



Backend con Python/Django

Proyecto final

Objetivo

Implementar algunos desarrollos web con Python como lenguaje de programación, dentro del *framework* Django.

Descripción

El aprendedor implementará un sistema que le permita a los usuarios subir, bajar, visualizar y administrar videos vía *streaming* en una plataforma digital, a través del uso de Python/Django.

Instrucciones

En el curso encontrarás cuatro avances, uno por semana, para que construyas tu proyecto final que deberás entregar en la semana cinco. Aunque estos avances no son evaluables, se sugiere que los realices para tener una mejor organización.

Duración: 2 horas.

En este avance 1 tendrás un contacto con el lenguaje de programación Python: su instalación, usabilidad y utilidad. Además, llevarás a cabo una interacción inicial con el lenguaje de programación en sí.

Instrucciones

Pro-Gol watch es una empresa dedicada a monitorear videos de competiciones futbolísticas en todo el mundo (copas, torneos, mundiales y eventos similares). Su objetivo es detectar tendencias en los partidos, jugadores promesas, estrategias, etc., para analizarlas y, con la información recabada, dar sugerencias a los entrenadores/directivos de los clubes de fútbol que lo soliciten.

Luis Santoy es un joven emprendedor de 25 años, siempre ávido de conocimiento y de nuevos retos. Desde hace cinco años forma parte del equipo de tecnologías de información de **Pro-Gol watch** como desarrollador junior. Desde que estudiaba tuvo la oportunidad de realizar sus prácticas en la empresa y, por sus excelentes resultados, se le contrató de manera permanente.

La empresa le solicitó a Luis desarrollar un proyecto que cumpliera con todas las necesidades actuales de la compañía, pero con una nueva herramienta: Python. Para esto se organizó un proyecto dividido en cuatro etapas.

En primer lugar, se le pidió a Luis que instalara todo lo necesario para empezar a programar con esta tecnología, así como realizar su primera interacción entre usuario y plataforma a través de ciclos y condicionales.

Para lograr lo anterior, Luis deberá seguir estos pasos:

1. Definir el nombre del proyecto (por ejemplo, “Mis videos online”), pues esto ayudará a referenciarlo durante todo el proceso. Debe ser un nombre creativo y fácil de recordar para los usuarios.
2. Instalar los programas necesarios para empezar a programar en Python (Python, PyCharm, Insomnia Rest Client). Se pueden descargar en los siguientes enlaces:



Python. (s.f.). *Download Python*. Recuperado de
<https://www.python.org/downloads/>

Visual Studio Code. (s.f.). *Code Editing. Redefined*. Recuperado de
<https://code.visualstudio.com/>

Avance 1



Kong. (s.f.). *Download Insomnia*. Recuperado de <https://insomnia.rest/download>

El uso y descarga del software deberá apegarse a los términos y condiciones del sitio oficial del fabricante y su uso será responsabilidad de quien lo descargue. Tecmilenio no tiene licencia ni posee los derechos sobre dicho software.

3. Desarrollar una aplicación en la que un usuario pueda subir desde 1 hasta N videos y que solo le permita cargar archivos de máximo 3 MB. Todo lo anterior debe contemplar algunas excepciones, por ejemplo, posibles errores por parte del usuario.

Cambia la lógica del código del avance 1:

1. Crea tu primer archivo codificado en Python (archivo con extensión **.py**) que pueda realizar lo siguiente:

- a. Pedirle a un usuario su *Id* (número de nómina).
- b. Pedirle a un usuario su nombre.
- c. Preguntarle a un usuario la cantidad de videos que subirá.

2. Para cumplir con el punto anterior deberás:

- a. Definir las variables (y su tipo de dato) que guardarán la nómina, nombre del usuario y cantidad de videos a subir.
- b. En cada línea o sentencia del código documentarás, por medio de comentarios, la funcionalidad que realiza.

3. Una vez que el usuario introduzca la información requerida se deberá desplegar el siguiente mensaje con sus datos:

- a. Bienvenido **[nombre del usuario]**, tu número de nómina es **[número de nómina]** y estás intentando subir **[cantidad de videos]** videos. ¿Es correcta la información? Sí/No.
- b. Uso de condicionales. En caso de que el usuario conteste “Sí”, se le pedirá la siguiente información para cada uno de los “N” videos que desea subir:
 - i. Título del video.
 - ii. Nombre del video.
 - iii. Extensión del video (.mpg, .mov, etc).
 - iv. Tamaño (en megas y no mayor a 3).

Avance 1



NOTA. Para realizar la petición de los “N” videos utiliza ciclos (FOR o WHILE).

c. Uso de condicionales. En caso de que el usuario conteste “No”, se le preguntará si desea salir del sistema. De ser afirmativo, se mostrará el siguiente letrero **Muchas gracias por haber usado nuestro sistema, hasta pronto.** Si el usuario contesta que “No” quiere salir del sistema, se le pedirá nuevamente la información de entrada:

- i. Pedir su id (número de nómina).
- ii. Pedir su nombre.
- iii. Preguntar la cantidad de videos a subir.

