**Prueba técnica para desarrollador de software**

Nivel: .net, angular, plsql

Duración: 1 horas

Parte teórica

* **¿Qué son los procedimientos almacenados y las funciones en PL/SQL? ¿Cuál es la diferencia entre ellos?**

Son bloques de código que ejecutan operaciones, por ejemplo: Actualizar registros, realizar tareas de validaciones, realizar cálculos, etc.

Tiene como función sólo la ejecución del código pero no retorna valores.

Las funciones también ejecutan operaciones pero en este caso se retorna un valor que puede ser usado en alguna otra porción de código.  
  
De acuerdo con esto, la diferencia principal es que las funciones devuelven un valor que puede ser usado posteriormente y asignarlo a variables, mientras que los procedimientos almacenados sólo se ejecutan pero no retornan valores.

* **¿Indique un ejemplo de cómo usar un cursor en PL/SQL para recorrer los resultados de una consulta y mostrarlos en la consola que incluya manejo de excepciones y parámetros de entrada?**

Como ejemplo de cursor, vamos a seleccionar de una tabla de clientes sus Ids, todos aquellos que son hombres   
  
DECLARE

    gender CHAR := 'M';

    CURSOR customer\_cursor IS

        SELECT CUSTOMER\_ID

        FROM CUSTOMERS

        WHERE GENDER = gender;

    v\_customer\_id CUSTOMERS.CUSTOMER\_ID%TYPE;

BEGIN

    OPEN customer\_cursor;

    LOOP

        FETCH customer\_cursor INTO v\_customer\_id;

        EXIT WHEN customer\_cursor%NOTFOUND;

        DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || v\_customer\_id);

    END LOOP;

    CLOSE customer\_cursor;

EXCEPTION

    WHEN OTHERS THEN

        DBMS\_OUTPUT.PUT\_LINE('Ha ocurrido un error: ' || SQLERRM);

END;

* **¿indique un disparador que se active después de insertar una fila en una tabla y actualice otra tabla relacionada.?**

Para esto se tiene como ejemplo las siguientes tablas   
   
SALES

| ID | PRODUCT\_ID | QUANTITY\_SOLD | SALE\_DATE |
| --- | --- | --- | --- |
| 1 | 101 | 5 | 2024-09-01 |
| 2 | 102 | 3 | 2024-09-01 |

PRODUCTS

| ID | PRODUCT\_NAME | INVENTORY |
| --- | --- | --- |
| 101 | APPLE A | 100 |
| 102 | APPLE B | 50 |

CREATE OR REPLACE TRIGGER update\_inventory

AFTER INSERT ON SALES

FOR EACH ROW

BEGIN

UPDATE PRODUCTS

SET INVENTORY = INVENTORY - :NEW.QUANTITY\_SOLD

WHERE PRODUCT\_ID = :NEW.PRODUCT\_ID;

END;

/  
  
TABLAS DESPUES DE LA INSERCIÓN

SALES

| ID | PRODUCT\_ID | QUANTITY\_SOLD | SALE\_DATE |
| --- | --- | --- | --- |
| 1 | 101 | 5 | 2024-09-01 |
| 2 | 102 | 3 | 2024-09-01 |
| 3 | 101 | 10 | 2024-09-01 |

PRODUCTS

| ID | PRODUCT\_NAME | INVENTORY |
| --- | --- | --- |
| 101 | APPLE A | 90 |
| 102 | APPLE B | 50 |

La cantidad en inventario del producto con PRODUCT\_ID 101 se ha reducido en 10 unidades, pasando de 100 a 90, después de insertar una nueva venta para ese producto.

* **¿Qué es la recursividad en PL/SQL y cómo se implementa?**

Es un procedimiento o función que se llama así mismo para resolver un problema, dividiéndolos en problemas más simples   
  
Se implementa de la siguiente manera:   
  
**A. caso base:** Condición que impide que la función o procedimiento continúe llamándose a sí misma indefinidamente.

**B. Caso recursivo:** Punto donde la función o procedimiento se llama a sí misma con nuevos parámetros , resolviendo una versión reducida del problema original , cada llamada se acerca al caso base

C. **Ejemplo de Implementación:**

Imprimir los números de forma descendente desde n hasta 1.

CREATE OR REPLACE FUNCTION count\_down(n IN NUMBER) IS

BEGIN

IF n < 1 THEN

RETURN;

ELSE

DBMS\_OUTPUT.PUT\_LINE(n);

count\_down(n - 1);

END IF;

END;

/

Caso Base: Se detiene cuando es n es 1

Caso Recursivo: Se llama a sí misma para ir decrementando los números.

* **¿Qué son las transacciones autónomas en PL/SQL y en qué casos son útiles?**

Las transacciones autónomas es un proceso que se ejecuta por sí solo, independientemente de otro proceso que esté en curso. Son útiles cuando se necesita realizar operaciones que deben ser independientes a la transición principal . Por ejemplo, para registrar errores o logs sin que el resultado de estos registros dependa del éxito o fracaso de la transacción principal.

* **¿Qué son los cursores dinámicos en PL/SQL y en qué se diferencian de los cursores estáticos?**

Los cursores dinámicos son consultas cuyo contenido no se conoce hasta el momento de la ejecución, son consultas que pueden variar y deben construirse de acuerdo a la lógica del programa en tiempo real.   
  
La diferencia es que los cursores estáticos son consultas fijas que no cambian durante la ejecución , la consulta se indica cuando se declara el cursor y no se puede modificar mientras se está ejecutando

* **¿Qué técnicas de optimización de rendimiento se pueden aplicar en el desarrollo de código,**

**PL/SQL?**Pueden utilizar diferente técnicas, como optimización de consulta SQL, Minimizar las llamadas a DB, declarar sólo las variables necesarias y utilizar el tipo de dato correcto para minimizar el uso de memoria, manejo de excepciones, minimizar el tamaño de los bloques , optimización de recursos y como se menciona en respuestas anterior, usando recursividad para no usar código de forma repetitiva.

Parte práctica

Duración: 4 horas

Objetivo: Evaluar las capacidades lógicas, de resolución de problemas, de ejecución limpia en

código y de integración de APIs RESTful entre un Backend y Frontend.

Enunciado:

Se le pide al candidato que desarrolle una aplicación web que permita a los usuarios controlar un

ascensor. La aplicación debe tener las siguientes características:

1. El Backend debe proporcionar una API RESTful para controlar el ascensor.

2. El Frontend debe usar la API RESTful para interactuar con el Backend.

3. La aplicación debe permitir al usuario:

4. Llamar al ascensor a un piso específico.

5. Abrir y cerrar las puertas del ascensor.

6. Iniciar y detener el ascensor.

Requisitos técnicos:

El Backend debe estar escrito en .net

El Frontend debe estar escrito en angular js

La API RESTful debe seguir las convenciones de RESTful.

El uso de Motor de base de datos es opcional.

A tener en cuenta:

La aplicación debe simular a varias personas llamando el ascensor en diferentes pisos y el ascensor

debe responder a las peticiones, según su orden, y teniendo en cuenta que si pasa por un piso

solicitado (no cumpliendo el orden de solicitud) debe parar en ese piso.

| **CONTROL DE DOCUMENTOS** | | | | |
| --- | --- | --- | --- | --- |
| **FECHA** | **ELABORADO** | **REVISADO** | **APROBADO** | **CONTROL DE CAMBIOS** |
| 22/03/2023 | Directora Talento Humano | Directora HSEQ | Directora HSEQ | Versión 01 |
| 29/07/2024 | Directora Talento Humano | Directora HSEQ | Directora HSEQ | Versión 02 Clasificación tipo de información |