**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/<pageId>.
* Si no existe ninguna página mostrar un "No hay páginas aún".
* Para crear, editar o borrar las fotos debes estar registrado como Administrador.

**Piezas sugeridas**

| **Te recomendamos incluir:** |
| --- |
| * 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.** |
| --- |
| * **Inicio:** Al momento de ingresar a la app en la ruta base ‘/’ * Visualizar el home del blog. * Poder listar todas las páginas del blog, poder ver en detalle cada una, poder crear, editar o borrar páginas del blog. * Las páginas están formadas por un título, un contenido en editor de texto avanzado (ckeditor por ejemplo), una imagen, fecha de posteo de la imagen. * Tener una app de registro donde se puedan registrar usuarios en el route accounts/signup, un usuario está compuesto por: email - contraseña - nombre de usuario * Tener una app de login en el route accounts/login/ la cual permite loggearse con los datos de administrador o de usuario normal. * Tener una app de perfiles en el route accounts/profile/ la cual muestra la info de nuestro usuario y permite poder modificar y/o borrar: imagen - nombre - descripción - un link a una página web - email y contraseña * Contar con un admin en route admin/ donde se puedan manejar las apps y los datos en las apps. * Tener una app de mensajería en el route messages/ para que los perfiles se puedan contactar entre sí. |

**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). |