

Manual de Usuario para la Aplicación de Gestión de Empresa de Material Informático

1. Introducción.

- Esta aplicación permite gestionar una empresa de material informático, incluyendo departamentos, empleados, clientes, dispositivos, componentes y sus respectivas compras. Proporciona un menú interactivo para realizar operaciones CRUD (Crear, Leer, Actualizar) sobre estos elementos.

2. Requisitos del Sistema.

- Java JDK 8 o superior.
- Entorno Eclipse IDE (opcional).
- Terminal de comandos para ejecutar la aplicación.

3. Instrucciones de Uso.

3.1. Inicio de la Aplicación.

- Ejecute el archivo AppProyEmpMatInfCRUD.java
- Se mostrará un menú con las opciones disponibles.

3.2. Menú Principal.

- ----- MENÚ -----
 1. Listar departamentos
 2. Insertar departamento
 3. Insertar empleado
 4. Listar empleados
 5. Buscar empleados por departamento
 6. Listar clientes
 7. Insertar cliente
 8. Listar dispositivos
 9. Insertar dispositivo
 10. Listar componentes
 11. Insertar componente
 12. Realizar compra de dispositivo

13. Realizar compra de componente
14. Listar compras de dispositivos
15. Listar compras de componentes
16. Informe de compras por departamento
17. Salir

3.3. Operaciones Disponibles.

- **Departamentos:**
 - **Listar departamentos (Opción 1):** Muestra todos los departamentos registrados.
 - **Insertar departamento (Opción 2):** Solicita ID y nombre para crear un nuevo departamento.
- **Empleados:**
 - **Insertar empleado (Opción 3):** Solicita datos personales y departamento para registrar un nuevo empleado.
 - **Listar empleados (Opción 4):** Muestra todos los empleados registrados.
 - **Buscar empleados por departamento (Opción 5):** Filtra empleados por ID de departamento.
- **Clientes:**
 - **Listar clientes (Opción 6):** Muestra todos los clientes registrados.
 - **Insertar cliente (Opción 7):** Solicita datos personales para registrar un nuevo cliente.
- **Dispositivos:**
 - **Listar dispositivos (Opción 8):** Muestra todos los dispositivos disponibles.
 - **Insertar dispositivo (Opción 9):** Solicita datos técnicos y de inventario para añadir un nuevo dispositivo.
- **Componentes:**
 - **Listar componentes (Opción 10):** Muestra todos los componentes disponibles.
 - **Insertar componente (Opción 11):** Solicita descripción y datos de inventario para añadir un nuevo componente.
- **Compras:**
 - **Realizar compra de dispositivo (Opción 12):** Registra una compra de dispositivo, verificando stock y asociando empleado y cliente.

- **Realizar compra de componente (Opción 13):** Registra una compra de componente, verificando stock y asociando empleado y cliente.
- **Listar compras de dispositivos (Opción 14):** Muestra el historial de compras de dispositivos.
- **Listar compras de componentes (Opción 15):** Muestra el historial de compras de componentes.
- **Informes:**
 - **Informe de compras por departamento (Opción 16):** Genera un reporte de compras agrupadas por departamento.
- **Salir:**
 - **Salir (Opción 17):** Termina la ejecución del programa.

4. Consideraciones Importantes.

- Al insertar datos, asegúrese de que los IDs no estén duplicados.
- Para las compras, verifique que exista suficiente stock antes de realizarlas.
- El formato de fecha para las compras debe ser YYYY-MM-DD.
- Los apellidos secundarios son opcionales (puede dejarlos en blanco).

5. Ejemplos de Uso.

- **Ejemplo 1: Insertar un nuevo departamento.**
 1. Seleccione la opción 2 del menú.
 2. Ingrese el ID del departamento (ej: 1).
 3. Ingrese el nombre del departamento (ej: "Ventas").
 4. El sistema confirmará la inserción.
- **Ejemplo 2: Realizar una compra de dispositivo.**
 1. Seleccione la opción 12 del menú.
 2. Ingrese:
 - a. ID del dispositivo
 - b. ID del cliente
 - c. ID del empleado
 - d. Cantidad
 - e. Fecha (YYYY-MM-DD)
 3. El sistema verificará disponibilidad y registrará la compra.

