

INF275-10156 - SEMINARIO DE PRÁCTICA DE INFORMATICA

Trabajo Práctico N°2

Rolando Andrés Palermo

DNI: 33.311.173 / Legajo: VIN013533

Profesor: PABLO ALEJANDRO VIRGOLINI

19 de Mayo de 2024

01 OBJETIVO

Esta actividad busca que seas capaz de plantear una solución problemática que pueda resolverse mediante la realización de un proyecto informático.

Para llevar adelante este desafío, vas a poder aplicar muchos de los conceptos más importantes abordados durante el desarrollo del módulo 2, que recupera tu trabajo en materias troncales de la carrera.

Durante el proceso, lograrás aplicar tus conocimientos para alcanzar los siguientes objetivos:

- Reconocer y aplicar el proceso unificado de desarrollo.
- Utilizar UML para representar y comunicar los aspectos del sistema.
- Diseñar un modelo de datos relacional y realiza un diagrama entidad-relación.
- Emplear consultas SQL para gestionar una base de datos.
- Plantear los requerimientos de comunicación del sistema.

En este segundo TP deberás retomar los requerimientos planteados para el sistema a desarrollar y avanzar con el proyecto.

02 SITUACIÓN PROBLEMÁTICA

Lee con atención el siguiente caso teniendo presente cada uno de los contenidos que hemos desarrollado en el módulo. Una vez leído, tendrás que resolver preguntas cerradas en base al mismo. Haz clic sobre el siguiente enlace para descargar el enunciado:

[Click aquí](#)

03 CONSIGNA

Considera que continúas trabajando en la organización cuyo problema definiste resolver con un proyecto informático, y estás en el equipo de desarrollo que va a llevar adelante el mismo.

Recuerda que el proyecto informático definido debe ser distinto al considerado en la lectura y que el entregable final debe cumplir con los siguientes objetivos:

Realizar el análisis del modelo de negocios para definir y justificar el proyecto.

Aplicar el proceso unificado de desarrollo (PUD) para garantizar la calidad, escalabilidad y eficiencia en el ciclo de desarrollo.

Utilizar una base de datos MySQL para la persistencia de los datos.

Emplear Java como lenguaje de programación para el desarrollo del sistema.

A los fines del trabajo, para la implementación de la base de datos y el desarrollo con Java, puedes presentar un prototipo, que es “es la creación de un modelo operacional que incluya solo algunas características del sistema final” (Kendall & Kendall, 2011).

El concepto de operacional es clave, ya que no se trata de un simple modelo, sino que permite desarrollar módulos que se van integrando en la versión final del sistema.

Esta segunda actividad te permitirá aplicar de forma completa los modelos de análisis, diseño, implementación y pruebas del PUD. Es fundamental asegurar la trazabilidad del sistema, seleccionando adecuadamente los artefactos a presentar en cada etapa, utilizando correctamente el lenguaje unificado de desarrollo (UML) para representar y comunicar los diferentes aspectos del sistema.

A continuación, deberás diseñar un modelo de datos relacional que represente las entidades, atributos y relaciones específicas del sistema. Es importante aplicar los principios de normalización para asegurar que las tablas en la base de datos estén estructuradas de manera eficiente y plantear la realización de un diagrama entidad-relación. Recuerda asegurar que las estructuras de datos estén alineadas con el diseño del sistema y cumplan con los requisitos de integridad y rendimiento.

Finalmente, podrás definir cuáles son los protocolos y estándares adecuados para la comunicación entre el sistema y otros sistemas externos o la base de datos: entorno de red, infraestructura física, control de enlace de datos, etc.

Los entregables a presentar son los siguientes:

- Etapa de análisis.
- Etapa de diseño.
- Etapa de implementación.
- Etapa de pruebas.
- Definición de base de datos para el sistema.
- Diagrama entidad-relación de la base de datos.
- Creación de las tablas MySQL.
- Inserción, consulta y borrado de registros.
- Presentación de las consultas SQL.
- Definiciones de comunicación.

Es importante que puedas asegurar tu participación activa en actividades propuestas: foros, encuentros sincrónicos, aportes en espacios comunes y comunicación. En la rúbrica de la última actividad hay un criterio específico que evalúa estos aspectos.

Te recomiendo apoyarte en las lecturas y bibliografía propuesta en las materias: Análisis y Diseño de Software, Base de Datos I, Comunicaciones.

Referencia:

Kendall, K., & Kendall, J. (2011). Análisis y diseño de sistemas. Pearson Education.

04 FORMATO ENTREGABLE

La actividad propuesta debe ser elaborada en forma individual y respetar los criterios de presentación mencionados a continuación. Se debe realizar una entrega incremental, considerando las correcciones realizadas en la entrega previa.

El informe se debe presentar en un documento en formato PDF (papel A4, letra Calibri 11, espaciado simple). Debe incluir las respuestas a las consignas de forma prolija y ordenada, con su correspondiente explicación para justificar las decisiones adoptadas. Incluir una portada y respetar el siguiente criterio para el nombre del archivo:

«APELLIDO-NOMBRE»-AP2.PDF

La realización de todos los diagramas requiere utilizar una herramienta de libre elección, pero asegurando la prolijidad y claridad de presentación. Todos los diagramas deben incluirse en el documento a entregar con su correspondiente explicación.

Para la persistencia de los datos, deberás utilizar una base de datos MySQL, y para el desarrollo del prototipo, una IDE de Java. La explicación debe estar en el documento a entregar e incluir un enlace a la versión completa (código y archivos asociados) en Github: <https://github.com/Links to an external site..>

Se evaluará la precisión en las respuestas, capacidad de síntesis, redacción, ortografía y calidad de la presentación.

En caso de requerirlo, es importante revisar el cumplimiento de las normas de estilo American Psychological Association (APA).

03 RESOLUCIÓN

Proyecto Informático: Sistema de Gestión de Ventas para SQ Deportes

Etapa de Análisis

Modelo de Negocios:

Contexto:

SQ Deportes es una tienda que vende artículos deportivos. El sistema actual de gestión de ventas y stock es manual, lo que genera ineficiencias y errores en la gestión de inventarios, pedidos y ventas. Se propone desarrollar un sistema informático que automatice estas tareas.

Objetivos del Proyecto:

Automatizar el proceso de ventas, desde la consulta de productos hasta la facturación.

Gestionar el inventario de manera eficiente, manteniendo un registro actualizado del stock.

Facilitar la generación de reportes de ventas y stock.

Mejorar la experiencia del cliente proporcionando información precisa y rápida.

Justificación:

La implementación de este sistema reducirá errores humanos, optimizará el tiempo de atención al cliente, mejorará la gestión del inventario y proporcionará información en tiempo real para la toma de decisiones.

