 Grupo Bancolombia Capital	Proceso	Prueba Técnica
	Gerencia	Gerencia Servicio Negocios Fiduciarios
	Sección	Sección Mejoramiento Negocios Fiduciarios

Generalidades:

Como parte del proceso de evaluación de conocimientos que estás realizando para la Sección Mejoramiento GSNF te invitamos a realizar la siguiente prueba técnica, en la cual evaluaremos conocimientos en SQL, Java (POO), análisis de problemas y desarrollo de interfaces gráficas mediante ejercicios completamente prácticos. Se requiere que el candidato tome capturas de pantalla del proceso y entregue el código fuente al finalizar.

Instrucciones Generales:

- Todos los ejercicios deben realizarse de manera práctica.
- Se debe tomar capturas de pantalla de cada paso clave.
- Al finalizar la prueba, se debe entregar el código fuente con comentarios explicativos.
- Posterior a la entrega, el candidato será citado a una sustentación donde deberá explicar su solución y responder preguntas sobre su implementación.

Sección 1: SQL

Ejercicio 1: Consulta SQL

Dada la siguiente estructura de base de datos:


```
CREATE TABLE Clientes (
    id INT PRIMARY KEY,
    nombre VARCHAR(100),
    email VARCHAR(100),
    fecha_registro DATE
);

CREATE TABLE Pedidos (
    id INT PRIMARY KEY,
    cliente_id INT,
    monto DECIMAL(10,2),
    fecha_pedido DATE,
    FOREIGN KEY (cliente_id) REFERENCES Clientes(id)
);
```

1. Escribe una consulta SQL para obtener el nombre de cada cliente con el total de dinero que ha gastado en pedidos. Solo incluir clientes que han hecho al menos un pedido.
2. Agrega una condición para mostrar solo los clientes que hayan gastado más de 1000.
3. Ejecuta la consulta en un entorno SQL y toma una captura de pantalla con los resultados.

Atención

La información contenida en este documento es propiedad de BANCOLOMBIA S.A. Y no debe ser reproducida, utilizada o divulgada al exterior de la organización o con propósitos externos a los de la organización sin la autorización respectiva.

 Grupo Bancolombia Capital	Proceso	Prueba Técnica
	Gerencia	Gerencia Servicio Negocios Fiduciarios
	Sección	Sección Mejoramiento Negocios Fiduciarios

Ejercicio 2: Corrección de consulta

Se tiene la siguiente consulta:

```
SELECT nombre, SUM(monto)
FROM Clientes, Pedidos
WHERE Clientes.id = Pedidos.cliente_id;
```

1. Identifica y explica los errores en la consulta.
2. Corrige la consulta para que funcione correctamente.
3. Ejecuta la consulta corregida y toma una captura de pantalla con los resultados.

Sección 2: Java con Programación Orientada a Objetos

Ejercicio 1: Modelado de clases

Diseña una clase en Java para representar un "Producto" con los siguientes atributos:

- ID (entero)
- Nombre (cadena)
- Precio (decimal)
- Cantidad en stock (entero)

Incluye:

- Constructor con todos los atributos
- Métodos getters y setters
- Un método para calcular el valor total del stock del producto
- Implementa la clase en un entorno de desarrollo, prueba su funcionamiento y toma una captura de pantalla con los resultados.

Ejercicio 2: Análisis de código


Analiza el siguiente código en Java:

```
public class Cliente {
    private String nombre;
    private int edad;

    public Cliente(String nombre, int edad) {
        nombre = nombre;
    }
}
```

Atención

La información contenida en este documento es propiedad de BANCOLOMBIA S.A. Y no debe ser reproducida, utilizada o divulgada al exterior de la organización o con propósitos externos a los de la organización sin la autorización respectiva.

 Grupo Bancolombia Capital	Proceso	Prueba Técnica
	Gerencia	Gerencia Servicio Negocios Fiduciarios
	Sección	Sección Mejoramiento Negocios Fiduciarios

```

        edad = edad;
    }

    public void mostrarInformacion() {
        System.out.println("Cliente: " + nombre + ", Edad: " + edad);
    }
}

```

1. Identifica los errores y explica por qué ocurren.
2. Corrige el código y proporciona una versión funcional.
3. Ejecuta el código corregido y toma una captura de pantalla con la salida en consola.

Sección 3: Análisis de Problemas

Ejercicio 1: Lógica de programación

Escribe un algoritmo en Java que reciba una lista de enteros y devuelva el segundo número más grande de la lista.

Ejemplo:

```

Entrada: [4, 1, 7, 3, 9]
Salida: 7

```

1. Implementa el algoritmo en Java y prueba diferentes entradas.
2. Toma capturas de pantalla de la ejecución con distintos casos de prueba.

Ejercicio 2: Resolución de problema

Se está desarrollando un sistema de reservas para un cine. Un cliente solo puede reservar si hay al menos 2 asientos disponibles en la misma fila. Diseña un método en Java que reciba una matriz 2D de asientos (1 = ocupado, 0 = libre) y devuelva si un cliente puede hacer una reserva.

Ejemplo:


```

Entrada:
[
  [1, 0, 0, 1],
  [0, 1, 1, 0],
  [0, 0, 0, 0]
]
Salida: true (Hay una fila con al menos dos asientos libres juntos)

```

Atención

La información contenida en este documento es propiedad de BANCOLOMBIA S.A. Y no debe ser reproducida, utilizada o divulgada al exterior de la organización o con propósitos externos a los de la organización sin la autorización respectiva.

 Grupo Bancolombia Capital	Proceso	Prueba Técnica
	Gerencia	Gerencia Servicio Negocios Fiduciarios
	Sección	Sección Mejoramiento Negocios Fiduciarios

1. Implementa la solución en Java.
2. Prueba la solución con distintos escenarios.
3. Toma capturas de pantalla con los resultados.

Sección 4: Desarrollo de Interfaz Gráfica y Generación de Archivos

Ejercicio 1: Creación de una Interfaz Gráfica

Desarrolla una aplicación en Java con Swing o JavaFX que contenga:

- Un diseño visual atractivo con un logo de la empresa.
- Un formulario donde el usuario pueda ingresar información de un producto (nombre, precio, cantidad en stock).
- Un botón para guardar la información ingresada en memoria.
- Ejecuta la aplicación y toma capturas de pantalla mostrando la interfaz en funcionamiento

Ejercicio 2: Generación de Archivo XLSX con Contraseña

Modifica la aplicación del ejercicio anterior para que, al presionar el botón de guardar, se genere un archivo Excel (.xlsx) con la información ingresada.

- El archivo debe estar bloqueado con una contraseña definida en el código.
- Usa la librería Apache POI para la generación del archivo Excel.
- Genera el archivo y toma una captura de pantalla mostrando su contenido bloqueado con la contraseña.

Entregable:

1. Todos los archivos de código fuente deben ser entregados en un repositorio ZIP o GitHub.
2. Adjuntar un documento con las capturas de pantalla tomadas durante el proceso.
3. Se recomienda agregar comentarios en el código explicando su funcionamiento.
4. Posterior a la entrega, el candidato será citado a una sustentación donde deberá explicar su solución y responder preguntas sobre su implementación.

Atención

La información contenida en este documento es propiedad de BANCOLOMBIA S.A. Y no debe ser reproducida, utilizada o divulgada al exterior de la organización o con propósitos externos a los de la organización sin la autorización respectiva.