Guía para Estudiantes - Curso Python para la Web

1. Hacer Fork y Clonar el Repositorio

- Abre el navegador y ve a: https://github.com/FreakJazz/python-ws
- Haz clic en el botón **"Fork"** (arriba a la derecha). Esto creará una copia del repositorio en tu cuenta.
- Abre tu repositorio forkeado y copia el enlace HTTPS **(botón verde 'Code').**
- Abre Visual Studio Code o terminal y escribe los siguientes comandos (reemplaza **TU_USUARIO** con tu usuario de GitHub):

```
git clone https://github.com/TU_USUARIO/python-ws.git
cd python-ws
```

2. Conectar tu Repositorio con el del Profesor (upstream)

Esto es para que puedas actualizar tu repositorio con las nuevas clases que se van a ir subiendo cada día.

Ejecuta este comando dentro de la carpeta del proyecto:

```
git remote add upstream https://github.com/FreakJazz/python-ws.git
```

Verifica que todo esté correcto con:

```
git remote -v
```

Debe mostrar dos líneas: 'origin' (tu copia) y 'upstream' (el del repositorio de origen).

3. Subir tus Tareas

Se han creado las carpetas por clase (clase1, clase2, etc.).

Dentro de cada clase debes subir tu archivo con el siguiente formato:

nombre_apellido.py

Ejemplo para la clase 1:

```
clase1/jazmin_rodriguez.py
```

Pasos para subir tu tarea:

1. Copia tu archivo dentro de la carpeta correcta (por ejemplo clase2/).

2. Abre la terminal y ejecuta:

```
git add .
git commit -m "Entrega clase 2 - Jazmin Rodriguez"
git push origin main
```

3. Entra a tu repositorio en GitHub y haz clic en "Compare & pull request".

4. Verifica que el repositorio base sea el del profesor y haz clic en "Create pull request".

4. Actualizar tu Código con Nuevas Clases

Antes de comenzar cada nueva clase, asegúrate de tener la versión más actualizada del repositorio del profesor.

1. Ejecuta estos comandos en la terminal:

```
git fetch upstream
```

```
git merge upstream/main
```

2. Luego puedes subir esos cambios a tu copia en GitHub también:

```
git push origin main
```

Esto evitará errores y te permitirá ver las carpetas nuevas creadas.

5. Recomendaciones Finales

- Siempre trabaja dentro de la carpeta correcta (clase1, clase2, etc).
- Usa nombres de archivo con tu nombre y apellido separados por guión bajo: ejemplo juan_perez.py
- Nunca edites ni borres archivos de otros compañeros.
- Si algo no funciona, vuelve a leer los pasos o pide ayuda.
- No necesitas saber programar para seguir estos pasos, solo copiar, pegar y subir tu archivo.

6. Ventajas de hacer un fork

- Te permite trabajar en una copia personal sin afectar el proyecto original.
- Puedes subir tus tareas y practicar sin miedo a dañar nada.
- Facilita enviar tus avances al profesor mediante Pull Requests.
- Permite mantener tu copia siempre actualizada con los nuevos cambios.
- Es el mismo método que usan los desarrolladores profesionales en proyectos reales.