Etapa de Diseño

Casos de Uso

CU001 - Cliente consulta al vendedor disponibilidad de un producto (Producto disponible)

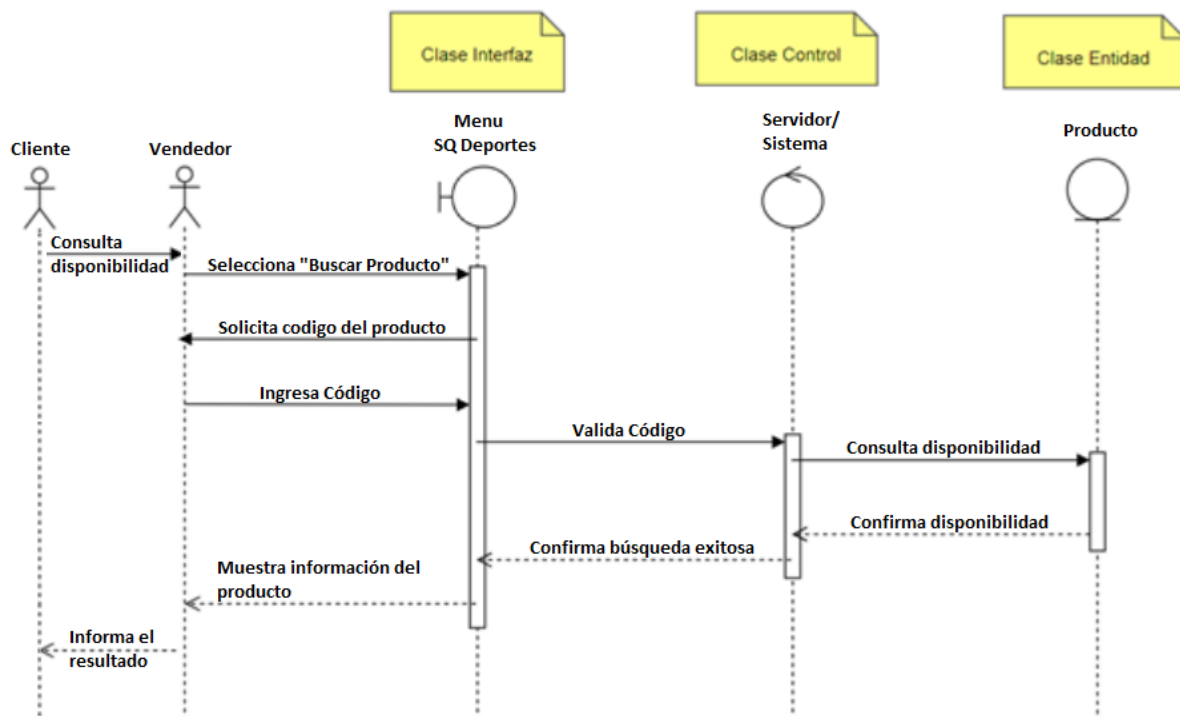
Actores: Cliente, Vendedor

Descripción: Permite al cliente obtener información detallada sobre un producto.

Flujo Principal: El cliente consulta al vendedor disponibilidad de un producto.

- El vendedor selecciona "Buscar producto".
- El sistema pide ingreso del código de producto.
- El vendedor ingresa el código
- El sistema Valida el código
- El sistema consulta disponibilidad de stock a la clase producto.
- La clase Producto confirma disponibilidad.
- El sistema muestra la información del producto.
- El vendedor proporciona la información al cliente.

Diagrama de secuencia correspondiente al CU001



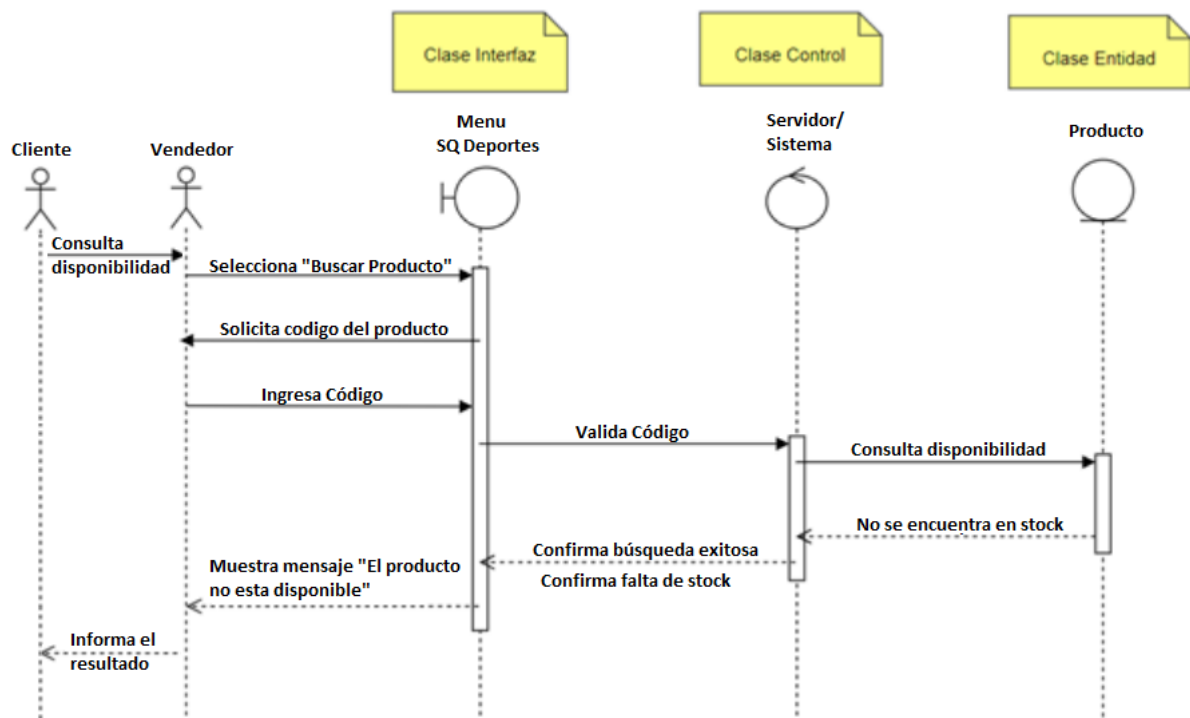
Flujo Alternativo:

CU001(Alternativo) - Cliente consulta al vendedor disponibilidad de un producto (Producto NO disponible).

- El vendedor selecciona "Buscar producto".
- El sistema pide ingreso del código de producto.
- El vendedor ingresa el código
- El sistema Valida el código
- El sistema consulta disponibilidad de stock a la clase producto.
- La clase Producto informa que no hay disponibilidad.
- El sistema notifica al vendedor que el producto no está disponible.
- El vendedor informa al cliente que el producto no está disponible.

