

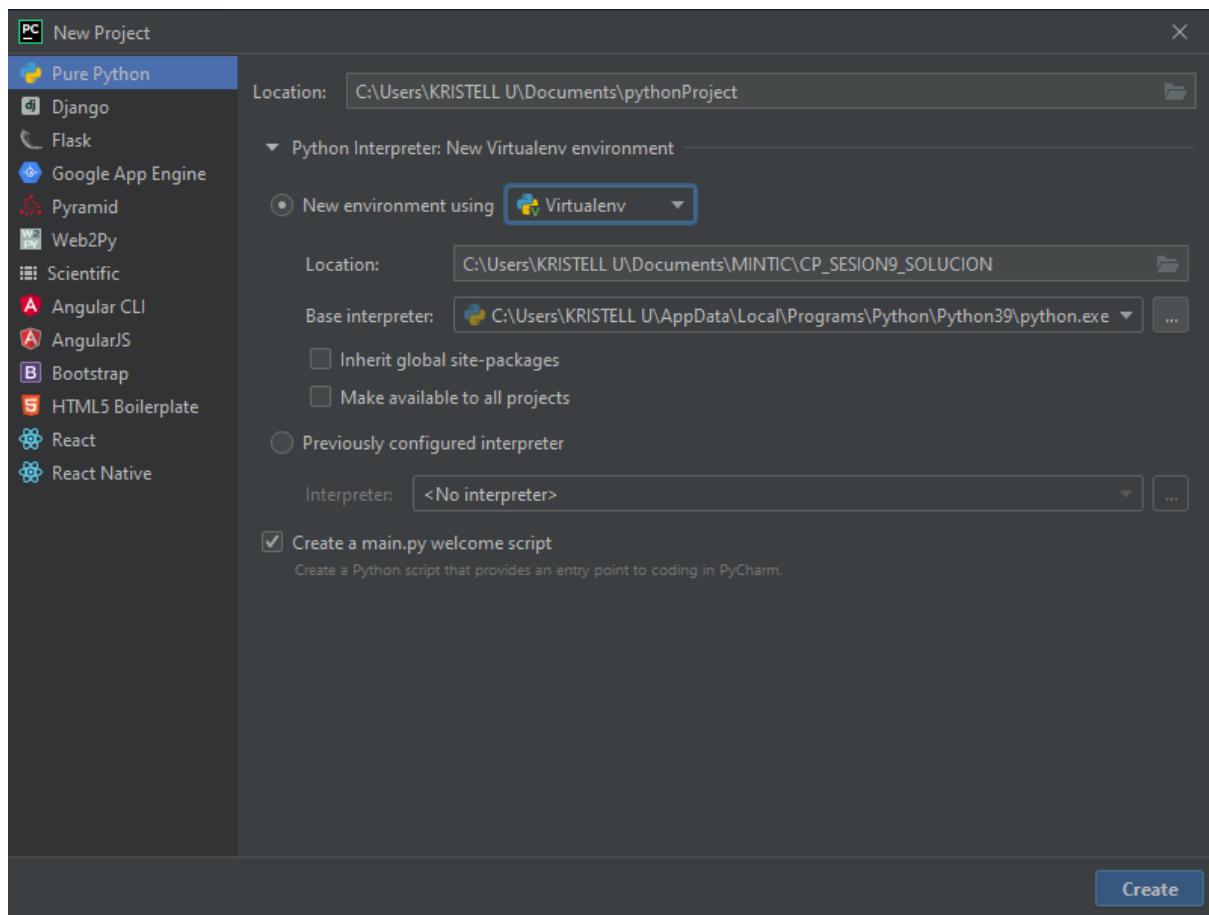
Sesión # 8 Componente Práctico

Tecnologías y lenguajes para el desarrollo del Back-end

Dado que ya tenemos Flask instalado, vamos a instalar PyCharm, un IDE que facilitará el proceso de desarrollo de las aplicaciones.

- Para usarlo con Flask vas a requerir la edición Professional (tienes 1 mes gratis) con esta licencia. <https://www.jetbrains.com/pycharm/download>

Vamos a crear un nuevo proyecto para esta sesión (escoge la ubicación de tu preferencia).



Una vez creado, podrás ver que PyCharm añadió una serie de carpetas para el proyecto.

Vamos a abrir la carpeta de nuestro proyecto en VSCode y organizaremos los documentos con los que venimos trabajando en las sesiones anteriores (muy parecido al patrón MVC) de la siguiente manera:

1. Crea una carpeta static y añade tres subcarpetas en ella:
 - a. css – Incluir el archivo css miEstilo.css que generamos en las sesiones anteriores.

- b. Images – Incluir la imagen para ver el password con la que trabajamos en las sesiones anteriores (view.png).
 - c. js – Incluir el archivo JavaScript creado y renombrarlo como `validacion.js`
- 2. Crea una carpeta `template` y mueve el archivo `.html` dentro de ella.
- 3. Vamos a modificar ahora el archivo `.html` para corregir las rutas a los archivos previamente usados.
 - a. Modifica la ruta a cada archivo de la forma:

```
{{ url_for('static', filename='css/miEstilo.css') }}
```

```
{{ url_for('static', filename='js/validacion.js') }}
```

```
{{ url_for('static', filename='images/view.png') }}
```
- 4. Finalmente, vamos a modificar el archivo `app.py` y usando la función `render_template` vamos a llamar a la página HTML que realizará el registro a nuestra aplicación. Ejemplo:

```
from flask import Flask, render_template
app = Flask(__name__)
@app.route('/')
def index():
    return render_template('sesion.html')
```

El árbol de archivos debe quedar así:

