# Evaluación Analista de Desarrollo Xplora 2024

**Transact-SQL y Procedimientos Almacenados:**

**Dada una tabla Sales con las columnas SaleID, ProductID, CustomerID, SaleDate, Amount, crea un procedimiento almacenado que:**

* Calcule las ventas totales por cliente para un período de fechas determinado.
* Permita actualizar la cantidad de una venta específica.
* Devuelva la lista de ventas ordenada por cantidad en orden descendente.

***Creación de Vistas:***

**Crea una vista en SQL Server que muestre el total de ventas por producto y por cliente en el último año, agrupando por ProductID y CustomerID.**

**Desarrollo Frontend con Angular y Angular Material :**

**Desarrolla una aplicación en Angular 15 que permita a los usuarios gestionar sus perfiles.**

**Requerimientos:**

* Implementa un formulario de edición de perfil usando Angular Material, con validaciones en tiempo real.
* Los perfiles deben incluir campos para Nombre, Email, Teléfono, y Dirección.
* Utiliza Bootstrap para asegurar que el formulario sea responsive y se vea bien en diferentes tamaños de pantalla.
* Asegúrate de que la aplicación cumpla con las mejores prácticas de accesibilidad.

**Validación y Manejo de Errores:**

**Agrega manejo de errores en la aplicación para capturar y mostrar mensajes de error cuando falle la conexión al backend o se ingresen datos inválidos.**

**Desarrollo de una API REST:**

**Desarrolla una API REST en .NET 6 que permita la gestión de productos.**

* Implementa los endpoints para CRUD (Crear, Leer, Actualizar, Eliminar) de productos utilizando DAPPER para la gestión de datos.
* Asegúrate de que los endpoints incluyan validaciones para evitar la inserción de datos incorrectos (por ejemplo, que el precio de un producto no sea negativo).
* Implementa un sistema de logging para registrar los errores que ocurran durante las operaciones CRUD.

**Gestión de Incidencias:**

**Simula una incidencia reportada por un cliente: "Los usuarios no pueden actualizar su perfil después de la última actualización del sistema."**

* Analiza el problema y propone una solución.
* Documenta el proceso de identificación, resolución y pruebas de la incidencia.

**Atención de Requerimientos:**

**Un cliente ha solicitado una nueva funcionalidad en la API: la capacidad de buscar productos por nombre con búsqueda parcial (similar a "LIKE" en SQL).**

* Añade un nuevo endpoint en la API para realizar esta búsqueda.
* Asegúrate de que la búsqueda sea eficiente y esté bien documentada.

**1. Análisis del Problema**

**Reporte Inicial:**

* **Descripción:** Los usuarios han reportado que no pueden actualizar su perfil tras la última actualización del sistema.
* **Prioridad:** Alta
* **Impacto:** Afecta a todos los usuarios que tratan actualizar su perfil.

**Paso 1: Reproducir el Error**

* **Acción:** Intentar actualizar el perfil de un usuario en un entorno de pruebas o en producción.
* **Resultado:** Al intentar guardar los cambios, se muestra un mensaje de error o no se actualiza el perfil sin ningún mensaje.

**Paso 2: Revisión de Logs**

* **Acción:** Revisar los logs del sistema para identificar posibles errores.
* **Resultado:** Se encuentra una excepción en los logs que indica un problema de validación o un error en la consulta a la base de datos.

**Paso 3: Análisis del Código**

* **Acción:** Revisar el código relacionado con la actualización del perfil, incluyendo el controlador, el servicio y toda lógica de validación.
* **Resultado:** Se encuentra que la validación de ciertos campos es demasiado restrictiva o que el código de la consulta a la base de datos no está manejando correctamente los datos nulos o vacíos.

**2. Propuesta de Solución**

**Identificación del Problema:**

* **Causa:** Después de la última actualización, se agregaron nuevas validaciones para los campos del perfil, pero no se consideraron ciertos escenarios en los que los datos pueden ser opcionales o estar en un formato diferente al esperado.
* **Impacto Específico:** Las validaciones adicionales están rechazando cambios legítimos o la consulta SQL está fallando debido a datos inesperados.

**Solución Propuesta:**

1. **Ajuste de Validaciones:** Modificar la lógica de validación para que sea menos restrictiva o para que maneje adecuadamente los valores opcionales.
2. **Revisión de Consultas SQL:** Verificar y corregir cualquier consulta SQL que pueda estar fallando debido a la estructura de datos modificada.
3. **Pruebas Automatizadas:** Implementar pruebas unitarias y de integración que cubran estos casos específicos de actualización de perfil para evitar problemas similares en el futuro.

