PARTE TEORICA (10 PUNTOS)

1. Para qué se puede usar Python en lo que respecta a datos. Dar 5 casos y explicar brevemente

Python se utiliza en el manejo y análisis de datos por su simplicidad y gran cantidad de librerías especializadas. Algunos usos son:

1. Análisis de datos: con librerías como Pandas y NumPy se pueden limpiar, transformar y analizar grandes volúmenes de información.
2. Visualización de datos: mediante herramientas como Matplotlib o Seaborn se pueden crear gráficos e interpretar resultados de manera visual.
3. Machine Learning: con Scikit-learn o TensorFlow se desarrollan modelos predictivos para clasificar o predecir datos.
4. Automatización de reportes: se pueden programar scripts que procesen datos y generen informes automáticamente.
5. Conexión a bases de datos: Python puede leer y escribir información desde bases de datos SQL o NoSQL, facilitando la gestión de la información empresarial.

2. ¿Cómo se diferencian Flask de Django? Argumentar.

Flask es un microframework que ofrece lo esencial para construir una aplicación web de forma sencilla y rápida. Es más flexible y ligero, ideal para proyectos pequeños o medianos.  
Django, en cambio, es un framework completo que incluye herramientas integradas para autenticación, administración y ORM. Es más estructurado y se usa en proyectos grandes que requieren una arquitectura más robusta.  
En resumen, Flask ofrece libertad y simplicidad, mientras que Django brinda una estructura más completa y predefinida.

3. ¿Qué es un API? Explicar en sus propias palabras

Un API (Application Programming Interface) es una interfaz que permite que diferentes aplicaciones o sistemas se comuniquen entre sí. Define un conjunto de reglas para enviar y recibir información sin necesidad de conocer cómo funciona internamente el otro sistema. Por ejemplo, una aplicación móvil puede usar un API para obtener datos desde un servidor remoto.

4. ¿Cuál es la principal diferencia entre REST y WebSockets?

REST es un protocolo basado en solicitudes HTTP, donde el cliente realiza una petición y el servidor responde. Es ideal para operaciones puntuales, como crear o consultar datos.  
WebSockets permiten una comunicación en tiempo real y bidireccional entre cliente y servidor, manteniendo una conexión abierta constante. Se utilizan en chats, videojuegos en línea o monitoreos en vivo.

5. Describir un ejemplo de API comercial y como funciona – usar otros ejemplos no vistos en el curso.

Un ejemplo de API comercial es la API de Stripe, usada para procesar pagos en línea. Las empresas integran esta API en sus sitios web para que los usuarios puedan pagar con tarjeta sin que la empresa maneje directamente los datos financieros.  
El sistema funciona enviando una solicitud segura a los servidores de Stripe, los cuales validan el pago y devuelven una respuesta con el estado de la transacción (exitosa o rechazada).