

JOSE LUIS CÁRDENAS BARROSO

GRADO SUPERIOR DE DESARROLLO DE  
APLICACIONES MULTIPLATAFORMA

MÓDULO PROYECTO INTERMODULAR

ANÁLISIS Y DISEÑO

## Sumario

1. Análisis.....	4
1.1. Requisitos Funcionales.....	4
1.2. Requisitos No Funcionales.....	7
1.3. Casos de Uso.....	8
1.3.1. Actores del Sistema.....	8
1.3.2. Casos de Uso Principales.....	8
CU-01: Iniciar Sesión en el Sistema.....	8
CU-02: Gestionar Clientes.....	9
CU-03: Gestionar Vehículos.....	14
CU-04: Gestionar Presupuestos.....	19
CU-05: Generar y Gestionar Facturas.....	25
CU-06: Gestionar Proveedores.....	31
CU-07: Gestionar Suministros/Productos.....	34
CU-08: Gestionar Usuarios (Administrador).....	38
CU-09: Gestionar Reparaciones.....	41
CU-10: Gestionar Empleados.....	57
1.3.3. Matriz de Trazabilidad Requisitos - Casos de Uso.....	68
2. Diseño del Sistema.....	70
2.1. Arquitectura del Software.....	70
2.2. Modelado de Datos.....	71
2.2.1 Modelo Entidad-Relación (MER).....	71
2.2.2 Diagrama de Clases (UML).....	72
2.2.3 Diseño Relacional (Tablas).....	74
2.3. Diseño de Interfaces (Prototipos).....	82
2.3.1 Boceto 1: Ventana de Login.....	82
2.3.2 Boceto 2: Ventana Principal.....	83
2.3.3 Boceto 3: Panel de Gestión de Clientes.....	85
2.3.4 Boceto 4: Panel de Gestión de Vehículos.....	86
2.3.5 Boceto 5: Panel de Gestión de Reparaciones.....	87
2.3.6 Boceto 6: Panel de Gestión de Proveedores.....	88
2.3.7 Boceto 7: Panel de Gestión de Productos.....	89
2.3.8 Boceto 8: Panel de Gestión de Presupuestos.....	90
2.3.9 Boceto 9: Panel de Gestión de Facturas.....	91
2.3.10 Boceto 10: Panel de Empleados.....	92
2.3.11 Boceto 11: Panel de Ayuda.....	93
2.3.12 Boceto 12: Modal Nuevo Presupuesto.....	94
2.3.13 Boceto 13: Modal Nueva Reparación.....	96
2.3.14 Boceto 14: Modal Detalles Reparación.....	97
2.3.15 Boceto 15: Modal Histórico de Cliente.....	98

3. Control de Versiones.....	99
4. Creación del proyecto.....	100
Anexo I: Diagrama de casos de uso.....	102
Anexo II: Diagrama E/R.....	103
Anexo III: Diagrama de Clases.....	104
Anexo IV: Script de Base de Datos.....	105
Anexo V: Repositorio GitHub.....	122

## Índice de figuras

Figura 1: Boceto panel Login.....	82
Figura 2: Boceto Ventana Principal.....	84
Figura 3: Boceto panel Clientes.....	85
Figura 4: Boceto panel Vehículos.....	86
Figura 5: Boceto panel Reparaciones.....	87
Figura 6: Boceto panel Proveedores.....	88
Figura 7: Boceto panel Productos.....	89
Figura 8: Boceto panel Presupuestos.....	90
Figura 9: Boceto panel Facturas.....	91
Figura 10: Boceto panel Empleados.....	92
Figura 11: Boceto panel Ayuda.....	93
Figura 12: Boceto nuevo Presupuesto.....	95
Figura 13: Boceto nueva Reparación.....	96
Figura 14: Boceto Detalles Reparación.....	97
Figura 15: Boceto Histórico de Clientes.....	98
Figura 16: Historial de commits hasta la fecha.....	99
Figura 17: Creación del proyecto y la estructura de clases.....	100

## Índice de tablas

Tabla 1: Requisitos Funcionales.....	4
Tabla 2: Requisitos No Funcionales.....	7
Tabla 3: Trazabilidad Requisitos - Casos de Uso.....	68

# Informe de Análisis y Diseño: APPTaller

AÑO ACADÉMICO:	2025/2026
CICLO:	Desarrollo de Aplicaciones Multiplataforma
ALUMNO/A:	José Luis Cárdenas Barroso
FECHA:	02/01/2026

# 1. Análisis

En esta sección se definen los requisitos del sistema.

## 1.1. Requisitos Funcionales

*Tabla 1: Requisitos Funcionales*

ID	Módulo	Requisito
RF-01.1	Usuarios	El sistema debe permitir la autenticación de usuarios (login).
RF-01.2	Usuarios	El sistema debe gestionar diferentes niveles de acceso (roles).
RF-02.1	Clientes	El sistema debe permitir el Alta, Baja y Modificación (ABM) de clientes.
RF-02.2	Clientes	El sistema debe permitir la búsqueda avanzada de clientes (por DNI, nombre, etc.)
RF-02.3	Clientes	El sistema debe mostrar un historial de servicios (reparaciones/facturas) asociados a un cliente.
RF-03.1	Vehículos	El sistema debe permitir el registro de vehículos, asociándolos a un cliente propietario.
RF-03.2	Vehículos	El sistema debe almacenar datos técnicos del vehículo (matrícula, bastidor, marca, modelo).
RF-03.3	Vehículos	El sistema debe mostrar un historial de reparaciones y mantenimientos por vehículo.
RF-04.1	Proveedores y Suministros	El sistema debe permitir el ABM de proveedores.
RF-04.2	Proveedores y Suministros	El sistema debe gestionar un control básico de suministros (piezas, aceites, filtros, etc.).

ID	Módulo	Requisito
RF-05.1	Servicios	El sistema debe permitir la creación de presupuestos detallados, incluyendo mano de obra y piezas/suministros.
RF-05.2	Servicios	El sistema debe permitir cambiar el estado de un presupuesto (Pendiente, Aprobado, Rechazado).
RF-05.3	Servicios	El sistema debe permitir la conversión de un presupuesto "Aprobado" en una factura.
RF-05.4	Servicios	El sistema debe gestionar la emisión de facturas, con numeración automática.
RF-05.5	Servicios	El sistema debe permitir la generación e impresión de la factura final en formato PDF.
RF-06.1	Reparaciones	El sistema debe permitir registrar la entrada de vehículos al taller con sus datos iniciales.
RF-06.2	Reparaciones	El sistema debe permitir asignar/reasignar mecánicos a las reparaciones.
RF-06.3	Reparaciones	El sistema debe gestionar el estado de las reparaciones (EN_COLA, EN_PROCESO, PAUSADA, FINALIZADA, ENTREGADA).
RF-06.4	Reparaciones	El sistema debe permitir registrar el trabajo de mano de obra realizado en cada reparación.
RF-06.5	Reparaciones	El sistema debe permitir registrar los productos usados en cada reparación.
RF-06.6	Reparaciones	El sistema debe descontar automáticamente el stock al usar productos en reparaciones.
RF-06.7	Reparaciones	El sistema debe permitir finalizar reparaciones con validación de trabajos registrados.

ID	Módulo	Requisito
RF-06.8	Reparaciones	El sistema debe gestionar la entrega de vehículos con checklist y actualización de kilometraje.
RF-07.1	Empleados	El sistema debe permitir el registro de empleados con sus datos personales y profesionales.
RF-07.2	Empleados	El sistema debe permitir vincular empleados con usuarios del sistema.
RF-07.3	Empleados	El sistema debe permitir dar de alta y baja empleados manteniendo el historial.
RF-07.4	Empleados	El sistema debe permitir consultar la carga de trabajo por empleado.
RF-07.5	Empleados	El sistema debe permitir consultar el historial completo de cada empleado.
RF-08.1	Inventario	El sistema debe registrar todos los movimientos de stock (entradas, salidas, ajustes).
RF-08.2	Inventario	El sistema debe generar alertas de productos con stock bajo.

## 1.2. Requisitos No Funcionales

Tabla 2: Requisitos No Funcionales

ID	Categoría	Requisito
RNF-01	Plataforma	La aplicación será de escritorio, desarrollada en Java (JDK 23).
RNF-02	Interfaz	La interfaz de usuario se desarrollará con la biblioteca <b>Java Swing</b> , implementando el sistema de diseño Material Design a través de la librería <i>material-ui-swing</i> , garantizando consistencia visual y feedback interactivo.
RNF-03	Arquitectura	El código se estructurará siguiendo el patrón <b>MVC</b> .
RNF-04	Persistencia	Los datos se almacenarán en una base de datos <b>MySQL</b> .
RNF-05	Acceso a Datos	La comunicación con la base de datos se realizará mediante <b>JDBC</b> y se abstraerá utilizando el patrón <b>DAO (Data Access Object)</b>
RNF-06	Informes	Los informes (facturas, presupuestos) se generarán utilizando la biblioteca <b>JasperReports</b> (o iReports).
RNF-07	Entorno	El entorno de desarrollo principal será <b>NetBeans</b> .
RNF-08	Calidad	Se realizarán pruebas unitarias con <b>JUnit</b> para validar la lógica de negocio.
RNF-09	Control de Versiones	El código fuente se gestionará mediante <b>GitHub</b> .
RNF-SEG	Seguridad	El sistema no almacenará contraseñas en texto plano. Se utilizará obligatoriamente el algoritmo de hashing <b>BCrypt</b> (cost factor 10) cumpliendo estándares OWASP.

## 1.3. Casos de Uso

### 1.3.1. Actores del Sistema

- **Empleado:** Usuario general del taller que gestiona clientes, vehículos, presupuestos y facturas. Tiene permisos para realizar todas las operaciones CRUD del sistema.
- **Administrador:** Rol con permisos extendidos que incluye todas las funcionalidades del Empleado más la gestión de usuarios del sistema (futuro).

En el [Anexo I](#) se puede ver el diagrama de casos de uso.

### 1.3.2. Casos de Uso Principales

#### CU-01: Iniciar Sesión en el Sistema

**Actor:** Empleado, Administrador

**Descripción:** Permite la autenticación de usuarios para acceder al sistema.

**Precondiciones:**

- El usuario debe estar registrado en la base de datos
- La aplicación debe estar conectada a la base de datos MySQL

**Postcondiciones:**

- El usuario accede a la ventana principal con permisos según su rol
- Se registra la sesión activa

**Flujo Principal:**

1. El sistema muestra la ventana de login
2. El usuario ingresa su nombre de usuario
3. El usuario ingresa su contraseña
4. El usuario presiona el botón "Acceder"
5. El sistema valida las credenciales contra la base de datos
6. El sistema verifica el hash de la contraseña
7. El sistema identifica el rol del usuario
8. El sistema carga la ventana principal con los permisos correspondientes

**Flujos Alternativos:**

*FA-01: Credenciales incorrectas*

- Las credenciales no son válidas
  - El sistema muestra mensaje: "Usuario o contraseña incorrectos"
  - El sistema limpia el campo de contraseña
  - Vuelve al paso 2

#### *FA-02: Usuario bloqueado (futuro)*

- El usuario ha excedido intentos fallidos
  - El sistema muestra mensaje: "Usuario bloqueado temporalmente"
  - Fin del caso de uso

#### *FA-03: Error de conexión*

- No hay conexión con la base de datos
  - El sistema muestra mensaje: "Error de conexión. Verifique la BD"
  - Fin del caso de uso

### **CU-02: Gestionar Clientes**

**Actor:** Empleado

**Descripción:** Permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre la base de datos de clientes del taller.

#### **Precondiciones:**

- El usuario debe haber iniciado sesión
- El panel de Clientes debe estar activo

#### **Postcondiciones:**

- Los datos del cliente se almacenan/modifican/eliminan en la base de datos
- El historial de modificaciones queda registrado

### **CU-02.1: Dar de Alta un Cliente**

#### **Flujo Principal:**

1. El empleado hace clic en el botón "Nuevo Cliente"
2. El sistema limpia todos los campos del formulario
3. El sistema habilita el campo DNI para edición

4. El empleado ingresa los datos del cliente:
  - DNI (obligatorio)
  - Nombre completo (obligatorio)
  - Dirección
  - Teléfono
  - Email
5. El empleado hace clic en "Guardar"
6. El sistema valida el formato del DNI (8 dígitos + letra)
7. El sistema verifica que el DNI no exista en la base de datos
8. El sistema valida el formato del email (si se proporciona)
9. El sistema asigna automáticamente la fecha\_registro (fecha actual)
10. El sistema guarda el nuevo cliente en la base de datos
11. El sistema muestra mensaje: "Cliente guardado correctamente"
12. El sistema actualiza la tabla de clientes
13. El sistema limpia el formulario

#### Flujos Alternativos:

##### FA-01: DNI duplicado

- El DNI ya existe en la base de datos
  - El sistema muestra mensaje: "Ya existe un cliente con este DNI"
  - El sistema marca el campo DNI en rojo
  - Vuelve al paso 4

##### FA-02: Formato DNI inválido

- El formato del DNI es incorrecto
  - El sistema muestra mensaje: "Formato de DNI inválido (ej: 12345678A)"
  - Vuelve al paso 4

##### FA-03: Email duplicado

- El email ya está registrado
  - El sistema muestra advertencia: "Este email ya está en uso"
  - Permite continuar o modificar

##### FA-04: Campos obligatorios vacíos

- Faltan campos obligatorios

- El sistema marca en rojo los campos vacíos
- El sistema muestra mensaje: "Complete los campos obligatorios"
- Vuelve al paso 4

### **CU-02.2: Buscar Cliente**

#### **Flujo Principal:**

1. El empleado ingresa un criterio de búsqueda en el campo "Buscar"
2. El empleado puede buscar por:
  - DNI (parcial o completo)
  - Nombre (parcial, sin distinción mayúsculas)
  - Teléfono
3. El sistema realiza la búsqueda en tiempo real (mientras escribe)
4. El sistema muestra los resultados en la tabla
5. El sistema indica el número de resultados encontrados

#### **Flujos Alternativos:**

##### *FA-01: Sin resultados*

- No se encuentran coincidencias
  - El sistema muestra mensaje: "No se encontraron clientes"
  - La tabla queda vacía

##### *FA-02: Búsqueda vacía*

- El campo de búsqueda está vacío
  - El sistema muestra todos los clientes ordenados por nombre

### **CU-02.3: Modificar Cliente**

#### **Flujo Principal:**

1. El empleado busca el cliente (ver CU-02.2)
2. El empleado selecciona una fila de la tabla
3. El sistema carga los datos del cliente en el formulario
4. El sistema deshabilita el campo DNI (no se puede modificar la PK)
5. El empleado modifica los campos deseados

6. El empleado hace clic en "Guardar"
7. El sistema valida los datos modificados
8. El sistema actualiza el registro en la base de datos
9. El sistema muestra mensaje: "Cliente actualizado correctamente"
10. El sistema actualiza la tabla de clientes

**Flujos Alternativos:***FA-01: Email duplicado*

- El nuevo email ya pertenece a otro cliente
  - El sistema muestra advertencia: "Email en uso por otro cliente"
  - Permite modificar o cancelar

*FA-02: Cancelar modificación*

- El empleado hace clic en "Cancelar" o "Nuevo"
  - El sistema descarta los cambios
  - El sistema limpia el formulario

**CU-02.4: Eliminar Cliente****Flujo Principal:**

1. El empleado busca el cliente (ver CU-02.2)
2. El empleado selecciona una fila de la tabla
3. El sistema carga los datos del cliente en el formulario
4. El empleado hace clic en "Eliminar"
5. El sistema verifica si el cliente tiene vehículos asociados
6. El sistema verifica si el cliente tiene presupuestos/facturas
7. El sistema muestra diálogo de confirmación:
  - "¿Está seguro de eliminar al cliente [Nombre]?"
  - "Esta acción no se puede deshacer"
8. El empleado confirma la eliminación
9. El sistema elimina el cliente de la base de datos
10. El sistema muestra mensaje: "Cliente eliminado correctamente"
11. El sistema actualiza la tabla
12. El sistema limpia el formulario

**Flujos Alternativos:**

*FA-01: Cliente con vehículos asociados*

- El cliente tiene vehículos registrados
  - El sistema muestra advertencia: "El cliente tiene X vehículo(s) asociado(s)"
  - El sistema pregunta: "¿Desea eliminar también los vehículos?"
  - Si acepta: elimina en cascada (se eliminarán presupuestos/facturas)
  - Si cancela: vuelve al paso 3

*FA-02: Cliente con facturas*

- El cliente tiene facturas emitidas
  - El sistema muestra error: "No se puede eliminar. Cliente con facturas registradas"
  - El sistema sugiere: "Consideré desactivar el cliente en lugar de eliminarlo"
  - Fin del caso de uso

*FA-03: Cancelar eliminación*

- El empleado cancela en el diálogo
  - El sistema no realiza cambios
  - Vuelve al paso 3

**CU-02.5: Consultar Historial de Cliente**

**Flujo Principal:**

1. El empleado selecciona un cliente de la tabla
2. El empleado hace clic en el botón "Ver Historial"
3. El sistema abre una ventana modal con pestañas:
  - Pestaña "Vehículos": Lista de vehículos del cliente
  - Pestaña "Presupuestos": Historial de presupuestos (con estados)
  - Pestaña "Facturas": Facturas emitidas al cliente
4. En cada pestaña se muestran:
  - Fecha
  - Detalles principales
  - Total (en caso de presupuestos/facturas)
5. El empleado puede hacer doble clic en cualquier registro para ver detalles
6. El empleado cierra la ventana modal

**Flujos Alternativos:**

**FA-01: Cliente sin historial**

- El cliente no tiene registros
- Cada pestaña muestra: "No hay registros disponibles"

**CU-03: Gestionar Vehículos****Actor:** Empleado**Descripción:** Permite registrar, modificar, eliminar y consultar vehículos asociados a clientes del taller.**Precondiciones:**

- El usuario debe haber iniciado sesión
- Debe existir al menos un cliente registrado
- El panel de Vehículos debe estar activo

**Postcondiciones:**

- Los datos del vehículo se almacenan/modifican/eliminan en la base de datos
- El vehículo queda asociado a un cliente propietario

**CU-03.1: Registrar Nuevo Vehículo****Flujo Principal:**

1. El empleado hace clic en el botón "Nuevo Vehículo"
2. El sistema limpia todos los campos del formulario
3. El empleado ingresa los datos del vehículo:
  - Bastidor (VIN - obligatorio, 17 caracteres alfanuméricos)
  - Matrícula (obligatorio)
  - Marca
  - Modelo
  - Color
  - Fecha de compra
4. El empleado busca y selecciona el cliente propietario:
  - Opción A: Escribe el DNI directamente
  - Opción B: Hace clic en "Buscar Cliente" (abre diálogo de búsqueda)
5. El sistema muestra los datos del cliente seleccionado
6. El empleado hace clic en "Guardar"

7. El sistema valida el formato del bastidor (17 caracteres)
8. El sistema verifica que bastidor y matrícula no existan
9. El sistema verifica que el cliente existe
10. El sistema guarda el vehículo vinculado al cliente
11. El sistema crea el primer registro en Historial\_Propiedad
12. El sistema muestra mensaje: "Vehículo registrado correctamente"
13. El sistema actualiza la tabla de vehículos

**Flujos Alternativos:***FA-01: Bastidor duplicado*

- El bastidor ya existe
  - El sistema muestra mensaje: "Este vehículo ya está registrado"
  - El sistema muestra los datos del vehículo existente
  - Permite ver el historial del vehículo

*FA-02: Matrícula duplicada*

- La matrícula ya existe
  - El sistema muestra advertencia: "Esta matrícula ya está registrada"
  - El sistema muestra el vehículo con esa matrícula
  - Sugiere: "¿Es un cambio de matrícula?"

*FA-03: Cliente no seleccionado*

- No se ha seleccionado propietario
  - El sistema marca el campo en rojo
  - El sistema muestra mensaje: "Debe seleccionar un cliente propietario"
  - Vuelve al paso 4

*FA-04: Formato bastidor inválido*

- El bastidor no tiene 17 caracteres
  - El sistema muestra mensaje: "El bastidor debe tener 17 caracteres"
  - Vuelve al paso 3

**CU-03.2: Cambiar Propietario de Vehículo****Flujo Principal:**

1. El empleado selecciona un vehículo de la tabla
2. El sistema carga los datos en el formulario
3. El sistema muestra el propietario actual
4. El empleado hace clic en "Cambiar Propietario"
5. El sistema abre un diálogo de cambio de propietario
6. El empleado busca y selecciona el nuevo cliente
7. El empleado indica la fecha de cambio de propiedad
8. El empleado hace clic en "Confirmar Cambio"
9. El sistema actualiza el campo propietario\_actual\_dni en Vehiculos
10. El sistema cierra el registro anterior en Historial\_Propiedad (fecha\_fin)
11. El sistema crea un nuevo registro en Historial\_Propiedad con el nuevo propietario
12. El sistema muestra mensaje: "Propietario actualizado correctamente"
13. El sistema actualiza el formulario con los nuevos datos

#### **Flujos Alternativos:**

##### *FA-01: Mismo propietario*

- Se selecciona el mismo cliente que ya es propietario
  - El sistema muestra advertencia: "Este cliente ya es el propietario actual"
  - Permite cancelar o continuar (por si es un error en el registro)

##### *FA-02: Fecha inválida*

- La fecha es anterior a la última fecha\_inicio del historial
  - El sistema muestra error: "La fecha debe ser posterior a [última\_fecha]"
  - Vuelve al paso 7

#### **CU-03.3: Consultar Historial de Vehículo**

#### **Flujo Principal:**

1. El empleado selecciona un vehículo de la tabla
2. El empleado hace clic en "Ver Historial"
3. El sistema abre una ventana modal con pestañas:
  - Pestaña "Propietarios": Historial de propiedad
  - Pestaña "Reparaciones": Presupuestos/facturas del vehículo
  - Pestaña "Mantenimientos": Servicios realizados
4. En Pestaña "Propietarios" se muestra:

- Cliente (DNI, Nombre)
- Fecha inicio
- Fecha fin (vacío si es propietario actual)

5. En Pestaña "Reparaciones" se muestra:

- Fecha
- Tipo (Presupuesto/Factura)
- Descripción
- Estado
- Total

6. El empleado puede exportar el historial a PDF

7. El empleado cierra la ventana

#### **Flujos Alternativos:**

##### *FA-01: Vehículo sin reparaciones*

- No hay presupuestos ni facturas
  - Se muestra: "No hay reparaciones registradas para este vehículo"

#### **CU-03.4: Modificar Datos de Vehículo**

##### **Flujo Principal:**

1. El empleado selecciona un vehículo de la tabla
2. El sistema carga los datos en el formulario
3. El sistema deshabilita los campos: bastidor y matrícula (no modificables)
4. El empleado modifica los campos permitidos:
  - Marca
  - Modelo
  - Color
  - Fecha de compra
5. El empleado hace clic en "Guardar"
6. El sistema valida los datos
7. El sistema actualiza el registro en la base de datos
8. El sistema muestra mensaje: "Vehículo actualizado correctamente"
9. El sistema actualiza la tabla

**Nota:** El cambio de propietario se gestiona mediante CU-03.2

## CU-03.5: Eliminar Vehículo

### Flujo Principal:

1. El empleado selecciona un vehículo de la tabla
2. El empleado hace clic en "Eliminar"
3. El sistema verifica si el vehículo tiene presupuestos o facturas
4. El sistema muestra diálogo de confirmación con advertencia
5. El empleado confirma la eliminación
6. El sistema elimina el vehículo (ON DELETE CASCADE en BD)
7. El sistema elimina automáticamente:
  - Registros de Historial\_Propiedad
  - Presupuestos asociados (y sus detalles)
  - Facturas asociadas
8. El sistema muestra mensaje: "Vehículo eliminado correctamente"
9. El sistema actualiza la tabla

