

INSTITUTO DE EDUCACIÓN SECUNDARIA

ISIDRA DE GUZMÁN



ANEXOS PRÁCTICA DE EVALUACIÓN CONTINUA (PEC)

Acceso a Datos (JDBC)

Alumno/a: Diego Luengo Gil
Grupo: 2º DAM
Módulo: Acceso a Datos
Profesor/a: Ruth Lospitao Ruiz
Fecha: 21/11/2025
Curso académico: 2025/2026

Índice

Índice.....	2
1. Modelo	2
Cliente	3
Producto.....	3
Venta	4
DetalleVenta	4
2. Repositorios (Data Access).....	5
ClienteRepository.....	5
ProductoRepository	6
VentaRepository.....	6
3. Servicios (Business Logic).....	7
ClienteService.....	7
ProductoService	7
VentaService	8
4. Controladores (Controller).....	8
MainController	8
ClienteController	8
ProductoController	9
VentaController.....	9
5. Vistas (Views)	9
MainView	9
ClienteView	9
ProductoView	10
VentaView	10
6. Utilidades y Excepciones.....	10
JPAUtil	11
Util.....	11
Excepciones.....	11

1. Modelo

Clases que representan los datos persistentes del sistema.

Cliente

Campo	Tipo	Visibilidad	Descripción
id	int	private	Identificador único (PK).
nombre	String	private	Nombre completo.
email	String	private	Correo electrónico.
telefono	String	private	Teléfono de contacto.
dinero	double	private	Saldo disponible.

Método	Retorno	Visibilidad	Descripción
getId() / setId(int)	int / void	public	Getter/Setter ID.
getNombre() / setNombre(String)	String / void	public	Getter/Setter Nombre.
getEmail() / setEmail(String)	String / void	public	Getter/Setter Email.
getTelefono() / setTelefono(String)	String / void	public	Getter/Setter Teléfono.
getDinero() / setDinero(double)	double / void	public	Getter/Setter Dinero.

Producto

Campo	Tipo	Visibilidad	Descripción
id	int	private	Identificador único (PK).
nombre	String	private	Nombre del producto.
descripcion	String	private	Descripción detallada.
precio	double	private	Precio unitario.

stock	int	private	Cantidad en inventario.	
Método		Retorno	Visibilidad	Descripción
getId() / setId(int)		int / void	public	Getter/Setter ID.
getNombre() / setNombre(String)		String / void	public	Getter/Setter Nombre.
getDescripcion() / setDescripcion(String)		String / void	public	Getter/Setter Descripción.
getPrecio() / setPrecio(double)		double / void	public	Getter/Setter Precio.
getStock() / setStock(int)		int / void	public	Getter/Setter Stock.

Venta

Campo		Visibilidad	Descripción	
id		private	ID de la venta.	
cliente		private	Cliente asociado.	
detalles		private	Lista de detalles.	
Método		Retorno	Visibilidad	Descripción
getId() / setId(int)		int / void	public	Getter/Setter ID.
getCliente() / setCliente(Cliente)		Cliente / void	public	Getter/Setter Cliente.
getDetalles() / setDetalles(List)		List / void	public	Getter/Setter Detalles.
addDetalle(DetalleVenta)		void	public	Agrega detalle a la lista.
removeDetalle(DetalleVenta)		void	public	Elimina detalle de la lista.

DetalleVenta

Campo	Tipo	Visibilidad	Descripción
id	int	private	ID del detalle.
venta	Venta	private	Venta asociada.
producto	Producto	private	Producto asociado.
cantidad	int	private	Cantidad vendida.
precioUnitario	double	private	Precio al momento de venta.
Método	Retorno	Visibilidad	Descripción
getVenta() / setVenta(Venta)	Venta / void	public	Getter/Setter Venta.
getProducto() / setProducto(Producto)	Producto / void	public	Getter/Setter Producto.
getCantidad()	int	public	Obtiene cantidad.
getPrecioUnitario()	double	public	Obtiene precio.

2. Repositorios (Data Access)

Clases encargadas de la persistencia y comunicación con la base de datos vía JPA.

ClienteRepository

Método	Retorno	Visibilidad	Descripción
existeCliente(int)	boolean	public	Verifica existencia por ID.
agregarDinero(int, double)	boolean	public	Suma saldo transaccionalmente.
restarDinero(int, double)	boolean	public	Resta saldo transaccionalmente.
contarClientes()	int	public	Cuenta total de clientes.

buscarPorCampo(String, Object)	List<Cliente>	public	Busca dinámicamente por campo.
eliminarCliente(int)	boolean	public	Elimina cliente por ID.
actualizarClienteCampo(int, String, Object)	boolean	public	Actualiza un campo específico.
obtenerClientes()	List<Cliente>	public	Retorna todos los clientes.
insertarCliente(Cliente)	boolean	public	Persiste un nuevo cliente.

ProductoRepository

Método	Retorno	Visibilidad	Descripción
contarProductos()	int	public	Cuenta total de productos.
buscarPorCampo(String, Object)	List<Producto>	public	Busca dinámicamente por campo.
actualizarProductoCampo(int, String, Object)	boolean	public	Actualiza un campo específico.
eliminarProducto(int)	boolean	public	Elimina producto por ID.
obtenerProductos()	List<Producto>	public	Retorna todos los productos.
insertarProducto(Producto)	boolean	public	Persiste nuevo producto.

