* **Pregunta de resolución de problemas en Python:**

"Desarrolla un script en Python que nos permita calcular la factorial de un número dado. Además, ¿cómo asegurarías que el script maneje correctamente casos especiales como la factorial de 0 y números negativos?"

**Respuesta:**

**En este Script de Python inicialmente defino una función para que calcule la factorial de un número, si el número es 0 o uno devolverá su factorial correspondiente que s 1 en ambos casos.**

**Luego ejecuto un control antes de llamar la función lo cual asegura que el número del que se calcula es factorial es un entero positivo, mientras el numero ingresado sea menor que 0 el script seguirá pidiendo un numero entero positivo, en caso que el número ingresado sea un decimal la string del input será casteada a float y después a entero y finalmente una vez se obtenga el numero entero positivo se calculara la factorial y se devolverá el resultado en un print.**

def factorial(num):

    if num in [0, 1]:

        return 1

    else:

        resultado= 1

        for i in range(1, num+1):

            resultado \*=i

        return resultado

num= int(float(input('introduce un numero entero positivo: ')))

while num < 0:

    num= int(float(input('introduce un numero entero positivo: ')))

if num >=0:

    print(f'Su factorial es {factorial(num)}')

* **Pregunta sobre SQL:**

"Imagina que tenemos dos bases de datos diferentes para nuestros productos: una en nuestro Sistema de Gestión de Almacén (SGA) y otra en Navision. ¿Cómo enlazarías y compararías los datos de estos sistemas utilizando SQL? Proporciona un ejemplo de consulta que muestre esta integración y explica qué tipo de JOIN usarías."

**Respuesta:**

Suponiendo que tengo acceso a las bases de datos con los permisos adecuados para acceder desde mi entorno de trabajo, la forma correcta de enlazar las bases de datos dependerá del sistema de gestión de bases de datos que se esté utilizando, Algunos sistemas de gestion permiten la creación de enlaces entre bases de datos, por ejemplo, Oracle utiliza "Database Links", mientras que SQL Server utiliza "Linked Servers". Al configurar estos enlaces para establecer la conexión entre las bases de datos:

En SQL server podría crear un "Linked Servers" de la siguiente manera:

EXEC sp\_addlinkedserver

@server= 'NuevoServer',

@srvproduct= ' ',

@provider= 'SQLNCLI', #proveedor dependerá de la versión de SQL

@datasrc= 'NombreServidorSegunadaBaseDatos'

Después de crear el "Linked Servers" se podría realizar la consulta se la siguiente manera:

SELECT \*

FROM NuevoServer.BaseDatosGestionAlmacen.dbo.tabla\_sga as sga

LEFT JOIN tabla\_navision as tn

ON sga.id\_sga = tn.id\_navision

En esta consulta estoy considerando que las tablas se encuentran relacionadas por una columna Id y que la información predominante se encuentra en la tabla SGA ya que al utilizar el LEFT JOIN los datos de esta tabla permanecerán además de los datos de la tabla Navision donde el id exista en ambas tablas. Además, estoy haciendo un SELECT \* para mirar toda la información ya que no se especifica cuáles columnas se quieren obtener o si se desea realizar alguna operación.

* **Pregunta sobre la centralita Asterisk:**

"Queremos tener un registro del número de llamadas recibidas a través de nuestra centralita Asterisk para poder analizar el volumen de llamadas de nuestros clientes. ¿Cómo diseñarías y gestionarías una base de datos para almacenar esta información? Además, ¿qué consideraciones tendríamos que tener en cuenta para asegurar la integridad y eficiencia de los datos almacenados?"

**Respuesta:**

Para diseñar una base de datos para almacenar las llamadas de los clientes lo haría de la siguiente manera:

* Dividir la información en varias tablas

**Tabla\_llamadas**

ID\_llamada: definida como Primary Key(PK) para identificar cada llamada.

ID\_Cliente: definida como Foreing Key(FK) para identificar cada cliente y poder relacionar las tablas

ID\_Empleado: definida como Foreing Key(FK) para relacionar con tabla empleados

Fecha\_Hora\_inicio: fecha y hora de la llamada definida como timestamp y Primary Key

Fecha\_Hora\_final: fecha y hora al finalizar la llamada definida como timestamp

Teléfono: número de teléfono del cliente

Estado: estado de la llamada (Si fue contestada y resuelta o hay que contactar al cliente por alguna razón)

**Tabla\_Clientes**

ID\_Cliente: definida PK para identificar cada cliente

Nombre: Nombre del cliente

Apellido: Apellido del cliente

Direccion: Direccion del cliente

CodigoPostal: Codigo postal del cliente

**Tabla\_empleados**

ID\_Empleado: definida PK para identificar cada empleado.