### Flujos Alternativos:

#### FA-01: Vehículo con facturas

- El vehículo tiene facturas emitidas
  - El sistema muestra error: "No se puede eliminar. Vehículo con facturas registradas"
  - El sistema muestra el número de facturas
  - Sugiere archivar en lugar de eliminar
  - Fin del caso de uso

#### FA-02: Cancelar eliminación

- El empleado cancela
  - No se realizan cambios
  - Vuelve al paso 1

**CU-04: Gestionar Presupuestos****Actor:** Empleado**Descripción:** Permite crear, modificar y gestionar presupuestos de reparación para clientes, incluyendo mano de obra y productos.**Precondiciones:**

- El usuario debe haber iniciado sesión
- Deben existir clientes y vehículos registrados
- Deben existir productos en el inventario
- El panel de Presupuestos debe estar activo

**Postcondiciones:**

- El presupuesto queda registrado con estado inicial PENDIENTE
- Se almacenan los detalles de mano de obra y productos
- El total se calcula automáticamente

**CU-04.1: Crear Nuevo Presupuesto****Flujo Principal:**

1. El empleado hace clic en "Nuevo Presupuesto"
2. El sistema abre un formulario de presupuesto con tres secciones:
  - Sección A: Datos generales
  - Sección B: Mano de obra
  - Sección C: Productos/Piezas
3. **Sección A - Datos Generales:**
  - El empleado busca y selecciona el cliente
  - El sistema carga los vehículos del cliente en un combo
  - El empleado selecciona el vehículo
  - El sistema asigna automáticamente:
    - Fecha de emisión (fecha actual)
    - Estado: PENDIENTE
  - El empleado ingresa descripción general del trabajo
4. **Sección B - Mano de Obra:**
  - El empleado hace clic en "Añadir Línea de Trabajo"
  - El sistema muestra un formulario para:

- Descripción del trabajo (ej: "Revisión de frenos")
- Tiempo estimado en horas (ej: 2.5)
- Tarifa por hora (ej: 45.00€) - valor predeterminado configurable
- El empleado guarda la línea
- El sistema añade la línea a una tabla de mano de obra
- El sistema calcula: subtotal\_mano\_obra = suma(tiempo × tarifa)
- El empleado puede añadir más líneas (repetir desde paso anterior)

#### 5. Sección C - Productos/Piezas:

- El empleado hace clic en "Añadir Producto"
- El sistema muestra un diálogo de búsqueda de productos:
  - Búsqueda por código o nombre
  - Tabla con: ID, Nombre, Stock disponible, Precio unitario
- El empleado selecciona un producto
- El sistema muestra un formulario para:
  - Cantidad usada (validar contra stock)
  - Precio venta unitario (precargado, modificable)
  - Descuento % (opcional, default 0)
- El empleado confirma
- El sistema añade el producto a una tabla de productos
- El sistema calcula: subtotal\_productos = suma((precio × cantidad) × (1 - descuento))
- El empleado puede añadir más productos (repetir)

#### 6. Cálculo Total:

- El sistema calcula automáticamente:
  - Total Mano de Obra
  - Total Productos
  - Total Estimado = Total Mano de Obra + Total Productos
- Los totales se muestran en tiempo real al añadir/modificar líneas

#### 7. El empleado hace clic en "Guardar Presupuesto"

#### 8. El sistema valida:

- Cliente y vehículo seleccionados
- Al menos una línea de mano de obra o un producto

#### 9. El sistema guarda en la base de datos:

- Registro en tabla Presupuestos
- Líneas en Detalle\_Mano\_Obra

- Líneas en Detalle\_Productos
- 10. El sistema asigna un ID de presupuesto automático
- 11. El sistema muestra mensaje: "Presupuesto #[ID] creado correctamente"
- 12. El sistema cierra el formulario
- 13. El sistema actualiza la tabla de presupuestos

### **Flujos Alternativos:**

#### *FA-01: Stock insuficiente*

- La cantidad solicitada excede el stock disponible
  - El sistema muestra advertencia: "Stock insuficiente. Disponible: X unidades"
  - El sistema permite:
    - Opción A: Reducir cantidad
    - Opción B: Guardar presupuesto con nota de "pendiente de suministro"
    - Opción C: Cancelar

#### *FA-02: Cliente sin vehículos*

- El cliente seleccionado no tiene vehículos registrados
  - El sistema muestra mensaje: "El cliente no tiene vehículos registrados"
  - El sistema ofrece: "¿Desea registrar un vehículo ahora?"
  - Si acepta: abre CU-03.1
  - Si cancela: vuelve al paso 3

#### *FA-03: Presupuesto vacío*

- No se han añadido líneas de trabajo ni productos
  - El sistema muestra error: "Debe añadir al menos una línea de trabajo o producto"
  - Vuelve al paso 4 o 5

#### *FA-04: Cancelar presupuesto*

- El empleado hace clic en "Cancelar"
  - El sistema muestra confirmación: "¿Descartar presupuesto sin guardar?"
  - Si confirma: cierra el formulario sin guardar
  - Si cancela: vuelve al paso 3

### **CU-04.2: Modificar Presupuesto**

**Flujo Principal:**

1. El empleado busca el presupuesto en la tabla
2. El empleado selecciona un presupuesto con estado PENDIENTE
3. El empleado hace clic en "Editar"
4. El sistema abre el formulario precargado con los datos actuales
5. El empleado puede:
  - Modificar la descripción general
  - Añadir/eliminar líneas de mano de obra
  - Añadir/eliminar/modificar productos
6. El sistema recalcula el total automáticamente
7. El empleado hace clic en "Guardar Cambios"
8. El sistema actualiza el presupuesto en la base de datos
9. El sistema muestra mensaje: "Presupuesto actualizado"

**Flujos Alternativos:**
*FA-01: Presupuesto no modificable*

- El presupuesto tiene estado APROBADO o RECHAZADO
  - El sistema muestra mensaje: "No se puede modificar un presupuesto [ESTADO]"
  - El sistema permite solo visualización (sin guardar cambios)

*FA-02: Presupuesto facturado*

- El presupuesto ya tiene una factura asociada
  - El sistema muestra error: "Este presupuesto ya fue facturado"
  - Deshabilita el botón "Editar"
  - Solo permite visualización

**CU-04.3: Cambiar Estado de Presupuesto**
**Flujo Principal:**

1. El empleado selecciona un presupuesto de la tabla
2. El empleado hace clic en el combo "Estado" del formulario
3. El sistema muestra las opciones:
  - PENDIENTE (amarillo)
  - APROBADO (verde)

- RECHAZADO (rojo)
4. El empleado selecciona el nuevo estado
  5. Si selecciona RECHAZADO:
    - El sistema solicita motivo del rechazo (opcional)
  6. El empleado hace clic en "Guardar Estado"
  7. El sistema actualiza el campo estado en la tabla Presupuestos
  8. El sistema muestra mensaje: "Estado actualizado a [NUEVO\_ESTADO]"
  9. El sistema actualiza el indicador visual en la tabla (color)

#### **Flujos Alternativos:**

##### *FA-01: Presupuesto ya facturado*

- El presupuesto tiene factura asociada
  - El sistema deshabilita la opción de cambiar estado
  - El sistema muestra: "No se puede cambiar el estado de un presupuesto facturado"

##### *FA-02: Cambio a PENDIENTE desde APROBADO*

- Se intenta cambiar de APROBADO a PENDIENTE
  - El sistema muestra advertencia: "¿Está seguro? El cliente ya había aceptado"
  - Si confirma: permite el cambio
  - Si cancela: mantiene APROBADO

#### **CU-04.4: Consultar Presupuesto**

#### **Flujo Principal:**

1. El empleado selecciona un presupuesto de la tabla
2. El sistema muestra todos los datos en el formulario (modo lectura):
  - Datos del cliente y vehículo
  - Descripción del trabajo
  - Tabla de mano de obra con subtotal
  - Tabla de productos con subtotal
  - Total estimado
  - Estado actual
  - Fecha de emisión
3. El empleado puede:

- Imprimir presupuesto en PDF
  - Enviar por email al cliente (futuro)
4. El empleado cierra la vista

**Flujos Alternativos:***FA-01: Generar PDF del presupuesto*

- El empleado hace clic en "Imprimir PDF"
  - El sistema genera el documento con JasperReports
  - El sistema incluye: logo del taller, datos, líneas y total
  - El sistema abre el PDF en el visor predeterminado

**CU-04.5: Eliminar Presupuesto****Flujo Principal:**

1. El empleado selecciona un presupuesto de la tabla
2. El empleado hace clic en "Eliminar"
3. El sistema verifica que el presupuesto NO tenga factura asociada
4. El sistema muestra diálogo de confirmación:
  - "¿Eliminar presupuesto #[ID]?"
  - "Se eliminarán todos los detalles asociados"
5. El empleado confirma
6. El sistema elimina en cascada:
  - Registro de Presupuestos
  - Registros de Detalle\_Mano\_Obra
  - Registros de Detalle\_Productos
7. El sistema muestra mensaje: "Presupuesto eliminado"
8. El sistema actualiza la tabla

**Flujos Alternativos:***FA-01: Presupuesto facturado*

- El presupuesto tiene una factura asociada
  - El sistema muestra error: "No se puede eliminar. Presupuesto facturado"
  - Fin del caso de uso

*FA-02: Cancelar eliminación*

- El empleado cancela
  - No se realizan cambios
  - Vuelve al paso 1

### CU-05: Generar y Gestionar Facturas

**Actor:** Empleado

**Descripción:** Permite convertir presupuestos aprobados en facturas, generando documentos PDF con formato profesional y numeración automática.

**Precondiciones:**

- El usuario debe haber iniciado sesión
- Debe existir un presupuesto con estado APROBADO
- El presupuesto NO debe tener factura asociada
- El panel de Facturas o Presupuestos debe estar activo

**Postcondiciones:**

- Se crea una factura vinculada al presupuesto
- Se asigna un número de factura correlativo
- El stock de productos se descuenta automáticamente
- Se genera el documento PDF

#### CU-05.1: Facturar Presupuesto

**Flujo Principal:**

1. El empleado selecciona un presupuesto con estado APROBADO
2. El empleado hace clic en "Generar Factura"
3. El sistema verifica:
  - Estado = APROBADO
  - Sin factura asociada
  - Stock suficiente de todos los productos
4. El sistema muestra un diálogo de confirmación con vista previa:
  - Cliente y vehículo
  - Desglose de mano de obra

- Desglose de productos
  - Total a cobrar
  - Campo para: Método de pago (Efectivo, Tarjeta, Transferencia)
  - Campo para: Fecha de cobro (opcional)
5. El empleado revisa los datos
  6. El empleado selecciona método de pago
  7. El empleado hace clic en "Confirmar y Generar Factura"
  8. El sistema realiza las siguientes operaciones (transacción atómica):
    - Genera el siguiente número de factura (formato: FACT-2025-0001)
    - Crea el registro en tabla Facturas:
      - id\_factura (auto-incremental)
      - numero\_factura (único)
      - id\_presupuesto (FK)
      - fecha\_emision (fecha actual)
      - fecha\_cobro (si se indicó)
      - total\_cobrado (copiado del presupuesto)
      - metodo\_pago
    - Copia los detalles del presupuesto a la factura
    - Actualiza el stock de productos:
      - Para cada producto en Detalle\_Productos:
        - cantidad\_stock = cantidad\_stock - cantidad\_usada
    - Crea registro de auditoría (opcional)
  9. El sistema genera el PDF de la factura usando JasperReports con:
    - Logotipo del taller
    - Datos del taller (dirección, CIF, teléfono)
    - Número de factura
    - Fecha de emisión
    - Datos del cliente
    - Datos del vehículo
    - Tabla de mano de obra con subtotal
    - Tabla de productos con subtotal
    - Total general
    - Método de pago
  10. El sistema muestra mensaje: "Factura #[NÚMERO] generada correctamente"
  11. El sistema abre el PDF en el visor predeterminado
  12. El sistema actualiza la tabla de presupuestos (marcando que está facturado)

13. El sistema actualiza la tabla de facturas

### **Flujos Alternativos:**

#### *FA-01: Presupuesto no aprobado*

- El estado del presupuesto es PENDIENTE o RECHAZADO
  - El sistema muestra error: "Solo se pueden facturar presupuestos APROBADOS"
  - El sistema sugiere: "Cambio el estado a APROBADO primero"
  - Fin del caso de uso

#### *FA-02: Presupuesto ya facturado*

- El presupuesto ya tiene una factura
  - El sistema muestra mensaje: "Este presupuesto ya fue facturado"
  - El sistema muestra el número de factura existente
  - El sistema ofrece: "¿Desea ver la factura?"
  - Si acepta: abre el PDF existente
  - Fin del caso de uso

#### *FA-03: Stock insuficiente*

- Uno o más productos no tienen stock suficiente
  - El sistema muestra error detallado:
    - "Stock insuficiente para:"
    - Lista de productos con stock actual vs necesario
  - El sistema ofrece dos opciones:
    - Opción A: "Actualizar stock y reintentar"
    - Opción B: "Facturar sin descontar stock" (con advertencia)
  - Si elige A: redirige a CU-07 (Gestionar Suministros)
  - Si elige B: continúa sin descontar stock (requiere confirmación adicional)

#### *FA-04: Error al generar PDF*

- Fallo en JasperReports
  - El sistema muestra error: "No se pudo generar el PDF"
  - La factura queda guardada en BD
  - El sistema ofrece: "Reintentar generación de PDF"
  - Los cambios en BD NO se revierten (la factura existe)

#### *FA-05: Cancelar facturación*

- El empleado cancela en el diálogo de confirmación
  - No se realizan cambios en la BD
  - Vuelve al paso 1

*FA-06: Error de numeración*

- Error al generar número de factura (duplicado por concurrencia)
  - El sistema reintenta con el siguiente número disponible
  - Si persiste: muestra error y cancela operación

### **CU-05.2: Consultar Facturas**

#### **Flujo Principal:**

1. El empleado accede al panel de Facturas
2. El sistema muestra una tabla con todas las facturas:
  - Número de factura
  - Fecha de emisión
  - Cliente (nombre)
  - Vehículo (matrícula)
  - Total cobrado
  - Método de pago
  - Estado de cobro (Cobrada/Pendiente)
3. El empleado puede filtrar por:
  - Rango de fechas
  - Cliente (nombre o DNI)
  - Número de factura
  - Método de pago
  - Estado de cobro
4. El empleado selecciona una factura de la tabla
5. El sistema muestra los detalles completos en un panel lateral:
  - Datos del cliente y vehículo
  - Fecha de emisión y cobro
  - Desglose completo de líneas
  - Presupuesto origen (enlace)
  - Total
6. El empleado puede:

- Ver PDF
- Reimprimir factura
- Ver presupuesto asociado
- Marcar como cobrada (si estaba pendiente)

### **Flujos Alternativos:**

#### *FA-01: Buscar facturas sin resultados*

- Los filtros no producen resultados
  - El sistema muestra: "No se encontraron facturas con estos criterios"
  - Permite limpiar filtros

#### *FA-02: Reimprimir factura*

- El empleado hace clic en "Reimprimir"
  - El sistema regenera el PDF desde la BD
  - El sistema abre el PDF
  - El sistema registra la reimpresión (auditoría)

### **CU-05.3: Registrar Cobro de Factura**

#### **Flujo Principal:**

1. El empleado selecciona una factura con fecha\_cobro = NULL
2. El empleado hace clic en "Registrar Cobro"
3. El sistema muestra un diálogo con:
  - Número de factura
  - Cliente
  - Total a cobrar
  - Campo: Fecha de cobro (llenado con fecha actual)
  - Campo: Método de pago utilizado (si no estaba registrado)
4. El empleado confirma o modifica la fecha
5. El empleado hace clic en "Confirmar Cobro"
6. El sistema actualiza la factura:
  - fecha\_cobro = fecha indicada
  - metodo\_pago (si se modificó)
7. El sistema muestra mensaje: "Cobro registrado correctamente"

8. El sistema actualiza la tabla (la factura ya no aparece como pendiente)

#### **Flujos Alternativos:**

*FA-01: Factura ya cobrada*

- La factura ya tiene fecha de cobro
  - El sistema muestra: "Esta factura ya fue cobrada el [FECHA]"
  - Permite modificar la fecha de cobro (con confirmación)

#### **CU-05.4: Anular Factura (Caso Excepcional)**

**Precondición:** Solo el Administrador puede anular facturas

#### **Flujo Principal:**

1. El administrador selecciona una factura
2. El administrador hace clic en "Anular Factura"
3. El sistema muestra advertencia:
  - "ADVERTENCIA: Esta acción afecta la contabilidad"
  - "Se recomienda crear una factura rectificativa en su lugar"
  - Campo obligatorio: Motivo de anulación
4. El administrador ingresa el motivo
5. El administrador confirma
6. El sistema:
  - NO elimina la factura (por trazabilidad)
  - Añade un campo estado = 'ANULADA'
  - Revierte el descuento de stock (devuelve productos al inventario)
  - Registra la anulación con fecha, usuario y motivo
7. El sistema muestra mensaje: "Factura anulada. Stock revertido"
8. La factura aparece en la tabla con indicador visual de ANULADA

#### **Flujos Alternativos:**

*FA-01: Factura antigua*

- La factura tiene más de 30 días
  - El sistema muestra advertencia adicional:
    - "Factura antigua. Consulte con el departamento contable"
  - Requiere doble confirmación

FA-02: *C Cancelar anulación*

- El administrador cancela
  - No se realizan cambios
  - Fin del caso de uso

**CU-06: Gestión de Proveedores****Actor:** Empleado**Descripción:** Permite administrar la base de datos de proveedores de suministros y piezas del taller.**Precondiciones:**

- El usuario debe haber iniciado sesión
- El panel de Proveedores debe estar activo

**Postcondiciones:**

- Los datos del proveedor se almacenan/modifican/eliminan en la base de datos

**CU-06.1: Registrar Nuevo Proveedor****Flujo Principal:**

1. El empleado hace clic en "Nuevo Proveedor"
2. El sistema limpia el formulario
3. El empleado ingresa los datos:
  - CIF (obligatorio, 9 caracteres)
  - Nombre comercial (obligatorio)
  - Dirección
  - Teléfono
  - Email
  - Persona de contacto
4. El empleado hace clic en "Guardar"
5. El sistema valida el formato del CIF español
6. El sistema verifica que el CIF no exista
7. El sistema valida el formato del email (si se proporciona)

8. El sistema guarda el proveedor en la base de datos
9. El sistema muestra mensaje: "Proveedor registrado correctamente"
10. El sistema actualiza la tabla de proveedores

**Flujos Alternativos:***FA-01: CIF duplicado*

- El CIF ya existe en la base de datos
  - El sistema muestra error: "Ya existe un proveedor con este CIF"
  - El sistema muestra los datos del proveedor existente
  - Vuelve al paso 3

*FA-02: Formato CIF inválido*

- El formato del CIF es incorrecto
  - El sistema muestra mensaje: "Formato CIF inválido (ej: A12345678)"
  - Vuelve al paso 3

*FA-03: Campos obligatorios vacíos*

- Faltan campos obligatorios
  - El sistema marca en rojo los campos vacíos
  - El sistema muestra mensaje: "Complete los campos obligatorios"
  - Vuelve al paso 3

**CU-06.2: Modificar Proveedor****Flujo Principal:**

1. El empleado busca el proveedor (por CIF o nombre)
2. El empleado selecciona un proveedor de la tabla
3. El sistema carga los datos en el formulario
4. El sistema deshabilita el campo CIF (no modificable)
5. El empleado modifica los campos deseados
6. El empleado hace clic en "Guardar"
7. El sistema valida los datos modificados
8. El sistema actualiza el registro
9. El sistema muestra mensaje: "Proveedor actualizado correctamente"

**CU-06.3: Eliminar Proveedor****Flujo Principal:**

1. El empleado selecciona un proveedor
2. El empleado hace clic en "Eliminar"
3. El sistema verifica si el proveedor tiene productos asociados
4. El sistema muestra diálogo de confirmación:
  - Si tiene productos: "El proveedor tiene X productos asociados"
  - "¿Desea eliminar el proveedor?"
5. El empleado confirma
6. El sistema actualiza productos del proveedor (proveedor\_cif = NULL)
7. El sistema elimina el proveedor
8. El sistema muestra mensaje: "Proveedor eliminado"

**Flujos Alternativos:***FA-01: Cancelar eliminación*

- El empleado cancela
  - No se realizan cambios
  - Fin del caso de uso

**CU-06.4: Consultar Productos de Proveedor****Flujo Principal:**

1. El empleado selecciona un proveedor
2. El empleado hace clic en "Ver Productos"
3. El sistema abre una ventana modal con:
  - Lista de todos los productos del proveedor
  - Stock actual
  - Precio unitario
  - Última fecha de actualización
4. El empleado puede hacer doble clic en un producto para editarlo
5. El empleado cierra la ventana

**CU-07: Gestionar Suministros/Productos****Actor:** Empleado**Descripción:** Permite administrar el inventario de piezas y suministros del taller, controlando stock y precios.**Precondiciones:**

- El usuario debe haber iniciado sesión
- Debe existir al menos un proveedor registrado
- El panel de Suministros debe estar activo

**Postcondiciones:**

- Los datos del producto se almacenan/modifican/eliminan
- El stock se actualiza correctamente

**CU-07.1: Registrar Nuevo Producto****Flujo Principal:**

1. El empleado hace clic en "Nuevo Producto"
2. El sistema muestra el formulario
3. El empleado ingresa los datos:
  - ID producto / Código (obligatorio, único)
  - Nombre (obligatorio)
  - Descripción
  - Cantidad en stock (obligatorio, default 0)
  - Precio unitario (obligatorio)
  - Proveedor (selección desde combo)
4. El empleado hace clic en "Guardar"
5. El sistema valida:
  - ID único
  - Stock  $\geq 0$
  - Precio  $> 0$
  - Proveedor existe
6. El sistema guarda el producto

7. El sistema muestra mensaje: "Producto registrado correctamente"
8. El sistema actualiza la tabla de productos

#### **Flujos Alternativos:**

##### *FA-01: ID duplicado*

- El ID ya existe
  - El sistema muestra error: "Ya existe un producto con este código"
  - Sugiere el siguiente código disponible
  - Vuelve al paso 3

##### *FA-02: Stock negativo*

- Se intenta ingresar stock negativo
  - El sistema muestra error: "El stock no puede ser negativo"
  - Vuelve al paso 3

##### *FA-03: Precio inválido*

- El precio es  $\leq 0$ 
  - El sistema muestra error: "El precio debe ser mayor que cero"
  - Vuelve al paso 3

#### **CU-07.2: Actualizar Stock**

#### **Flujo Principal:**

1. El empleado selecciona un producto de la tabla
2. El empleado hace clic en "Actualizar Stock"
3. El sistema muestra un diálogo con:
  - Nombre del producto
  - Stock actual
  - Campo: Tipo de movimiento (Entrada / Salida)
  - Campo: Cantidad
  - Campo: Motivo (opcional)
4. El empleado selecciona el tipo de movimiento
5. El empleado ingresa la cantidad
6. El empleado hace clic en "Confirmar"
7. El sistema valida:

- Si es SALIDA: stock\_actual >= cantidad
  - Si es ENTRADA: cantidad > 0
8. El sistema actualiza el stock:
- Si ENTRADA: stock = stock + cantidad
  - Si SALIDA: stock = stock - cantidad
9. El sistema registra el movimiento (auditoría)
10. El sistema muestra mensaje: "Stock actualizado: [NUEVO\_STOCK] unidades"
11. El sistema actualiza la tabla

#### **Flujos Alternativos:**

##### *FA-01: Stock insuficiente para salida*

- Se intenta sacar más cantidad de la disponible
  - El sistema muestra error: "Stock insuficiente. Disponible: [X] unidades"
  - Vuelve al paso 5

##### *FA-02: Cantidad inválida*

- La cantidad es <= 0
  - El sistema muestra error: "Ingrese una cantidad válida"
  - Vuelve al paso 5

#### **CU-07.3: Modificar Producto**

#### **Flujo Principal:**

1. El empleado selecciona un producto
2. El empleado hace clic en "Editar"
3. El sistema carga los datos en el formulario
4. El sistema deshabilita el campo ID (no modificable)
5. El empleado modifica los campos deseados:
  - Nombre
  - Descripción
  - Precio unitario
  - Proveedor
  - (El stock se modifica mediante CU-07.2)
6. El empleado hace clic en "Guardar"
7. El sistema valida los cambios

8. El sistema actualiza el producto
9. El sistema muestra mensaje: "Producto actualizado"

#### CU-07.4: Eliminar Producto

##### Flujo Principal:

1. El empleado selecciona un producto
2. El empleado hace clic en "Eliminar"
3. El sistema verifica si el producto está en presupuestos/facturas
4. Si NO está en uso:
  - El sistema muestra confirmación: "¿Eliminar producto [NOMBRE]?"
  - El empleado confirma
  - El sistema elimina el producto
5. Si ESTÁ en uso:
  - El sistema muestra advertencia: "El producto está en presupuestos/facturas"
  - El sistema sugiere: "No se puede eliminar, considere desactivarlo"
  - El sistema NO permite la eliminación

##### Flujos Alternativos:

###### FA-01: Producto en uso

- El producto tiene referencias en Detalle\_Productos
  - El sistema muestra error: "No se puede eliminar. Producto en uso"
  - Muestra número de presupuestos que lo usan
  - Sugiere marcar como "descatalogado" (campo adicional futuro)
  - Fin del caso de uso

#### CU-07.5: Consultar Productos con Stock Bajo

##### Flujo Principal:

1. El empleado hace clic en "Alertas de Stock"
2. El sistema muestra un diálogo con:

- Campo: Stock mínimo (default: 5 unidades)
- 3. El empleado puede ajustar el umbral
- 4. El sistema consulta productos donde cantidad\_stock < umbral
- 5. El sistema muestra una tabla con:
  - Código
  - Nombre
  - Stock actual
  - Proveedor
  - Acción sugerida: "Reabastecer"
- 6. El empleado puede:
  - Exportar lista a PDF/Excel
  - Seleccionar productos para actualizar stock
  - Contactar proveedor (generar email automático - futuro)
- 7. El empleado cierra el diálogo

**Flujos Alternativos:**

*FA-01: Sin productos con stock bajo*

- Todos los productos tienen stock suficiente
  - El sistema muestra: "✓ Todos los productos tienen stock adecuado"

**CU-08: Gestionar Usuarios (Administrador)**

**Actor:** Administrador

**Descripción:** Permite al administrador crear, modificar, eliminar y gestionar usuarios del sistema.

**Precondiciones:**

- El usuario debe haber iniciado sesión con rol ADMINISTRADOR
- El panel de Usuarios debe estar activo

**Postcondiciones:**

- Los usuarios son creados/modificados/eliminados
- Las contraseñas se almacenan hasheadas con BCrypt

## CU-08.1: Crear Nuevo Usuario

### Flujo Principal:

1. El administrador hace clic en "Nuevo Usuario"
2. El sistema muestra el formulario
3. El administrador ingresa:
  - Username (obligatorio, único)
  - Contraseña temporal (obligatorio, mín. 8 caracteres)
  - Confirmar contraseña
  - Rol: Empleado / Administrador
4. El administrador hace clic en "Crear"
5. El sistema valida:
  - Username único
  - Contraseñas coinciden
  - Contraseña cumple política (mín. 8 caracteres)
6. El sistema hashea la contraseña con BCrypt
7. El sistema guarda el usuario
8. El sistema muestra mensaje: "Usuario creado. Contraseña temporal: [XXX]"
9. El sistema sugiere: "El usuario debe cambiar la contraseña en el primer acceso"

### Flujos Alternativos:

#### FA-01: Username duplicado

- El username ya existe
  - El sistema muestra error: "El nombre de usuario ya está en uso"
  - Vuelve al paso 3

#### FA-02: Contraseñas no coinciden

- Las contraseñas no son iguales
  - El sistema muestra error: "Las contraseñas no coinciden"
  - Vuelve al paso 3

#### FA-03: Contraseña débil

- La contraseña no cumple la política
  - El sistema muestra: "La contraseña debe tener al menos 8 caracteres"
  - Vuelve al paso 3

## CU-08.2: Modificar Usuario

### Flujo Principal:

1. El administrador selecciona un usuario
2. El administrador puede:
  - Cambiar el rol
  - Resetear contraseña
  - Activar/Desactivar usuario (futuro)
3. Si elige "Resetear Contraseña":
  - El sistema genera una contraseña temporal
  - El sistema hashea y actualiza
  - El sistema muestra la contraseña temporal
4. Si cambia el rol:
  - El sistema actualiza el campo rol
  - El sistema muestra confirmación
5. El sistema guarda los cambios

## CU-08.3: Eliminar Usuario

### Flujo Principal:

1. El administrador selecciona un usuario
2. El administrador hace clic en "Eliminar"
3. El sistema verifica que no sea el usuario actual
4. El sistema muestra confirmación
5. El administrador confirma
6. El sistema elimina el usuario
7. El sistema muestra mensaje: "Usuario eliminado"

### Flujos Alternativos:

#### FA-01: Eliminar usuario actual

- Se intenta eliminar el usuario con sesión activa
  - El sistema muestra error: "No puede eliminar su propio usuario"

- Fin del caso de uso

### CU-09: Gestionar Reparaciones

**Actor:** Empleado

**Descripción:** Permite gestionar el ciclo de vida completo de una reparación, desde la entrada del vehículo al taller hasta su entrega al cliente. Este es el caso de uso central del sistema, donde se registra el trabajo real realizado, los productos utilizados y se controla el estado de cada reparación.

**Precondiciones:**

- El usuario debe haber iniciado sesión con rol Empleado
- Deben existir clientes y vehículos registrados en el sistema
- Debe haber al menos un empleado activo para asignar
- El panel de Reparaciones debe estar activo

**Postcondiciones:**

- La reparación queda registrada con estado inicial EN\_COLA
- El vehículo queda asociado a la reparación
- Se registran todos los trabajos y productos utilizados
- El stock se actualiza automáticamente al usar productos
- La reparación puede ser facturada cuando esté FINALIZADA

#### CU-09.1: Registrar Entrada de Vehículo al Taller

**Flujo Principal:**

1. El empleado hace clic en "Nueva Reparación"
2. El sistema muestra un formulario con tres secciones:
  - **Sección A:** Datos del cliente y vehículo
  - **Sección B:** Datos de entrada
  - **Sección C:** Asignación y prioridad
3. **Sección A - Cliente y Vehículo:**
  - El empleado busca y selecciona el cliente
  - El sistema carga los vehículos del cliente en un combo

- El empleado selecciona el vehículo
- El sistema muestra los datos del vehículo (marca, modelo, matrícula)

#### 4. Sección B - Datos de Entrada:

- El sistema registra automáticamente:
  - Fecha y hora de entrada (actual)
  - Usuario de recepción (usuario actual)
- El empleado ingresa:
  - Kilometraje de entrada (obligatorio)
  - Nivel de combustible: Vacío / 1/4 / 1/2 / 3/4 / Lleno
  - Diagnóstico inicial (descripción del problema reportado)
  - Fecha salida estimada (opcional)

#### 5. Sección C - Asignación y Prioridad:

- El empleado selecciona:
  - Empleado asignado (mecánico responsable)
  - Prioridad: Baja / Normal / Alta / Urgente
- El empleado puede vincular un presupuesto existente (opcional):
  - Si existe presupuesto APROBADO: se asocia automáticamente
  - Si no existe: la reparación se registra sin presupuesto

#### 6. El sistema asigna automáticamente:

- Estado inicial: EN\_COLA
- ID de reparación (autoincremental)

#### 7. El empleado hace clic en "Registrar Entrada"

#### 8. El sistema valida los datos obligatorios

#### 9. El sistema guarda la reparación en la base de datos

#### 10. El sistema muestra mensaje: "Vehículo registrado en taller - Reparación #[ID]"

#### 11. El sistema actualiza la tabla de reparaciones

#### 12. El sistema actualiza el contador del dashboard "Vehículos en Taller"

#### Flujos Alternativos:

##### FA-01: Cliente sin vehículos

- El cliente no tiene vehículos registrados
  - El sistema muestra mensaje: "El cliente no tiene vehículos registrados"
  - El sistema ofrece: "¿Desea registrar un vehículo ahora?"
  - Si acepta: abre CU-03.1 (Registrar Nuevo Vehículo)
  - Si cancela: vuelve al paso 3

*FA-02: Presupuesto pendiente*

- Existe un presupuesto PENDIENTE para este vehículo
  - El sistema muestra advertencia: "Existe presupuesto pendiente sin aprobar"
  - El sistema pregunta: "¿Desea vincular el presupuesto PENDIENTE?"
  - Si acepta: vincula el presupuesto (aunque no esté aprobado)
  - Si rechaza: registra sin presupuesto

*FA-03: Vehículo ya en taller*

- El vehículo ya tiene una reparación activa (no entregada)
  - El sistema muestra error: "El vehículo ya está en el taller"
  - El sistema muestra: "Reparación #[ID] - Estado: [ESTADO]"
  - El sistema ofrece: "¿Desea ver la reparación existente?"
  - Si acepta: abre la reparación existente
  - Fin del caso de uso

*FA-04: Empleado no disponible*

- Todos los empleados están al máximo de carga de trabajo
  - El sistema muestra advertencia: "Todos los empleados tienen carga alta"
  - El sistema muestra carga de trabajo de cada empleado
  - Permite asignar de todas formas o poner en cola sin asignar

*FA-05: Campos obligatorios vacíos*

- Faltan datos obligatorios
  - El sistema marca en rojo los campos vacíos
  - El sistema muestra mensaje: "Complete los campos obligatorios"
  - Vuelve al paso 3

**CU-09.2: Asignar/Reasignar Empleado a Reparación**

**Flujo Principal:**

1. El empleado selecciona una reparación de la tabla
2. El empleado hace clic en "Asignar/Cambiar Empleado"
3. El sistema muestra un diálogo con:
  - Empleado actual (si hay)
  - Lista de empleados disponibles con su carga de trabajo:

- Nombre del empleado
  - Reparaciones activas asignadas
  - Indicador visual de carga (Verde/Amarillo/Rojo)
4. El empleado selecciona el nuevo mecánico responsable
  5. El empleado puede añadir un comentario sobre el cambio (opcional)
  6. El empleado hace clic en "Confirmar Asignación"
  7. El sistema actualiza el campo empleado\_asignado\_id
  8. El sistema registra el cambio en el historial de la reparación
  9. El sistema muestra mensaje: "Mecánico asignado correctamente"
  10. El sistema notifica al nuevo empleado asignado (futuro)

#### **Flujos Alternativos:**

##### *FA-01: Reparación ya finalizada*

- La reparación está en estado FINALIZADA o ENTREGADA
  - El sistema muestra error: "No se puede reasignar una reparación finalizada"
  - Fin del caso de uso

##### *FA-02: Mismo empleado*

- Se selecciona el mismo empleado ya asignado
  - El sistema muestra advertencia: "Este empleado ya está asignado"
  - Permite continuar o cancelar

#### **CU-09.3: Iniciar Trabajo en Reparación**

#### **Flujo Principal:**

1. El empleado selecciona una reparación con estado EN\_COLA
2. El empleado hace clic en "Iniciar Trabajo"
3. El sistema verifica:
  - Estado actual = EN\_COLA o PAUSADA
  - Empleado asignado existe
4. El sistema muestra confirmación:
  - "¿Iniciar trabajo en Reparación #[ID]?"
  - Datos del vehículo
  - Diagnóstico inicial
5. El empleado confirma

6. El sistema actualiza:
  - Estado → EN\_PROCESO
  - Registra fecha/hora de inicio de trabajo
7. El sistema muestra mensaje: "Trabajo iniciado"
8. El sistema actualiza la tabla con indicador visual (color amarillo)
9. El sistema habilita las opciones:
  - Registrar trabajo realizado
  - Registrar productos usados
  - Pausar trabajo
  - Finalizar trabajo

#### **Flujos Alternativos:**

*FA-01: Reparación sin empleado asignado*

- No hay empleado asignado
  - El sistema muestra error: "Debe asignar un empleado primero"
  - El sistema redirige a CU-09.2
  - Vuelve al paso 1

*FA-02: Estado no válido*

- La reparación está EN\_PROCESO, FINALIZADA o ENTREGADA
  - El sistema muestra error: "La reparación no puede ser iniciada en estado [ESTADO]"
  - Fin del caso de uso

#### **CU-09.4: Registrar Trabajo Realizado (Mano de Obra)**

##### **Flujo Principal:**

1. El empleado selecciona una reparación EN\_PROCESO o FINALIZADA
2. El empleado hace clic en "Añadir Trabajo Realizado"
3. El sistema muestra un formulario modal con:
  - Empleado que realizó el trabajo (combo con empleados activos)
  - Descripción del trabajo (ej: "Cambio de pastillas de freno delanteras")

- Fecha/hora inicio (opcional, prellenado con fecha actual)
  - Fecha/hora fin (opcional)
  - Tiempo empleado en horas (obligatorio, ej: 2.5)
  - Tarifa por hora (prellenado con tarifa del empleado, modificable)
  - Subtotal (calculado automáticamente: tiempo × tarifa)
4. El empleado completa los campos
  5. El sistema calcula el subtotal en tiempo real
  6. El empleado hace clic en "Añadir"
  7. El sistema valida:
    - Tiempo empleado > 0
    - Tarifa > 0
    - Empleado seleccionado existe
  8. El sistema guarda la línea en Reparacion\_Detalle\_Mano\_Obra
  9. El sistema añade la línea a la tabla de trabajos realizados
  10. El sistema recalcula el costo total de la reparación
  11. El sistema muestra mensaje: "Trabajo añadido correctamente"
  12. El empleado puede seguir añadiendo más líneas (repetir desde paso 2)

### **Flujos Alternativos:**

#### *FA-01: Tiempo inválido*

- El tiempo empleado es <= 0
  - El sistema muestra error: "El tiempo debe ser mayor que cero"
  - Vuelve al paso 4

#### *FA-02: Tarifa inválida*

- La tarifa es <= 0
  - El sistema muestra error: "La tarifa debe ser mayor que cero"
  - Vuelve al paso 4

#### *FA-03: Modificar línea existente*

- El empleado selecciona una línea de la tabla
- El sistema carga los datos en el formulario
- El empleado modifica los datos
- El empleado hace clic en "Actualizar"
- El sistema valida y actualiza la línea

#### *FA-04: Eliminar línea*

- El empleado selecciona una línea
- El empleado hace clic en "Eliminar"
- El sistema muestra confirmación
- Si confirma: elimina la línea y recalcula totales

**FA-05: Cancelar**

- El empleado hace clic en "Cancelar"
  - El sistema cierra el formulario sin guardar
  - Vuelve al paso 1

**CU-09.5: Registrar Productos Usados**

**Flujo Principal:**

1. El empleado selecciona una reparación EN PROCESO o FINALIZADA
2. El empleado hace clic en "Añadir Producto"
3. El sistema muestra un diálogo de búsqueda de productos:
  - Campo de búsqueda (por código o nombre)
  - Tabla con resultados:
    - ID Producto
    - Nombre
    - Categoría
    - Stock disponible
    - Precio unitario
  - Filtros: Por categoría, por proveedor, solo con stock
4. El empleado busca el producto
5. El sistema muestra los productos que coinciden
6. El empleado selecciona un producto de la tabla
7. El sistema muestra un formulario con:
  - Producto seleccionado (código, nombre)
  - Stock actual disponible
  - Cantidad a usar (obligatorio)
  - Precio venta unitario (prellenado, modificable)
  - Descuento % (opcional, default 0)
  - Subtotal (calculado: cantidad × precio × (1 - descuento/100))
8. El empleado ingresa la cantidad

9. El sistema valida:

- Cantidad > 0
- Cantidad <= stock disponible
- Precio > 0

10. El sistema calcula el subtotal en tiempo real

11. El empleado hace clic en "Añadir"

12. El sistema guarda en Reparacion\_Detalle\_Productos

13. El sistema descuenta automáticamente el stock (trigger de BD)

14. El sistema registra el movimiento en Movimientos\_Stock:

- Tipo: SALIDA
- Motivo: "Usado en reparación #[ID]"

15. El sistema añade el producto a la tabla de productos usados

16. El sistema recalcula el costo total de la reparación

17. El sistema muestra mensaje: "Producto añadido - Stock actualizado"

18. El empleado puede seguir añadiendo más productos (repetir desde paso 2)

### **Flujos Alternativos:**

#### *FA-01: Stock insuficiente*

- La cantidad solicitada excede el stock disponible
  - El sistema muestra error: "Stock insuficiente. Disponible: [X] unidades"
  - El sistema ofrece opciones:
    - Opción A: Reducir cantidad al stock disponible
    - Opción B: Registrar de todas formas (requiere confirmación, no descuenta stock)
    - Opción C: Ir a actualizar stock (abre CU-07.2)
    - Opción D: Cancelar
  - Si elige B: se añade con stock\_descontado = FALSE
  - Si elige C: redirige a CU-07.2, luego vuelve aquí

#### *FA-02: Producto ya añadido*

- El producto ya está en la lista de productos usados
  - El sistema pregunta: "Este producto ya fue añadido. ¿Desea aumentar cantidad?"
  - Si acepta: suma la nueva cantidad a la existente
  - Si rechaza: permite seleccionar otro producto

*FA-03: Producto sin stock*

- El producto seleccionado tiene stock = 0
  - El sistema muestra advertencia: "Producto sin stock"
  - El sistema ofrece:
    - Opción A: Registrar de todas formas (sin descontar stock)
    - Opción B: Actualizar stock primero
  - Si elige A: requiere confirmación adicional

*FA-04: Modificar cantidad de producto añadido*

- El empleado selecciona un producto de la tabla
- El empleado modifica la cantidad
- El sistema:
  - Calcula diferencia: cantidad\_nueva - cantidad\_anterior
  - Ajusta el stock según la diferencia
  - Registra movimiento de ajuste

*FA-05: Eliminar producto usado*

- El empleado selecciona un producto de la tabla
- El empleado hace clic en "Eliminar"
- El sistema muestra confirmación: "¿Devolver producto al stock?"
- Si confirma:
  - Elimina la línea
  - **Devuelve el stock** (stock += cantidad\_usada)
  - Registra movimiento: ENTRADA (devolución)
  - Recalcula totales

*FA-06: Sin resultados de búsqueda*

- La búsqueda no produce resultados
  - El sistema muestra: "No se encontraron productos"
  - Permite modificar búsqueda o crear nuevo producto

**CU-09.6: Cambiar Estado de Reparación**

**Flujo Principal:**

1. El empleado selecciona una reparación de la tabla

2. El empleado hace clic en el combo "Estado" o en botones de acción rápida
3. El sistema muestra las opciones de estado válidas según estado actual:
  - Desde **EN\_COLA**: → EN\_PROCESO
  - Desde **EN\_PROCESO**: → PAUSADA, FINALIZADA
  - Desde **PAUSADA**: → EN\_PROCESO, FINALIZADA
  - Desde **FINALIZADA**: → ENTREGADA
  - **ENTREGADA**: Estado final, no permite cambios
4. El empleado selecciona el nuevo estado
5. Si selecciona **PAUSADA**:
  - El sistema solicita motivo de pausa (obligatorio)
  - Opciones comunes: "Esperando piezas", "Esperando autorización cliente", "Otro"
6. Si selecciona **FINALIZADA**:
  - El sistema verifica:
    - Hay al menos una línea de mano de obra registrada
    - La fecha de salida real se puede registrar (opcional)
  - El sistema pregunta: "¿Registrar fecha/hora de finalización?"
  - Si acepta: registra fecha\_salida\_real = NOW()
7. Si selecciona **ENTREGADA**:
  - El sistema verifica:
    - Estado actual = FINALIZADA
    - Existe factura asociada y está PAGADA (obligatorio)
  - El sistema solicita:
    - Usuario que entrega (prellenado con usuario actual)
    - Confirmación del cliente (checkbox)
  - El sistema actualiza el kilometraje del vehículo (si es mayor)
8. El empleado confirma el cambio
9. El sistema actualiza el campo estado en la tabla Reparaciones
10. El sistema registra el cambio en el historial con:
  - Usuario que realizó el cambio
  - Fecha/hora del cambio
  - Estado anterior → Estado nuevo
11. El sistema muestra mensaje: "Estado actualizado a [NUEVO\_ESTADO]"
12. El sistema actualiza el indicador visual en la tabla (colores):
  - EN\_COLA: Gris
  - EN\_PROCESO: Amarillo

- PAUSADA: Naranja
- FINALIZADA: Azul
- ENTREGADA: Verde

### **Flujos Alternativos:**

#### *FA-01: Cambio no permitido*

- El cambio de estado solicitado no es válido
  - El sistema muestra error: "No se puede cambiar de [ESTADO\_ACTUAL] a [ESTADO\_NUEVO]"
  - Muestra el flujo válido de estados
  - Fin del caso de uso

#### *FA-02: Finalizar sin trabajos registrados*

- Se intenta finalizar sin líneas de mano de obra
  - El sistema muestra advertencia: "No hay trabajos registrados"
  - El sistema pregunta: "¿Finalizar de todas formas?"
  - Si rechaza: vuelve al paso 2
  - Si acepta: requiere confirmación adicional

#### *FA-03: Entregar sin factura*

- La reparación no tiene factura asociada
  - El sistema muestra error: "Debe facturar antes de entregar"
  - El sistema ofrece: "¿Desea facturar ahora?"
  - Si acepta: abre CU-05.1 (Facturar)
  - Si rechaza: vuelve al paso 2

#### *FA-04: Entregar con factura pendiente*

- La factura existe pero no está pagada
  - El sistema muestra advertencia: "La factura no está cobrada"
  - El sistema pregunta: "¿Entregar de todas formas?"
  - Si acepta: requiere autorización (solo admin puede confirmar)
  - Si rechaza: vuelve al paso 2

### **CU-09.7: Finalizar Reparación**

#### **Flujo Principal:**

1. El empleado selecciona una reparación EN\_PROCESO o PAUSADA

2. El empleado hace clic en "Finalizar Reparación"
3. El sistema verifica:
  - Hay al menos una línea de mano de obra
  - Hay al menos un producto usado (opcional, advertencia)
4. El sistema muestra un diálogo de confirmación con resumen:
  - **Datos del vehículo:** Matrícula, marca, modelo
  - **Tiempo en taller:** Días transcurridos desde entrada
  - **Resumen de trabajos:**
    - Número de líneas de mano de obra
    - Total horas trabajadas
    - Costo mano de obra
  - **Resumen de productos:**
    - Cantidad de productos usados
    - Costo total productos
  - **Costo total estimado:** Mano obra + Productos
  - **Comparación con presupuesto** (si existe):
    - Presupuestado: [X]€
    - Real: [Y]€
    - Diferencia: [Z]€ (positiva o negativa)
5. El empleado revisa el resumen
6. El empleado puede:
  - Añadir observaciones finales sobre el trabajo (opcional)
  - Registrar fecha/hora de finalización (prellenado con actual)
7. El empleado hace clic en "Confirmar Finalización"
8. El sistema actualiza:
  - Estado → FINALIZADA
  - fecha\_salida\_real = fecha indicada
  - trabajos\_realizados = texto de observaciones
9. El sistema muestra mensaje: "Reparación finalizada correctamente"
10. El sistema habilita el botón "Generar Factura"
11. El sistema deshabilita:
  - Añadir más trabajos
  - Añadir más productos
  - Cambiar empleado asignado
12. El sistema actualiza estadísticas del empleado
13. El sistema envía notificación al cliente (futuro): "Su vehículo está listo"

**Flujos Alternativos:**
*FA-01: Sin trabajos registrados*

- No hay líneas de mano de obra
  - El sistema muestra error: "Debe registrar al menos un trabajo realizado"
  - El sistema ofrece: "¿Desea añadir ahora?"
  - Si acepta: abre CU-09.4
  - Si rechaza: cancela la finalización

*FA-02: Sin productos usados*

- No hay productos registrados
  - El sistema muestra advertencia: "No se han registrado productos"
  - El sistema pregunta: "¿Es correcto? (Reparación solo de mano de obra)"
  - Si confirma: continúa
  - Si rechaza: vuelve al paso 1

*FA-03: Diferencia significativa con presupuesto*

- El costo real excede el presupuesto en más del 20%
  - El sistema muestra alerta: "El costo real supera el presupuesto en [X]%"
  - El sistema sugiere: "Contacte al cliente para autorización"
  - Requiere confirmación adicional para continuar

*FA-04: Cancelar finalización*

- El empleado hace clic en "Cancelar"
  - El sistema descarta los cambios
  - La reparación sigue en estado actual
  - Vuelve al paso 1

**CU-09.8: Entregar Vehículo al Cliente**
**Flujo Principal:**

1. El empleado selecciona una reparación FINALIZADA
2. El empleado verifica que la factura está cobrada
3. El empleado hace clic en "Entregar Vehículo"
4. El sistema verifica:
  - Estado = FINALIZADA

- Factura existe y estado = PAGADA
5. El sistema muestra un diálogo de entrega con:
- **Datos del vehículo:** Matrícula, bastidor, marca, modelo
  - **Datos del cliente:** Nombre, teléfono
  - **Checklist de entrega:**
    - [ ] Vehículo lavado
    - [ ] Documentación revisada
    - [ ] Cliente informado de trabajos realizados
    - [ ] Cliente firma conforme
  - **Kilometraje de salida** (prellenado con el de entrada)
  - **Observaciones de entrega** (opcional)
  - **Firma digital del cliente** (futuro)
6. El empleado completa el checklist
7. El empleado puede actualizar el kilometraje si es diferente
8. El empleado añade observaciones si es necesario
9. El empleado hace clic en "Confirmar Entrega"
10. El sistema valida que todos los items del checklist están marcados
11. El sistema actualiza:
- Estado → ENTREGADA
  - usuario\_entrega = usuario actual
  - Fecha/hora de entrega
12. El sistema actualiza el kilometraje del vehículo en la tabla Vehiculos:
- Si kilometraje\_salida > kilometraje\_actual: actualiza
13. El sistema registra la entrega en el historial
14. El sistema muestra mensaje: "Vehículo entregado correctamente"
15. El sistema remueve el vehículo del contador "Vehículos en Taller"
16. El sistema genera un comprobante de entrega (PDF) con:
- Datos del vehículo
  - Trabajos realizados (resumen)
  - Fecha de entrega
  - Firma del empleado y cliente
17. El sistema ofrece: "¿Desea imprimir el comprobante?"