VentaRepository

Método	Retorno	Visibilidad	Descripción
realizarVentaCompleta(int, List<DetalleVenta>)	void	public	Transacción compleja (Venta + Detalles + Stock + Saldo).
obtenerResumenVentas()	List<String>	public	Consulta resumen formateado.

obtenerDetallesVenta(int)	List<String>	public	Consulta detalles formateados.
calcularTotalVenta(int)	double	public	Calcula total sumando detalles.

3. Servicios (Business Logic)

Capa intermedia que contiene la lógica de negocio y orquesta los repositorios.

ClienteService

Atributo	Tipo	Descripción
clienteRepository	ClienteRepository	Dependencia de datos.
Método	Descripción	
listarClientes()	Llama a repo para obtener todos.	
buscarClientePorId(int)	Busca por campo ID.	
insertarCliente(Cliente)	Valida y llama a insertar.	
actualizarCliente(int, String, Object)	Llama a actualizar campo.	
eliminarCliente(int)	Llama a eliminar.	
agregarDinero(int, double)	Llama a operación saldo.	
restarDinero(int, double)	Llama a operación saldo.	

ProductoService

Atributo	Tipo	Descripción
productoRepository	ProductoRepository	Dependencia de datos.
Método	Descripción	
listarProductos()	Llama a repo para obtener todos.	
buscarProductoPorId(int)	Busca por campo ID.	
insertarProducto(Producto)	Llama a insertar.	
actualizarProducto(int, String, Object)	Llama a actualizar campo.	

eliminarProducto(int)	Llama a eliminar.
-----------------------	-------------------

VentaService

Atributo	Tipo	Descripción
ventaRepository	VentaRepository	Dependencia de datos.
Método	Descripción	
realizarVenta(int, List)	Ejecuta transacción de venta.	
listarResumenVentas()	Obtiene resumen.	
listarDetallesVenta(int)	Obtiene detalles.	
obtenerTotalVenta(int)	Obtiene total.	

4. Controladores (Controller)

Gestionan el flujo de la aplicación, interactuando entre Vistas y Servicios.

MainController

Atributo	Tipo	Descripción
mainView	MainView	Vista principal.
productoController	ProductoController	Sub-controlador.
clienteController	ClienteController	Sub-controlador.
ventaController	VentaController	Sub-controlador.
Método	Descripción	
iniciar()	Bucle principal de la aplicación.	

ClienteController

Método	Descripción
iniciar()	Menú de gestión de clientes.
listarClientes()	Muestra listado.

<code>insertarCliente()</code>	Pide datos e inserta.
<code>buscarClientePorId()</code>	Pide ID y busca.
<code>eliminarCliente()</code>	Pide ID y elimina.
<code>actualizarCliente()</code>	Menú de actualización.
<code>agregarDinero()</code>	Gestión de saldo.
<code>restarDinero()</code>	Gestión de saldo.

ProductoController

Método	Descripción
<code>iniciar()</code>	Menú de gestión de productos.
<code>listarProductos()</code>	Muestra listado.
<code>buscarProductoPorId()</code>	Pide ID y busca.
<code>insertarProducto()</code>	Pide datos e inserta.
<code>actualizarProducto()</code>	Menú de actualización.
<code>eliminarProducto()</code>	Pide ID y elimina.

VentaController

Método	Descripción
<code>menuConsultas()</code>	Menú de consultas de ventas.
<code>crearVenta()</code>	Flujo principal: selección cliente, selección productos, cobro.

5. Vistas (Views)

Clases encargadas de la interacción con el usuario (entrada/salida por consola).

MainView

Método	Descripción
<code>mostrarMenuPrincipal()</code>	Muestra opciones generales.
<code>mostrarMensaje(String)</code>	Imprime mensaje genérico.

ClienteView

Método	Descripción
mostrarMenu()	Menú CRUD clientes.
mostrarClientes(List)	Tabla de clientes.
pedirDatosCliente()	Formulario de nuevo cliente.
pedirIdCliente()	Input de ID.
pedirOpcionActualizar()	Submenú campos.

ProductoView

Método	Descripción
mostrarMenu()	Menú CRUD productos.
mostrarProductos(List)	Tabla de productos.
pedirDatosProducto()	Formulario nuevo producto.
pedirIdProducto()	Input de ID.
pedirOpcionActualizar()	Submenú campos.

VentaView

Método	Descripción
mostrarMenuConsultas()	Menú consultas.
mostrarResumenVentas(List)	Tabla resumen.
mostrarDetallesVenta(...)	Detalle factura.
pedirIdVenta()	Input ID.
pedirCantidad(int)	Input cantidad con max.
pedirSiNo(String)	Confirmación S/N.
mostrarError(String)	Mensaje de error (syserr).

6. Utilidades y Excepciones

JPAUtil

Método	Descripción
getEntityManager()	Obtiene un EntityManager nuevo.
close()	Cierra la factoría emf.

Util

Método	Descripción
pedirNumero()	Lectura segura de enteros.
pedirTexto()	Lectura de cadenas no vacías.
pedirDecimal()	Lectura de doubles.
pedirSiNo()	Selección binaria.

Excepciones

1. **DatoInvalidoException**: Checked Exception para validaciones de dominio (datos vacíos, negativos, etc.).
2. **SaldoInsuficienteException**: Checked Exception específica para fallos en el proceso de cobro de una venta.

**INSTITUTO DE EDUCACIÓN SECUNDARIA
ISIDRA DE GUZMÁN**