Diagrama de secuencia correspondiente al CU001(Alternativo)

En este caso damos por hecho que la búsqueda fue exitosa solo que en la cantidad disponible del producto es 0 lo cual refleje en el diagrama como "Confirma falta de stock" y procede a mostrar un mensaje de "Producto no disponible".



(Los siguientes casos de uso no se diagramaron para no hacer tan extenso el trabajo practico).

CU0002 - Vendedor genera pedido de compra

Actores: Vendedor

Descripción: Permite al vendedor generar un pedido de compra.

Flujo Principal:

- El vendedor selecciona "Generar Pedido de Compra".
- El sistema solicita los detalles del pedido.
- El vendedor completa los detalles del pedido.
- El sistema valida la información.
- El sistema Registra el pedido
- El pedido genera una confirmación.
- El sistema Notifica al vendedor que el pedido fue realizado con éxito.

CU0003 - Cajero realizar el cobro de los productos

Actores: Cajero

Descripción: Permite al cajero realizar el cobro de los productos.

Flujo Principal:

- El cajero selecciona "Realizar Cobro"
- El sistema solcicita datos del cliente
- El cajero ingresa los datos del cliente
- El sistema muestra el pedido de compra.
- El sistema solicita método de pago

- El sistema Procesa el pago.
- El sistema genera la factura.
- El cajero entrega la factura al cliente.

Etapa de diseño

Descripción General

El diagrama de clases muestra la estructura estática del sistema, incluyendo las clases principales, sus atributos, métodos y las relaciones entre ellas.

Clases Principales:

1- Personas(Padre de Cliente, Empleado, Proveedor)

- PK - idPersonas
- nombre
- dirección
- teléfono
- localidad
- provincia

2- Cliente

- PK -idCliente
- DNI
- cuentaCorriente
- Metodos:
- +altaCliente
- +bajaCliente
- +modificarCliente
- +visualizarCliente
- +listarClientes

3- Empleado

- PK -idEmpleado
- CUIL
- rol(cajero, vendedor, administrador)
- Metodos:
- +altaEmpleado
- +bajaEmpleado
- +modificarEmpleado
- +visualizarEmpleado
- +listarEmpleados

4- Proveedor

- PK -idProveedor
- CUIT
- Metodos:
- +altaProveedor

- +bajaProveedor
- +modificarProveedor
- +visualizarProveedor
- +listarProveedores

5- Producto

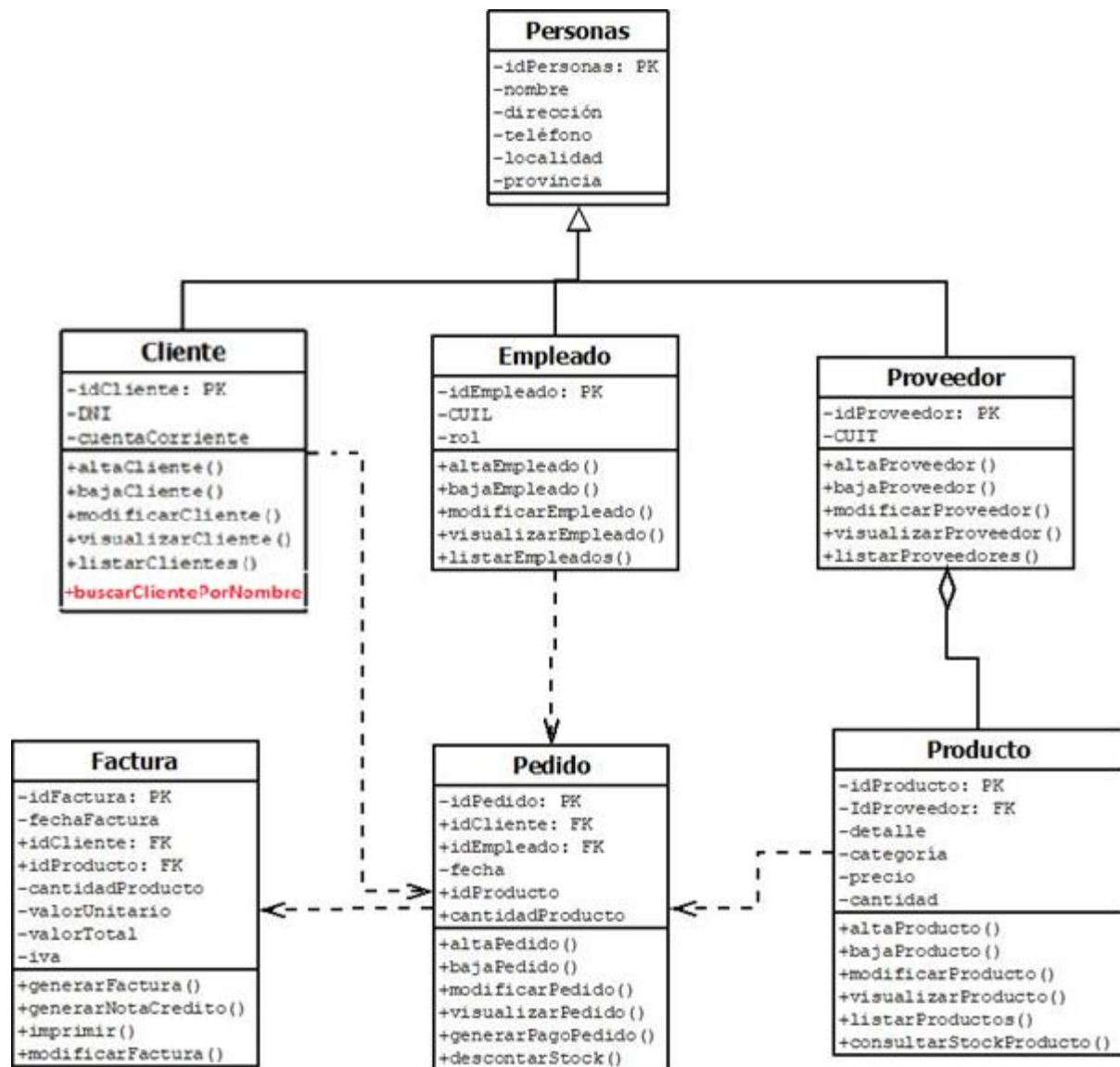
- PK - idProducto
- FK - IdProveedor
- detalle
- categoría
- precio
- cantidad
- Metodos:
- +altaProducto
- +bajaProducto
- +modificarProducto
- +visualizarProducto
- +listarProductos
- +consultarStockProducto

6- Pedido

- PK - idPedido
- FK - idCliente
- FK - idEmpleado
- fecha
- idProducto
- cantidadProducto
- Metodos:
- +altaPedido
- +bajaPedido
- +modificarPedido
- +visualizarPedido
- +generarPagoPedido
- +descontarStock

7- Factura

- idFactura
- fechaFactura
- idCliente
- idProducto
- cantidadProducto
- valorUnitario
- valorTotal
- iva
- Metodos:
- +generarFactura
- +generarNotaCredito
- +imprimir
- +modificarFactura



Subsistema de Diseño

El subsistema de diseño incluye los componentes y módulos del sistema, destacando cómo se organiza la funcionalidad del sistema en subsistemas lógicos.