#### **Flujos Alternativos:**

*FA-01: Factura no cobrada*

- La factura existe pero no está pagada

- El sistema muestra error: "La factura debe estar cobrada antes de entregar"
- El sistema muestra datos de la factura:
  - Número de factura
  - Total
  - Estado: PENDIENTE
- El sistema ofrece opciones:
  - Opción A: "Registrar cobro ahora" (abre CU-05.3)
  - Opción B: "Entregar sin cobrar" (solo Administrador)
- Si elige A: redirige a CU-05.3, luego vuelve aquí
- Si elige B: requiere contraseña de administrador

*FA-02: Sin factura*

- La reparación no tiene factura asociada
  - El sistema muestra error: "Debe facturar antes de entregar"
  - El sistema ofrece: "¿Desea facturar ahora?"
  - Si acepta: abre CU-05.1
  - Si rechaza: cancela la entrega

*FA-03: Checklist incompleto*

- No todos los items están marcados
  - El sistema muestra advertencia: "Complete el checklist de entrega"
  - Permite completar o entregar de todas formas (con confirmación)

*FA-04: Cliente no presente*

- El empleado indica que el cliente no está presente
  - El sistema pregunta: "¿Entrega autorizada por teléfono?"
  - Si sí: permite continuar sin firma
  - Si no: cancela la entrega

*FA-05: Cancelar entrega*

- El empleado hace clic en "Cancelar"
  - El sistema descarta el proceso
  - La reparación sigue FINALIZADA
  - Vuelve al paso 1

**CU-09.9: Consultar Detalles de Reparación**

**Flujo Principal:**

1. El empleado selecciona una reparación de la tabla
2. El empleado hace clic en "Ver Detalles" o doble clic en la fila
3. El sistema abre una ventana modal con pestañas:
  - **Pestaña "Datos Generales":**
    - ID Reparación, Estado, Prioridad
    - Cliente (con enlace a CU-02)
    - Vehículo (con enlace a CU-03)
    - Empleado asignado
    - Presupuesto asociado (si existe, con enlace)
    - Factura asociada (si existe, con enlace)
    - Fechas: entrada, salida estimada, salida real
    - Kilometraje entrada, nivel combustible
    - Diagnóstico inicial
  - **Pestaña "Trabajos Realizados":**
    - Tabla con líneas de mano de obra:
      - Empleado, Descripción, Horas, Tarifa, Subtotal
    - Total mano de obra
  - **Pestaña "Productos Usados":**
    - Tabla con productos:
      - Código, Nombre, Cantidad, Precio unit., Descuento, Subtotal
    - Total productos
  - **Pestaña "Costos":**
    - Desglose completo:
      - Mano de obra: [X]€
      - Productos: [Y]€
      - Subtotal: [X+Y]€
      - IVA (21%): [Z]€
      - Total: [TOTAL]€
    - Si hay presupuesto:
      - Comparación presupuesto vs real
      - Diferencia en € y %
  - **Pestaña "Historial":**
    - Cronología de cambios de estado
    - Usuario, Fecha/hora, Estado anterior → nuevo
    - Observaciones de cada cambio

4. El empleado puede:
  - Imprimir la orden de trabajo (PDF)
  - Exportar detalles a Excel
  - Copiar información al portapapeles
5. El empleado cierra la ventana

**Flujos Alternativos:**

*FA-01: Reparación en curso*

- La reparación está EN PROCESO
  - Todos los datos son editables
  - Muestra botones de acción: Pausar, Finalizar

*FA-02: Reparación finalizada*

- La reparación está FINALIZADA o ENTREGADA
  - Los datos son solo lectura
  - Muestra botón "Generar Factura" (si no hay factura)

**CU-10: Gestionar Empleados**

**Actor:** Administrador

**Descripción:** Permite al administrador gestionar los empleados del taller, incluyendo sus datos personales, profesionales, asignación de usuarios del sistema y control de altas/bajas.

**Precondiciones:**

- El usuario debe haber iniciado sesión con rol ADMINISTRADOR
- El panel de Empleados debe estar activo

**Postcondiciones:**

- Los datos del empleado se almacenan/modifican en la base de datos
- Se puede vincular un empleado con un usuario del sistema
- Se mantiene el historial de altas y bajas

**CU-10.1: Registrar Nuevo Empleado****Flujo Principal:**

1. El administrador hace clic en "Nuevo Empleado"
2. El sistema muestra un formulario con tres secciones:
  - **Sección A:** Datos personales
  - **Sección B:** Datos profesionales
  - **Sección C:** Acceso al sistema (opcional)
3. **Sección A - Datos Personales:**
  - El administrador ingresa:
    - DNI (obligatorio, único)
    - Nombre (obligatorio)
    - Apellidos (obligatorio)
    - Teléfono
    - Email (opcional, único)
    - Dirección
4. **Sección B - Datos Profesionales:**
  - El administrador ingresa:
    - Cargo (obligatorio): Mecánico, Jefe de Taller, Recepcionista, etc.
    - Fecha de alta (obligatorio, prellenado con fecha actual)
    - Salario base (opcional, confidencial)
    - Foto de perfil (opcional, upload de imagen)
5. **Sección C - Acceso al Sistema:**
  - El administrador puede:
    - **Opción A:** Vincular a un usuario existente (combo)
    - **Opción B:** Crear nuevo usuario vinculado (abre CU-08.1)
    - **Opción C:** No dar acceso al sistema (usuario\_id = NULL)
  - Si selecciona crear usuario:
    - Se abre formulario de creación de usuario
    - El username se sugiere automáticamente: primera letra nombre + apellidos (ej: jcardenas)
    - Se genera contraseña temporal
6. El administrador hace clic en "Guardar Empleado"
7. El sistema valida:
  - Formato DNI correcto (8 dígitos + letra)
  - DNI no existe en la base de datos
  - Email no existe (si se proporciona)
  - Cargo válido

- Fecha de alta <= fecha actual
8. El sistema guarda el empleado en la base de datos
9. Si se creó usuario vinculado:
- El sistema guarda el usuario en tabla Usuarios
  - El sistema vincula: empleado.usuario\_id = usuario.id\_usuario
10. El sistema muestra mensaje: "Empleado registrado correctamente"
11. Si se creó usuario, muestra: "Usuario: [username] - Contraseña temporal: [XXX]"
12. El sistema sugiere: "Imprima esta información para entregársela al empleado"
13. El sistema actualiza la tabla de empleados
14. El sistema limpia el formulario

### **Flujos Alternativos:**

#### *FA-01: DNI duplicado*

- El DNI ya existe en la base de datos
  - El sistema muestra error: "Ya existe un empleado con este DNI"
  - El sistema muestra los datos del empleado existente
  - El sistema pregunta: "¿Es un reingreso? (empleado dado de baja anteriormente)"
  - Si sí: carga datos del empleado inactivo para reactivar
  - Si no: vuelve al paso 3

#### *FA-02: Email duplicado*

- El email ya está registrado
  - El sistema muestra advertencia: "Este email ya está en uso"
  - El sistema muestra: "Usado por empleado: [Nombre]"
  - Permite modificar o continuar sin email

#### *FA-03: Formato DNI inválido*

- El formato del DNI es incorrecto
  - El sistema muestra mensaje: "Formato DNI inválido (ej: 12345678A)"
  - Resalta el campo en rojo
  - Vuelve al paso 3

#### *FA-04: Salario inválido*

- El salario es negativo
  - El sistema muestra error: "El salario debe ser mayor o igual a cero"

- Vuelve al paso 4

*FA-05: Error al crear usuario*

- Falla la creación del usuario vinculado
  - El sistema guarda el empleado sin usuario
  - El sistema muestra advertencia: "Empleado creado pero sin usuario"
  - El sistema sugiere: "Puede crear el usuario después en CU-08"

*FA-06: Cancelar registro*

- El administrador hace clic en "Cancelar"
  - El sistema muestra confirmación: "¿Descartar datos del empleado?"
  - Si confirma: limpia el formulario sin guardar
  - Si cancela: vuelve al paso 3

### **CU-10.2: Modificar Datos de Empleado**

#### **Flujo Principal:**

1. El administrador busca un empleado (por DNI o nombre)
2. El administrador selecciona un empleado de la tabla
3. El sistema carga los datos en el formulario
4. El sistema deshabilita el campo DNI (no modificable - PK)
5. El administrador puede modificar:
  - Nombre, apellidos
  - Teléfono, email, dirección
  - Cargo
  - Salario base
  - Foto de perfil (cambiar imagen)
  - Usuario vinculado (cambiar o desvincular)
6. El administrador hace clic en "Guardar Cambios"
7. El sistema valida los datos modificados
8. El sistema actualiza el registro en la base de datos
9. El sistema registra la modificación en el historial (auditoría)
10. El sistema muestra mensaje: "Empleado actualizado correctamente"
11. El sistema actualiza la tabla

#### **Flujos Alternativos:**

*FA-01: Email duplicado*

- El nuevo email ya pertenece a otro empleado
  - El sistema muestra advertencia: "Email en uso por otro empleado"
  - Permite modificar o conservar el anterior

*FA-02: Cambiar usuario vinculado*

- El administrador desvincula el usuario actual
  - El sistema pregunta: "¿Desea desactivar también el usuario?"
  - Si sí: marca el usuario como activo=FALSE
  - Si no: el usuario queda sin empleado vinculado
- El administrador vincula a otro usuario
  - El sistema verifica que el usuario no esté vinculado a otro empleado
  - Si está vinculado: muestra error y no permite
  - Si está libre: actualiza la vinculación

*FA-03: Modificar empleado inactivo*

- El empleado seleccionado está inactivo (fecha\_baja != NULL)
  - El sistema muestra advertencia: "Empleado dado de baja el [FECHA]"
  - El sistema pregunta: "¿Desea reactivarlo?"
  - Si sí: continúa con la modificación y limpia fecha\_baja
  - Si no: solo permite consulta (lectura)

*FA-04: Cancelar modificación*

- El administrador hace clic en "Cancelar"
  - El sistema descarta los cambios
  - El sistema recarga los datos originales

**CU-10.3: Dar de Baja Empleado**

**Flujo Principal:**

1. El administrador selecciona un empleado activo de la tabla
2. El administrador hace clic en "Dar de Baja"
3. El sistema muestra un diálogo de confirmación con:
  - Datos del empleado (nombre, cargo)
  - Advertencias:

- "Esta acción no elimina al empleado del sistema"
- "Solo marca como inactivo manteniendo el historial"
- Campo obligatorio: Motivo de la baja
  - Opciones: Renuncia, Despido, Jubilación, Otro
- Campo obligatorio: Fecha de baja (llenado con fecha actual)
- Campo opcional: Observaciones

4. El sistema verifica:

- El empleado tiene reparaciones activas asignadas

5. Si tiene reparaciones activas:

- El sistema muestra advertencia: "El empleado tiene [X] reparaciones activas"
- El sistema lista las reparaciones:
  - ID Reparación, Vehículo (matrícula), Estado
- El sistema pregunta: "¿Reasignar reparaciones a otro empleado?"
- Si acepta: muestra lista de empleados disponibles
- El administrador selecciona empleado sustituto para cada reparación

6. El administrador completa los campos obligatorios

7. El administrador hace clic en "Confirmar Baja"

8. El sistema actualiza:

- fecha\_baja = fecha indicada
- activo = FALSE
- Si tiene usuario vinculado: marca usuario.activo = FALSE

9. Si había reparaciones:

- El sistema reasigna las reparaciones al empleado sustituto
- Registra los cambios en el historial de cada reparación

10. El sistema registra la baja en el historial del empleado

11. El sistema muestra mensaje: "Empleado dado de baja correctamente"

12. El sistema actualiza la tabla (el empleado aparece inactivo)

13. El sistema ofrece: "¿Desea imprimir certificado de baja?"

### Flujos Alternativos:

FA-01: Sin motivo de baja

- No se ha seleccionado motivo
  - El sistema marca el campo en rojo
  - El sistema muestra mensaje: "El motivo de baja es obligatorio"
  - Vuelve al paso 3

*FA-02: Fecha de baja futura*

- La fecha de baja es posterior a la fecha actual
  - El sistema muestra advertencia: "Fecha de baja futura"
  - El sistema pregunta: "¿Programar baja para esa fecha?"
  - Si acepta: guarda pero mantiene activo=TRUE hasta esa fecha
  - Si rechaza: permite cambiar la fecha

*FA-03: Empleado sin reparaciones*

- El empleado no tiene reparaciones activas
  - Salta el paso 5
  - Continúa directamente al paso 6

*FA-04: No se pueden reasignar reparaciones*

- No hay otros empleados disponibles
  - El sistema muestra error: "No hay empleados disponibles para reasignar"
  - El sistema sugiere:
    - Opción A: Crear nuevo empleado primero
    - Opción B: Finalizar reparaciones antes de dar de baja
  - Cancela la baja

*FA-05: Cancelar baja*

- El administrador hace clic en "Cancelar"
  - El sistema descarta el proceso
  - El empleado sigue activo
  - Vuelve al paso 1

*FA-06: Reactivar empleado*

- Si más tarde se necesita reactivar:
  - El administrador modifica el empleado (CU-10.2)
  - Limpia fecha\_baja = NULL
  - Marca activo = TRUE
  - El sistema pregunta si reactivar también el usuario

**CU-10.4: Consultar Carga de Trabajo por Empleado**

**Flujo Principal:**

1. El administrador hace clic en "Carga de Trabajo"
2. El sistema muestra un diálogo con:
  - Filtros:
    - Fecha desde / hasta (prellenado: mes actual)
    - Solo empleados activos (checkbox, marcado por defecto)
    - Cargo (todos, o filtrar por cargo específico)
  - Botón "Generar Informe"
3. El administrador ajusta los filtros si es necesario
4. El administrador hace clic en "Generar Informe"
5. El sistema consulta la base de datos y calcula:
  - Para cada empleado:
    - Reparaciones asignadas en el período
    - Reparaciones finalizadas vs asignadas (%)
    - Total horas de mano de obra realizadas
    - Reparaciones activas actualmente
    - Tiempo promedio por reparación
    - Valoración de carga: Baja / Media / Alta / Muy Alta
6. El sistema muestra una tabla con:
  - Foto, Nombre, Cargo
  - Rep. Asignadas, Rep. Finalizadas, % Finalización
  - Horas Trabajadas, Promedio días/reparación
  - Rep. Activas Ahora
  - Indicador de Carga (color: Verde/Amarillo/Rojo)
7. El sistema ordena por carga de trabajo (mayor a menor)
8. El administrador puede:
  - Hacer clic en un empleado para ver detalle:
    - Lista de todas sus reparaciones
    - Gráficos de productividad
  - Exportar el informe a Excel/PDF
  - Imprimir el informe
9. El administrador cierra el diálogo

**Flujos Alternativos:**

*FA-01: Sin datos en el período*

- No hay reparaciones en el período seleccionado
  - El sistema muestra mensaje: "No hay datos para el período seleccionado"
  - Permite modificar los filtros
  - Vuelve al paso 3

*FA-02: Ver detalle de empleado*

- El administrador hace clic en un empleado
  - El sistema abre una ventana modal con:
    - Gráfico de reparaciones por mes
    - Gráfico de tipos de trabajo realizados
    - Lista detallada de reparaciones:
      - Fecha entrada, Vehículo, Cliente, Estado, Horas, Días en taller
    - Estadísticas:
      - Tiempo promedio por reparación
      - Reparación más rápida / más lenta
      - Productos más utilizados
  - El administrador puede imprimir este detalle

*FA-03: Comparar empleados*

- El administrador selecciona varios empleados (checkbox)
  - El administrador hace clic en "Comparar"
  - El sistema muestra gráficos comparativos:
    - Reparaciones finalizadas (barras)
    - Horas trabajadas (barras)
    - Tiempo promedio (líneas)
    - Tasa de finalización (%)

### **CU-10.5: Consultar Historial de Empleado**

#### **Flujo Principal:**

1. El administrador selecciona un empleado de la tabla
2. El administrador hace clic en "Ver Historial"
3. El sistema abre una ventana modal con pestañas:
  - **Pestaña "Datos Generales":**
    - Foto, Nombre completo, DNI
    - Cargo actual, Fecha de alta

- Usuario vinculado (si existe)
  - Estado: Activo / Inactivo
  - Antigüedad: [X] años [Y] meses
  - **Pestaña "Reparaciones":**
    - Historial completo de reparaciones asignadas:
      - Fecha, Vehículo, Cliente, Estado, Resultado
    - Totales:
      - Reparaciones asignadas: [X]
      - Reparaciones finalizadas: [Y]
      - Tasa de éxito: [Z]%
  - **Pestaña "Trabajos Realizados":**
    - Detalles de mano de obra registrada:
      - Fecha, Reparación, Descripción, Horas, Costo
    - Totales:
      - Total horas trabajadas
      - Total facturado por su trabajo
  - **Pestaña "Estadísticas":**
    - Gráficos:
      - Reparaciones por mes (líneas)
      - Tipos de trabajo realizados (torta)
      - Evolución de productividad (barras)
    - Promedios:
      - Tiempo promedio por reparación
      - Horas promedio por semana
  - **Pestaña "Cambios y Eventos":**
    - Cronología de cambios:
      - Cambios de cargo
      - Cambios de salario
      - Períodos de baja (si aplica)
      - Modificaciones de datos
    - Usuario que realizó cada cambio
    - Fecha/hora de cada cambio
4. El administrador puede:
- Exportar historial completo a PDF
  - Imprimir certificado laboral
  - Generar carta de recomendación (plantilla)

5. El administrador cierra la ventana

### Flujos Alternativos:

#### FA-01: Empleado sin historial

- El empleado no tiene reparaciones asignadas
  - Las pestañas "Reparaciones" y "Trabajos" muestran: "Sin datos"
  - Solo muestra datos generales y eventos

#### FA-02: Empleado inactivo

- El empleado está dado de baja
  - El sistema muestra en rojo: "EMPLEADO INACTIVO"
  - Muestra: "Dado de baja el: [FECHA]"
  - Muestra: "Motivo: [MOTIVO]"
  - Todos los datos son de solo lectura

### 1.3.3. Matriz de Trazabilidad Requisitos - Casos de Uso

Tabla 3: Trazabilidad Requisitos - Casos de Uso

Requisito	Caso de Uso	Descripción
RF-01.1	CU-01	Autenticación de usuarios
RF-01.2	CU-08	Gestión de roles
RF-02.1	CU-02.1, CU-02.3, CU-02.4	ABM de clientes
RF-02.2	CU-02.2	Búsqueda avanzada de clientes
RF-02.3	CU-02.5	Historial de servicios por cliente
RF-03.1	CU-03.1	Registro de vehículos
RF-03.2	CU-03.1, CU-03.4	Almacenamiento de datos técnicos
RF-03.3	CU-03.3	Historial de reparaciones por vehículo
RF-04.1	CU-06.1, CU-06.2, CU-06.3	ABM de proveedores
RF-04.2	CU-07.1, CU-07.2, CU-07.3	Control de suministros
RF-05.1	CU-04.1	Creación de presupuestos detallados
RF-05.2	CU-04.3	Cambio de estado de presupuesto
RF-05.3	CU-05.1	Conversión de presupuesto a factura
RF-05.4	CU-05.1	Emisión de facturas con numeración
RF-05.5	CU-05.1	Generación e impresión de PDF
RF-06.1	CU-09.1	Registro de entrada de vehículos al taller
RF-06.2	CU-09.2	Asignación de mecánicos a reparaciones
RF-06.3	CU-09.6	Gestión de estados de reparación
RF-06.4	CU-09.4	Registro de trabajo real realizado
RF-06.5	CU-09.5	Registro de productos usados
RF-06.6	CU-09.5	Descuento automático de stock
RF-06.7	CU-09.7	Finalización y validación de reparaciones
RF-06.8	CU-09.8	Entrega de vehículos con checklist
RF-07.1	CU-10.1	Registro de empleados
RF-07.2	CU-10.1	Vinculación empleado-usuario
RF-07.3	CU-10.3	Control de altas y bajas de personal
RF-07.4	CU-10.4	Ánalysis de carga de trabajo
RF-07.5	CU-10.5	Consulta de historial de empleado
RF-08.1	CU-07.2, Triggers	Auditoría de movimientos de stock



## ANÁLISIS Y DISEÑO

IES AUGUSTOBRIGA

G.S. DAM

Pág.  
69/123

**Requisito**

RF-08.2 CU-07.5

**Caso de Uso**

Alertas de stock bajo

**Descripción**

## 2. Diseño del Sistema

En esta sección se detalla **cómo** se construirá la aplicación para cumplir con los requisitos del análisis.

