



**República Bolivariana de Venezuela**  
**Ministerio del Poder Popular para la Educación Universitaria**  
**Universidad Nacional Experimental Para Las Telecomunicaciones e Informática**  
**PNF Ingeniería Informática**  
**Base de Datos I**  
**Sección 6B**

## **FASE 1 DISEÑO DE TABLAS Y RELACIONES**

### **"CONSTRUCCIÓN RENTABLE"**

**Profesora:**  
**Yuly Delgado**

**Ingeniero en Formación:**  
**Mijael Engelmann**  
**C.I:V-31.222.463**

# Índice

## 1. Introducción

## 2. Análisis de Requisitos

- 1. Cliente
- 2. Equipo
- 3. Contrato de Alquiler
- 4. Mantenimiento
- 5. Logística de Entrega y Recogida
- 6. Facturación
- 7. Seguimiento de Ubicación
- 8. Comunicación
- 9. Informes y Estadísticas
- 10. Funcionalidades Avanzadas

## 3. Diagrama DER

## 4. Descripción de las Relaciones entre Tablas del Diagrama DER

- 1. Cliente
- 2. Equipo
- 3. Contrato de Alquiler
- 4. Mantenimiento
- 5. Logística de Entrega y Recogida
- 6. Facturación
- 7. Seguimiento de Ubicación
- 8. Comunicación
- 9. Informes y Estadísticas
- 10. Funcionalidades Avanzadas

## 5. Scripts SQL

## 6. Justificación de Diseño

- 1. Normalización Aplicada
- 2. Elecciones de Diseño

## 7. Conclusión

## 8. Glosario de Términos

## 9. Link al Repositorio en GitHub

# Introducción

El caso de estudio "Construcción Rentable" se enfoca en los desafíos que enfrenta una empresa dedicada al alquiler de maquinaria y equipos de construcción. En este sector, la gestión eficiente de los recursos es fundamental para garantizar la disponibilidad de los equipos, programar mantenimientos adecuados y asegurar la rentabilidad del negocio. Actualmente, la empresa opera con un sistema manual que combina hojas de cálculo, documentos en papel y un software básico de facturación. Aunque este enfoque es funcional, presenta varios problemas: resulta difícil mantener un control preciso del inventario, gestionar contratos de alquiler, planificar mantenimientos (tanto preventivos como correctivos), registrar clientes y coordinar las entregas y recogidas de los equipos.

El principal desafío de la empresa radica en la falta de integración entre sus procesos y la ausencia de automatización. Esto genera ineficiencias diarias, retrasos en la atención al cliente y un alto riesgo de errores en la facturación y el control de inventario. Además, al no contar con un sistema centralizado, el acceso a la información crítica es lento, lo que dificulta la toma de decisiones estratégicas y sobrecarga al equipo administrativo con tareas repetitivas.

Para abordar estos problemas, el proyecto propone diseñar e implementar un sistema integral de gestión que unifique y automatice los procesos mediante una base de datos relacional en PostgreSQL. Este sistema centralizará y optimizará la administración del inventario de maquinaria, la gestión de datos de clientes, los contratos de alquiler, el mantenimiento preventivo y correctivo de los equipos, y la logística de entregas y recolecciones. Este enfoque integrado permitirá eliminar redundancias, mejorar la precisión de los registros y facilitar el acceso a información estratégica en tiempo real.

El sistema debe ser escalable para adaptarse al crecimiento futuro del negocio, incluir capas de seguridad robustas para proteger los datos sensibles y ofrecer una interfaz intuitiva que reduzca la curva de aprendizaje del equipo. Con estas mejoras, se espera reducir los costos operativos, agilizar los procesos internos y mejorar la experiencia del cliente mediante un servicio más rápido y confiable.

# Análisis de Requisitos

A continuación, se presentan las entidades principales y sus atributos clave, junto con las características adicionales implementadas para optimizar el diseño:

## 1. Cliente

**Propósito:** Registrar y gestionar la información de los clientes que alquilan equipos.

### Atributos Clave:

- cliente\_id (PK): Identificador único del cliente.
- nombre: Nombre completo del cliente.
- razon\_social: Razón social (en caso de ser una empresa).
- cif\_nif: CIF o NIF del cliente.
- direccion: Dirección física del cliente.
- teléfono: Teléfono de contacto.
- email: Correo electrónico.
- tipo\_cliente: Tipo de cliente (empresa constructora, particular).
- historial\_alquileres: Historial de alquileres realizados por el cliente.

### Características Adicionales:

- Se agregó un índice en la columna cif\_nif para optimizar las búsquedas por identificador fiscal.
- Se creó una vista simplificada llamada Vista\_Clientes, que incluye solo los datos más relevantes del cliente (nombre, CIF/NIF, email, etc.), facilitando consultas frecuentes.

## 2. Equipo

**Propósito:** Gestionar el inventario de equipos disponibles para alquilar.

### Atributos Clave:

- equipo\_id (PK): Identificador único del equipo.
- tipo: Tipo de equipo (excavadora, grúa, andamio, etc.).
- marca: Marca del equipo.

- modelo: Modelo del equipo.
- año\_fabricacion: Año de fabricación.
- numero\_serie: Número de serie del equipo.
- estado: Estado actual del equipo (disponible, alquilado, en mantenimiento).
- horas\_uso: Horas de uso acumuladas.
- ubicacion\_actual: Ubicación física del equipo.
- historial\_mantenimiento: Historial de mantenimientos realizados.
- certificaciones\_seguridad: Certificaciones de seguridad del equipo.

### **Características Adicionales:**

- Se agregaron índices en las columnas estado y numero\_serie para optimizar las consultas relacionadas con la disponibilidad y el seguimiento de los equipos.
- Se creó una vista simplificada llamada Vista\_Equipos, que muestra información clave del equipo (tipo, marca, modelo, estado, ubicación).