Componentes del Subsistema de Diseño

Gestión de Clientes:

- Módulos: Alta, Baja, Modificación, Visualización, Listado de Clientes.

Gestión de Empleados:

- Módulos: Alta, Baja, Modificación, Visualización, Listado de Empleados.

Gestión de Proveedores:

- Módulos: Alta, Baja, Modificación, Visualización, Listado de Proveedores.

Gestión de Productos:

- Módulos: Alta, Baja, Modificación, Visualización, Listado de Productos, Consulta de Stock.

Gestión de Pedidos:

- Módulos: Alta, Baja, Modificación, Visualización, Generación de Pago, Descuento de Stock.

Gestión de Facturas:

- Módulos: Generación de Factura, Generación de Nota de Crédito, Impresión, Modificación de Factura.

Estos componentes y módulos forman la base del sistema propuesto, proporcionando una estructura clara y organizada para la implementación del sistema de gestión de ventas para SQ Deportes.

Etapa de Implementación

Requisitos del Sistema

Requerimientos No Funcionales (RNF):

RNF01: El sistema debe estar desarrollado en Java.

RNF02: El sistema debe contar con una base de datos MySQL.

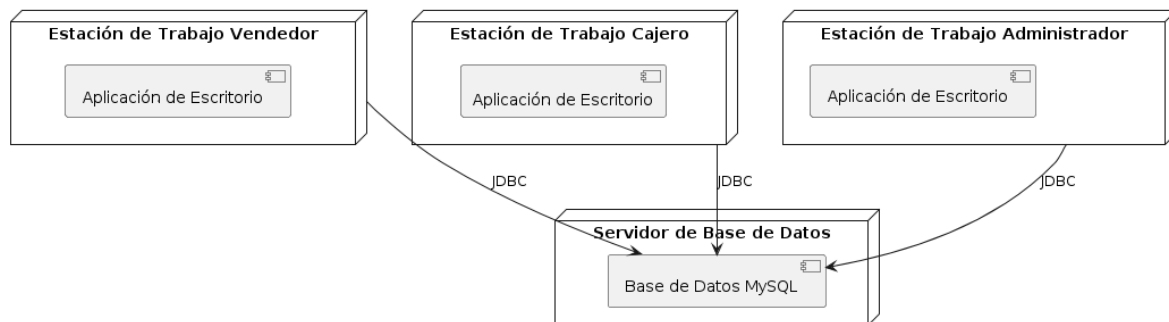
RNF03: El sistema debe utilizar XAMPP como entorno de desarrollo para gestionar la base de datos.

Servidor de Base de Datos: Este nodo contiene la base de datos MySQL, gestionada por XAMPP. La base de datos almacena toda la información relacionada con clientes, empleados, proveedores, productos, pedidos y facturas.

Estación de Trabajo Vendedor: Los vendedores utilizan una aplicación de escritorio desarrollada en Java para realizar consultas de productos, gestionar pedidos y manejar inventarios. La aplicación se conecta directamente a la base de datos MySQL a través de JDBC (Java Database Connectivity).

Estación de Trabajo Cajero: Los cajeros utilizan la misma aplicación de escritorio para realizar cobros, gestionar facturas y procesar pagos. Esta aplicación también se conecta a la base de datos MySQL a través de JDBC.

Estación de Trabajo Administrador: Los administradores utilizan la aplicación de escritorio para gestionar la información de clientes, empleados, proveedores y generar reportes, conectándose a la base de datos MySQL mediante JDBC.



Etapa de Pruebas

En esta sección, se presenta un plan de pruebas detallado adaptado al proyecto de gestión de ventas para SQ Deportes. Se abordarán los casos de uso seleccionados previamente y se definirán las pruebas correspondientes, junto con los criterios de aceptación y el tratamiento de defectos.

Plan de Pruebas

El plan de pruebas incluirá pruebas unitarias, de integración y del sistema, centrándose en asegurar que el sistema funcione correctamente de acuerdo con los requisitos definidos. A continuación, se detalla el plan de pruebas para los casos de uso CU001, CU002 y CU003.

Tabla 1: Plan de Pruebas

Caso de Uso	Código Prueba	Tipo de Prueba	Técnica Propuesta	Observaciones
CU001	CU001C01	Componente	Cobertura	Aplicación de caja blanca al método consultarDisponibilidad() de la clase Producto.
CU001	CU001F02	Componente	Partición de equivalencia-frontera	Aplicación de caja negra al método consultarDisponibilidad() de la clase Producto.
CU001	CU001S03	Sistema	Pruebas funcionales	Verificación de requerimientos funcionales para el caso de uso.
CU002	CU002C01	Componente	Cobertura	Aplicación de caja blanca al método generarPedido() de la clase Pedido.
CU002	CU002F02	Componente	Partición de equivalencia-frontera	Aplicación de caja negra al método generarPedido() de la clase Pedido.
CU002	CU002S03	Sistema	Pruebas funcionales	Verificación de requerimientos funcionales para el caso de uso.
CU003	CU003C01	Componente	Cobertura	Aplicación de caja blanca al método realizarCobro() de la clase Factura.

CU003	CU003F02	Componente	Partición de equivalencia-frontera	Aplicación de caja negra al método realizarCobro() de la clase Factura.
CU003	CU003S03	Sistema	Pruebas funcionales	Verificación de requerimientos funcionales para el caso de uso.

Casos de Prueba

A continuación, se presenta un caso de prueba específico para el caso de uso CU001, aplicando la técnica de partición de equivalencia y análisis de frontera al método consultarDisponibilidad() de la clase Producto.

Tabla 2: Requerimientos

Requerimiento	Descripción
RFS01	El sistema debe mostrar un mensaje de alerta si el producto no está disponible.
RFS02	El sistema debe mostrar un mensaje informativo si el producto está disponible.