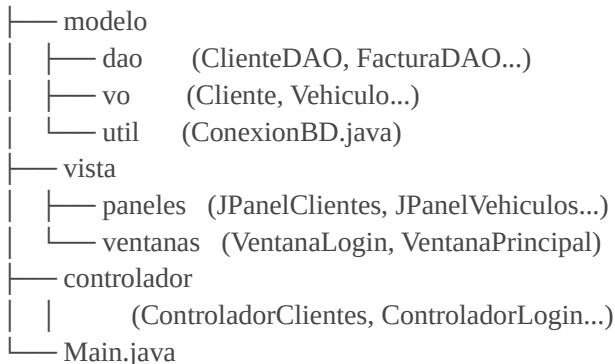
### 2.1. Arquitectura del Software

El proyecto utilizará una arquitectura **Modelo-Vista-Controlador (MVC)**. Esta arquitectura separa la lógica de negocio, los datos y la interfaz de usuario en tres componentes interconectados.

- **Modelo**
  - **Clases de Entidad (POJOs)**: Cliente, Vehiculo, Factura, Producto, etc.
  - **Clases DAO (Data Access Object)**: ClienteDAO, VehiculoDAO, etc. Serán las únicas clases responsables de interactuar con la base de datos (consultas SQL con JDBC).
- **Vista**: Es la interfaz de usuario.
  - **Clases Swing (JFrame, JPanel)**: VentanaLogin, VentanaPrincipal, PanelClientes, PanelPresupuestos, etc.. Serán responsables de mostrar los datos y capturar las acciones del usuario.
- **Controlador**: Actúa como intermediario.
  - **Clases de Controlador**: ControladorClientes, ControladorLogin, etc. Reciben los eventos de la Vista (ej. "clic en botón Guardar"), interactúan con el Modelo (ej. clienteDAO.actualizar(cliente)) y actualizan la Vista con los nuevos datos.

#### Estructura de Paquetes Propuesta:

com.apptaller



## 2.2. Modelado de Datos

### 2.2.1 Modelo Entidad-Relación (MER)

A continuación, se adjunta el Modelo Entidad-Relación (MER) que representa la estructura conceptual de la base de datos de "APPTaller". El diagrama define las entidades principales del sistema, sus atributos y las relaciones entre ellas.

En el [Anexo II](#) se puede ver el diagrama E/R.

#### Descripción del Modelo:

El modelo se centra en el flujo de trabajo del taller, conectando clientes y vehículos con los servicios de presupuestos y facturación, e integrando la gestión de proveedores y productos.

- **Entidades Principales:**

- **Clientes:** Almacena los datos de los propietarios de los vehículos. Su clave principal (PK) es el *dni*.
- **Vehiculos:** Registra los vehículos del taller. Su PK es el *bastidor*.
- **Presupuestos:** Es la entidad central de la operativa. Contiene la información de un servicio propuesto, identificado por *id\_presupuesto*.
- **Facturas :** Almacena los datos de una factura emitida, con PK *id\_factura*.
- **Proveedores :** Guarda la información de los proveedores de suministros, con PK *cif*.
- **Productos :** Representa los suministros o piezas del inventario, con PK *id\_producto*.

- **Relaciones y Cardinalidades:**

- **Relación Cliente-Vehículo:** Un *Cliente* puede ser *es\_dueño\_actual\_de* cero o muchos (0..\*) *Vehiculos*. Un *Vehiculo* tiene un único propietario actual (1). Además, la entidad asociativa **Historial\_Propiedad** resuelve una relación M:N (*propiedad\_por\_cliente*) para registrar los propietarios anteriores de un vehículo.

- **Relación con Servicios:**

- Un *Cliente* encarga cero o muchos (0..\*) *Presupuestos*.

- Un *Vehiculo requiere\_presupuesto* cero o muchos (0..\*) *Presupuestos*.
- Un *Presupuesto* se asocia con un (1) *Cliente* y un (1) *Vehiculo* .
- **Relación Presupuesto-Factura:** La relación *genera* es de tipo uno-a-cero-o-uno (1:0..1). Esto significa que un *Presupuesto* puede generar (0..1) *Facturas* , pero una *Factura* se genera siempre a partir de un (1) único *Presupuesto*.
- **Detalles del Presupuesto:** Un *Presupuesto* se desglosa en dos tipos de detalles:
  - **Detalle\_Mano\_Obra :** Mediante la relación *detalla\_mano obra*, un *Presupuesto* tiene una o muchas (1..\*) líneas de mano de obra.
  - **Detalle\_Productos :** Esta es una entidad asociativa que resuelve la relación muchos-a-muchos (M:N) *contiene\_productos* entre *Presupuestos* y *Productos*. Permite registrar la *cantidad\_usada* y el *precio\_venta\_unitario* de cada pieza.
- **Relación de Suministro:** Un *Proveedor* vende uno o muchos (1..\*) *Productos*, y un *Producto* es suministrado por un único *Proveedor* (1).
- **Usuario-Emppleado:** Relación 1:0..1 (un empleado puede tener usuario del sistema o no)
- **Empleado-Reparación:** Relación 1:N (un empleado puede estar asignado a muchas reparaciones)
- **Reparación-Vehículo:** Relación N:1
- **Reparación-Cliente:** Relación N:1
- **Reparación-Presupuesto:** Relación 0..1:1 (una reparación puede tener un presupuesto asociado)
- **Factura-Reparación:** Relación 1:1 (modificación de la relación original con Presupuesto)

## 2.2.2 Diagrama de Clases (UML)

El siguiente diagrama de clases define la estructura estática del sistema. Representa las clases de entidad (que se convertirán en los POJOs del paquete modelo.vo), sus atributos, y las relaciones que existen entre ellas.

En el Anexo III se puede ver el diagrama de clases.

### **Descripción de las Clases y Relaciones:**

El diagrama se organiza en torno a tres núcleos principales: la gestión de clientes/vehículos, la gestión de servicios (presupuestos/facturas) y la gestión de inventario.

- **Núcleo Cliente-Vehículo:**
  - **Clientes:** Es la entidad principal que almacena los datos del cliente, identificada por su dni.
  - **Vehiculos :** Almacena la información de los vehículos, identificados por el *bastidor*.
  - **Relaciones Cliente-Vehículo:** Existen dos relaciones clave:
    1. *posee\_actualmente*: Una relación uno-a-muchos (1:N) que indica que un *Cliente* puede tener 0..\* (cero o muchos) *Vehiculos*. Un *Vehiculo* pertenece a un único cliente actual (reflejado en el atributo *propietario\_actual\_dni*).
    2. *Historial\_Propiedad*: Es una clase asociativa que resuelve una relación muchos-a-muchos (M:N) histórica (*fue\_dueño\_en\_período*). Permite que un vehículo haya pertenecido a varios clientes a lo largo del tiempo y viceversa, registrando una *fecha\_inicio* y *fecha\_fin*.
- **Núcleo de Inventario:**
  - **Proveedores:** Entidad que guarda los datos de los proveedores, identificada por su cif.
  - **Productos:** Almacena los suministros y piezas del taller (aceite, filtros, etc.), identificados por *id\_producto*.
  - **Relación vende:** Es una relación uno-a-muchos (1:N) donde un *Proveedor* suministra 1..\* (uno o muchos) *Productos* , como indica el atributo *proveedor\_cif* en *Productos*.
- **Núcleo de Servicios (Presupuestos y Facturas):**
  - **Presupuestos:** Es la clase central del negocio. Se vincula directamente con un *Cliente* y un *Vehiculo*. Contiene el *estado* (PENDIENTE, APROBADO, RECHAZADO).
  - **Detalle\_Productos:** Es una clase asociativa que resuelve la relación muchos-a-muchos (M:N) *usa/incluido* en entre *Presupuestos* y *Productos*. Almacena los detalles de qué productos se usan en un presupuesto, con su *cantidad\_usada* y *precio\_venta\_unitario*.

- **Detalle\_Mano\_Obra:** Modela una relación uno-a-muchos (1:N) *incluye* con *Presupuestos*. Un presupuesto puede tener 1..\* (una o muchas) líneas de mano de obra, cada una con su *descripcion\_trabajo*, *tiempo\_empleado\_horas* y *tarifa\_por\_hora*.
- **Facturas:** Es el documento final. Mantiene una relación uno-a-uno opcional (1:0..1) *es\_facturado\_por* con *Presupuestos*. Un *Presupuesto* puede tener 0..1 (cero o una) *Factura*, y una *Factura* se genera a partir de un único *id\_presupuesto*.
- **Núcleo Operativo (Reparaciones)**
  - **Reparaciones** (entidad central del trabajo real)
  - **Reparacion\_Detalle\_Mano\_Obra** (trabajo realmente ejecutado)
  - **Reparacion\_Detalle\_Productos** (productos realmente usados)
- **Núcleo de Gestión de Personal**
  - **Usuarios** (autenticación y roles)
  - **Empleados** (datos personales y profesionales)

### 2.2.3 Diseño Relacional (Tablas)

A continuación, se describe la estructura de las tablas de la base de datos apptaller\_db, diseñada para MySQL.

- **Usuarios**
  - id\_usuario (INT, PK, Auto-incremental)
  - username (VARCHAR, Único, No Nulo)
  - password\_hash (VARCHAR, No Nulo)
  - rol (ENUM('Empleado', 'Administrador'), No Nulo)

(Nota: Esta tabla se deduce de los requisitos funcionales RF-01.1, RF-01.2 y del Caso de Uso 'Gestionar Usuarios' ).

- **Empleados**
  - id\_empleado (INT, PK, Auto-incremental)

- usuario\_id (INT, FK a Usuarios(id\_usuario), Único, Nullable) - Permite vincular el empleado a un usuario del sistema. NULL si no tiene acceso.
- dni (VARCHAR(10), Único, No Nulo)
- nombre (VARCHAR(100), No Nulo)
- apellidos (VARCHAR(100), No Nulo)
- telefono (VARCHAR(15))
- email (VARCHAR(100), Único)
- direccion (VARCHAR(255))
- cargo (VARCHAR(50), Default 'Mecánico') - Ejemplos: Mecánico, Jefe de Taller, Recepcionista
- fecha\_alta (DATE, No Nulo)
- fecha\_baja (DATE, Nullable) - NULL si el empleado está activo
- foto\_perfil (VARCHAR(255)) - Ruta relativa a la imagen de perfil
- salario\_base (DECIMAL(10,2))
- activo (BOOLEAN, Default TRUE) - Indica si el empleado está activo o dado de baja
- created\_at (TIMESTAMP, Default CURRENT\_TIMESTAMP)

- **Clients**

- dni (VARCHAR(10), PK)
- nombre (VARCHAR(100), No Nulo)
- direccion (VARCHAR(255))
- telefono (INT)
- email (VARCHAR(100))

- fecha\_registro (DATE, No Nulo)
- **Vehiculos**
  - bastidor (VARCHAR(50), PK)
  - matricula (VARCHAR(10), Único, No Nulo)
  - marca (VARCHAR(50))
  - modelo (VARCHAR(50))
  - color (VARCHAR(30))
  - propietario\_actual\_dni (VARCHAR(10), FK a Clientes(dni))
  - fecha\_compra (DATE)
- **Historial\_Propiedad** (Tabla asociativa para el historial)
  - id\_registro (INT, PK, Auto-incremental)
  - vehiculo\_bastidor (VARCHAR(50), FK a Vehiculos(bastidor))
  - cliente\_dni (VARCHAR(10), FK a Clientes(dni))
  - fecha\_inicio (DATE, No Nulo)
  - fecha\_fin (DATE)
- **Proveedores**
  - cif (VARCHAR(10), PK)
  - nombre (VARCHAR(100), No Nulo)
  - direccion (VARCHAR(255))
  - telefono (VARCHAR(15))
  - email (VARCHAR(100))
  - contacto (VARCHAR(100))

- **Productos** (Suministros y piezas)
  - id\_producto (VARCHAR(50), PK)
  - nombre (VARCHAR(100), No Nulo)
  - descripcion (TEXT)
  - cantidad\_stock (INT, No Nulo, Default 0)
  - precio\_unitario (DECIMAL(10, 2), No Nulo)
  - proveedor\_cif (VARCHAR(10), FK a Proveedores(cif))
  
- **Presupuestos**
  - id\_presupuesto (INT, PK, Auto-incremental)
  - vehiculo\_bastidor (VARCHAR(50), FK a Vehiculos(bastidor))
  - cliente\_dni (VARCHAR(10), FK a Clientes(dni))
  - fecha\_emision (DATE, No Nulo)
  - estado (ENUM('PENDIENTE', 'APROBADO', 'RECHAZADO'), No Nulo)
  - descripcion\_trabajo (TEXT)
  - total\_estimado (DECIMAL(10, 2))
  
- **Detalle\_Mano\_Obra** (Líneas de trabajo del presupuesto)
  - id\_detalle (INT, PK, Auto-incremental)
  - id\_presupuesto (INT, FK a Presupuestos(id\_presupuesto))
  - descripcion\_trabajo (VARCHAR(255), No Nulo)
  - tiempo\_empleado\_horas (DECIMAL(5, 2), No Nulo)
  - tarifa\_por\_hora (DECIMAL(10, 2), No Nulo)

- **Detalle\_Productos** (Productos usados en el presupuesto)

- id\_presupuesto (INT, PK, FK a Presupuestos(id\_presupuesto))
- id\_producto (VARCHAR(50), PK, FK a Productos(id\_producto))
- cantidad\_usada (INT, No Nulo)
- precio\_venta\_unitario (DECIMAL(10, 2), No Nulo)
- descuento (DECIMAL(5, 2), Default 0)

- **Facturas**

- id\_factura (INT, PK, Auto-incremental)
- numero\_factura (VARCHAR(20), Único, No Nulo)
- id\_presupuesto (INT, Único, FK a Presupuestos(id\_presupuesto)) (*Único para garantizar la relación 1:1*)
- fecha\_emision (DATE, No Nulo)
- fecha\_cobro (DATE)
- total\_cobrado (DECIMAL(10, 2), No Nulo)
- metodo\_pago (VARCHAR(50))

- **Reparaciones**

Tabla central del sistema operativo que registra las órdenes de trabajo. Cada reparación representa la entrada de un vehículo al taller, el trabajo realizado y su entrega.

- id\_reparacion (INT, PK, Auto-incremental)
- vehiculo\_bastidor (VARCHAR(50), FK a Vehiculos(bastidor), No Nulo)
- cliente\_dni (VARCHAR(10), FK a Clientes(dni), No Nulo)
- empleado\_asignado\_id (INT, FK a Empleados(id\_empleado), No Nulo) - Mecánico responsable
- id\_presupuesto (INT, FK a Presupuestos(id\_presupuesto), Nullable) - Puede haber reparaciones sin presupuesto previo
- fecha\_entrada (DATETIME, No Nulo)

- fecha\_salida\_estimada (DATE)
- fecha\_salida\_real (DATETIME)
- kilometraje\_entrada (INT)
- nivel\_combustible (ENUM: 'Vacío', '1/4', '1/2', '3/4', 'Lleno')
- estado (ENUM: 'EN\_COLA', 'EN\_PROCESO', 'PAUSADA', 'FINALIZADA', 'ENTREGADA', No Nulo, Default 'EN\_COLA')
  - EN\_COLA: Vehículo en espera, no se ha empezado
  - EN\_PROCESO: Se está reparando actualmente
  - PAUSADA: Trabajo pausado (esperando piezas, autorización, etc.)
  - FINALIZADA: Reparación completada, pendiente de facturar/entregar
  - ENTREGADA: Vehículo entregado al cliente
- prioridad (ENUM: 'Baja', 'Normal', 'Alta', 'Urgente', Default 'Normal')
- diagnostico (TEXT) - Diagnóstico inicial del problema reportado
- trabajos\_realizados (TEXT) - Descripción detallada de los trabajos ejecutados
- observaciones (TEXT)
- usuario\_recepcion (INT, FK a Usuarios(id\_usuario)) - Usuario que registró la entrada usu
- usuario\_entrega (INT, FK a Usuarios(id\_usuario)) - Usuario que entregó el vehículo
- created\_at (TIMESTAMP, Default CURRENT\_TIMESTAMP)
- updated\_at (TIMESTAMP, Default CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP)

- **Reparacion\_Detalle\_Mano\_Obra**

Tabla que registra el trabajo real de mano de obra ejecutado en cada reparación. Permite que múltiples empleados trabajen en diferentes tareas de una misma reparación. Puede diferir del presupuesto si hubo cambios durante el trabajo.

- id\_detalle (INT, PK, Auto-incremental)
- id\_reparacion (INT, FK a Reparaciones(id\_reparacion), No Nulo)

- empleado\_id (INT, FK a Empleados(id\_empleado), No Nulo) - Mecánico que realizó este trabajo específico
- descripcion\_trabajo (VARCHAR(255), No Nulo) - Ej: "Cambio de pastillas de freno delanteras"
- fecha\_inicio (DATETIME)
- fecha\_fin (DATETIME)
- tiempo\_empleado\_horas (DECIMAL(5,2), No Nulo) - Horas realmente trabajadas
- tarifa\_por\_hora (DECIMAL(10,2), No Nulo)
- subtotal (DECIMAL(10,2), GENERATED/COMPUTED) - Calculado como:  $\text{tiempo\_empleado\_horas} \times \text{tarifa\_por\_hora}$

- **Reparacion\_Detalle\_Productos**

Tabla asociativa que registra los productos y piezas **realmente usados** en cada reparación. Cuando se añade un producto aquí, el sistema descuenta automáticamente el stock mediante triggers.

- id\_reparacion (INT, PK, FK a Reparaciones(id\_reparacion))
- id\_producto (VARCHAR(50), PK, FK a Productos(id\_producto))
- cantidad\_usada (INT, No Nulo)
- precio\_venta\_unitario (DECIMAL(10,2), No Nulo) - Precio al que se vende al cliente
- descuento (DECIMAL(5,2), Default 0) - Porcentaje de descuento (0-100)
- subtotal (DECIMAL(10,2), GENERATED/COMPUTED) - Calculado como:  $\text{cantidad\_usada} \times \text{precio\_venta\_unitario} \times (1 - \text{descuento}/100)$
- fecha\_aplicacion (DATETIME, Default CURRENT\_TIMESTAMP)

- **Movimientos\_Stock**

Tabla de auditoría que registra todos los cambios en el inventario de productos. Permite trazabilidad completa de entradas, salidas y ajustes de stock. Cumple con el requisito RF-08.1.

- id\_movimiento (INT, PK, Auto-incremental)
- id\_producto (VARCHAR(50), FK a Productos(id\_producto), No Nulo)
- tipo\_movimiento (ENUM: 'ENTRADA', 'SALIDA', 'AJUSTE', 'DEVOLUCION', No Nulo)

- ENTRADA: Recepción de nuevos productos del proveedor
- SALIDA: Uso en reparaciones o venta
- AJUSTE: Correcciones de inventario (mermas, errores, etc.)
- DEVOLUCION: Devolución de productos no usados
- cantidad (INT, No Nulo) - Cantidad del movimiento (siempre positiva)
- stock\_anterior (INT, No Nulo) - Stock antes del movimiento
- stock\_posterior (INT, No Nulo) - Stock después del movimiento
- motivo (VARCHAR(255)) - Descripción del motivo del movimiento
- usuario\_id (INT, FK a Usuarios(id\_usuario)) - Usuario que realizó el movimiento
- fecha\_movimiento (DATETIME, Default CURRENT\_TIMESTAMP)

El script SQL completo para la creación de esta estructura en MySQL se puede consultar en el [Anexo IV: Script de Base de Datos.](#)

## 2.3. Diseño de Interfaces (Prototipos)

La interfaz de usuario se diseñará siguiendo los principios de **Material Design**. Esto se logrará mediante la implementación de la biblioteca *Material-UI-Swing*, que proporciona componentes Swing personalizados (botones, campos de texto, sombras) y gestiona la paleta de colores y tipografías. Los siguientes bocetos de baja fidelidad representan la **estructura y disposición** de las ventanas principales.

### 2.3.1 Boceto 1: Ventana de Login

Conforme al flujo de autenticación, esta será la primera ventana de la aplicación. Será una ventana modal simple que actúa como una "tarjeta" (Card) con elevación. Contendrá el logo, un *MaterialTextField* para "Usuario", un *MaterialPasswordField* para "Contraseña" y un *MaterialButton* para "Acceder".

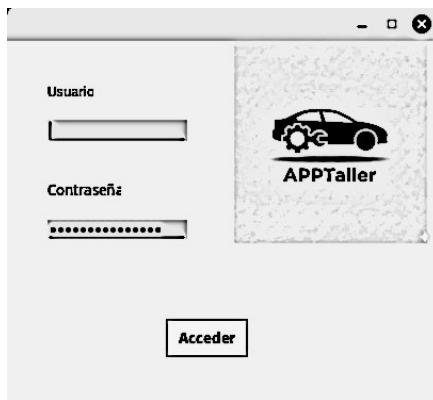


Figura 1: Boceto panel Login

### 2.3.2 Boceto 2: Ventana Principal

Una vez autenticado, el usuario accederá a esta ventana, que actúa como el contenedor principal de la aplicación.

- **Descripción:** Un JFrame maximizado que sigue una disposición estándar de Material Design.
- **Componentes:**
  1. **Barra de Aplicación (App Bar) Superior:** Un panel (*JPanel* o *JToolbar*) en la parte superior que mostrará el título de la sección actual (ej. "APP Taller - Inicio") y un ícono con el logo de la empresa.
  2. **Panel de Navegación Lateral (Nav Drawer):** Un panel a la izquierda con los botones de navegación. Estos serán *MaterialButton* de tipo "text" (sin fondo) con un ícono y el nombre de la sección:
    - Inicio
    - Clientes
    - Vehículos
    - Presupuestos
    - Facturas
    - Proveedores
    - Suministros
    - Ayuda
  3. **Área de Panel Central:**
    - **Descripción:** Este será el JPanel que se carga en el área de contenido principal al iniciar sesión.
    - **Contenido:**
      - Un título de bienvenida con la foto del empleado y la fecha y hora.
      - Una parte central que contendrá tres paneles:

- Un panel para mostrar un resumen numérico de los vehículos que hay en el taller ahora mismo.
- Un panel para mostrar una lista con lo que hay que hacer.
- Un panel para mostrar los productos con bajo stock.
- Una parte inferior que servirá como Block de Notas para escribir notas rápidas.

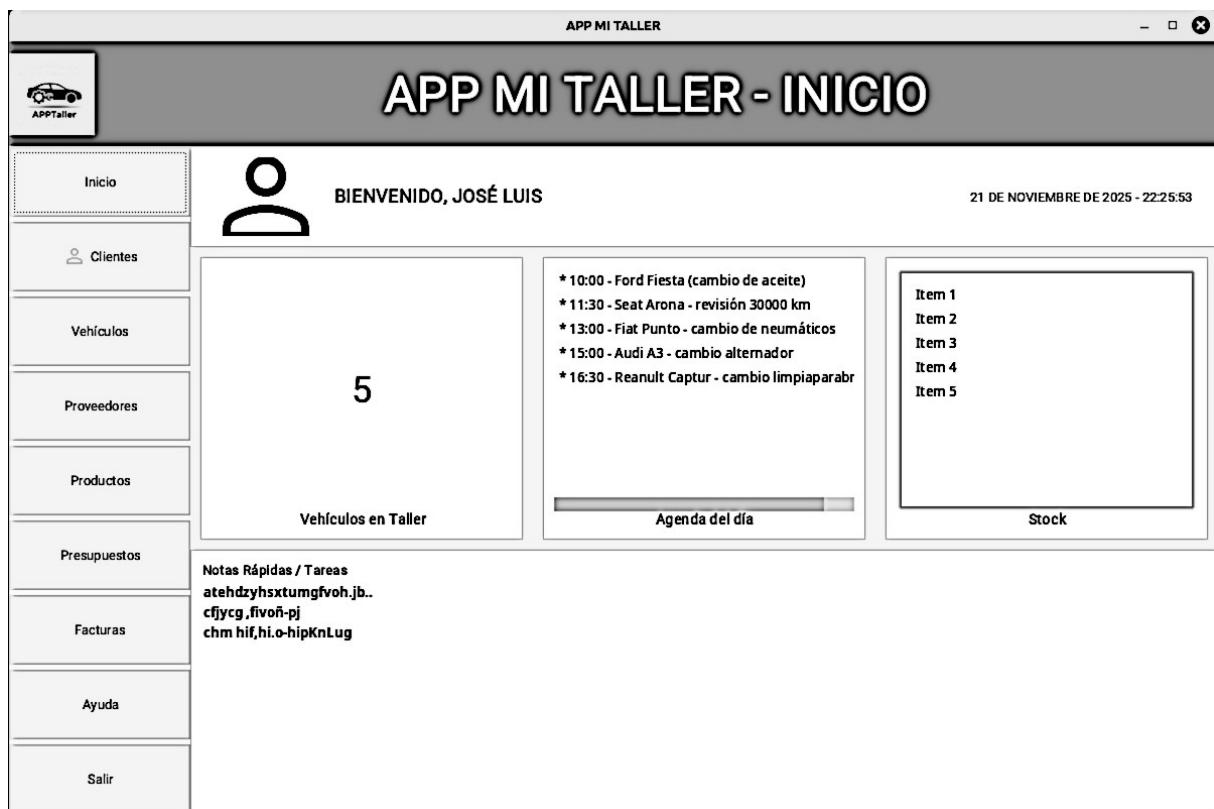


Figura 2: Boceto Ventana Principal

### 2.3.3 Boceto 3: Panel de Gestión de Clientes

Cuando el usuario selecciona el botón de *Clientes* en el menú lateral, el panel central muestra la interfaz de gestión (CRUD) correspondiente.