**3. Documentación del Proceso**

**3.1. Identificación del Problema**

* **Fecha de Incidencia:** 22/08/2024
* **Descripción:** Usuarios reportan que no pueden actualizar sus perfiles después de una actualización.
* **Logs:** Se detectó una excepción relacionada con la validación de campos del perfil de usuario.

**3.2. Resolución del Problema**

* **Análisis:** Se identificó que las nuevas validaciones y/o consultas SQL eran demasiado restrictivas.
* **Acciones Tomadas:** Se ajustaron las reglas de validación y se revisaron las consultas SQL para manejar adecuadamente los valores opcionales y diferentes formatos de datos.

**3.3. Pruebas Realizadas**

* **Pruebas Unitarias:** Se añadieron pruebas unitarias para verificar que las validaciones permiten la actualización de perfil en todos los escenarios posibles.
* **Pruebas de Integración:** Se realizaron pruebas de integración para asegurar que la funcionalidad completa del perfil sigue operando correctamente.
* **Pruebas en Producción:** Se realizó una prueba en un entorno de pre-producción para verificar la solución antes de desplegar en producción.

**3.4. Despliegue**

* **Fecha de Despliegue:** 23/08/2024
* **Responsable:** Rafhael Pillaca Pariona
* **Notas:** Se recomienda monitorizar los logs después del despliegue para asegurar que no surgen nuevos problemas.

**4. Pruebas Posteriores**

**Acción:** Tras aplicar la solución, realizar pruebas exhaustivas en los entornos de desarrollo, staging, y producción.

* **Resultado Esperado:** Los usuarios deben poder actualizar sus perfiles sin problemas, y los logs deben estar libres de errores relacionados.
* **Verificación de Clientes:** Contactar con los usuarios que reportaron el problema para confirmar que se ha solucionado.

**Optimización de Consultas:**

**Se te proporciona una consulta SQL que realiza operaciones complejas en una base de datos grande con varias tablas relacionadas: Customers, Orders, OrderDetails, y Products. La consulta original busca obtener el total gastado por cada cliente en productos de la categoría "Electronics" durante el año 2023. La consulta incluye múltiples uniones (JOIN), filtrado por fecha y categoría, agrupación por cliente, y ordenación por el total gastado en orden descendente.**

SELECT

C.CustomerID,

C.CustomerName,

SUM(O.TotalAmount) AS TotalSpent

FROM

Customers C

JOIN

Orders O ON C.CustomerID = O.CustomerID

JOIN

OrderDetails OD ON O.OrderID = OD.OrderID

JOIN

Products P ON OD.ProductID = P.ProductID

WHERE

O.OrderDate BETWEEN '2023-01-01' AND '2023-12-31'

AND P.Category = 'Electronics'

GROUP BY

C.CustomerID,

C.CustomerName

ORDER BY

TotalSpent DESC;

* Analiza la consulta anterior y describe los problemas potenciales que podrían surgir en términos de rendimiento
* Optimiza la consulta original para mejorar su rendimiento. Considera la creación de índices adecuados, el uso de subconsultas para filtrar datos tempranamente, y la reducción de la cantidad de datos intermedios procesados.

Solución:  
Problemas Potenciales :

* Uso de múltiples join sin con tablas que no estén indexadas.
* Join innecesario con la tabla products cuando podría hacerce el join directamente usando una subconsulta donde en ella se haga un join entre la tabla OrderDetails y la tabla Products y a la vez filtrar por la columna Category directamente en ella.
* Filtrado tardio a la tabla Orders por medio de las fechas, y también filtrado tardio a la tabla Category .

Creación de índices en columnas utilizadas para el filtrado en las consultas :   
CREATE INDEX IX\_Orders\_OrderDate ON Orders(OrderDate);

CREATE INDEX IX\_Products\_Category ON Products(Category);

CREATE INDEX IX\_OrderDetails\_ProductID ON OrderDetails(ProductID);

CREATE INDEX IX\_Orders\_CustomerID ON Orders(CustomerID);

Consulta Optimizada :  
  
SELECT C.CustomerID

, C.CustomerName

, SUM(O.TotalAmount) AS TotalSpent

FROM Customers C

JOIN (

SELECT O.CustomerID

, O.OrderID

, O.TotalAmount

FROM Orders O

WHERE O.OrderDate BETWEEN '2023-01-01' AND '2023-12-31'

) O

ON C.CustomerID = O.CustomerID

JOIN (

SELECT OD.OrderID

, OD.ProductID

FROM OrderDetails OD

JOIN Products P

ON OD.ProductID = P.ProductID

WHERE P.Category = 'Electronics'

) OD

ON O.OrderID = OD.OrderID

GROUP BY C.CustomerID

, C.CustomerName

ORDER BY TotalSpent DESC;