Particiones de Equivalencia Válidas

Clase de Equivalencia Válida	Descripción
CEV1	$0 \leq \text{stock}$
CEV2	$0 \leq \text{stock}$

Particiones de Equivalencia Inválidas

Clase de Equivalencia Inválida	Descripción
CEI1	$\text{stock} < 0$

Análisis de Valores de Frontera

Tabla 3: Análisis de Valores de Frontera

Clase de Equivalencia	Límite Frontera	Límite Inferior	Límite Superior
CEV1	0	-1	1
CEV2	1	0	2
CEI1	-1	-2	0

Comportamiento del Sistema

Tabla 4: Comportamiento del Sistema

Dato de Entrada	Comportamiento Esperado	Mensaje Emitido por el Sistema
-2	Error, valor no válido.	Error: stock no puede ser negativo.
-1	Error, valor no válido.	Error: stock no puede ser negativo.
0	Producto no disponible.	Producto no disponible.
1	Producto disponible.	Producto disponible: 1 en stock.
2	Producto disponible.	Producto disponible: 2 en stock.

Procedimiento de Prueba

El procedimiento de prueba detalla los pasos a seguir para ejecutar cada caso de prueba. A continuación, se describe el procedimiento para el caso de prueba CU001F02.

Tabla 5: Procedimiento de Prueba para CU001F02

Paso	Descripción
1	Iniciar la aplicación de escritorio y acceder a la funcionalidad de búsqueda de productos.
2	Ingresar el código del producto a consultar.
3	Ejecutar el método consultarDisponibilidad().
4	Verificar el mensaje emitido por el sistema en base a los datos de entrada definidos.
5	Registrar los resultados observados y compararlos con los resultados esperados.

Tratamiento de Defectos

Tabla 6: Tratamiento de Defectos

ID Defecto	Descripción del Defecto	Prioridad	Estado	Responsable	Fecha
D001	El sistema permite valores negativos para el stock.	Alta	Abierto	Equipo Dev	2024-06-01
D002	Mensaje incorrecto cuando el stock es 0.	Media	Abierto	Equipo QA	2024-06-01
D003	La aplicación se cierra inesperadamente al consultar stock.	Crítica	Abierto	Equipo Dev	2024-06-01

Evaluación de Pruebas

Tabla 7: Evaluación de Pruebas

ID Prueba	Resultado Esperado	Resultado Observado	Estado	Observaciones
CU001F02	Producto no disponible.	Producto no disponible.	Aprobado	Comportamiento conforme a lo esperado.

CU002F02	Pedido generado con éxito.	Producto no disponible.	Aprobado	Comportamiento conforme a lo esperado.
CU003F02	Factura generada y cobro realizado.	Error al generar factura.	Fallido	Error al procesar el pago.

Este plan de pruebas proporciona una guía detallada para verificar y validar la funcionalidad del sistema de gestión de ventas para SQ Deportes. Las pruebas cubren tanto componentes individuales como el sistema completo, asegurando que el software cumpla con los requisitos definidos y funcione correctamente en diversos escenarios.

Definición de Base de Datos para el sistema

Para el prototipo del sistema de gestión de ventas para SQ Deportes, se utilizará una base de datos relacional MySQL. Esta elección se basa en los requerimientos no funcionales que destacan el buen rendimiento, la flexibilidad y la escalabilidad de MySQL, lo que lo hace adecuado para el desarrollo ordenado del proyecto completo. La base de datos será fundamental para la persistencia de datos clave, como inventario de productos, registros de ventas, información de clientes, proveedores y usuarios.

Objetivos de la Base de Datos:

Almacenar datos relacionados con productos, ventas, clientes, proveedores y usuarios.
 Mantener un historial completo de las actividades y transacciones realizadas en el sistema.
 Facilitar la generación de informes para analizar el rendimiento de las ventas y el inventario.
 Permitir la identificación de tendencias, problemas y oportunidades para mejorar la eficiencia del negocio.

Diseño del Prototipo de Base de Datos:

Modelo Vista Controlador (MVC):

Se aplicará el patrón de diseño Modelo Vista Controlador para organizar la estructura del sistema. Las clases asociadas al modelo se traducirán en tablas de la base de datos.

Representación de Clases en Tablas:

Cada clase del modelo será representada por una tabla en la base de datos.
 Se incluirán tablas para productos, clientes, empleados, pedidos, proveedores y usuarios, entre otras entidades relevantes.
 Relaciones entre Tablas:

Se establecerán relaciones entre las diferentes tablas para reflejar la estructura de datos del sistema. Por ejemplo, una venta estará relacionada con productos y clientes, y un producto estará relacionado con proveedores.

Tablas del Prototipo de Base de Datos:

Tabla de Productos:

Contendrá información sobre los productos disponibles en la tienda, como nombre, descripción, precio y cantidad en stock.

Tabla de Clientes:

Almacenará datos de los clientes, incluyendo nombre, dirección, número de teléfono y dirección de correo electrónico.

Tabla de Empleados:

Registrará información de los empleados de la tienda, como nombre, cargo, número de empleado y dirección de correo electrónico.

Tabla de Pedidos:

Mantendrá un registro de los pedidos realizados por los clientes, incluyendo detalles como fecha, cliente, empleado responsable y estado del pedido.

Tabla de Facturas:

Almacenará información sobre las facturas generadas para los pedidos, incluyendo detalles como fecha de emisión, cliente, productos vendidos, total y estado de pago.