Este panel sigue un patrón de diseño dividido en dos áreas:

- **Gestión de Clientes (Arriba):** Contiene los campos de texto (JTextField) para la búsqueda del cliente (DNI, Nombre, Dirección, etc.) y los botones de acción para “Buscar”, “Nuevo”, “Informe” y “Eliminar”.
- **Tabla (abajo):** Muestra una tabla (JTable) con el listado de todos los registros.



Figura 3: Boceto panel Clientes

### 2.3.4 Boceto 4: Panel de Gestión de Vehículos

Cuando el usuario selecciona el botón de *Vehículos* en el menú lateral, el panel central muestra la interfaz de gestión (CRUD) correspondiente.

Este panel sigue un patrón de diseño dividido en dos áreas:

- **Gestión de Vehículos (Arriba):** Contiene los campos de texto (JTextField) para la búsqueda del vehículo (bastidor, matrícula,...) y los botones de acción para “Buscar”, “Nuevo”, “Informe” y “Eliminar”.
- **Tabla (abajo):** Muestra una tabla (JTable) con el listado de todos los registros.



Figura 4: Boceto panel Vehículos

### 2.3.5 Boceto 5: Panel de Gestión de Reparaciones

Cuando el usuario selecciona el botón de *Reparaciones* en el menú lateral, el panel central muestra la interfaz de gestión (CRUD) correspondiente.

Este panel sigue un patrón de diseño dividido en dos áreas:

- **Parte superior:** Contiene los indicadores por colores de los vehículos en el taller según su estado, y los botones para añadir una nueva reparación, los detalles de una reparación en concreto, el estado de una reparación y para asignar una reparación a un mecánico.
- **Tabla (abajo):** Muestra una tabla (JTable) con el listado de todos los registros.

ID	Vehículo	Cliente	Mecánico	Prioridad	Estado	Fecha Entrada
REP-001	Ford Fiesta	Juan Pérez	Carlos M.	ALTA	EN_PROCESO	20/11/2025
REP-002	Audi A3	Ana Gomez	-	MEDIA	EN_COLA	21/11/2025
REP-003	Seat León	Pedro Ruiz	Carlos M.	BAJA	FINALIZADA	19/11/2025

Figura 5: Boceto panel Reparaciones

### 2.3.6 Boceto 6: Panel de Gestión de Proveedores

Cuando el usuario selecciona el botón de *Proveedores* en el menú lateral, el panel central muestra la interfaz de gestión (CRUD) correspondiente.

Este panel sigue un patrón de diseño dividido en dos áreas:

- **Gestión de Proveedores (Arriba):** Contiene los campos de texto (*JTextField*) para la búsqueda del proveedor (CIF, nombre,...) y los botones de acción para “Buscar”, “Nuevo”, “Informe” y “Eliminar”.
- **Tabla (abajo):** Muestra una tabla (*JTable*) con el listado de todos los registros.

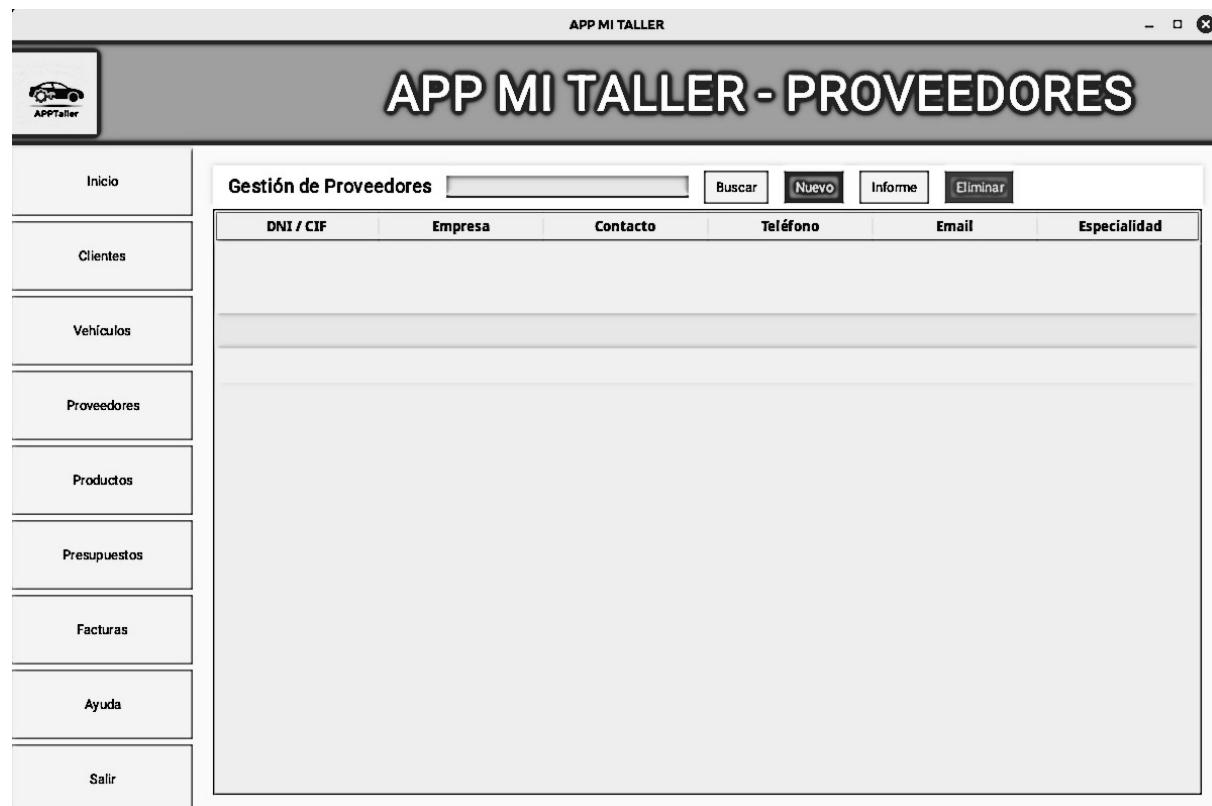


Figura 6: Boceto panel Proveedores

### 2.3.7 Boceto 7: Panel de Gestión de Productos

Cuando el usuario selecciona el botón de *Productos* en el menú lateral, el panel central muestra la interfaz de gestión (CRUD) correspondiente.

Este panel sigue un patrón de diseño dividido en dos áreas:

- **Gestión de Productos (Arriba):** Contiene los campos de texto (*JTextField*) para la búsqueda del producto (ID/Referencia,...) y los botones de acción para “Buscar”, “Nuevo”, “Informe” y “Eliminar”.
- **Tabla (abajo):** Muestra una tabla (*JTable*) con el listado de todos los registros.



Figura 7: Boceto panel Productos

### 2.3.8 Boceto 8: Panel de Gestión de Presupuestos

Cuando el usuario selecciona el botón de *Presupuestos* en el menú lateral, el panel central muestra la interfaz de gestión (CRUD) correspondiente.

Este panel sigue un patrón de diseño dividido en dos áreas:

- **Gestión de Presupuestos (Arriba):** Contiene los campos de texto (JTextField) para la búsqueda del producto (N.º Presupuesto, Cliente,...) y los botones de acción para “Buscar”, “Nuevo”, “Informe” y “Eliminar”.
- **Tabla (abajo):** Muestra una tabla (JTable) con el listado de todos los registros.



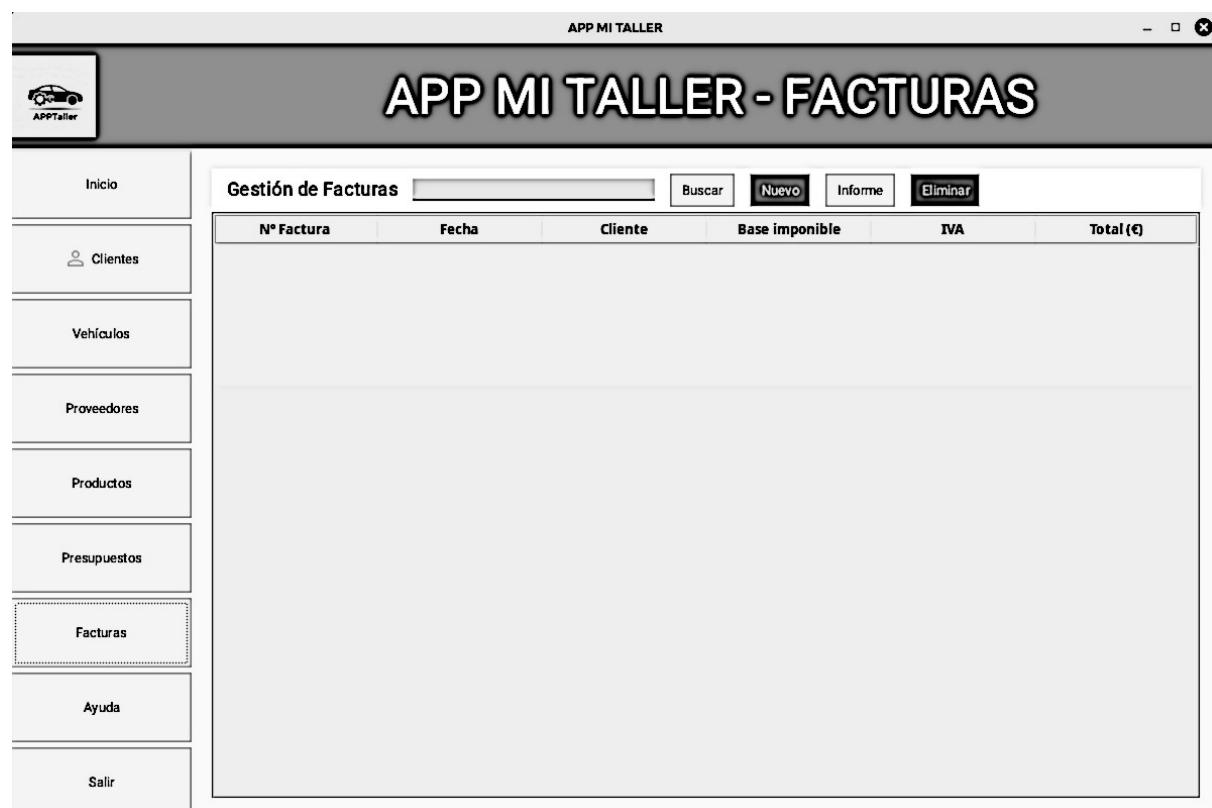
Figura 8: Boceto panel Presupuestos

## **2.3.9 Boceto 9: Panel de Gestión de Facturas**

Cuando el usuario selecciona el botón de *Facturas* en el menú lateral, el panel central muestra la interfaz de gestión (CRUD) correspondiente.

Este panel sigue un patrón de diseño dividido en dos áreas:

- **Gestión de Facturas (Arriba):** Contiene los campos de texto (*JTextField*) para la búsqueda de la factura (N.<sup>º</sup> Factura, Cliente,...) y los botones de acción para “Buscar”, “Nuevo”, “Informe” y “Eliminar”.
  - **Tabla (abajo):** Muestra una tabla (*JTable*) con el listado de todos los registros.



*Figura 9: Boceto panel Facturas*

### 2.3.10 Boceto 10: Panel de Empleados

Cuando el usuario selecciona el botón de *Empleados* en el menú lateral, el panel central muestra la interfaz de gestión (CRUD) correspondiente.

Este panel sigue un patrón de diseño dividido en dos áreas:

- **Gestión de Empleados (Arriba):** Contiene los campos de texto (JTextField) para la búsqueda del empleado (DNI, Nombre, Dirección, etc.) y los botones de acción para “Buscar”, “Nuevo”, “Informe” y “Eliminar”.
- **Tabla (abajo):** Muestra una tabla (JTable) con el listado de todos los registros.

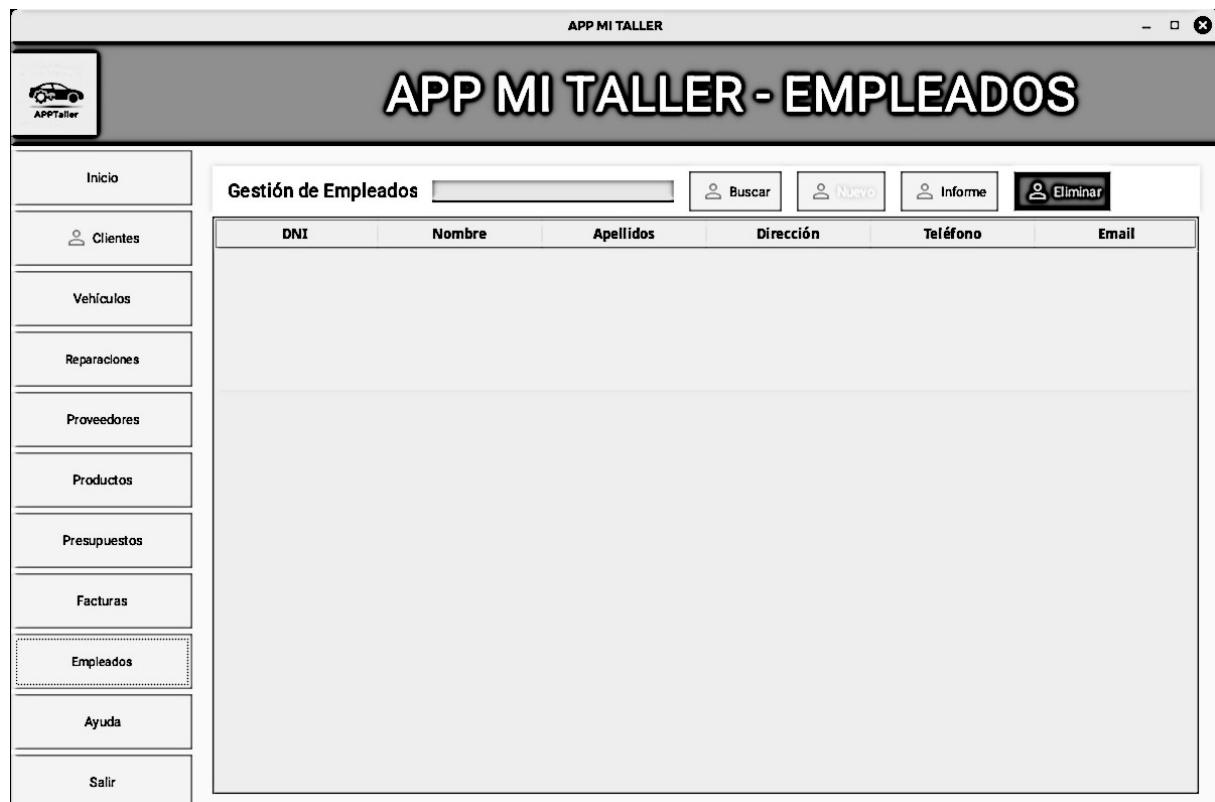


Figura 10: Boceto panel Empleados

### 2.3.11 Boceto 11: Panel de Ayuda

Cuando el usuario selecciona el botón de *Ayuda* en el menú lateral, el panel central muestra la interfaz de gestión (CRUD) correspondiente.

Este panel sigue un patrón de diseño dividido en dos áreas:

- **Índice de temas de ayuda (Izquierda):** Una lista que muestra los temas de ayuda.
- **Visor de contenido (Derecha):** Un visor donde se muestra la explicación.

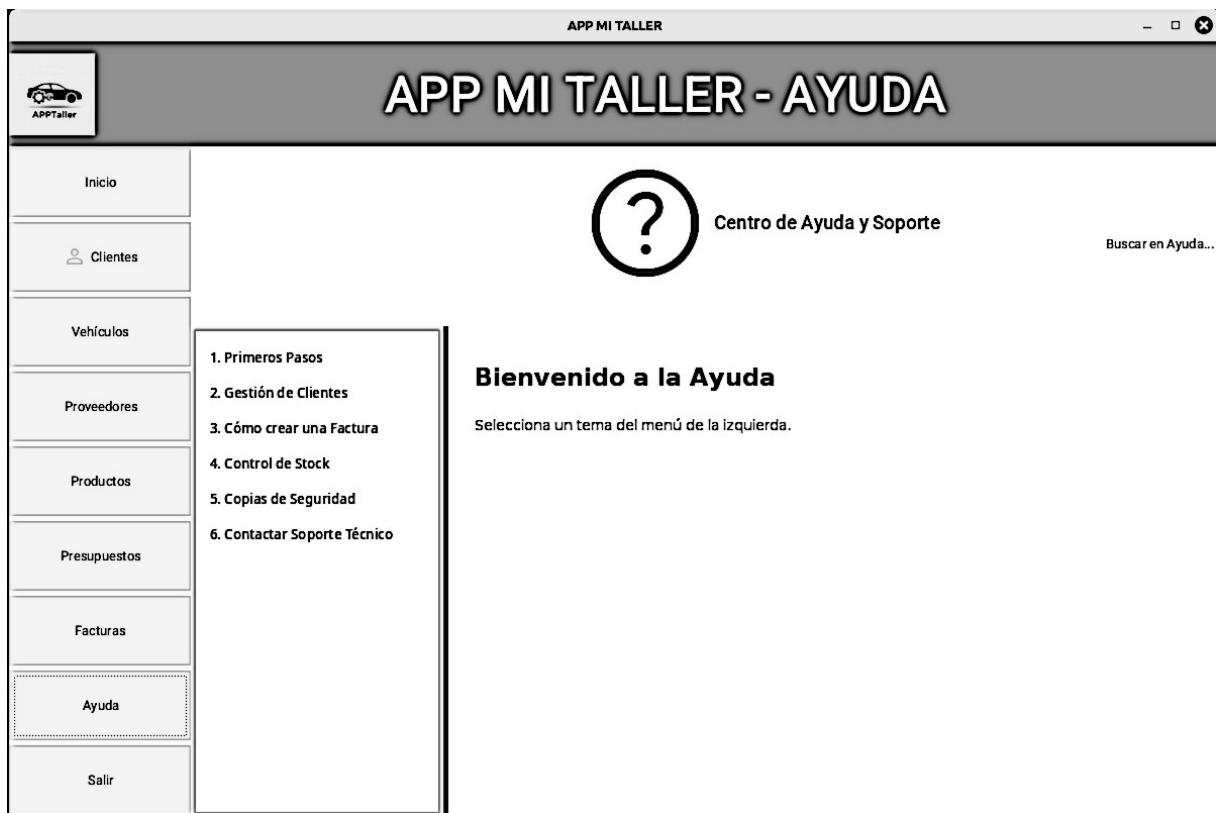


Figura 11: Boceto panel Ayuda

### 2.3.12 Boceto 12: Modal Nuevo Presupuesto

Al pinchar en el botón *Nuevo* del panel de *Presupuestos* se abre un diálogo modal para crear un nuevo presupuesto. Este diálogo está compuesto por:

- Una parte superior con los datos del cliente y del vehículo, con un botón para añadir un nuevo cliente, la fecha y el estado del vehículo.
- A continuación la zona de mano de obra, con una tabla donde se detallan los conceptos.
- Seguidamente la zona para añadir los materiales y piezas que se usarán en la reparación.
- La zona de resumen, con los importes antes y después de impuestos.
- Un botón para cancelar la creación del presupuesto y otro para guardarla.

**Nuevo Presupuesto**

Datos del Cliente y Vehículo																												
Cliente	Vehículo	Fecha	Estado																									
<input type="button" value="Ite..."/>	<input type="button" value="Nuevo"/>	<input type="button" value="Ite..."/>	<input type="button" value="19/12/25, 1"/>																									
Avería																												
<b>Mano de Obra</b> <table border="1"> <thead> <tr> <th>Descripción</th> <th>Horas</th> <th>Precio/Hora</th> <th>Total</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>				Descripción	Horas	Precio/Hora	Total																					
Descripción	Horas	Precio/Hora	Total																									
<b>Materiales y piezas</b> <table border="1"> <thead> <tr> <th>Referencia</th> <th>Concepto</th> <th>Cantidad</th> <th>Precio Unidad</th> <th>Total</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>				Referencia	Concepto	Cantidad	Precio Unidad	Total																				
Referencia	Concepto	Cantidad	Precio Unidad	Total																								
<b>Resumen</b> <table border="1"> <tbody> <tr> <td>Base Imponible</td> <td>0.00 €</td> </tr> <tr> <td>IVA(21%)</td> <td>0.00 €</td> </tr> <tr> <td><b>TOTAL</b></td> <td><b>0.00 €</b></td> </tr> </tbody> </table>				Base Imponible	0.00 €	IVA(21%)	0.00 €	<b>TOTAL</b>	<b>0.00 €</b>																			
Base Imponible	0.00 €																											
IVA(21%)	0.00 €																											
<b>TOTAL</b>	<b>0.00 €</b>																											
<input type="button" value="Cancelar"/>		<input type="button" value="Guardar Presupuesto"/>																										

*Figura 12: Boceto nuevo Presupuesto*

### 2.3.13 Boceto 13: Modal Nueva Reparación

Al pinchar en el botón *Nueva* del panel de *Reparaciones* se abre un diálogo modal para crear una nueva reparación. Este diálogo está compuesto por:

- Una parte superior con los datos del cliente y del vehículo.
- Una parte central con los datos de la inspección del vehículo a la entrada del taller, con el kilometraje, el nivel de combustible, la descripción de la avería y un apartado de notas.
- Una parte inferior donde se le asigna un mecánico, la prioridad de la reparación y la fecha prevista de entrega.
- Un botón para cancelar la creación de la reparación y otro para guardarla.

**Recepción de Vehículo - Nueva Reparación**

Datos del Cliente y Vehículo	
Cliente	Juan Pérez
Vehículo	1234-BCD
Inspección de Entrada	
Kilometraje	45786
Combustible	<input type="range" value="25"/> <div style="display: flex; justify-content: space-around;"> <span>0</span> <span>25</span> <span>50</span> <span>75</span> <span>100</span> </div>
Descripción de la avería	El coche no arranca
Notas	Deja documentación.
Asignación y Prioridad	
Mecánico asignado	Pedro García
Prioridad	Normal
Fecha estimada entrega	20/12/25, 17:15
<input type="button" value="Cancelar"/> <input type="button" value="Crear Orden Reparación"/>	

Figura 13: Boceto nueva Reparación

### 2.3.14 Boceto 14: Modal Detalles Reparación

Al pinchar en el botón *Detalles* del panel de *Reparaciones* se abre un diálogo modal para ver una reparación. Este diálogo está compuesto por pestañas (TabbedPane):

- Datos Generales
- Trabajos Realizados (tabla de mano de obra)
- Productos Usados (tabla de productos)
- Costos (desglose completo)
- Historial (cambios de estado)

En la parte inferior 3 botones de acción: Salir, Imprimir y Editar (por si hubiera que corregir algo).