## **3. Contrato de Alquiler**

**Propósito:** Gestionar los contratos de alquiler de equipos.

### **Atributos Clave:**

- contrato\_id (PK): Identificador único del contrato.
- cliente\_id (FK): Cliente que realiza el contrato.
- equipo\_id (FK): Equipo alquilado.
- fecha\_inicio: Fecha de inicio del alquiler.
- fecha\_fin: Fecha de finalización del alquiler.
- lugar\_entrega: Lugar de entrega del equipo.
- lugar\_recogida: Lugar de recogida del equipo.
- precio\_alquiler: Precio total del alquiler.
- condiciones\_contrato: Condiciones específicas del contrato.
- kilometraje\_inicio: Kilometraje o horas de uso al inicio del alquiler.
- kilometraje\_fin: Kilometraje o horas de uso al final del alquiler.

### **Características Adicionales:**

- Se agregaron índices en las columnas cliente\_id y equipo\_id para optimizar las consultas relacionadas con contratos específicos.
- Se creó una vista simplificada llamada Vista\_Contratos, que muestra información clave del contrato (cliente, equipo, fechas, precio).

## **4. Mantenimiento**

**Propósito:** Gestionar el mantenimiento de los equipos.

### **Atributos Clave:**

- mantenimiento\_id (PK): Identificador único del mantenimiento.
- equipo\_id (FK): Equipo al que se le realiza el mantenimiento.
- tipo\_mantenimiento: Tipo de mantenimiento (preventivo, correctivo).
- fecha\_mantenimiento: Fecha en que se realizó el mantenimiento.
- taller\_tecnico: Taller o técnico responsable.
- descripcion\_trabajo: Descripción del trabajo realizado.
- costo\_mantenimiento: Costo del mantenimiento.
- proxima\_fecha\_mantenimiento: Fecha estimada para el próximo mantenimiento.

### **Características Adicionales:**

- Se agregó un índice en la columna equipo\_id para optimizar las consultas relacionadas con el historial de mantenimiento.
- Se creó una vista simplificada llamada Vista\_Mantenimientos, que muestra información clave del mantenimiento (equipo, tipo, fecha, costo).

## **5. Logística de Entrega y Recogida**

**Propósito:** Gestionar la logística de entrega y recogida de equipos.

### **Atributos Clave:**

- logística\_id (PK): Identificador único de la logística.
- contrato\_id (FK): Contrato de alquiler asociado.
- equipo\_id (FK): Equipo a entregar o recoger.

- direccion\_entrega: Dirección de entrega del equipo.
- fecha\_hora\_entrega: Fecha y hora de entrega.
- direccion\_recogida: Dirección de recogida del equipo.
- fecha\_hora\_recogida: Fecha y hora de recogida.
- transportista\_vehiculo: Transportista o vehículo utilizado.

#### **Características Adicionales:**

- Se agregaron índices en las columnas contrato\_id y equipo\_id para optimizar las consultas relacionadas con entregas y recogidas.
- Se creó una vista simplificada llamada Vista\_Logistica, que muestra información clave de la logística (contrato, equipo, direcciones, fechas).

### **6. Facturación**

**Propósito:** Gestionar la facturación de los contratos de alquiler.

#### **Atributos Clave:**

- factura\_id (PK): Identificador único de la factura.
- contrato\_id (FK): Contrato de alquiler asociado.
- fecha\_factura: Fecha de emisión de la factura.
- importe: Importe total de la factura.
- estado\_factura: Estado de la factura (pagada, pendiente).

#### **Características Adicionales:**

- Se agregó un índice en la columna contrato\_id para optimizar las consultas relacionadas con facturas específicas.
- Se creó una vista simplificada llamada Vista\_Facturas, que muestra información clave de la factura (contrato, fecha, importe, estado).

### **7. Seguimiento de Ubicación**

**Propósito:** Gestionar el seguimiento de la ubicación de los equipos.

#### **Atributos Clave:**

- seguimiento\_id (PK): Identificador único del seguimiento.

- equipo\_id (FK): Equipo al que se le realiza el seguimiento.
- ubicacion\_actual: Ubicación actual del equipo.
- fecha\_hora\_actualizacion: Fecha y hora de la última actualización.

#### **Características Adicionales:**

- Se agregó un índice en la columna equipo\_id para optimizar las consultas relacionadas con el historial de ubicaciones.
- Se creó una vista simplificada llamada Vista\_Seguimiento\_Ubicacion, que muestra información clave del seguimiento (equipo, ubicación, fecha).

### **8. Comunicación**

**Propósito:** Gestionar la comunicación con los clientes.

#### **Atributos Clave:**

- comunicacion\_id (PK): Identificador único de la comunicación.
- cliente\_id (FK): Cliente al que se le envía la comunicación.
- tipo\_comunicacion: Tipo de comunicación (email, SMS, llamada).
- fecha\_hora\_comunicacion: Fecha y hora de la comunicación.
- mensaje: Contenido del mensaje.

#### **Características Adicionales:**

- Se agregó un índice en la columna cliente\_id para optimizar las consultas relacionadas con el historial de comunicaciones.
- Se creó una vista simplificada llamada Vista\_Comunicaciones, que muestra información clave de la comunicación (cliente, tipo, fecha, mensaje).

### **9. Informes y Estadísticas**

**Propósito:** Generar informes y estadísticas sobre los alquileres, ingresos, mantenimiento y disponibilidad de equipos.

#### **Atributos Clave:**

- informe\_id (PK): Identificador único del informe.
- tipo\_informe: Tipo de informe (alquileres por equipo, ingresos, mantenimiento, disponibilidad).



- fecha\_generacion: Fecha de generación del informe.
- datos\_informe: Datos del informe en formato JSON (para flexibilidad).

#### **Características Adicionales:**

- Se agregó un índice en la columna contrato\_id para optimizar las consultas relacionadas con informes específicos.
- Se creó una vista simplificada llamada Vista\_Informes, que muestra información clave del informe (tipo, fecha, contrato).

### **10. Funcionalidades Avanzadas**

**Propósito:** Gestionar funcionalidades avanzadas como la integración con sistemas de gestión de almacenes (WMS), seguros, control de acceso y aplicaciones móviles.

#### **Atributos Clave:**

- funcionalidad\_id (PK): Identificador único de la funcionalidad.
- tipo\_funcionalidad: Tipo de funcionalidad (WMS, seguros, control de acceso, app móvil).
- descripcion: Descripción de la funcionalidad.
- configuracion: Configuración de la funcionalidad en formato JSON.

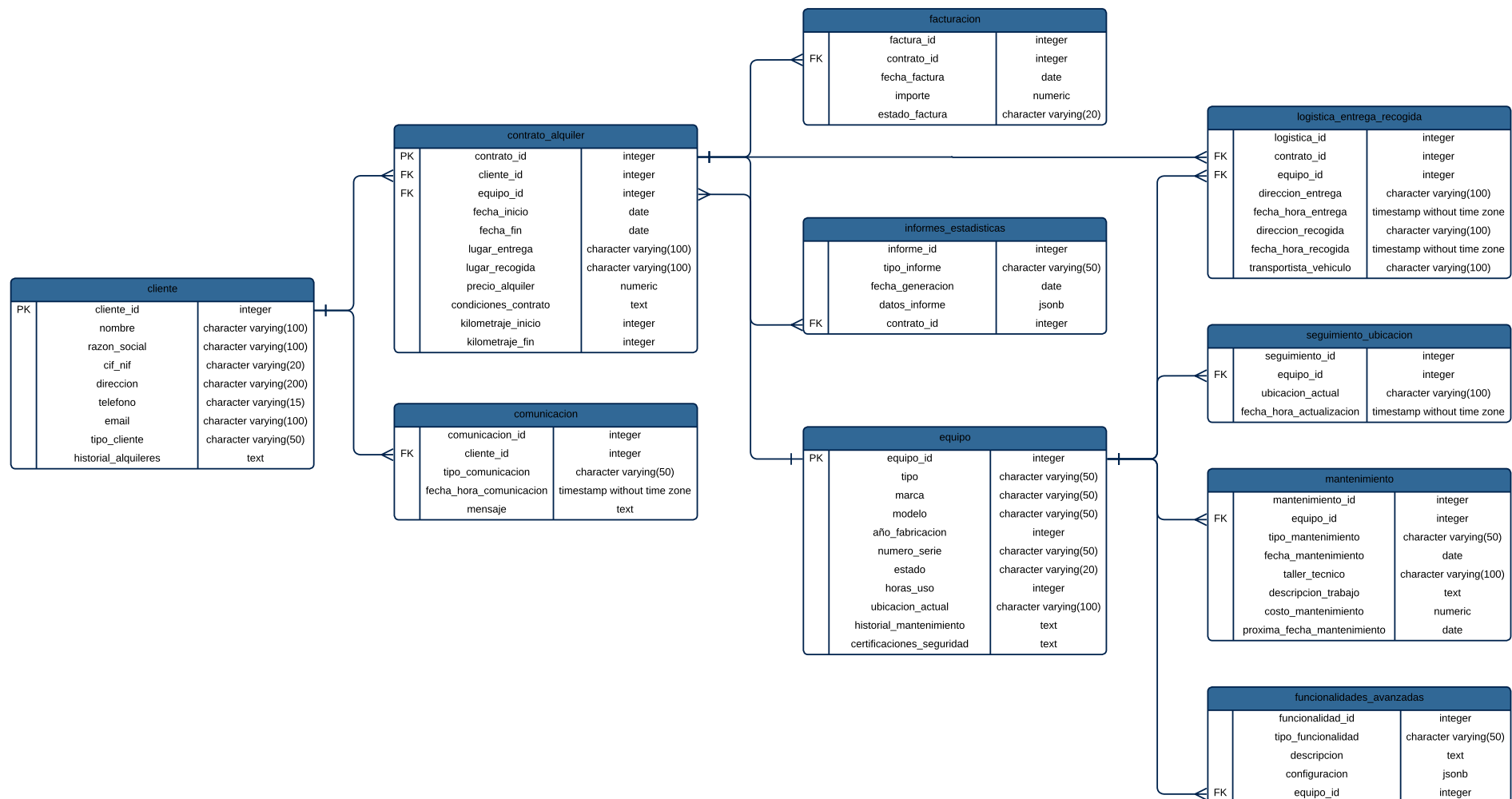
#### **Características Adicionales:**

- Se agregó un índice en la columna equipo\_id para optimizar las consultas relacionadas con funcionalidades específicas.
- Se creó una vista simplificada llamada Vista\_Funcionalidades\_Avanzadas, que muestra información clave de la funcionalidad (tipo, descripción, equipo).

# Diagrama DER

El **Diagrama Entidad-Relación (DER)** representa gráficamente las entidades, atributos y relaciones del sistema. A continuación, se describe el proceso para crearlo:

## Modelo Entidad - Relación



# Descripción de las Relaciones entre Tablas del Diagrama DER

## 1. Cliente

### Atributos Clave:

- **cliente\_id (PK):** Identificador único del cliente.
- **nombre:** Nombre del cliente.
- **razon\_social:** Razón social (en caso de ser una empresa).
- **cif\_nif:** CIF/NIF del cliente.
- **direccion:** Dirección del cliente.
- **teléfono:** Teléfono de contacto.
- **email:** Correo electrónico.
- **tipo\_cliente:** Tipo de cliente (empresa constructora, particular).
- **historial\_alquileres:** Historial de alquileres del cliente.

**Descripción:** Representa a los clientes de la empresa.

### Relaciones:

- **1 a Muchos con Contrato de Alquiler:** Un cliente puede tener varios contratos de alquiler.
- **1 a Muchos con Comunicación:** Un cliente puede recibir varias comunicaciones.

## 2. Equipo

### Atributos Clave:

- **equipo\_id (PK):** Identificador único del equipo.
- **tipo:** Tipo de equipo (excavadora, grúa, andamio, etc.).
- **marca:** Marca del equipo.
- **modelo:** Modelo del equipo.
- **año\_fabricacion:** Año de fabricación.
- **numero\_serie:** Número de serie del equipo.
- **estado:** Estado del equipo (disponible, alquilado, en mantenimiento).
- **horas\_uso:** Horas de uso del equipo.

- **ubicacion\_actual:** Ubicación actual del equipo.
- **historial\_mantenimiento:** Historial de mantenimiento del equipo.
- **certificaciones\_seguridad:** Certificaciones de seguridad del equipo.

**Descripción:** Representa los equipos disponibles para alquiler.

**Relaciones:**

- **1 a Muchos con Contrato de Alquiler:** Un equipo puede estar en varios contratos.
- **1 a Muchos con Mantenimiento:** Un equipo puede tener varios registros de mantenimiento.
- **1 a Muchos con Seguimiento de Ubicación:** Un equipo puede tener varios registros de seguimiento.
- **1 a Muchos con Funcionalidades Avanzadas:** Un equipo puede tener varias funcionalidades asociadas.

### 3. Contrato de Alquiler

**Atributos Clave:**

- **contrato\_id (PK):** Identificador único del contrato.
- **cliente\_id (FK):** Cliente que realiza el contrato.
- **equipo\_id (FK):** Equipo alquilado.
- **fecha\_inicio:** Fecha de inicio del alquiler.
- **fecha\_fin:** Fecha de fin del alquiler.
- **lugar\_entrega:** Lugar de entrega del equipo.
- **lugar\_recogida:** Lugar de recogida del equipo.
- **precio\_alquiler:** Precio del alquiler.
- **condiciones\_contrato:** Condiciones del contrato.
- **kilometraje\_inicio:** Kilometraje/horas de uso al inicio del alquiler.
- **kilometraje\_fin:** Kilometraje/horas de uso al final del alquiler.

**Descripción:** Representa los contratos de alquiler de equipos.

**Relaciones:**

- **Muchos a 1 con Cliente:** Cada contrato pertenece a un cliente.
- **Muchos a 1 con Equipo:** Cada contrato está asociado a un equipo.

- **1 a 1 con Logística de Entrega y Recogida:** Cada contrato tiene una única logística.
- **1 a Muchos con Facturación:** Un contrato puede generar varias facturas.
- **1 a Muchos con Informes y Estadísticas:** Un contrato puede generar varios informes.

#### 4. Mantenimiento

##### Atributos Clave:

- **mantenimiento\_id (PK):** Identificador único del mantenimiento.
- **equipo\_id (FK):** Equipo al que se le realiza el mantenimiento.
- **tipo\_mantenimiento:** Tipo de mantenimiento (preventivo, correctivo).
- **fecha\_mantenimiento:** Fecha del mantenimiento.
- **taller\_tecnico:** Taller o técnico responsable.
- **descripcion\_trabajo:** Descripción del trabajo realizado.
- **costo\_mantenimiento:** Costo del mantenimiento.
- **proxima\_fecha\_mantenimiento:** Próxima fecha de mantenimiento.

**Descripción:** Representa el mantenimiento de los equipos.

##### Relaciones:

- **Muchos a 1 con Equipo:** Cada mantenimiento está asociado a un equipo.

#### 5. Logística de Entrega y Recogida

##### Atributos Clave:

- **logistica\_id (PK):** Identificador único de la logística.
- **contrato\_id (FK):** Contrato de alquiler asociado.
- **equipo\_id (FK):** Equipo a entregar o recoger.
- **direccion\_entrega:** Dirección de entrega.
- **fecha\_hora\_entrega:** Fecha y hora de entrega.
- **direccion\_recogida:** Dirección de recogida.
- **fecha\_hora\_recogida:** Fecha y hora de recogida.
- **transportista\_vehiculo:** Transportista o vehículo utilizado.

**Descripción:** Representa la logística de entrega y recogida de equipos.

**Relaciones:**

- **1 a 1 con Contrato de Alquiler:** Cada logística está asociada a un contrato.

## 6. Facturación

**Atributos Clave:**

- **factura\_id (PK):** Identificador único de la factura.
- **contrato\_id (FK):** Contrato de alquiler asociado.
- **fecha\_factura:** Fecha de la factura.
- **importe:** Importe de la factura.
- **estado\_factura:** Estado de la factura (pagada, pendiente).

**Descripción:** Representa las facturas generadas por los contratos de alquiler.

**Relaciones:**

- **Muchos a 1 con Contrato de Alquiler:** Cada factura está asociada a un contrato.

## 7. Seguimiento de Ubicación

**Atributos Clave:**

- **seguimiento\_id (PK):** Identificador único del seguimiento.
- **equipo\_id (FK):** Equipo al que se le realiza el seguimiento.
- **ubicacion\_actual:** Ubicación actual del equipo.
- **fecha\_hora\_actualizacion:** Fecha y hora de la última actualización.

**Descripción:** Representa el seguimiento de la ubicación de los equipos.

**Relaciones:**

- **Muchos a 1 con Equipo:** Cada seguimiento está asociado a un equipo.

## 8. Comunicación

**Atributos Clave:**

- **comunicacion\_id (PK):** Identificador único de la comunicación.
- **cliente\_id (FK):** Cliente al que se le envía la comunicación.

- **tipo\_comunicacion:** Tipo de comunicación (email, SMS, llamada).
- **fecha\_hora\_comunicacion:** Fecha y hora de la comunicación.
- **mensaje:** Contenido del mensaje.

**Descripción:** Representa la comunicación con los clientes.

**Relaciones:**

- **Muchos a 1 con Cliente:** Cada comunicación está asociada a un cliente.

## 9. Informes y Estadísticas

**Atributos Clave:**

- **informe\_id (PK):** Identificador único del informe.
- **tipo\_informe:** Tipo de informe (alquileres por equipo, ingresos, mantenimiento, disponibilidad).
- **fecha\_generacion:** Fecha de generación del informe.
- **datos\_informe:** Datos del informe en formato JSON (para flexibilidad).