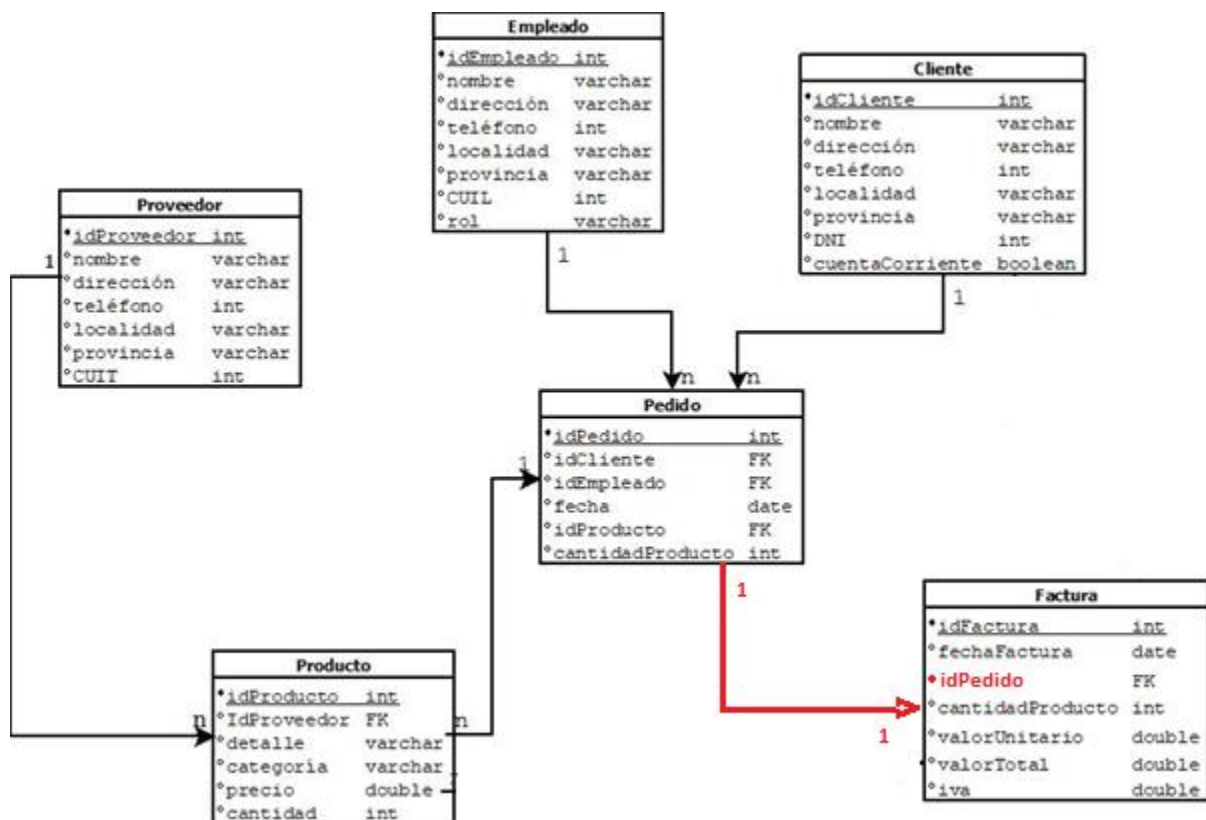
Tabla de Proveedores:

Registrará información sobre los proveedores de la tienda, como nombre, dirección, número de contacto y productos suministrados.

Tabla de Usuarios:

Mantendrá un registro de los usuarios del sistema, incluyendo nombre de usuario, contraseña y nivel de acceso.

Diagrama entidad-relación



Creación de las tablas MySQL.

Modelo de Datos Relacional

```
CREATE DATABASE sqdeportes;
USE sqdeportes;
```

-- Tabla Proveedores

```
CREATE TABLE Proveedores (
    idProveedor INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    direccion VARCHAR(255),
    telefono INT(20),
    localidad VARCHAR(100),
    provincia VARCHAR(100),
    CUIT INT(20)
);
```

-- Tabla Productos

```
CREATE TABLE Productos (
    idProducto INT AUTO_INCREMENT PRIMARY KEY,
    detalle VARCHAR(100) NOT NULL,
```

```
categoria VARCHAR(100),  
precio DECIMAL(10, 2) NOT NULL,  
cantidad INT NOT NULL,  
idProveedor INT,  
FOREIGN KEY (idProveedor) REFERENCES Proveedores(idProveedor)  
);
```

-- Tabla Clientes

```
CREATE TABLE Clientes (  
idCliente INT AUTO_INCREMENT PRIMARY KEY,  
nombre VARCHAR(100) NOT NULL,  
direccion VARCHAR(255),  
telefono INT(20),  
localidad VARCHAR(100),  
provincia VARCHAR(100),  
DNI INT(20),  
cuentaCorriente BOOLEAN  
);
```

-- Tabla Empleados

```
CREATE TABLE Empleados (  
idEmpleado INT AUTO_INCREMENT PRIMARY KEY,  
nombre VARCHAR(100) NOT NULL,  
direccion VARCHAR(255),  
telefono INT(20),  
localidad VARCHAR(100),  
provincia VARCHAR(100),  
rol VARCHAR(50),  
CUIL INT(20)  
);
```

-- Tabla Pedidos

```
CREATE TABLE Pedidos (  
idPedido INT AUTO_INCREMENT PRIMARY KEY,  
fechaPedido DATE NOT NULL,  
cantidad INT(50) NOT NULL,  
idCliente INT,  
idEmpleado INT,  
idProducto INT,  
FOREIGN KEY (idCliente) REFERENCES Clientes(idCliente),  
FOREIGN KEY (idEmpleado) REFERENCES Empleados(idEmpleado),  
FOREIGN KEY (idProducto) REFERENCES Productos(idProducto)  
);
```

-- Tabla Facturas

```
CREATE TABLE Facturas (  
idFactura INT AUTO_INCREMENT PRIMARY KEY,  
fechaFactura DATE NOT NULL,  
idCliente INT,
```

```
idProducto INT,
cantidadProducto INT(50) NOT NULL,
valorunitario DECIMAL(10, 2) NOT NULL,
valortotal DECIMAL(10, 2) NOT NULL,
FOREIGN KEY (idCliente) REFERENCES Clientes(idCliente),
FOREIGN KEY (idProducto) REFERENCES Productos(idProducto)
);
```

Con el código proporcionado, se ejecutó el SQL y creo correctamente las siguientes tablas:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> clientes	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	16.0 KB	-
<input type="checkbox"/> empleados	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	16.0 KB	-
<input type="checkbox"/> facturas	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	32.0 KB	-
<input type="checkbox"/> pedidos	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	48.0 KB	-
<input type="checkbox"/> productos	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	32.0 KB	-
<input type="checkbox"/> proveedores	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	latin1_swedish_ci	16.0 KB	-
6 tablas	Número de filas	0	InnoDB	latin1_swedish_ci	160.0 KB	0 B

Inserción, Consulta y Borrado de Registros

Inserción de Registros

```
INSERT INTO Proveedores (nombre, direccion, telefono, localidad, provincia, CUIT)
VALUES ('Deportes SA', 'Calle Falsa 123', 1234567890, 'Ciudad X', 'Provincia Y', 20304050607),
('Sport Center', 'Avenida Principal 456', 987654321, 'Ciudad W', 'Provincia Z', 30705060809),
('Equipos Deportivos', 'Boulevard Central 789', 1122334455, 'Ciudad V', 'Provincia U', 40906070810);
```

	idProveedor	nombre	direccion	telefono	localidad	provincia	CUIT
<input type="checkbox"/> Editar Copiar Borrar	1	Deportes SA	Calle Falsa 123	1234567890	Ciudad X	Provincia Y	2147483647
<input type="checkbox"/> Editar Copiar Borrar	2	Sport Center	Avenida Principal 456	987654321	Ciudad W	Provincia Z	2147483647
<input type="checkbox"/> Editar Copiar Borrar	3	Equipos Deportivos	Boulevard Central 789	1122334455	Ciudad V	Provincia U	2147483647

```
INSERT INTO Productos (detalle, categoria, precio, cantidad, idProveedor)
VALUES ('Balón de fútbol', 'Deportes', 1500.00, 50, 1),
('Raqueta de tenis', 'Deportes', 2500.00, 30, 2),
('Pesas 5kg', 'Fitness', 300.00, 100, 3);
```

