del usuario.
- En la URL /login el usuario podrá hacer login en la plataforma
- En la URL /logout el usuario podrá hacer logout de la plataforma
- En la URL /signup el usuario podrá registrarse en la plataforma indicando su nombre, apellidos, nombre de usuario, e-mail y contraseña.

## **API REST**

Además del sitio web, se desea implementar un API REST que permita en un futuro desarrollar una app móvil para que los usuarios puedan gestionar sus blogs desde la app móvil.

La app móvil deberá tener los siguientes endpoints:

#### **API** de usuarios

- Endpoint que permita a cualquier usuario registrarse indicando su nombre, apellidos, nombre de usuario, e-mail y contraseña.
- Endpoint que permita ver el detalle de un usuario. Sólo podrán ver el endpoint de detalle de un usuario el propio usuario o un administrador.
- Endpoint que permita actualizar los datos de un usuario. Sólo podrán usar el endpoint de un usuario el propio usuario o un administrador.
- Endpoint que permita eliminar un usuario (para darse de baja). Sólo podrán usar el endpoint de un usuario el propio usuario o un administrador.

#### API de blogs

 Un endpoint que no requiera autenticación y devuelva el listado de blogs que hay en la plataforma con la URL de cada uno. Este endpoint debe permitir buscar blogs por el nombre del usuario y ordenarlos por nombre.

#### API de posts

- Un endpoint para poder leer los artículos de un blog de manera que, si el usuario no está autenticado, mostrará sólo los artículos publicados. Si el usuario está autenticado y es el propietario del blog o un administrador, podrá ver todos los artículos (publicados o no). En este endpoint se deberá mostrar únicamente el título del post, la imagen, el resumen y la fecha de publicación. Este endpoint debe permitir buscar posts por título o contenido y ordenarlos por título o fecha de publicación. Por defecto los posts deberán venir ordenados por fecha de publicación descendente.
- Un endpoint para crear posts en el cual el usuario deberá estar autenticado. En este endpoint el post quedará publicado automáticamente en el blog del usuario autenticado.
- Un endpoint de detalle de un post, en el cual se mostrará toda la información del POST. Si el post no es público, sólo podrá acceder al mismo el dueño del post o un administrador.
- Un endpoint de actualización de un post. Sólo podrá acceder al mismo el dueño del post o un administrador.
- Un endpoint de borrado de un post. Sólo podrá acceder al mismo el dueño del post o un administrador.

# **Extras opcionales**

#### General

 Personalizar el administrador de Django para que la plataforma sea más fácil de administrar pudiendo filtrar posts por categorías y mostrando los mismos datos que en el listado del API como columnas en el listado de posts. Personalizar también la página de detalle del post.

## Página web

- Integrar la página con el look and feel de la práctica de front-end avanzado
- Dotar de estilos CSS a la página
- Realizar filtrado por categorías en la página del blog de un usuario
- · Paginar los posts en la página del blog de un usuario

## **API Rest**

- Crear un endpoint para la subida de archivos que pueda ser usado para subir imágenes a la plataforma para luego utilizarlas como imágenes destacadas.
- Realizar filtrado por categorías en el endpoint de listado de posts
- En el API de blogs, mostrar el número de artículos que tiene cada blog
- Mostrar, en un campo 'autor o author', el nombre y apellidos del usuario en el endpoint de listado y detalle de artículos de un blog.