**Descripción:** Representa los informes y estadísticas generados por el sistema.

**Relaciones:**

- **Muchos a 1 con Contrato de Alquiler:** Cada informe está asociado a un contrato.

## 10. Funcionalidades Avanzadas

**Atributos Clave:**

- **funcionalidad\_id (PK):** Identificador único de la funcionalidad.
- **tipo\_funcionalidad:** Tipo de funcionalidad (WMS, seguros, control de acceso, app móvil).
- **descripcion:** Descripción de la funcionalidad.
- **configuracion:** Configuración de la funcionalidad en formato JSON.

**Descripción:** Representa las funcionalidades avanzadas del sistema.

**Relaciones:**

- **Muchos a 1 con Equipo:** Cada funcionalidad está asociada a un equipo.

# Scripts SQL

A continuación, se presenta el código SQL para crear las tablas en PostgreSQL.

## 1. Tabla Cliente

```
1. Tabla Cliente.sql X
1. Tabla Cliente.sql
1  -- Creación de la tabla Cliente
2  CREATE TABLE cliente (
3      cliente_id SERIAL PRIMARY KEY, -- Identificador único del cliente (clave primaria)
4      nombre VARCHAR(100) NOT NULL, -- Nombre del cliente
5      razon_social VARCHAR(100), -- Razón social (en caso de ser una empresa)
6      cif_nif VARCHAR(20) UNIQUE NOT NULL, -- CIF/NIF del cliente (único y obligatorio)
7      direccion VARCHAR(200), -- Dirección del cliente
8      telefono VARCHAR(15), -- Teléfono de contacto
9      email VARCHAR(100), -- Correo electrónico
10     tipo_cliente VARCHAR(50) CHECK (tipo_cliente IN ('empresa constructora', 'particular')), -- Tipo de cliente (empresa o particular)
11     historial_alquileres TEXT -- Historial de alquileres del cliente
12 );
13
14 -- Índice para optimizar búsquedas por CIF/NIF
15 CREATE INDEX idx_cliente_cif_nif ON cliente(cif_nif);
16
17 -- Vista simplificada de la tabla Cliente
18 CREATE VIEW Vista_clientes AS
19 SELECT
20     cliente_id,
21     nombre,
22     razon_social,
23     cif_nif,
24     email,
25     tipo_cliente
26 FROM
27     cliente;
```

## 2. Tabla Equipo

```
2. Tabla Equipo.sql X
2. Tabla Equipo.sql
1  -- Creación de la tabla Equipo
2  CREATE TABLE equipo (
3      equipo_id SERIAL PRIMARY KEY, -- Identificador único del equipo (clave primaria)
4      tipo VARCHAR(50) NOT NULL, -- Tipo de equipo (excavadora, grúa, etc.)
5      marca VARCHAR(50) NOT NULL, -- Marca del equipo
6      modelo VARCHAR(50) NOT NULL, -- Modelo del equipo
7      año_fabricacion INT NOT NULL, -- Año de fabricación del equipo
8      numero_serie VARCHAR(50) UNIQUE NOT NULL, -- Número de serie del equipo (único y obligatorio)
9      estado VARCHAR(20) NOT NULL CHECK (estado IN ('disponible', 'alquilado', 'en mantenimiento')), -- Estado del equipo
10     horas_uso INT CHECK (horas_uso >= 0), -- Horas de uso acumuladas (no puede ser negativo)
11     ubicacion_actual VARCHAR(100), -- Ubicación actual del equipo
12     historial_mantenimiento TEXT, -- Historial de mantenimientos realizados
13     certificaciones_seguridad TEXT -- Certificaciones de seguridad del equipo
14 );
15
16 -- Índices para optimizar búsquedas por estado y número de serie
17 CREATE INDEX idx_equipo_estado ON equipo(estado);
18 CREATE INDEX idx_equipo_numero_serie ON equipo(numero_serie);
19
20 -- Vista simplificada de la tabla Equipo
21 CREATE VIEW Vista_Equipos AS
22 SELECT
23     equipo_id,
24     tipo,
25     marca,
26     modelo,
27     estado,
28     ubicacion_actual
29 FROM
30     equipo;
```



### 3. Tabla Contrato de Alquiler

```
3. Tabla Contrato de Alquiler.sql X
3. Tabla Contrato de Alquiler.sql
1  -- Creación de la tabla Contrato de Alquiler
2  CREATE TABLE contrato_alquiler (
3      contrato_id SERIAL PRIMARY KEY, -- Identificador único del contrato (clave primaria)
4      cliente_id INT REFERENCES cliente(cliente_id), -- Clave foránea: Relación con la tabla Cliente
5      equipo_id INT REFERENCES equipo(equipo_id), -- Clave foránea: Relación con la tabla Equipo
6      fecha_inicio DATE NOT NULL, -- Fecha de inicio del alquiler
7      fecha_fin DATE NOT NULL CHECK (fecha_fin >= fecha_inicio), -- Fecha de fin del alquiler (debe ser mayor o igual a la fecha de inicio)
8      lugar_entrega VARCHAR(100), -- Lugar de entrega del equipo
9      lugar_recogida VARCHAR(100), -- Lugar de recogida del equipo
10     precio_alquiler NUMERIC(10, 2) NOT NULL CHECK (precio_alquiler >= 0), -- Precio del alquiler (no puede ser negativo)
11     condiciones_contrato TEXT, -- Condiciones específicas del contrato
12     kilometraje_inicio INT CHECK (kilometraje_inicio >= 0), -- Kilometraje/horas de uso al inicio del alquiler (no puede ser negativo)
13     kilometraje_fin INT CHECK (kilometraje_fin >= kilometraje_inicio) -- Kilometraje/horas de uso al final del alquiler (debe ser mayor o igual al inicial)
14 );
15
16 -- Índices para optimizar búsquedas por cliente y equipo
17 CREATE INDEX idx_contrato_cliente ON contrato_alquiler(cliente_id);
18 CREATE INDEX idx_contrato_equipo ON contrato_alquiler(equipo_id);
19
20 -- Vista simplificada de la tabla Contrato de Alquiler
21 CREATE VIEW Vista_Contratos AS
22 SELECT
23     contrato_id,
24     cliente_id,
25     equipo_id,
26     fecha_inicio,
27     fecha_fin,
28     precio_alquiler
29 FROM
30     contrato_alquiler;
```