4. Valida cada una de las capturas de información del usuario:

- a. Número de nómina: valor alfanumérico (A-Z, a-z, 0-9).
- b. Nombre del usuario: valor alfabético (A-Z, a-z).
- c. Cantidad de videos a subir: valor numérico (0-9).
- d. Título del video: valor alfanumérico (A-Z, a-z, 0-9).
- e. Nombre del video: valor alfanumérico (A-Z, a-z, 0-9).
- f. Extensión del video: valor alfanumérico (A-Z, a-z, 0-9).
- g. Tamaño: valor numérico (0-3).

5. En caso de que el valor colocado sea incorrecto, arrojar uno de los siguientes mensajes para cada validación (según sea el caso):

- a. Número de nómina: "Número en formato incorrecto. Debe capturar solo números y letras."
- b. Nombre del usuario: "Nombre de usuario en formato incorrecto. Debe capturar solo letras."
- c. Cantidad de videos a subir: "Cantidad de videos en formato incorrecto. Debe capturar solo números."
- d. Título del video: "Título del video en formato incorrecto. Debe capturar solo números y letras."
- e. Nombre del video: "Nombre del video en formato incorrecto. Debe capturar solo números y letras."

f. Extensión del video: "Extensión del video en formato incorrecto. Debe capturar solo números y letras."

g. Tamaño:

- i. Si el usuario introduce un valor alfanumérico, desplegar este mensaje: "Tamaño del video en formato incorrecto. Debe capturar solo números".
- ii. Si el usuario introduce un valor menor a 0 o mayor a 3, arrojar esta leyenda: "El archivo no debe pesar más de 3 MB".



NOTA. Para este último punto deberán manejarse al menos 2 excepciones.

6. Para cada uno de los puntos anteriores deberás crear al menos las siguientes funciones:

- a. Función para pedir **id**, nombre y cantidad de videos al usuario.
- b. Función para que el usuario capture título, nombre, extensión y megas de cada uno de los videos a subir.
- c. Función (o funciones) para validar capturas del usuario.
- d. Función para manejo de excepciones.
- e. Función que despliegue los letreros de las validaciones (deberá ser una sola función).

7. Validada la información, esta se guardará en la computadora, en un archivo llamado **salida.txt**, como una línea con el siguiente formato:

- i. Número de nómina | Nombre del usuario | Cantidad de videos| Título video 1 | Nombre del video 1 | Extensión video 1 | Tamaño video 1 | Título video 2 | Nombre del video 2 | Extensión video 2 | Tamaño video 2|.... |Título video N | Nombre del video N | Extensión video N | Tamaño video N |

Ejemplo: N087545 | Irma Guardado Segura | 5 | Ventajas del aprendizaje | ventajas aprender a aprender | mpg | 3 | ... | Python en 1, 2, 3 pasos | Python para principiantes | .mov | 3 |

Guarda este primer avance en tu archivo bajo el nombre **Primera etapa.py**, ya que será parte de tu entregable



NOTA. Para fines de esta práctica, el término "usuario" aplica para la persona que empleará el sistema, ya sea el mismo aprendedor o cualquier otra.

Avance 1

El entregable deberá contener:

1. Archivo **.py** con el código implementado.
2. Archivo con evidencia (capturas de pantalla) de haber probado la funcionalidad.
3. El archivo generado por el proyecto (con extensión **.txt**).

Consideraciones generales

Para el aprendedor

Para realizar este primer avance deberás conocer y practicar en Python los siguientes temas:

- Documentación de código usando comentarios de programación.
- Declaración de variables.
- Uso de diferentes tipos de datos.
- Conversión de datos.
- Entrada y salida de datos.
- Uso de archivos (ficheros).
- Condicionales (IF-ELSE) sencillos y anidados.
- Ciclos (FOR – WHILE).
- Uso de las excepciones.
- Uso de funciones.

Para el instructor

Para este avance del proyecto, asegúrese de que el aprendedor comprenda lo siguiente:

- Los conceptos básicos de programación en Python.
- El uso correcto y efectivo de los comentarios dentro de código fuente, además de su utilidad a largo plazo.
- La declaración de variables y su conversión a diferentes tipos de datos.
- El uso de la consola para entrada y salida de datos.
- El guardado de datos en un fichero.
- El uso de condicionales.
- El correcto uso de ciclos.
- Manejo de excepciones y validaciones.
- Construcción de funciones para modularizar el código.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

Avance 2

Duración: 2 horas.