Figura 14: Boceto Detalles Reparación

### 2.3.15 Boceto 15: Modal Histórico de Cliente

Al pinchar en el botón *Historial* del panel de *Clients* se abre un diálogo modal para ver el histórico de un cliente. Este diálogo está compuesto por pestañas (TabbedPane):

- Vehículos del cliente
- Presupuestos históricos
- Facturas emitidas

En la parte inferior 3 botones de acción: Salir y Editar (por si hubiera que corregir algo).

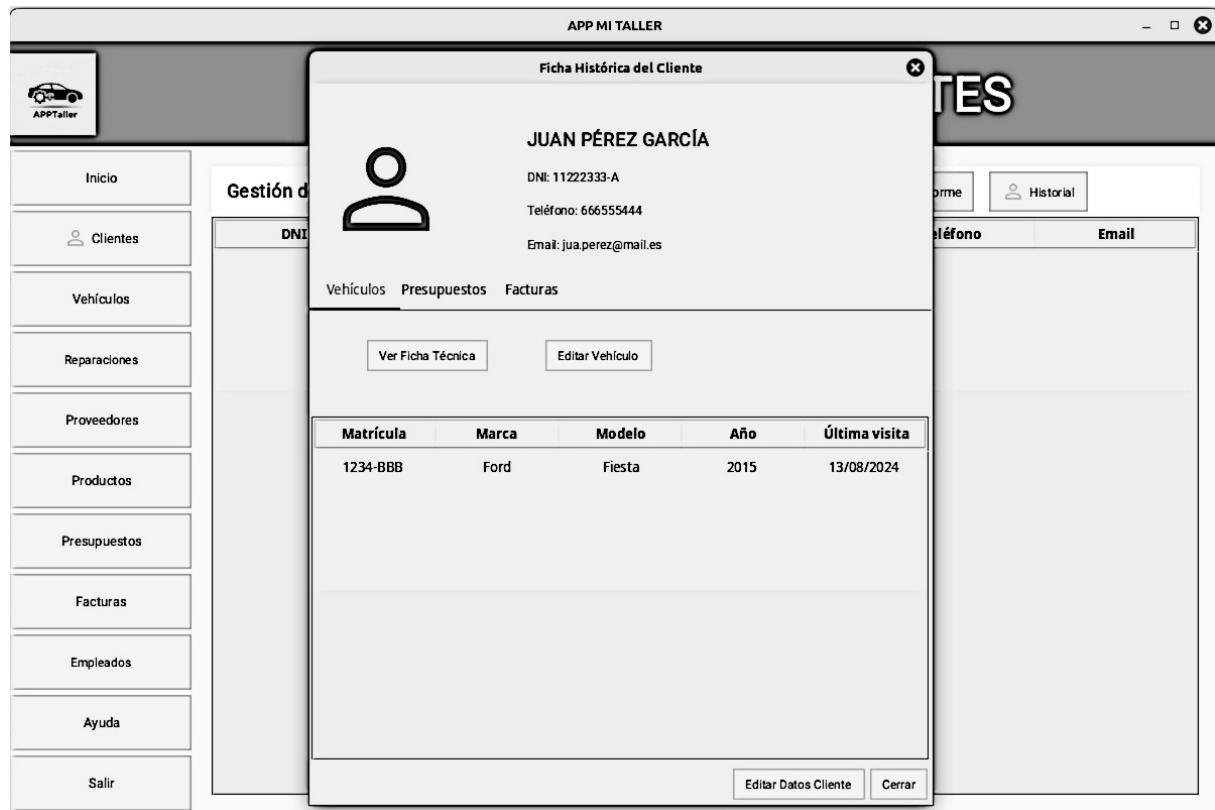


Figura 15: Boceto Histórico de Clientes

### 3. Control de Versiones

Para garantizar la integridad del código fuente, la trazabilidad de los cambios y la seguridad ante fallos, se ha implementado un sistema de control de versiones distribuido (DVCS) utilizando **Git**, con **GitHub** como repositorio remoto para el alojamiento y respaldo del proyecto.

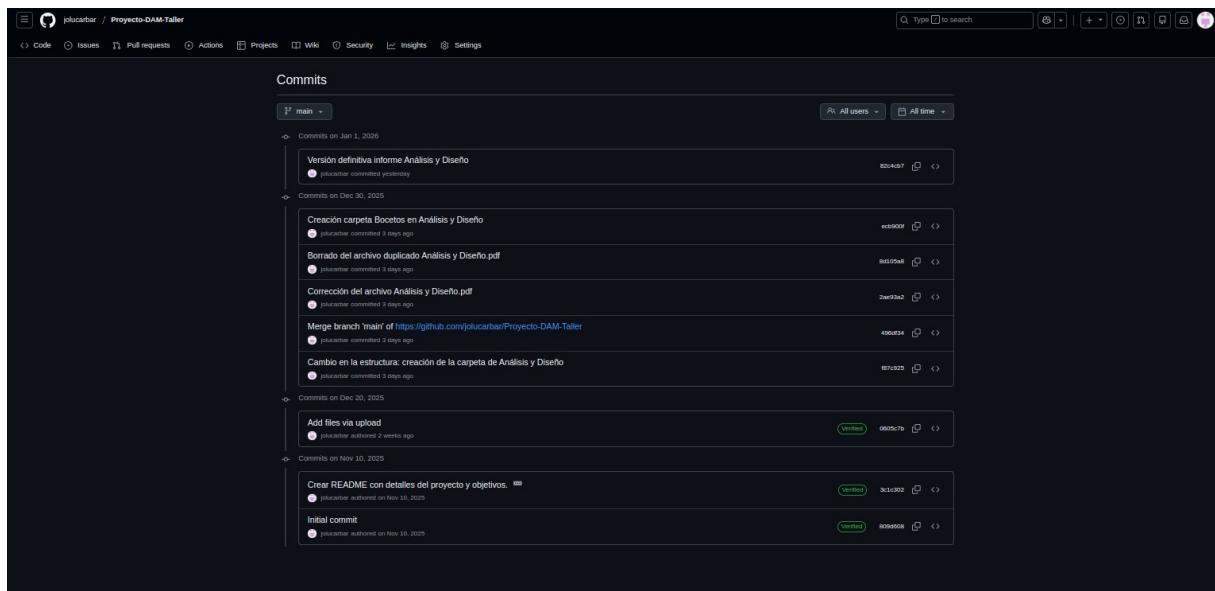


Figura 16: Historial de commits hasta la fecha

## 4. Creación del proyecto

A continuación se muestra una captura con la creación del proyecto y la estructura de clases.

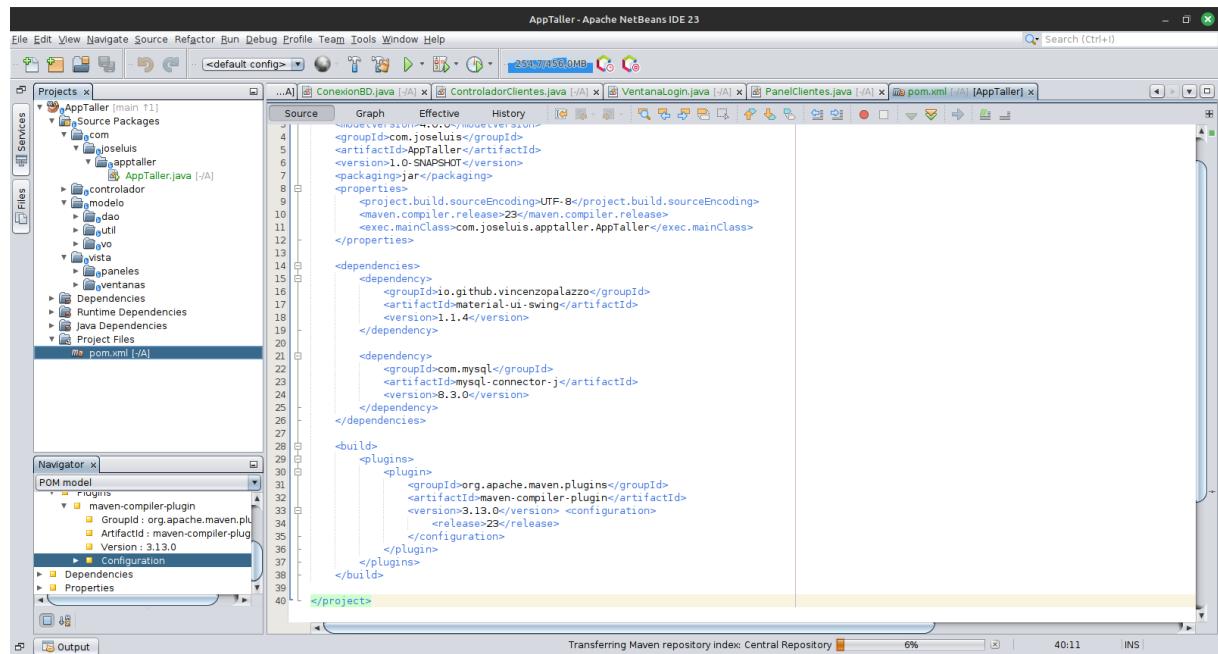
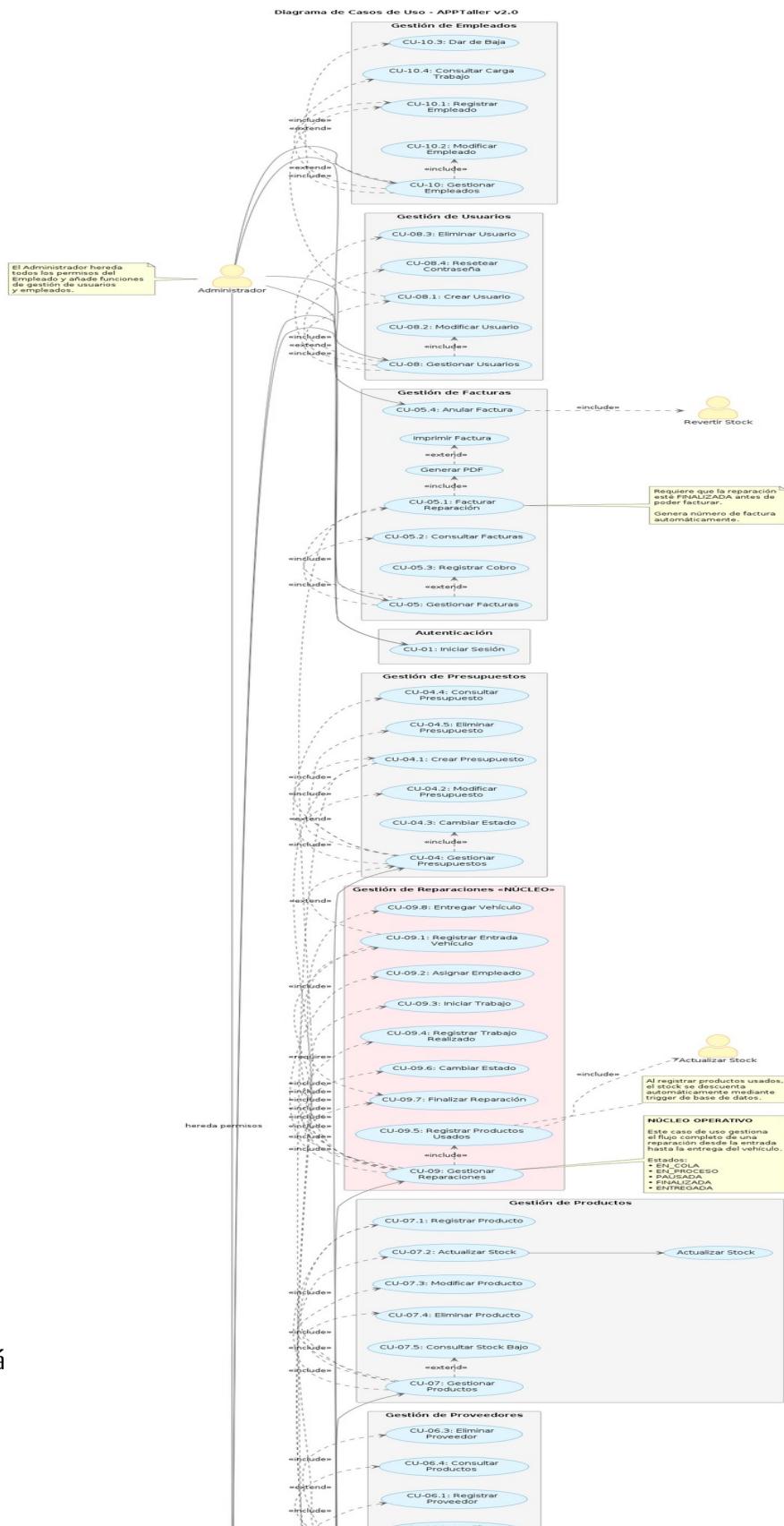
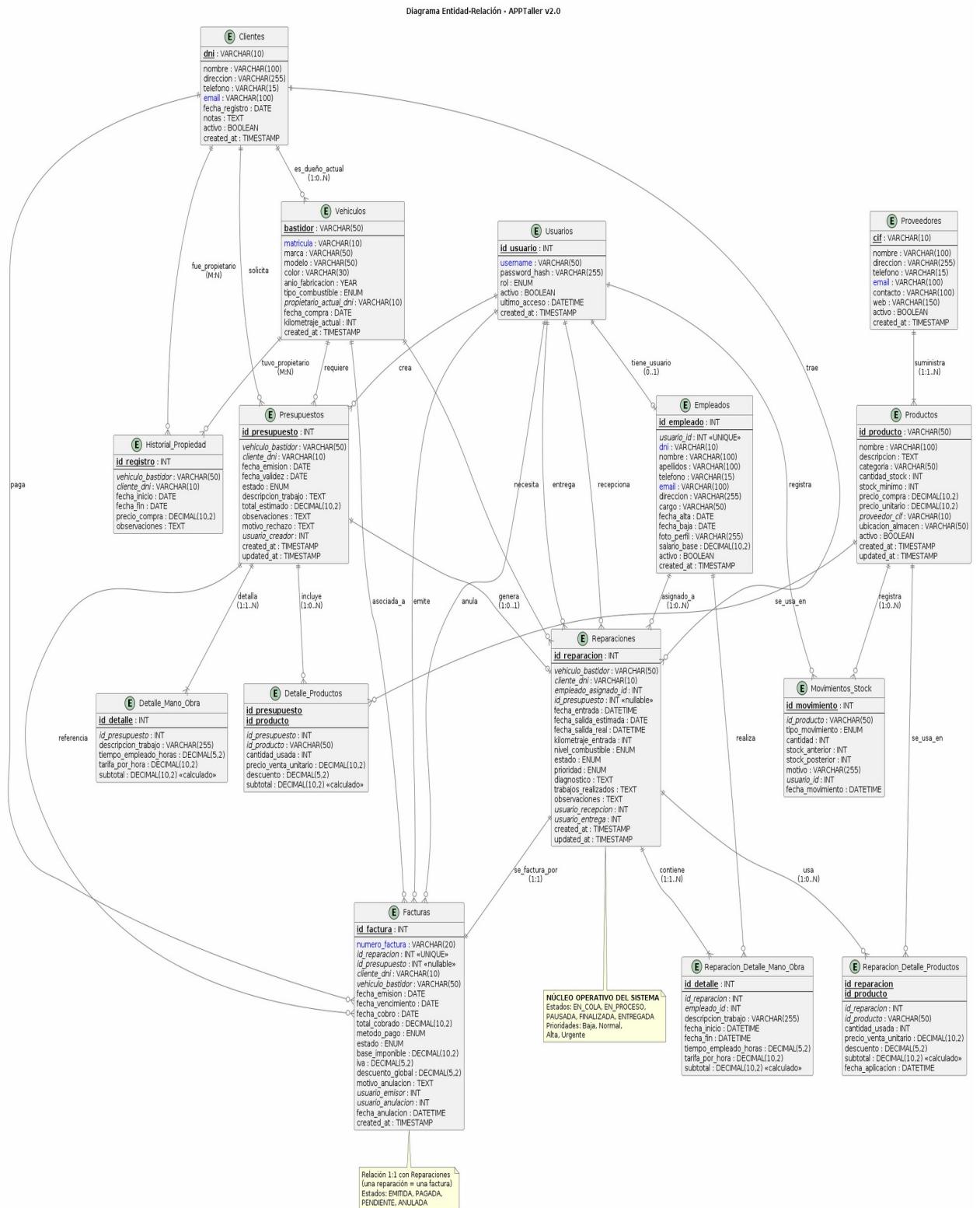


Figura 17: Creación del proyecto y la estructura de clases

# Anexo I: Diagrama de casos de uso

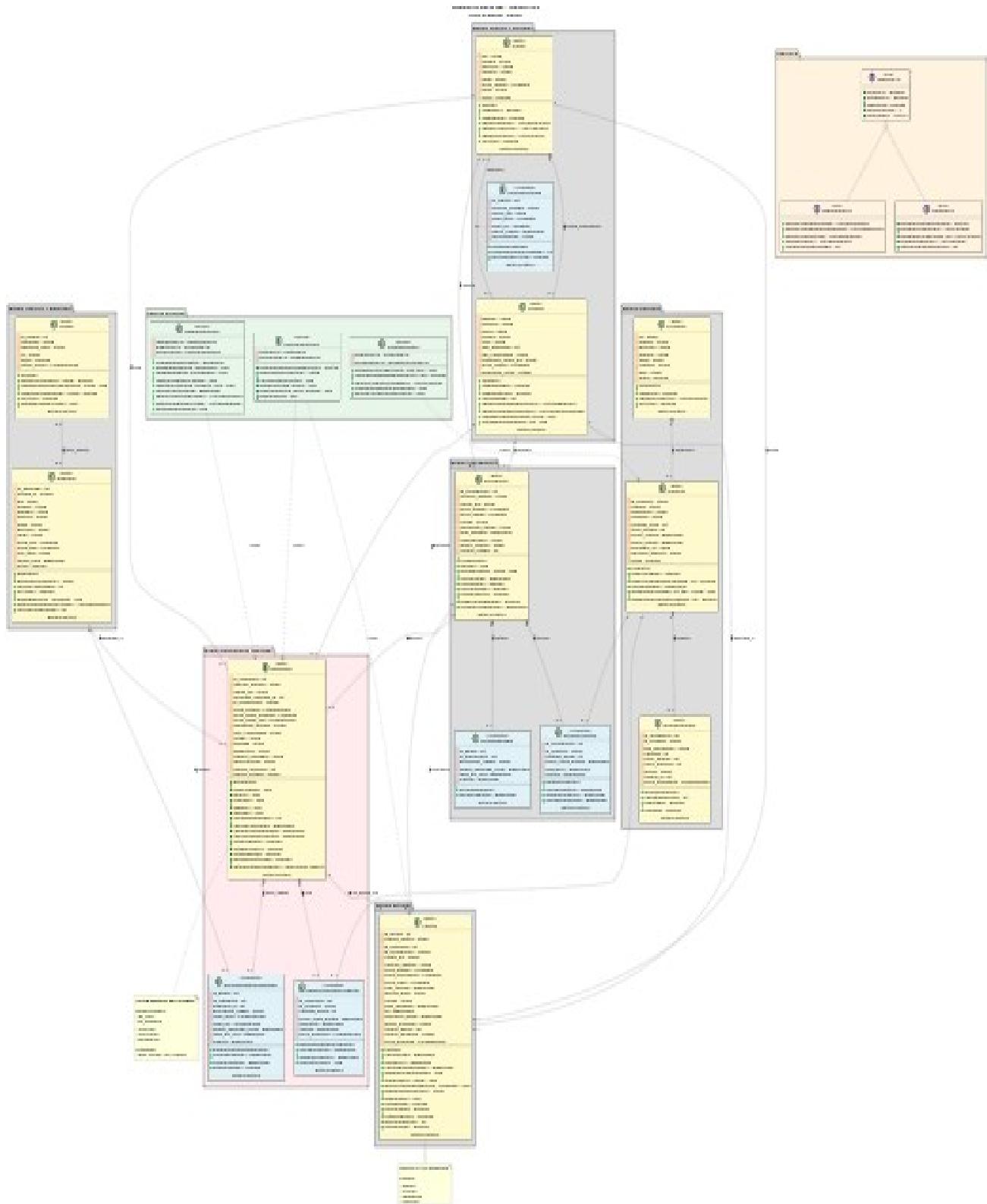


## Anexo II: Diagrama E/R



Color	Tipo
Entidad/Tabla	
campo	Clave Primaria
campo	Clave Foránea
campo	Único
«calculado»	Campo calculado
«nullable»	Puede ser NULL

## Anexo III: Diagrama de Clases



## Anexo IV: Script de Base de Datos

```
--  
=====  
==  
-- SCRIPT DE BASE DE DATOS - APPTaller  
-- Versión: 1.0  
-- Autor: José Luis Cárdenas Barroso  
-- Fecha: Diciembre 2025  
--  
=====  
==  
  
-- Creación de la base de datos  
CREATE DATABASE IF NOT EXISTS apptaller_db  
CHARACTER SET utf8mb4  
COLLATE utf8mb4_unicode_ci;  
  
USE apptaller_db;  
  
--  
=====  
==  
-- SECCIÓN 1: GESTIÓN DE USUARIOS Y EMPLEADOS  
--  
=====  
==  
  
-- Tabla de Usuarios (Autenticación y control de acceso)  
CREATE TABLE Usuarios (  
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    password_hash VARCHAR(255) NOT NULL,  
    rol ENUM('Empleado', 'Administrador') NOT NULL DEFAULT 'Empleado',  
    activo BOOLEAN DEFAULT TRUE,  
    ultimo_acceso DATETIME,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    INDEX idx_username (username),  
    INDEX idx_rol (rol)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- Tabla de Empleados (Datos personales y profesionales)
CREATE TABLE Empleados (
    id_empleado INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT UNIQUE COMMENT 'FK opcional a Usuarios. NULL si no tiene acceso
al sistema',
    dni VARCHAR(10) UNIQUE NOT NULL,
    nombre VARCHAR(100) NOT NULL,
    apellidos VARCHAR(100) NOT NULL,
    telefono VARCHAR(15),
    email VARCHAR(100) UNIQUE,
    direccion VARCHAR(255),
    cargo VARCHAR(50) DEFAULT 'Mecánico' COMMENT 'Ej: Mecánico, Jefe de Taller,
Repcionista',
    fecha_alta DATE NOT NULL,
    fecha_baja DATE,
    foto_perfil VARCHAR(255) COMMENT 'Ruta relativa a la imagen de perfil',
    salario_base DECIMAL(10, 2),
    activo BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (usuario_id) REFERENCES Usuarios(id_usuario) ON DELETE SET
NULL,
    INDEX idx_dni (dni),
    INDEX idx_activo (activo)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- =====
==  

-- SECCIÓN 2: GESTIÓN DE CLIENTES Y VEHÍCULOS  

-- =====
==
```

```
-- Tabla de Clientes
CREATE TABLE Clientes (
    dni VARCHAR(10) PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    direccion VARCHAR(255),
```

```

telefono VARCHAR(15),
email VARCHAR(100) UNIQUE,
fecha_registro DATE NOT NULL,
notas TEXT COMMENT 'Observaciones sobre el cliente',
activo BOOLEAN DEFAULT TRUE,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

INDEX idx_nombre (nombre),
INDEX idx_telefono (telefono),
INDEX idx_email (email)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

-- Tabla de Vehículos