### 4. Tabla Mantenimiento

```
4. Tabla Mantenimiento.sql X
4. Tabla Mantenimiento.sql
1  -- Creación de la tabla Mantenimiento
2  CREATE TABLE mantenimiento (
3      mantenimiento_id SERIAL PRIMARY KEY, -- Identificador único del mantenimiento (clave primaria)
4      equipo_id INT REFERENCES equipo(equipo_id), -- Clave foránea: Relación con la tabla Equipo
5      tipo_mantenimiento VARCHAR(50) NOT NULL CHECK (tipo_mantenimiento IN ('preventivo', 'correctivo')), -- Tipo de mantenimiento (preventivo o correctivo)
6      fecha_mantenimiento DATE NOT NULL, -- Fecha del mantenimiento
7      taller_tecnico VARCHAR(100), -- Taller o técnico responsable
8      descripcion_trabajo TEXT, -- Descripción del trabajo realizado
9      costo_mantenimiento NUMERIC(10, 2) CHECK (costo_mantenimiento >= 0), -- Costo del mantenimiento (no puede ser negativo)
10     proxima_fecha_mantenimiento DATE -- Fecha estimada del próximo mantenimiento
11 );
12
13 -- Índice para optimizar búsquedas por equipo
14 CREATE INDEX idx_mantenimiento_equipo ON mantenimiento(equipo_id);
15
16 -- Vista simplificada de la tabla Mantenimiento
17 CREATE VIEW Vista_Mantenimientos AS
18 SELECT
19     mantenimiento_id,
20     equipo_id,
21     tipo_mantenimiento,
22     fecha_mantenimiento,
23     costo_mantenimiento
24 FROM
25     mantenimiento;
```

## 5. Tabla Logística de Entrega y Recogida

```
5. Tabla Logística de Entrega y Recogida.sql X
5. Tabla Logística de Entrega y Recogida.sql
1  -- Creación de la tabla Logística de Entrega y Recogida
2  CREATE TABLE logistica_entrega_recogida (
3      logistica_id SERIAL PRIMARY KEY, -- Identificador único de la logística (clave primaria)
4      contrato_id INT UNIQUE REFERENCES contrato_alquiler(contrato_id), -- Clave foránea: Relación con la tabla Contrato de Alquiler (única)
5      equipo_id INT REFERENCES equipo(equipo_id), -- Clave foránea: Relación con la tabla Equipo
6      direccion_entrega VARCHAR(100), -- Dirección de entrega del equipo
7      fecha_hora_entrega TIMESTAMP, -- Fecha y hora de entrega
8      direccion_recogida VARCHAR(100), -- Dirección de recogida del equipo
9      fecha_hora_recogida TIMESTAMP CHECK (fecha_hora_recogida >= fecha_hora_entrega), -- Fecha y hora de recogida (debe ser mayor o igual a la de entrega)
10     transportista_vehiculo VARCHAR(100) -- Transportista o vehículo utilizado
11 );
12
13 -- Índices para optimizar búsquedas por contrato y equipo
14 CREATE INDEX idx_logistica_contrato ON logistica_entrega_recogida(contrato_id);
15 CREATE INDEX idx_logistica_equipo ON logistica_entrega_recogida(equipo_id);
16
17 -- Vista simplificada de la tabla Logística de Entrega y Recogida
18 CREATE VIEW Vista_Logistica AS
19 SELECT
20     logistica_id,
21     contrato_id,
22     equipo_id,
23     direccion_entrega,
24     fecha_hora_entrega,
25     direccion_recogida,
26     fecha_hora_recogida
27 FROM
28     logistica_entrega_recogida;
```

## 6. Tabla Facturación

```
6. Tabla Facturación.sql X
6. Tabla Facturación.sql
1  -- Creación de la tabla Facturación
2  CREATE TABLE facturacion (
3      factura_id SERIAL PRIMARY KEY, -- Identificador único de la factura (clave primaria)
4      contrato_id INT REFERENCES contrato_alquiler(contrato_id), -- Clave foránea: Relación con la tabla Contrato de Alquiler
5      fecha_factura DATE NOT NULL, -- Fecha de emisión de la factura
6      importe NUMERIC(10, 2) NOT NULL CHECK (importe >= 0), -- Importe total de la factura (no puede ser negativo)
7      estado_factura VARCHAR(20) CHECK (estado_factura IN ('pagada', 'pendiente')) -- Estado de la factura (pagada o pendiente)
8  );
9
10 -- Índice para optimizar búsquedas por contrato
11 CREATE INDEX idx_facturacion_contrato ON facturacion(contrato_id);
12
13 -- Vista simplificada de la tabla Facturación
14 CREATE VIEW Vista_Facturas AS
15 SELECT
16     factura_id,
17     contrato_id,
18     fecha_factura,
19     importe,
20     estado_factura
21 FROM
22     facturacion;
```

## 7. Tabla Seguimiento de Ubicación

```
7. Tabla Seguimiento de Ubicación.sql X
7. Tabla Seguimiento de Ubicación.sql
1 -- Creación de la tabla Seguimiento de Ubicación
2 CREATE TABLE seguimiento_ubicacion (
3     seguimiento_id SERIAL PRIMARY KEY, -- Identificador único del seguimiento (clave primaria)
4     equipo_id INT REFERENCES equipo(equipo_id), -- Clave foránea: Relación con la tabla Equipo
5     ubicacion_actual VARCHAR(100), -- Ubicación actual del equipo
6     fecha_hora_actualizacion TIMESTAMP CHECK (fecha_hora_actualizacion <= CURRENT_TIMESTAMP) -- Fecha y hora de la última actualización (no puede ser futura)
7 );
8
9 -- Índice para optimizar búsquedas por equipo
10 CREATE INDEX idx_seguimiento_equipo ON seguimiento_ubicacion(equipo_id);
11
12 -- Vista simplificada de la tabla Seguimiento de Ubicación
13 CREATE VIEW Vista_Seguimiento_Ubicacion AS
14 SELECT
15     seguimiento_id,
16     equipo_id,
17     ubicacion_actual,
18     fecha_hora_actualizacion
19 FROM
20     seguimiento_ubicacion;
```