	idProducto	detalle	categoria	precio	cantidad	idProveedor
<input type="checkbox"/> Editar Copiar Borrar	1	Balón de fútbol	Deportes	1500.00	50	1
<input type="checkbox"/> Editar Copiar Borrar	2	Raqueta de tenis	Deportes	2500.00	30	2
<input type="checkbox"/> Editar Copiar Borrar	3	Pesas 5kg	Fitness	300.00	100	3

```
INSERT INTO Clientes (nombre, direccion, telefono, localidad, provincia, DNI, cuentaCorriente)
VALUES ('Juan Pérez', 'Avenida Siempre Viva 742', 1234567890, 'Ciudad Z', 'Provincia A',
30123456789, TRUE),
```

('María García', 'Calle 123', 2345678901, 'Ciudad Y', 'Provincia B', 30234567890, FALSE),
('Pedro López', 'Calle Principal 456', 3456789012, 'Ciudad X', 'Provincia C', 30345678901, TRUE);

←T→		idCliente	nombre	direccion	telefono	localidad	provincia	DNI	cuentaCorriente
<input type="checkbox"/>	Editar Copiar Borrar	1	Juan Pérez	Avenida Siempreviva 742	1234567890	Ciudad Z	Provincia A	2147483647	1
<input type="checkbox"/>	Editar Copiar Borrar	2	María García	Calle 123	2147483647	Ciudad Y	Provincia B	2147483647	0
<input type="checkbox"/>	Editar Copiar Borrar	3	Pedro López	Calle Principal 456	2147483647	Ciudad X	Provincia C	2147483647	1

INSERT INTO Empleados (nombre, direccion, telefono, localidad, provincia, rol, CUIL)
VALUES ('Carlos Gómez', 'Calle 123', 987654321, 'Ciudad Y', 'Provincia B', 'Vendedor', 20123456789),
('Ana Martínez', 'Avenida Central 456', 876543210, 'Ciudad W', 'Provincia C', 'Cajero', 20234567890),
('Luis Fernández', 'Boulevard 789', 765432109, 'Ciudad V', 'Provincia D', 'Gerente', 20345678901);

←T→		idEmpleado	nombre	direccion	telefono	localidad	provincia	rol	CUIL
<input type="checkbox"/>	Editar Copiar Borrar	1	Carlos Gómez	Calle 123	987654321	Ciudad Y	Provincia B	Vendedor	2147483647
<input type="checkbox"/>	Editar Copiar Borrar	2	Ana Martínez	Avenida Central 456	876543210	Ciudad W	Provincia C	Cajero	2147483647
<input type="checkbox"/>	Editar Copiar Borrar	3	Luis Fernández	Boulevard 789	765432109	Ciudad V	Provincia D	Gerente	2147483647

INSERT INTO Pedidos (fechaPedido, cantidad, idCliente, idEmpleado, idProducto)
VALUES ('2024-05-19', 10, 1, 1, 1),
('2024-05-20', 5, 2, 2, 2),
('2024-05-21', 15, 3, 3, 3);

←T→		idPedido	fechaPedido	cantidad	idCliente	idEmpleado	idProducto
<input type="checkbox"/>	Editar Copiar Borrar	1	2024-05-19	10	1	1	1
<input type="checkbox"/>	Editar Copiar Borrar	2	2024-05-20	5	2	2	2
<input type="checkbox"/>	Editar Copiar Borrar	3	2024-05-21	15	3	3	3

INSERT INTO Facturas (fechaFactura, idPedido, cantidadProducto, valorUnitario, valorTotal)
VALUES ('2024-05-19', 1, 10, 150.00, 1500.00),
('2024-05-20', 2, 5, 250.00, 1250.00),
('2024-05-21', 3, 15, 300.00, 4500.00);

←T→		idFactura	fechaFactura	idPedido	cantidadProducto	valorunitario	valortotal
<input type="checkbox"/>	Editar Copiar Borrar	1	2024-05-19	1	10	150.00	1500.00
<input type="checkbox"/>	Editar Copiar Borrar	2	2024-05-20	2	5	250.00	1250.00
<input type="checkbox"/>	Editar Copiar Borrar	3	2024-05-21	3	15	300.00	4500.00

Consulta de Registros

SELECT * FROM Proveedores;

←T→	idProveedor	nombre	direccion	telefono	localidad	provincia	CUIT
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	Deportes SA	Calle Falsa 123	1234567890	Ciudad X	Provincia Y	2147483647
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	Sport Center	Avenida Principal 456	987654321	Ciudad W	Provincia Z	2147483647
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3	Equipos Deportivos	Boulevard Central 789	1122334455	Ciudad V	Provincia U	2147483647

SELECT * FROM Productos;

←T→	idProducto	detalle	categoria	precio	cantidad	idProveedor
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	Balón de fútbol	Deportes	1500.00	50	1
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	Raqueta de tenis	Deportes	2500.00	30	2
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3	Pesas 5kg	Fitness	300.00	100	3

SELECT * FROM Clientes;

←T→	idCliente	nombre	direccion	telefono	localidad	provincia	DNI	cuentaCorriente
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	Juan Pérez	Avenida Siempre Viva 742	1234567890	Ciudad Z	Provincia A	2147483647	1
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	María García	Calle 123	2147483647	Ciudad Y	Provincia B	2147483647	0
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3	Pedro López	Calle Principal 456	2147483647	Ciudad X	Provincia C	2147483647	1

SELECT * FROM Empleados;

...

SELECT * FROM Pedidos;

...

SELECT * FROM Facturas;

...

