



Prueba de desempeño Java

Nombre: _____

Team Leader: Kevin Wilfred Mejía Torres

Clan: Meta

Fecha: 07/04/2024

Reglas de la prueba :

- **Hora de finalización de la prueba - 1 :45 pm**
- **Comunicación:** Está prohibido hablar o comunicarse de cualquier forma con otros coders durante el examen, el team leader no atenderá dudas tecnicas, solo dudas logicas sobre el examen.
- **Integridad Académica:** Cualquier forma de trampa, incluido el plagio, copiado o uso de material no autorizado, resultará en una calificación de cero en el examen y puede llevar a sanciones adicionales según las políticas de RIWI.
- **Material Permitido:** Solo se permite ver material de apoyo como lo son diapositivas o ejercicios realizados en clase, si se requiere ver documentación oficial se debe avisar al team leader y con su respectiva supervisión.
- **Permanencia en el Aula:** Una vez iniciado el examen, no se permite salir del aula hasta haber entregado el examen y, de preferencia, hasta que haya transcurrido al menos la mitad del tiempo asignado.
- **Entrega:** Una vez finalizada la prueba se cargar a la plataforma moodle un archivo txt donde se encuentre el link del repositorio con el código del desarrollo.

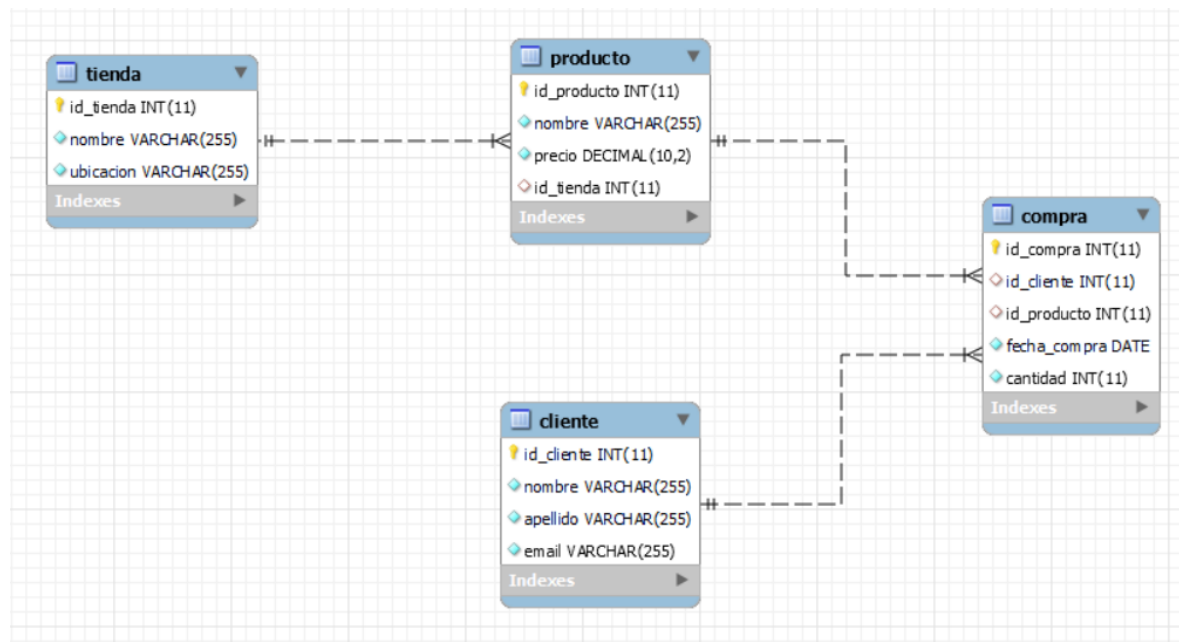
Recuerda :

« la disciplina tarde o temprano vencerá la inteligencia »

Enunciado del problema.

El centro comercial De Modal Outlet necesita que desarrolles una aplicación en Java utilizando JDBC que permita gestionar las operaciones del centro comercial. La aplicación debe permitir la administración de productos, clientes y compras. Cada compra estará asociada a un solo producto y puede tener múltiples clientes a través de las compras. Los clientes pueden realizar múltiples compras.

Modelo entidad Relacion





1. Creación de la Base de Datos y Tablas:

- Utiliza MySQL para crear una base de datos para el centro comercial.
- Define y crea las tablas para Tienda, Producto, Cliente, y Compra siguiendo el modelo de entidad-relación proporcionado, asegurándote de incluir la restricción ON DELETE CASCADE en las claves foráneas adecuadas. Esto garantizará que las eliminaciones en las tablas padres (como Tienda o Cliente) resulten en la eliminación automática de los registros relacionados (Productos o Compras, respectivamente).
- Mediante una alteracion a la tabla Producto agregar el campo stock como numerico.

2. Operaciones CRUD

Implementa las operaciones CRUD para cada una de las entidades (Producto, Cliente, Compra) utilizando JDBC:

- **Create:** Permite añadir nuevos productos, clientes y compras a la base de datos.
- **Read:** Consulta registros existentes, ademas se deben hacer los siguientes filtros : busca productos por nombre o tienda, clientes por nombre, todas las compras de un producto.
- **Update:** Actualiza registros, por ejemplo, cambia el precio de un producto, los datos de un cliente, o la cantidad de una compra.
- **Delete:** Elimina registros, por ejemplo, un producto, un cliente, o una compra.

3. Validaciones Específicas:

- Asegúrate de que las compras no superen el stock disponible de un producto.



- Implementa una lógica para actualizar el stock de los productos después de cada compra.
- Cuando se realice una compra se debe mostrar una factura con la información del producto vendido (nombre, precio), de la tienda (nombre, ubicación) y del cliente (nombre, apellido, email) además de un precio total el cual se debe calcular con el precio base del producto + el 19% (Precio con IVA).

4. Criterios de aceptación:

- El sistema debe estar desarrollado en Java con el paquete JDBC.
- Se debe subir el código a un repositorio de GitHub. Tenga en cuenta que se revisará el último commit realizado antes de la hora y fecha límite (1:45pm).
- Se debe subir al repositorio el script SQL de las creaciones de las tablas (Exportar la base de datos).
- El sistema debe estar desarrollado con el paquete de Java, swing (JOptionPane, JPanel).
- Las tiendas se deben llenar con información directamente en la base de datos.

Rubrica prueba de desempeño	
Prueba de desempeño 60%	
Funcionalidad de la aplicación	25%
Criterios de aceptación	20%
Buenas practicas de código	10%
Implementacion Modelo, Controlador, Entidad	5%
Total	60%