En este segundo avance interactuarás con condicionales, ciclos, funciones, excepciones y tendrás la oportunidad de empezar con la programación orientada a objetos.

Crearás todos los objetos necesarios para transformar tu código a sentencias para el manejo de clases y objetos.

Instrucciones

1. Crea una nueva versión de tu código y nómbrala **Segunda etapa.py**. En este archivo realizarás algunos cambios.

Crea una clase llamada “Persona” con los siguientes elementos:

a. Atributos:

- i. Nombre.
- ii. *Id* (nómina).

b. Funciones para:

- i. Capturar nombre.
- ii. Capturar id.
- iii. Imprimir nombre.
- iv. Imprimir id (nómina).

2. Crea una clase llamada “Videos” con estos componentes:

a. Atributos:

- i. Nombre del video.
- ii. Extensión del video.
- iii. Tamaño del video.

b. Funciones para:

- i. Capturar el nombre del video.
- ii. Capturar la extensión del video.
- iii. Capturar el tamaño del video.
- iv. Imprimir el nombre del video.
- v. Imprimir la extensión del video.
- vi. Imprimir el tamaño del video.

3. Crea un objeto del tipo “Persona” y otro del tipo “Videos”.

Avance 2

4. Utiliza estos objetos para capturar, imprimir y guardar la información de la persona y sus videos relacionados.

5. Asegúrate de que toda la funcionalidad de la etapa 1 sea igual a la de la etapa 2.

El entregable deberá contener:

1. Archivo **Segundo avance.py** con el código implementado en la segunda etapa.
2. Archivo con la evidencia (capturas de pantalla) de haber comprobado toda la funcionalidad.
3. Archivo de salida generado por el proyecto (**salida.txt**).

Consideraciones generales

Para el aprendedor

Para realizar este segundo avance deberás conocer y practicar en Python los siguientes temas:

- Manejo de clases y objetos.
- Desarrollo y uso de “Constructores”.
- Conceptos básicos de programación orientada a objetos.
- Uso de los atributos de una clase.
- Métodos de un objeto.
- Uso y manejo de encapsulamiento, herencia y polimorfismo.
- Importación de clases.
- Uso de módulos y paquetes.

Para el instructor

Para este avance del proyecto, asegúrese de que el aprendedor comprenda:

- Uso y manejo de clases y objetos.
- Conceptos de programación orientada a objetos: encapsulamiento, herencia y polimorfismo.
- Uso de módulos y paquetes.
- Atributos de una clase.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

Avance 3

Duración: 2 horas.

En este tercer avance conocerás el framework Django y entenderás cómo interactúa con Python; asimismo, aprenderás a hacer una base de datos en PostgreSQL.

Instrucciones

1. Una vez instalado Django, crea un proyecto llamado “Mis videos”.
2. Renombra el archivo creado en el avance anterior como **Avance 3.py**.
3. Asocia el archivo **Avance 3.py** al proyecto “Mis videos”.
4. Crea una base de datos llamada **Pro_Gol** que contenga las siguientes tablas y campos:
 - a. Tabla para guardar los datos del usuario:
 - i. Nombre. Tipo de dato alfanumérico de 50 caracteres.
 - ii. id (nómina). Campo llave, tipo de dato alfanumérico de 10 caracteres.
 - b. Tabla para guardar videos:
 - i. Id del video.
 - ii. Nombre del video. Tipo de dato alfanumérico de 50 caracteres.
 - iii. Extensión del video. Tipo de dato alfanumérico de 5 caracteres.
 - iv. Tamaño del video. Tipo de dato numérico.
 - c. Tabla para guardar los videos que subirá el usuario:
 - i. Id del usuario. Tipo de dato numérico.
 - ii. Id del video. Tipo de dato numérico.
5. El nombre de las tablas y campos deberán corresponder con la acción a realizar. **Ejemplo:** tabla de usuario **TBL_Usuario**.
6. Agrega la base de datos al proyecto “Mis videos” en Django.
7. Cambia la lógica de tu proyecto para guardar toda la información en la base de datos **Pro_Gol**.
 - a. Crea las funciones pertinentes.
 - b. Modifica los objetos antes construidos para adaptarlos a la interacción con la base de datos **Pro_Gol**.

El entregable deberá contener:

1. Archivo **Avance 3.py** con el código implementado en la tercera etapa.
2. Proyecto “Mis videos” creado en Django.
3. Base de datos y sus tablas correspondientes.

Avance 3

Consideraciones generales

Para el aprendedor

Para poder realizar este avance deberás conocer y practicar en Python los siguientes temas:

- Uso de framework Django.
- Construcción de base de datos en PostgreSQL.
- Acceso a datos.

Para el instructor

Para este avance del proyecto, asegúrese de que el aprendedor comprenda:

- Uso correcto del framework Django.
- Uso y manejo de PostgreSQL.
- Interacción entre Python/Django/PostgreSQL.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

Avance 4

Duración: 2 horas.