6. Solución de Problemas.

- **Error de conexión a la base de datos:** Verifique que MySQL está en ejecución y las credenciales sean correctas.
- **Error de stock insuficiente:** Verifique el stock disponible antes de realizar la compra.
- **Error de datos duplicados:** Asegúrese de usar IDs únicos para cada elemento.

7. Arquitectura y Clases del Sistema.

7.1. Clases Principales.

- **Departamento.java**
 - **Propósito:** Representa un departamento de la empresa.
 - **Atributos:**
 - **id:** Identificador único del departamento.
 - **nombre:** Nombre del departamento (máximo 70 caracteres).
 - **Funcionalidades:**
 - Constructor básico y completo.
 - Getters y setters con validación de longitud máxima.
 - Método toString() para representación en texto.
- **DepartamentoDAO.java**
 - **Propósito:** Gestiona la interacción con la tabla DEPARTAMENTO en la base de datos.
 - **Funcionalidades:**
 - Conexión a la base de datos (MySQL).
 - Listar todos los departamentos.
 - Insertar nuevos departamentos.
 - Buscar departamentos por ID.
 - Cierre seguro de conexiones.
- **Dispositivo.java**
 - **Propósito:** Representa un dispositivo informático de la empresa.
 - **Atributos:**
 - **id:** Identificador único.
 - **nombre:** Nombre del dispositivo (máximo 30 caracteres).

- **precio:** Valor monetario.
- **stock:** Cantidad disponible.
- **estado:** Condición del dispositivo.
- **Funcionalidades:**
 - Constructores básico y completo.
 - Getters y setters con validación.
 - Método toString() para visualización.
- **DispositivoDAO.java**
 - **Propósito:** Maneja operaciones con la tabla DISPOSITIVO en la base de datos.
 - **Funcionalidades:**
 - Listar todos los dispositivos.
 - Insertar nuevos dispositivos.
 - Buscar por ID.
 - Actualizar información de dispositivos existentes.
 - Gestión de conexiones a la base de datos.
- **Empleado.java**
 - **Propósito:** Representa a un empleado de la empresa.
 - **Atributos:**
 - **id:** Identificador único.
 - **dni:** Documento de identidad.
 - **nombre:** Nombre (máximo 30 caracteres).
 - **apellido1, apellido2:** Apellidos (máximo 30 caracteres cada uno).
 - **telefono:** Número de contacto.
 - **departamentoId:** Referencia al departamento asignado.
 - **Funcionalidades:**
 - Constructores básico y completo.
 - Getters y setters con validación de longitud.
 - Método toString() para visualización.
- **EmpleadoDAO.java**
 - **Propósito:** Gestiona la interacción con la tabla EMPLEADO en la base de datos.

- **Funcionalidades:**

- Listar todos los empleados.
- Insertar nuevos empleados.
- Buscar por ID.
- Buscar empleados por departamento.
- Obtener información del departamento asociado.
- Gestión de conexiones.

7.2. Configuración de Conexión a Base de Datos.

- Todas las clases DAO comparten la misma configuración:
 - Usuario: root
 - Contraseña: crisil1921
 - Servidor: localhost:3306
 - Base de datos: empresaMaterialInformatico

7.3. Patrones de Diseño Implementados.

- **DAO (Data Access Object):** Separación clara entre la lógica de negocio (clases modelo) y el acceso a datos (clases DAO).
- **Singleton implícito:** Cada DAO gestiona su propia conexión a la base de datos.
- **Encapsulación:** Todos los atributos son privados con acceso controlado mediante getters/setters.

7.4. Buenas Prácticas Implementadas.

- Validación de longitud en setters.
- Uso de PreparedStatement para prevenir SQL injection.
- Cierre seguro de recursos (try-with-resources).
- Documentación Javadoc completa.
- Separación de responsabilidades.

7.5. Limitaciones Conocidas.

- No implementa transacciones para operaciones múltiples.
- No tiene manejo avanzado de errores (solo imprime en consola).
- La conexión se establece al crear el DAO pero no se verifica periódicamente.

8. Recomendaciones para Extensión.

- Añadir métodos de actualización y eliminación en los DAO.
- Implementar un sistema de pooling de conexiones.
- Añadir logs más detallados.
- Implementar transacciones para operaciones críticas.
- Añadir validaciones más complejas en los setters.

9. Notas Finales.

- Este sistema está diseñado para ser intuitivo. Si sigue las instrucciones del menú y proporciona los datos solicitados, podrá gestionar eficientemente la empresa de material informático.