Nombre: Nombre del empleado

Apellido: Apellido del empleado

Tienda: lugar de trabajo del empleado

Para asegurar la integridad y la eficiencia de los datos se pueden tomar las siguientes consideraciones:

* Restringir los datos que son ingresados a las tablas, por ejemplo, los ID deben ser números enteros, deben ser únicos y no nulos, las fechas como timestamp y con el formato requerido.
* Usar la información de dos columnas como identificador y definirlas como Primary Key en el caso de la tabla de llamadas hará que las consultas sean más manejables y eficientes.
* Dividir la información en las tres tablas mencionadas facilitará la administración de los datos y mejorará el rendimiento.
* Se podrían utilizar procedimientos almacenados para restringir a los usuarios el acceso directo a las tablas de la base de datos.
* **Pregunta sobre la recopilación y análisis de precios:**

"Como parte de nuestro análisis de mercado, queremos recopilar y analizar los precios de nuestros productos en los sitios web de Amazon.es y Manoamano.es. ¿Cómo abordarías este problema y qué bibliotecas o herramientas utilizarías?"

**Respuesta:**  
Para recopilar los precios de los productos en las páginas mencionadas, emplearía técnicas de web scraping en Python. Utilizaría la biblioteca Selenium para simular interacciones humanas y obtener la página en el formato deseado. Posteriormente, pasaría la URL resultante a la biblioteca BeautifulSoup para extraer la estructura HTML de los elementos relevantes.

Una vez que tenga el HTML, procedería a almacenar la información de los productos en una lista de diccionarios. Para manejar y analizar eficientemente estos datos, emplearía la potente biblioteca Pandas para crear un DataFrame.

En esta etapa, tendría la flexibilidad de realizar análisis exploratorios de datos directamente en Python o, si es necesario, emplear la biblioteca SQLAlchemy para crear una tabla en una base de datos SQL y almacenar los datos recopilados, o podría trabajar conjuntamente desde Python con SQLAlchemy. Este enfoque brinda una solución integral, permitiendo tanto análisis en tiempo real como almacenamiento estructurado para futuras consultas y manipulaciones de datos.

* **Pregunta sobre resolución de problemas generales:**

"Imagina que enfrentamos un desafío técnico en el que necesitamos integrar datos de ventas en tiempo real desde nuestro sitio web con nuestro sistema de gestión de inventario. ¿Cuál sería tu enfoque para abordar esta integración técnica y qué tecnologías o herramientas considerarías para lograrlo?"

**Respuesta:**

Abordar el desafío de integrar datos de ventas en tiempo real desde el sitio web con el sistema de gestión de inventario implica considerar varios aspectos que están directamente vinculados a los requisitos específicos del proyecto. Una de las claves es evaluar el stack tecnológico utilizado.

Una estrategia viable para lograr esta integración es la utilización de una API que facilite la interacción entre el sitio web y las bases de datos. Este enfoque ofrece una solución eficiente para el intercambio de datos en tiempo real. No obstante, requeriría llevar a cabo una investigación más detallada y exhaustiva para desarrollar una API que se adapte perfectamente a los requisitos particulares del proyecto, permitiendo una integración fluida y eficaz entre los datos de ventas generados en tiempo real y el sistema de gestión de inventario.

* **Pregunta sobre habilidades de comunicación y trabajo en equipo:**

"Supongamos que estás trabajando en un proyecto en equipo para implementar una nueva funcionalidad en nuestro sistema de gestión de almacén. Durante una reunión, surge un desacuerdo sobre la mejor manera de abordar un problema técnico. ¿Cómo resolverías esta situación y qué acciones tomarías para garantizar que el proyecto avance de manera efectiva?"

**Respuesta:**

En una primera instancia, sería fundamental escuchar las propuestas de todas las partes involucradas en la funcionalidad del sistema de gestión del almacén, asegurándome que todos se sientan escuchados y respetados, una vez comprendidas todas las perspectivas, trabajaríamos juntos para encontrar un terreno común y buscar una solución que combine lo mejor de las ideas, además, propondría la realización de pruebas, siempre que sea factible. Este enfoque nos permitiría evaluar concretamente cada solución propuesta, asegurándonos de su viabilidad y eficacia.

El principal criterio orientador durante este proceso sería la fecha de entrega del proyecto. Dada su importancia, se priorizaría la elección de soluciones que se alineen de manera eficiente con la línea de tiempo establecida.

En situaciones donde realizar pruebas no sea posible, la toma de decisiones se basaría en la solución propuesta que mejor se ajuste a los objetivos temporales del proyecto. La consideración central sería garantizar el cumplimiento de los plazos establecidos, mientras se mantiene la disposición a mejorar los procesos en el futuro si es necesario.

Una vez que se haya llegado a una decisión, me aseguraría de que todos estén alineados y comprometidos con la acción acordada para garantizar que el proyecto avance de manera efectiva y se complete con éxito.