Consultar pedidos de un cliente específico

SELECT * FROM Pedidos WHERE idCliente = 1;

←T→	idPedido	fechaPedido	cantidad	idCliente	idEmpleado	idProducto
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	2024-05-19	10	1	1	1

Consultar facturas emitidas en una fecha específica

SELECT * FROM Facturas WHERE fechaFactura = '2024-05-20';

←T→	idFactura	fechaFactura	idPedido	cantidadProducto	valorunitario	valortotal
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	2024-05-20	2	5	250.00	1250.00

Consultar productos suministrados por un proveedor específico


SELECT * FROM Productos WHERE idProveedor = 1;

←T→	idProducto	detalle	categoria	precio	cantidad	idProveedor
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	Balón de fútbol	Deportes	1500.00	50	1

Borrado de Registros





Borrar un proveedor específico

DELETE FROM Proveedores WHERE idProveedor = 1;

				idProveedor	nombre	direccion	telefono	localidad	provincia	CUIT			
<input type="checkbox"/>		Editar		Copiar		Borrar	2	Sport Center	Avenida Principal 456	987654321	Ciudad W	Provincia Z	2147483647
<input type="checkbox"/>		Editar		Copiar		Borrar	3	Equipos Deportivos	Boulevard Central 789	1122334455	Ciudad V	Provincia U	2147483647







Borrar un producto específico

DELETE FROM Productos WHERE idProducto = 1;

←T→				idProducto	detalle	categoria	precio	cantidad	idProveedor
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	Raqueta de tenis	Deportes	2500.00	30	2
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	Pesas 5kg	Fitness	300.00	100	3






Borrar un cliente específico

DELETE FROM Clientes WHERE idCliente = 1;

<div>←T→</div>				idCliente	nombre	direccion	telefono	localidad	provincia	DNI	cuentaCorriente
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	María García	Calle 123	2147483647	Ciudad Y	Provincia B	2147483647	0
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	Pedro López	Calle Principal 456	2147483647	Ciudad X	Provincia C	2147483647	1

Borrar un empleado específico

DELETE FROM Empleados WHERE idEmpleado = 1;

←T→		idEmpleado	nombre	direccion	telefono	localidad	provincia	rol	CUIL
<input type="checkbox"/>	 Editar  Copiar  Borrar	2	Ana Martínez	Avenida Central 456	876543210	Ciudad W	Provincia C	Cajero	2147483647
<input type="checkbox"/>	 Editar  Copiar  Borrar	3	Luis Fernández	Boulevard 789	765432109	Ciudad V	Provincia D	Gerente	2147483647

Borrar un pedido específico

DELETE FROM Pedidos WHERE idPedido = 1;

...

Borrar una factura específica

...

DELETE FROM Facturas WHERE idFactura = 1;

Código SQL en github mediante el siguiente link:

<https://github.com/rolandoandres22/tp2-seminario-de-practica.git>

Definiciones de comunicación

Requerimientos de Comunicación del Sistema

A continuación, se detallan los aspectos clave para asegurar una comunicación eficiente y segura entre los componentes del sistema:

1. Entorno de Red

- Topología de Red: Se mantendrá una topología de red estrella, con dispositivos conectados a un switch central.
- Segmentación de Red: Utilización de VLANs para separar el tráfico entre servidores de base de datos y estaciones de trabajo.
- Seguridad de Red: Firewalls y sistemas de detección de intrusos (IDS) para proteger contra accesos no autorizados.

2. Infraestructura Física

- Servidores: Servidores en rack de alta disponibilidad con redundancia en hardware y conexiones de red.
- Estaciones de Trabajo: Computadoras de escritorio robustas y bien mantenidas para los usuarios finales.
- Cableado: Cableado de red de categoría 6 (Cat6) o superior.
- Dispositivos de Red: Switches gestionables de alta capacidad con soporte para QoS.

3. Protocolos de Comunicación

- Entre Aplicaciones de Escritorio y la Base de Datos:
 - MySQL: Utilizar el protocolo nativo de MySQL para la comunicación directa entre la aplicación de escritorio y la base de datos.
 - ODBC/JDBC: Como alternativa para la conexión a la base de datos desde aplicaciones de terceros si es necesario.
- Con Otros Sistemas Externos:
 - SOAP/REST: Para la integración con sistemas externos si la aplicación necesita comunicarse con otros servicios o APIs.
 - Correo Electrónico: SMTP para envío de correos y IMAP/POP3 para recepción.

4. Control de Enlace de Datos

- Ethernet: Utilizar Ethernet (IEEE 802.3) como el protocolo principal para la capa de enlace de datos.
- Wi-Fi: Opcionalmente, para estaciones de trabajo móviles o portátiles.
- Seguridad de Enlace:
 - WPA3: Para redes Wi-Fi, si se utilizan.
 - MAC Address Filtering: Para controlar el acceso a la red.

Interacciones entre los Diferentes Componentes

1. Aplicaciones de Escritorio y Servidor de Base de Datos

- Conexión Directa: Las aplicaciones de escritorio se conectarán directamente al servidor de base de datos MySQL utilizando el protocolo MySQL.
- Autenticación y Autorización: Gestión de usuarios y permisos en MySQL para asegurar que solo usuarios autorizados accedan a los datos.

- Encriptación: Utilizar SSL/TLS para encriptar las conexiones a la base de datos.

2. Integración con Sistemas Externos

- APIs SOAP/REST: Utilizar APIs para integrarse con sistemas externos como sistemas de pago, inventarios, etc.
- Servicios Web: En caso de necesidad de comunicación con servicios web, la aplicación de escritorio puede consumir servicios REST o SOAP.

Protocolos y Estándares

- MySQL: Para la comunicación de base de datos.
- ODBC/JDBC: Como alternativas para la conexión a la base de datos.
- SOAP/REST: Para la integración con sistemas externos.
- SSL/TLS: Para encriptar las conexiones a la base de datos.
- Ethernet/Wi-Fi: Para la comunicación de red.
- SMTP/IMAP/POP3: Para servicios de correo electrónico.
- WPA3: Para la seguridad de redes Wi-Fi.
- IEEE 802.3: Para comunicaciones Ethernet.

Detalle de Comunicación

Conexión de la Aplicación de Escritorio con MySQL

- Configuración del Cliente: Las aplicaciones de escritorio deben tener configurada la dirección IP del servidor MySQL, el puerto de conexión, y las credenciales de acceso.
- Conexión Segura: Configurar MySQL para usar SSL/TLS, asegurando que las conexiones entre las aplicaciones de escritorio y el servidor de base de datos estén encriptadas.

Configuración de la Red

- Segmentación: Usar VLANs para separar el tráfico de las estaciones de trabajo del tráfico del servidor de base de datos y otros servidores.
- QoS: Configurar Quality of Service (QoS) en los switches para priorizar el tráfico crítico de la base de datos.
- Firewall: Configurar reglas de firewall para permitir solo el tráfico necesario entre las estaciones de trabajo y el servidor de base de datos.

Conclusión:

El sistema debe garantizar una comunicación eficiente y segura entre las aplicaciones de escritorio y el servidor de base de datos. La infraestructura de red debe estar bien segmentada y protegida mediante firewalls y VLANs. El uso de SSL/TLS para encriptar las conexiones a la base de datos es crucial para mantener la seguridad de los datos. Además, la integración con sistemas externos puede realizarse mediante APIs SOAP o REST, según sea necesario.