```

CREATE TABLE Vehiculos (
bastidor VARCHAR(50) PRIMARY KEY,
matricula VARCHAR(10) NOT NULL UNIQUE,
marca VARCHAR(50),
modelo VARCHAR(50),
color VARCHAR(30),
anio_fabricacion YEAR,
tipo_combustible ENUM('Gasolina', 'Diesel', 'Híbrido', 'Eléctrico', 'GLP', 'Otro'),
propietario_actual_dni VARCHAR(10),
fecha_compra DATE,
kilometraje_actual INT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

```

```

FOREIGN KEY (propietario_actual_dni) REFERENCES Clientes(dni) ON DELETE SET
NULL,
INDEX idx_matricula (matricula),
INDEX idx_propietario (propietario_actual_dni),
INDEX idx_marca_modelo (marca, modelo)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

-- Tabla de Historial\_Propiedad (Relación M:N histórica)

```

CREATE TABLE Historial_Propiedad (
id_registro INT AUTO_INCREMENT PRIMARY KEY,
vehiculo_bastidor VARCHAR(50) NOT NULL,
cliente_dni VARCHAR(10) NOT NULL,
fecha_inicio DATE NOT NULL,
fecha_fin DATE,
precio_compra DECIMAL(10, 2),

```

observaciones TEXT,

```

FOREIGN KEY (vehiculo_bastidor) REFERENCES Vehiculos(bastidor) ON DELETE
CASCADE,
FOREIGN KEY (cliente_dni) REFERENCES Clientes(dni) ON DELETE CASCADE,
INDEX idx_vehiculo (vehiculo_bastidor),
INDEX idx_cliente (cliente_dni),
INDEX idx_fecha_inicio (fecha_inicio)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

--

=====

==

-- SECCIÓN 3: GESTIÓN DE PROVEEDORES E INVENTARIO

--

=====

==

-- Tabla de Proveedores

```

CREATE TABLE Proveedores (
    cif VARCHAR(10) PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    direccion VARCHAR(255),
    telefono VARCHAR(15),
    email VARCHAR(100) UNIQUE,
    contacto VARCHAR(100) COMMENT 'Nombre de la persona de contacto',
    web VARCHAR(150),
    activo BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

```

INDEX idx\_nombre (nombre)

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

-- Tabla de Productos (Suministros y piezas)

```

CREATE TABLE Productos (
    id_producto VARCHAR(50) PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT,
    categoria VARCHAR(50) COMMENT 'Ej: Aceites, Filtros, Neumáticos, Frenos, etc.',
    cantidad_stock INT NOT NULL DEFAULT 0,

```

```

stock_minimo INT DEFAULT 5 COMMENT 'Umbral para alertas de stock bajo',
precio_compra DECIMAL(10, 2),
precio_unitario DECIMAL(10, 2) NOT NULL COMMENT 'Precio de venta',
proveedor_cif VARCHAR(10),
ubicacion_almacen VARCHAR(50),
activo BOOLEAN DEFAULT TRUE,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
FOREIGN KEY (proveedor_cif) REFERENCES Proveedores(cif) ON DELETE SET
NULL,
INDEX idx_nombre (nombre),
INDEX idx_categoria (categoria),
INDEX idx_stock_bajo (cantidad_stock),
INDEX idx_proveedor (proveedor_cif),
CONSTRAINT chk_stock_no_negativo CHECK (cantidad_stock >= 0),
CONSTRAINT chk_precio_positivo CHECK (precio_unitario > 0)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```

-- Tabla de Movimientos_Stock (Auditoría de cambios en inventario)
CREATE TABLE Movimientos_Stock (
    id_movimiento INT AUTO_INCREMENT PRIMARY KEY,
    id_producto VARCHAR(50) NOT NULL,
    tipo_movimiento ENUM('ENTRADA', 'SALIDA', 'AJUSTE', 'DEVOLUCION') NOT
NULL,
    cantidad INT NOT NULL,
    stock_anterior INT NOT NULL,
    stock_posterior INT NOT NULL,
    motivo VARCHAR(255),
    usuario_id INT,
    fecha_movimiento DATETIME DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (id_producto) REFERENCES Productos(id_producto) ON DELETE
CASCADE,
FOREIGN KEY (usuario_id) REFERENCES Usuarios(id_usuario) ON DELETE SET
NULL,
INDEX idx_producto (id_producto),
INDEX idx_fecha (fecha_movimiento),
INDEX idx_tipo (tipo_movimiento)

```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
--
```

```
=====
```

```
--
```

```
-- SECCIÓN 4: GESTIÓN DE PRESUPUESTOS
```

```
--
```

```
=====
```

```
--
```

```
-- Tabla de Presupuestos
```

```
CREATE TABLE Presupuestos (
```

```
    id_presupuesto INT AUTO_INCREMENT PRIMARY KEY,
    vehiculo_bastidor VARCHAR(50) NOT NULL,
    cliente_dni VARCHAR(10) NOT NULL,
    fecha_emision DATE NOT NULL,
    fecha_validez DATE COMMENT 'Fecha hasta la que es válido el presupuesto',
    estado ENUM('PENDIENTE', 'APROBADO', 'RECHAZADO') NOT NULL DEFAULT
    'PENDIENTE',
    descripcion_trabajo TEXT,
    total_estimado DECIMAL(10, 2),
    observaciones TEXT,
    motivo_rechazo TEXT COMMENT 'Si estado=RECHAZADO, motivo del rechazo',
    usuario_creador INT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
```

```
    FOREIGN KEY (vehiculo_bastidor) REFERENCES Vehiculos(bastidor),
    FOREIGN KEY (cliente_dni) REFERENCES Clientes(dni),
    FOREIGN KEY (usuario_creador) REFERENCES Usuarios(id_usuario) ON DELETE
    SET NULL,
    INDEX idx_cliente (cliente_dni),
    INDEX idx_vehiculo (vehiculo_bastidor),
    INDEX idx_estado (estado),
    INDEX idx_fecha_emision (fecha_emision),
```

```
    CONSTRAINT chk_total_positivo CHECK (total_estimado >= 0)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

-- Tabla Detalle\_Mano\_Obra (Líneas de trabajo del presupuesto)

```
CREATE TABLE Detalle_Mano_Obra (
    id_detalle INT AUTO_INCREMENT PRIMARY KEY,
    id_presupuesto INT NOT NULL,
    descripcion_trabajo VARCHAR(255) NOT NULL,
    tiempo_empleado_horas DECIMAL(5, 2) NOT NULL,
    tarifa_por_hora DECIMAL(10, 2) NOT NULL,
    subtotal DECIMAL(10, 2) GENERATED ALWAYS AS (tiempo_empleado_horas *
tarifa_por_hora) STORED,
    FOREIGN KEY (id_presupuesto) REFERENCES Presupuestos(id_presupuesto) ON
DELETE CASCADE,
    INDEX idx_presupuesto (id_presupuesto),
    CONSTRAINT chk_tiempo_positivo CHECK (tiempo_empleado_horas > 0),
    CONSTRAINT chk_tarifa_positiva CHECK (tarifa_por_hora > 0)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

-- Tabla Detalle\_Productos (Productos usados en el presupuesto)

```
CREATE TABLE Detalle_Productos (
    id_presupuesto INT NOT NULL,
    id_producto VARCHAR(50) NOT NULL,
    cantidad_usada INT NOT NULL,
    precio_venta_unitario DECIMAL(10, 2) NOT NULL,
    descuento DECIMAL(5, 2) DEFAULT 0 CHECK (descuento >= 0 AND descuento <=
100),
    subtotal DECIMAL(10, 2) GENERATED ALWAYS AS (
        cantidad_usada * precio_venta_unitario * (1 - descuento/100)
    ) STORED,
    PRIMARY KEY (id_presupuesto, id_producto),
    FOREIGN KEY (id_presupuesto) REFERENCES Presupuestos(id_presupuesto) ON
DELETE CASCADE,
    FOREIGN KEY (id_producto) REFERENCES Productos(id_producto),
    CONSTRAINT chk_cantidad_positiva CHECK (cantidad_usada > 0)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

--

=====

==

-- SECCIÓN 5: GESTIÓN DE REPARACIONES (NUEVO - NÚCLEO OPERATIVO)

--

=====

--

-- Tabla de Reparaciones (Órdenes de trabajo)

CREATE TABLE Reparaciones (

```
    id_reparacion INT AUTO_INCREMENT PRIMARY KEY,
    vehiculo_bastidor VARCHAR(50) NOT NULL,
    cliente_dni VARCHAR(10) NOT NULL,
    empleado_asignado_id INT NOT NULL COMMENT 'Mecánico responsable',
    id_presupuesto INT NULL COMMENT 'FK opcional. Puede haber reparaciones sin
presupuesto previo',
```

-- Fechas y tiempos

```
    fecha_entrada DATETIME NOT NULL,
    fecha_salida_estimada DATE,
    fecha_salida_real DATETIME,
```

-- Datos del vehículo al entrar

```
    kilometraje_entrada INT,
    nivel_combustible ENUM('Vacío', '1/4', '1/2', '3/4', 'Lleno'),
```

-- Estado de la reparación

```
    estado ENUM(
        'EN_COLA',      -- Vehículo en espera, no se ha empezado
        'EN_PROCESO',   -- Se está reparando actualmente
        'PAUSADA',      -- Trabajo pausado (esperando piezas, autorización, etc.)
        'FINALIZADA',   -- Reparación completada, pendiente de facturar/entregar
        'ENTREGADA'     -- Vehículo entregado al cliente
    ) NOT NULL DEFAULT 'EN_COLA',
```

prioridad ENUM('Baja', 'Normal', 'Alta', 'Urgente') DEFAULT 'Normal',

-- Observaciones

```
    diagnostico TEXT COMMENT 'Diagnóstico inicial del problema',
    trabajos_realizados TEXT COMMENT 'Descripción detallada de los trabajos',
    observaciones TEXT,
```

-- Auditoría

usuario\_recepcion INT COMMENT 'Usuario que registró la entrada',

```

usuario_entrega INT COMMENT 'Usuario que entregó el vehículo',
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
FOREIGN KEY (vehiculo_bastidor) REFERENCES Vehiculos(bastidor),
FOREIGN KEY (cliente_dni) REFERENCES Clientes(dni),
FOREIGN KEY (empleado_asignado_id) REFERENCES Empleados(id_empleado),
FOREIGN KEY (id_presupuesto) REFERENCES Presupuestos(id_presupuesto) ON
DELETE SET NULL,
FOREIGN KEY (usuario_recepcion) REFERENCES Usuarios(id_usuario) ON DELETE
SET NULL,
FOREIGN KEY (usuario_entrega) REFERENCES Usuarios(id_usuario) ON DELETE
SET NULL,
INDEX idx_vehiculo (vehiculo_bastidor),
INDEX idx_cliente (cliente_dni),
INDEX idx_empleado (empleado_asignado_id),
INDEX idx_estado (estado),
INDEX idx_fecha_entrada (fecha_entrada),
INDEX idx_prioridad (prioridad),
CONSTRAINT chk_fecha_salida_posterior CHECK (
    fecha_salida_real IS NULL OR fecha_salida_real >= fecha_entrada
)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

-- Tabla de Reparacion\_Detalle\_Mano\_Obra (Trabajo real realizado en la reparación)  
-- Puede diferir del presupuesto si hubo cambios durante la reparación

```

CREATE TABLE Reparacion_Detalle_Mano_Obra (
    id_detalle INT AUTO_INCREMENT PRIMARY KEY,
    id_reparacion INT NOT NULL,
    empleado_id INT NOT NULL COMMENT 'Mecánico que realizó este trabajo específico',
    descripcion_trabajo VARCHAR(255) NOT NULL,
    fecha_inicio DATETIME,
    fecha_fin DATETIME,
    tiempo_empleado_horas DECIMAL(5, 2) NOT NULL,
    tarifa_por_hora DECIMAL(10, 2) NOT NULL,
    subtotal DECIMAL(10, 2) GENERATED ALWAYS AS (tiempo_empleado_horas *
    tarifa_por_hora) STORED,

```

```

FOREIGN KEY (id_reparacion) REFERENCES Reparaciones(id_reparacion) ON
DELETE CASCADE,
FOREIGN KEY (empleado_id) REFERENCES Empleados(id_empleado),
INDEX idx_reparacion (id_reparacion),
INDEX idx_empleado (empleado_id),

CONSTRAINT chk_tiempo_real_positivo CHECK (tiempo_empleado_horas > 0)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```

-- Tabla de Reparacion_Detalle_Productos (Productos realmente usados)
CREATE TABLE Reparacion_Detalle_Productos (
    id_reparacion INT NOT NULL,
    id_producto VARCHAR(50) NOT NULL,
    cantidad_usada INT NOT NULL,
    precio_venta_unitario DECIMAL(10, 2) NOT NULL,
    descuento DECIMAL(5, 2) DEFAULT 0 CHECK (descuento >= 0 AND descuento <=
100),
    subtotal DECIMAL(10, 2) GENERATED ALWAYS AS (
        cantidad_usada * precio_venta_unitario * (1 - descuento/100)
    ) STORED,
    fecha_aplicacion DATETIME DEFAULT CURRENT_TIMESTAMP,

```

```

PRIMARY KEY (id_reparacion, id_producto),
FOREIGN KEY (id_reparacion) REFERENCES Reparaciones(id_reparacion) ON
DELETE CASCADE,
FOREIGN KEY (id_producto) REFERENCES Productos(id_producto),

CONSTRAINT chk_cantidad_real_positiva CHECK (cantidad_usada > 0)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```

--
=====
==
-- SECCIÓN 6: GESTIÓN DE FACTURAS
--
=====

==
```

```

-- Tabla de Facturas (Modificada para vincular con Reparaciones)
CREATE TABLE Facturas (
    id_factura INT AUTO_INCREMENT PRIMARY KEY,

```

numero\_factura VARCHAR(20) NOT NULL UNIQUE COMMENT 'Formato: FACT-YYYY-NNNN',

-- Relaciones (una factura se basa en una reparación, que puede tener un presupuesto)  
 id\_reparacion INT NOT NULL UNIQUE COMMENT 'Relación 1:1 con Reparaciones',  
 id\_presupuesto INT NULL COMMENT 'Presupuesto origen (si existe)',

cliente\_dni VARCHAR(10) NOT NULL,  
 vehiculo\_bastidor VARCHAR(50) NOT NULL,

-- Fechas y montos  
 fecha\_emision DATE NOT NULL,  
 fecha\_vencimiento DATE,  
 fecha\_cobro DATE,  
 total\_cobrado DECIMAL(10, 2) NOT NULL,

-- Método de pago  
 metodo\_pago ENUM('Efectivo', 'Tarjeta', 'Transferencia', 'Financiación', 'Mixto'),

-- Estado  
 estado ENUM('EMITIDA', 'PAGADA', 'PENDIENTE', 'ANULADA') DEFAULT  
 'EMITIDA',  
 motivo\_anulacion TEXT,

-- IVA y descuentos  
 base\_imponible DECIMAL(10, 2),  
 iva DECIMAL(5, 2) DEFAULT 21.00,  
 descuento\_global DECIMAL(5, 2) DEFAULT 0,

-- Observaciones y auditoría  
 observaciones TEXT,  
 usuario\_emisor INT,  
 fecha\_anulacion DATETIME,  
 usuario\_anulacion INT,  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_reparacion) REFERENCES Reparaciones(id\_reparacion),  
 FOREIGN KEY (id\_presupuesto) REFERENCES Presupuestos(id\_presupuesto) ON  
 DELETE SET NULL,  
 FOREIGN KEY (cliente\_dni) REFERENCES Clientes(dni),  
 FOREIGN KEY (vehiculo\_bastidor) REFERENCES Vehiculos(bastidor),

FOREIGN KEY (usuario\_emisor) REFERENCES Usuarios(id\_usuario) ON DELETE SET NULL,

FOREIGN KEY (usuario\_anulacion) REFERENCES Usuarios(id\_usuario) ON DELETE SET NULL,

INDEX idx\_numero\_factura (numero\_factura),  
 INDEX idx\_cliente (cliente\_dni),  
 INDEX idx\_fecha\_emision (fecha\_emision),  
 INDEX idx\_estado (estado),  
 INDEX idx\_reparacion (id\_reparacion),

CONSTRAINT chk\_total\_factura\_positivo CHECK (total\_cobrado >= 0)  
 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

=====

==

-- SECCIÓN 7: TRIGGERS Y FUNCIONES AUXILIARES

--

=====

==

-- Trigger: Actualizar stock al crear/modificar detalle de reparación  
 DELIMITER //

```
CREATE TRIGGER after_reparacion_producto_insert
AFTER INSERT ON Reparacion_Detalle_Productos
FOR EACH ROW
BEGIN
    -- Descontar stock
    UPDATE Productos
    SET cantidad_stock = cantidad_stock - NEW.cantidad_usada
    WHERE id_producto = NEW.id_producto;
```

-- Registrar movimiento

```
INSERT INTO Movimientos_Stock (
    id_producto,
    tipo_movimiento,
    cantidad,
    stock_anterior,
```

```

stock_posterior,
motivo
)
SELECT
    NEW.id_producto,
    'SALIDA',
    NEW.cantidad_usada,
    cantidad_stock + NEW.cantidad_usada,
    cantidad_stock,
    CONCAT('Usado en reparación #', NEW.id_reparacion)
FROM Productos
WHERE id_producto = NEW.id_producto;
END//


-- Trigger: Actualizar kilometraje del vehículo al finalizar reparación
CREATE TRIGGER after_reparacion_finalizada
AFTER UPDATE ON Reparaciones
FOR EACH ROW
BEGIN
    IF NEW.estado = 'ENTREGADA' AND OLD.estado != 'ENTREGADA' THEN
        UPDATE Vehiculos
        SET kilometraje_actual = NEW.kilometraje_entrada
        WHERE bastidor = NEW.vehiculo_bastidor
        AND (kilometraje_actual IS NULL OR NEW.kilometraje_entrada > kilometraje_actual);
    END IF;
END//


-- Trigger: Validar que una factura solo se puede anular si no tiene más de 30 días
CREATE TRIGGER before_factura_anular
BEFORE UPDATE ON Facturas
FOR EACH ROW
BEGIN
    IF NEW.estado = 'ANULADA' AND OLD.estado != 'ANULADA' THEN
        IF DATEDIFF(NOW(), OLD.fecha_emision) > 30 THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'No se puede anular una factura con más de 30 días de
antigüedad';
        END IF;
        SET NEW.fecha_anulacion = NOW();
    END IF;
END//
```

DELIMITER ;

--

=====

==

-- SECCIÓN 8: VISTAS ÚTILES

--

=====

==

-- Vista: Vehículos actualmente en el taller

CREATE VIEW vw\_vehiculos\_en\_taller AS

SELECT

r.id\_reparacion,

r.vehiculo\_bastidor,

v.matricula,

v.marca,

v.modelo,

c.nombre AS nombre\_cliente,

c.telefono AS telefono\_cliente,

e.nombre AS nombre\_empleado,

e.apellidos AS apellidos\_empleado,

r.fecha\_entrada,

r.fecha\_salida\_estimada,

r.estado,

r.prioridad,

r.diagnostico,

DATEDIFF(NOW(), r.fecha\_entrada) AS dias\_en\_taller

FROM Reparaciones r

INNER JOIN Vehiculos v ON r.vehiculo\_bastidor = v.bastidor

INNER JOIN Clientes c ON r.cliente\_dni = c.dni

INNER JOIN Empleados e ON r.empleado\_asignado\_id = e.id\_empleado

WHERE r.estado IN ('EN\_COLA', 'EN\_PROCESO', 'PAUSADA', 'FINALIZADA')

ORDER BY r.prioridad DESC, r.fecha\_entrada ASC;

-- Vista: Productos con stock bajo

CREATE VIEW vw\_productos\_stock\_bajo AS

SELECT

p.id\_producto,

```

p.nombre,
p.categoría,
p.cantidad_stock,
p.stock_minimo,
(p.stock_minimo - p.cantidad_stock) AS cantidad_faltante,
pr.nombre AS proveedor,
pr.telefono AS telefono_proveedor,
pr.email AS email_proveedor
FROM Productos p
LEFT JOIN Proveedores pr ON p.proveedor_cif = pr.cif
WHERE p.cantidad_stock <= p.stock_minimo
AND p.activo = TRUE
ORDER BY (p.stock_minimo - p.cantidad_stock) DESC;

-- Vista: Resumen financiero de facturas
CREATE VIEW vw_resumen_facturas AS
SELECT
    DATE_FORMAT(fecha_emision, '%Y-%m') AS mes,
    COUNT(*) AS total_facturas,
    SUM(CASE WHEN estado = 'PAGADA' THEN 1 ELSE 0 END) AS facturas_pagadas,
    SUM(CASE WHEN estado = 'PENDIENTE' THEN 1 ELSE 0 END) AS
facturas_pendientes,
    SUM(total_cobrado) AS total_facturado,
    SUM(CASE WHEN estado = 'PAGADA' THEN total_cobrado ELSE 0 END) AS
total_cobrado,
    SUM(CASE WHEN estado = 'PENDIENTE' THEN total_cobrado ELSE 0 END) AS
total_pendiente
FROM Facturas
WHERE estado != 'ANULADA'
GROUP BY DATE_FORMAT(fecha_emision, '%Y-%m')
ORDER BY mes DESC;

-- Vista: Historial completo de vehículos
CREATE VIEW vw_historial_vehiculo AS
SELECT
    v.bastidor,
    v.matricula,
    'Reparación' AS tipo_registro,
    r.id_reparacion AS id_registro,
    r.fecha_entrada AS fecha,
    r.diagnóstico AS descripción,

```

```

COALESCE(f.total_cobrado, 0) AS importe,
r.estado
FROM Vehiculos v
LEFT JOIN Reparaciones r ON v.bastidor = r.vehiculo_bastidor
LEFT JOIN Facturas f ON r.id_reparacion = f.id_reparacion
UNION ALL
SELECT
    v.bastidor,
    v.matricula,
    'Cambio de Propietario' AS tipo_registro,
    hp.id_registro AS id_registro,
    hp.fecha_inicio AS fecha,
    CONCAT('Nuevo propietario: ', c.nombre) AS descripcion,
    hp.precio_compra AS importe,
    'Completado' AS estado
FROM Vehiculos v
LEFT JOIN Historial_Propiedad hp ON v.bastidor = hp.vehiculo_bastidor
LEFT JOIN Clientes c ON hp.cliente_dni = c.dni
ORDER BY fecha DESC;

```

```

-- =====
==  

-- SECCIÓN 9: DATOS INICIALES (OPCIONAL)
-- =====
==
```

```

-- Usuario administrador por defecto (contraseña: admin123 - CAMBIAR EN
PRODUCCIÓN)
-- Hash BCrypt de "admin123":
$2a$10$N9qo8uLOickgx2ZMRZoMyeIjZAgcfl7p92ldGxad68LJZdL17lhWy
INSERT INTO Usuarios (username, password_hash, rol) VALUES
('admin', '$2a$10$N9qo8uLOickgx2ZMRZoMyeIjZAgcfl7p92ldGxad68LJZdL17lhWy',
'Administrador');
```

```

-- Empleado de ejemplo vinculado al usuario admin
INSERT INTO Empleados (usuario_id, dni, nombre, apellidos, telefono, email, cargo,
fecha_alta) VALUES
```

(1, '12345678Z', 'José Luis', 'Cárdenas Barroso', '666777888', 'admin@apptaller.com', 'Jefe de Taller', CURDATE());

--

=====

==

-- FIN DEL SCRIPT

--

=====

==

## Anexo V: Repositorio GitHub

En el siguiente repositorio de GitHub puede encontrarse las imágenes de los diagramas, así como el script sql de la creación de la base de datos.

<https://github.com/jolucarbar/Proyecto-DAM-Taller>