## 8. Tabla Comunicación

```
8. Tabla Comunicación.sql X
8. Tabla Comunicación.sql
1 -- Creación de la tabla Comunicación
2 CREATE TABLE comunicacion (
3     comunicacion_id SERIAL PRIMARY KEY, -- Identificador único de la comunicación (clave primaria)
4     cliente_id INT REFERENCES cliente(cliente_id), -- Clave foránea: Relación con la tabla Cliente
5     tipo_comunicacion VARCHAR(50) CHECK (tipo_comunicacion IN ('email', 'SMS', 'llamada')), -- Tipo de comunicación (email, SMS o llamada)
6     fecha_hora_comunicacion TIMESTAMP CHECK (fecha_hora_comunicacion <= CURRENT_TIMESTAMP), -- Fecha y hora de la comunicación (no puede ser futura)
7     mensaje TEXT -- Contenido del mensaje
8 );
9
10 -- Índice para optimizar búsquedas por cliente
11 CREATE INDEX idx_comunicacion_cliente ON comunicacion(cliente_id);
12
13 -- Vista simplificada de la tabla Comunicación
14 CREATE VIEW Vista_Comunicaciones AS
15 SELECT
16     comunicacion_id,
17     cliente_id,
18     tipo_comunicacion,
19     fecha_hora_comunicacion,
20     mensaje
21 FROM
22     comunicacion;
```

## 9. Tabla Informes y Estadísticas

```
9. Tabla Informes y Estadísticas.sql X
9. Tabla Informes y Estadísticas.sql
1  -- Creación de la tabla Informes y Estadísticas
2  CREATE TABLE informes_estadisticas (
3      informe_id SERIAL PRIMARY KEY, -- Identificador único del informe (clave primaria)
4      tipo_informe VARCHAR(50) NOT NULL, -- Tipo de informe (alquileres, ingresos, etc.)
5      fecha_generacion DATE NOT NULL CHECK (fecha_generacion <= CURRENT_DATE), -- Fecha de generación del informe (no puede ser futura)
6      datos_informe JSONB, -- Datos del informe en formato JSON
7      contrato_id INT REFERENCES contrato_alquiler(contrato_id) -- Clave foránea: Relación con la tabla Contrato de Alquiler
8  );
9
10 -- Índice para optimizar búsquedas por contrato
11 CREATE INDEX idx_informes_contrato ON informes_estadisticas(contrato_id);
12
13 -- Vista simplificada de la tabla Informes y Estadísticas
14 CREATE VIEW Vista_informes AS
15 SELECT
16     informe_id,
17     tipo_informe,
18     fecha_generacion,
19     contrato_id
20 FROM
21     informes_estadisticas;
```

## 10. Tabla Funcionalidades Avanzadas

```
10. Tabla Funcionalidades Avanzadas.sql X
10. Tabla Funcionalidades Avanzadas.sql
1  -- Creación de la tabla Funcionalidades Avanzadas
2  CREATE TABLE funcionalidades_avanzadas (
3      funcionalidad_id SERIAL PRIMARY KEY, -- Identificador único de la funcionalidad (clave primaria)
4      tipo_funcionalidad VARCHAR(50) NOT NULL CHECK (tipo_funcionalidad IN ('WMS', 'seguros', 'control de acceso', 'app móvil')), -- Tipo de funcionalidad (WMS, seguros, etc.)
5      descripcion TEXT, -- Descripción de la funcionalidad
6      configuracion JSONB, -- Configuración de la funcionalidad en formato JSON
7      equipo_id INT REFERENCES equipo(equipo_id) -- Clave foránea: Relación con la tabla Equipo
8  );
9
10 -- Índice para optimizar búsquedas por equipo
11 CREATE INDEX idx_funcionalidades_equipo ON funcionalidades_avanzadas(equipo_id);
12
13 -- Vista simplificada de la tabla Funcionalidades Avanzadas
14 CREATE VIEW Vista_Funcionalidades_Avanzadas AS
15 SELECT
16     funcionalidad_id,
17     tipo_funcionalidad,
18     descripcion,
19     equipo_id
20 FROM
21     funcionalidades_avanzadas;
```

# Justificación de Diseño

## Normalización Aplicada

El diseño de la base de datos sigue las normas de normalización para reducir la redundancia y asegurar la integridad de los datos. A continuación, se detalla cómo se aplicaron las formas normales:

- **Primera Forma Normal (1NF):** Todas las tablas están en 1NF, lo que significa que cada columna contiene valores atómicos. Esto implica que no hay listas ni valores múltiples en una sola celda. Por ejemplo, en la tabla **Cliente**, cada campo como nombre, cif\_nif, o direccion almacena un único valor, lo que facilita la gestión y consulta de los datos.
- **Segunda Forma Normal (2NF):** No existen dependencias parciales en las tablas. Todos los atributos no clave dependen completamente de la clave primaria de cada tabla. Por ejemplo, en la tabla **Equipo**, todos los atributos como tipo, marca, modelo, y estado dependen únicamente de la clave primaria equipo\_id, lo que garantiza que no haya redundancias innecesarias.
- **Tercera Forma Normal (3NF):** Las tablas no contienen dependencias transitivas. Esto significa que todos los atributos no clave dependen directamente de la clave primaria y no de otros atributos no clave. Por ejemplo, en la tabla **Contrato de Alquiler**, los campos como fecha\_inicio, fecha\_fin, y precio\_alquiler dependen directamente de contrato\_id, sin intermediarios.

