

Python: Proyecto Final

Playground Final project

En forma **individual**, crearás una aplicación web estilo blog programada en Python en Django. Esta web tendrá admin, perfiles, registro, páginas y formularios.

User story/brief:

- ✓ Contar con algún acceso visible a la vista de "Acerca de mí" donde se contará acerca del dueño de la página manejado en el route about/.
- ✓ Contar con algún acceso visible a la vista de blogs que debe alojarse en el route pages/.
- ✓ Acceder a una pantalla que contendrá las páginas. Al clickear en "Leer más" debe navegar al detalle de la page mediante un route pages/<pageld>.
- ✓ Si no existe ninguna página mostrar un "No hay páginas aún".
- ✓ Para editar o borrar pages debes estar logueado.

Piezas sugeridas

Te recomendamos incluir:
<ul style="list-style-type: none">✓ NavBar✓ Home✓ About✓ Pages✓ Login✓ Signup✓ Messages✓ Profile✓ Logout✓ Get pages✓ Get page✓ Create page✓ Update Page✓ Delete page✓ Get profile✓ Update profile

Requisitos base

Los requisitos base serán parte de los criterios de evaluación para aprobar el proyecto.

Tener en cuenta que los requisitos están nombrados un modelo principal para que uds en caso de querer cambiar de blog a otra temática puedan siempre y cuando cumplan los requisitos.

- ✓ Entrega individual
- ✓ Subir a github
- ✓ Readme como la entrega 3
- ✓ Video de máximo 10 min que muestre la página y sus funcionalidades (con o sin audio)
 - programas que pueden utilizar freecam8, obs, filmora 12, etc.
- ✓ No agregar la Base de datos (el archivo db.sqlite3) en la entrega. Debería estar en el .gitignore
- ✓ Uso de herencia de templates. En el template base implementar la etiqueta nav de navegación que contenga los accesos que se crean necesarios para su página.
- ✓ Exista gitignore con:
 - <carpeta del entorno virtual>
 - __pycache__
 - db.sqlite3
 - media

Estos últimos son por el hecho de no compartir la info de tu bd y, aparte, las imágenes son archivos muy pesados que no es recomendable tenerlos en el repo. En cambio, las imágenes que sean parte del código del proyecto deberían guardarse en la carpeta static.
- ✓ Existencia del archivo requirements.txt actualizado.
- ✓ Tener en cuenta al manejar forms con imágenes hay que adaptar el template, y la vista...no solo el modelo y el formulario.
- ✓ Uso de mínimo 2 clases basadas en vista.
- ✓ Uso de mínimo un mixin en una CBV y un decorador en una view común.

- ✓ Una vista de inicio/home.
- ✓ Acceso a una vista "Acerca de mí"/"About"
- ✓ Crear un modelo principal (Blog/Post/Auto/Vendedor/Docente/etc) que contenga los siguiente campos como mínimo: 2 Charfield, 1 de texto enriquecido (usando ckeditor), 1 campo de imagen, 1 de fecha
- ✓ Vista de listado de los objetos del modelo principal (modelo a elección). En la cual cada objeto mostrará solo algunos de sus datos.
- ✓ Mensaje que da aviso en caso de no haber ningún objeto creado o al utilizar el buscador no encontrar tampoco algún objeto.
- ✓ Desde el listado:
 1. poder acceder a una vista que muestre el detalle de el objeto seleccionado

2. poder acceder a una vista de creación, una de edición y una de borrado de objetos.

- ✓ Registrar en el apartado de admin todos los modelos creados.
- ✓ Tener una app (accounts/cuentas/etc) para el manejo de todas las vistas relacionadas al usuario/autenticación.
- ✓ Desarrollar las vistas para un login, un logout y un registro para usuarios. En este último se debe solicitar: username, email, password.
- ✓ Crear una vista de perfil donde se muestran los datos del usuario:
 - nombre
 - apellido
 - email
 - avatar
 - biografía/link/fecha de cumpleaños/etc.
- ✓ Desde el perfil, crear un acceso a una vista de edición de estos datos. Agregar el cambio de password.
- ✓ Crear una app de mensajería con todo lo necesario para que los usuarios puedan comunicarse entre sí por mensajes. Todo en esta app queda a criterio del alumno/a siempre y cuando funcione correctamente.
- ✓ PUNTAZO A TENER EN CUENTA! PROBAR, PROBAR Y PROBAR ANTES DE SUBIR EL CÓDIGO A GITHUB... (no apurarse a hacer el commit y subir los cambios porque puede generar algún problema sin darnos cuenta)

Requisitos Extra

Los requisitos extra pro-coders no se incluyen en los criterios de evaluación.

Los requisitos extra son funcionalidades opcionales que no se incluyen en los criterios de evaluación, pero si te falta diversión y quieres agregar valor a tu proyecto... ¡bajo la única **condición** de que **lo que incluyas debe funcionar!**

- ✓ Messenger y like – integración otra db

Dont's

No es necesario ni recomendado.

- ✓ Utilizar Python puro para el proyecto final (se espera el uso de Django).