En este último avance construirás las pantallas para hacer más amigable tu proyecto. Esto volverá más atractivo e intuitivo el uso del sistema para los usuarios.

Instrucciones

1. Usando HTML construye una pantalla que le permita al usuario capturar:
 - a. Su id (número de nómina).
 - b. Su nombre completo.
 - c. Cantidad de videos a subir.
2. Botón de “Aceptar” que, al presionar, despliegue una ventana emergente de confirmación con el siguiente mensaje de validación:
 - a. Bienvenido **[nombre del usuario]**, tu número de nómina es **[número de nómina]** y estás intentando subir **[cantidad de videos]** videos. ¿Es correcta la información? Sí/No.
 - b. Si el usuario elige “Sí”, se le pedirá la siguiente información para los “N” videos que desea subir:
 - i. Título del video.
 - ii. Nombre del video.
 - iii. Extensión del video (.mpg, .mov, etc.).
 - iv. Tamaño (en megas, no mayor a 3).



NOTA. Para realizar la petición de los “N” videos utiliza ciclos (FOR o WHILE).

3. Todas las validaciones de los avances anteriores deberán aplicarse en esta etapa y el despliegue de los mensajes tendrá que ocurrir al momento de salir de los cuadros de captura.
4. Una vez validada la información deberá existir un botón de “Guardar”, que enviará la información a la base de datos.
5. Usando CSS prepara un diseño de estilos: colores, tipo de fuente, tamaño, etc. Este diseño debe ser atractivo para el usuario.

Avance 4



Toda la interacción con el usuario (*frontend*) será a través de HTML y CSS; por su parte, la lógica de negocio (*backend*) se desarrollará por medio de Python.

El entregable deberá contar con los siguientes elementos:

1. Proyecto en Django/Python.
2. Ejecución del proyecto con la evidencia (en pantallas o video) de su funcionamiento.
3. Base de datos en PostgreSQL.
4. Aplicación de estilos por medio de CSS.

Consideraciones generales

Para el aprendedor

Para poder realizar este cuarto avance deberás conocer y practicar en Python los siguientes temas:

- Manejo de formularios en HTML.
- Etiquetas en HTML.
- Aplicación de formato y estilos en CSS.
- Construcción de sitios web.
- Publicación de sitios web.

Para el instructor

Para este avance del proyecto, asegúrese de que el aprendedor comprenda:

- Uso y manejo general de HTML.
- Correcto uso de etiquetas en HTML.
- Uso de formatos y estilos en CSS.
- Construcción y publicación en sitios web.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

Criterios de evaluación

Rúbrica

Criterios de evaluación	Nivel de desempeño			%
	Altamente competente 100% - 86%	Competente 85% - 70%	Aún sin desarrollar la competencia 69% - 0%	
1. Desarrollo de backend para captura de datos de entrada y validación de datos capturados por el usuario.	15 - 14 Desarrolla correctamente la captura de datos de entrada. La validación de datos fue correcta.	13 - 11 Desarrolla correctamente la captura de datos de entrada, pero no validó los datos correctamente.	10 - 0 No desarrolla correctamente la captura de datos de entrada. La validación de datos fue incorrecta.	15
2. Creación de funciones, clases y objetos para el correcto funcionamiento de la aplicación.	15 - 14 Crea correctamente las funciones, clases y objetos en la aplicación.	13 - 11 Crea correctamente las funciones, pero presenta problemas para desarrollar las clases y objetos en la aplicación.	10 - 0 No crea correctamente las funciones, clases y objetos en la aplicación.	15
3. Integración en Django con Python y la base de datos en PostgreSQL.	30 - 26 Integra correctamente Django con Python y la base de datos en PostgreSQL.	25 - 22 Integra correctamente Django con Python, pero no así con la base de datos en PostgreSQL.	21 - 0 No integra correctamente Django con Python y la base de datos en PostgreSQL.	30
4. Desarrollo de backend para la interacción con la base de datos (creación y configuración).	30 - 26 Crea y configura correctamente la base de datos; además, la aplicación interactúa correctamente con ella.	25 - 22 Crea y configura correctamente la base de datos, pero la aplicación no interactúa correctamente con ella.	21 - 0 No crea ni configura correctamente la base de datos; además, la aplicación no interactúa correctamente con ella.	30

Criterios de evaluación

	10 - 9	8 - 5	4 - 0	
5. Diseño de pantallas utilizando HTML y CSS.	Realiza correctamente el diseño de las pantallas utilizando HTML y CSS.	Realiza el diseño de algunas pantallas utilizando HTML y CSS.	No realiza el diseño de las pantallas utilizando HTML y CSS.	10
Total				100%

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.