## Elecciones de Diseño

A continuación, se justifican las decisiones de diseño tomadas para cada tabla, explicando el razonamiento detrás de cada elección y los beneficios que aportan al sistema.

### Tabla Cliente

#### 1. Clave primaria ( `cliente_id`):

- **Razonamiento:** Se eligió un campo SERIAL como clave primaria para garantizar que cada cliente tenga un identificador único y autoincremental. Este enfoque es ideal para sistemas donde la cantidad de clientes puede crecer significativamente, ya que elimina la necesidad de asignar manualmente identificadores. Además, facilita la integración con otras tablas mediante claves foráneas, asegurando la integridad referencial.
- **Beneficio:** La unicidad del identificador permite evitar duplicados y simplifica las operaciones CRUD (Crear, Leer, Actualizar, Eliminar). También optimiza las consultas que involucran relaciones con otras tablas, como contratos y comunicaciones.

#### 2. Campo nombre ( `VARCHAR(100)`):

- **Razonamiento:** El tipo VARCHAR(100) fue seleccionado para almacenar nombres de clientes, permitiendo hasta 100 caracteres. Esta longitud es suficiente para nombres largos, incluidos aquellos con apellidos compuestos o razones sociales extensas. Además, el uso de VARCHAR optimiza el espacio en comparación con CHAR, ya que solo ocupa el espacio necesario.
- **Beneficio:** Facilita la búsqueda y clasificación de clientes por nombre, lo que es crucial para la atención al cliente y la generación de informes. También permite personalizar las interacciones con los clientes, mejorando su experiencia.

#### 3. Campo cif\_nif ( `VARCHAR(20)`, único):

- **Razonamiento:** El campo cif\_nif se define como VARCHAR(20) y se marca como UNIQUE para garantizar que no haya duplicados en los identificadores fiscales de los clientes. Esto es especialmente importante en contextos legales y fiscales, donde cada cliente debe ser identificado de manera única.

- **Beneficio:** Asegura que cada cliente tenga un identificador fiscal único, lo que es esencial para la facturación y la gestión legal. También facilita la validación de datos durante la creación de nuevos registros.

## Tabla Equipo

### 1. Clave primaria ( equipo\_id):

- **Razonamiento:** La clave primaria equipo\_id utiliza un campo SERIAL para garantizar la unicidad de cada equipo registrado en el sistema. Este identificador es esencial para gestionar el inventario y mantener un historial detallado de cada equipo.
- **Beneficio:** Facilita la identificación y seguimiento de equipos, lo que es crucial para la planificación de mantenimientos, entregas y recogidas. También optimiza las consultas relacionadas con el estado y ubicación de los equipos.

### 2. Campo numero\_ser ie ( VARCHAR(50), único):

- **Razonamiento:** El campo numero\_ser ie se define como VARCHAR(50) y se marca como UNIQUE para garantizar que cada equipo tenga un identificador físico único. Esto es esencial para el seguimiento y la trazabilidad de los equipos.
- **Beneficio:** Permite identificar inequívocamente cada equipo, lo que es crucial para la gestión del inventario y el mantenimiento. También facilita la resolución de problemas relacionados con equipos específicos.

### 3. Campo estado ( VARCHAR(20 )):

- **Razonamiento:** El campo estado almacena el estado actual del equipo (disponible, alquilado, en mantenimiento). Se utiliza VARCHAR(20) para permitir flexibilidad en la definición de estados.
- **Beneficio:** Facilita la gestión del inventario y la disponibilidad de equipos, lo que es crucial para la planificación de alquileres y mantenimientos. También permite generar informes sobre el estado general del inventario.

## Tabla Contrato de Alquiler

### 1. Clave primaria ( contrato\_id):

- **Razonamiento:** La clave primaria contrato\_id utiliza un campo SERIAL para garantizar la unicidad de cada contrato registrado en el sistema. Este identificador es esencial para gestionar los contratos de alquiler de equipos.
- **Beneficio:** Facilita la identificación y gestión de los contratos, lo que es fundamental para la optimización de los procesos operativos. Además, permite generar informes detallados sobre los contratos activos, históricos o pendientes.

### 2. Campo fecha\_inicio ( DATE):

- **Razonamiento:** El campo fecha\_inicio almacena la fecha de inicio del alquiler. Se utiliza DATE para garantizar que solo se almacenen fechas válidas, evitando errores relacionados con formatos incorrectos o datos inconsistentes.
- **Beneficio:** Permite gestionar el tiempo de alquiler de manera precisa, lo que es crucial para cumplir con los plazos acordados con los clientes. También facilita la generación de informes sobre los periodos de alquiler.

### 3. Campo precio\_alquiler ( NUMERIC(10, 2)):

- **Razonamiento:** El campo precio\_alquiler almacena el costo total del alquiler. Se utiliza NUMERIC(10, 2) para garantizar precisión en operaciones monetarias, evitando errores de redondeo.
- **Beneficio:** Garantiza que los valores monetarios sean precisos y consistentes, lo que es crucial para la gestión financiera del negocio. También permite generar informes detallados sobre los ingresos totales.



## Tabla Mantenimiento

### 1. Clave primaria ( `mantenimiento_id`):

- **Razonamiento:** La clave primaria `mantenimiento_id` utiliza un campo `SERIAL` para garantizar la unicidad de cada registro de mantenimiento en el sistema. Este identificador es esencial para gestionar el historial de mantenimientos de los equipos y mantener un seguimiento de
- **Beneficio:** Facilita la identificación y gestión de los registros de mantenimiento, lo que es fundamental para la optimización del inventario y la programación de futuros mantenimientos. Además, permite generar informes detallados sobre los costos y frecuencias de mantenimiento.

### 2. Campo `tipo_mantenimiento` ( `VARCHAR(50)`):

- **Razonamiento:** El campo `tipo_mantenimiento` almacena el tipo de mantenimiento realizado (preventivo o correctivo). Se utiliza `VARCHAR(50)` para permitir flexibilidad en la definición de tipos de mantenimiento.
- **Beneficio:** Facilita la clasificación y gestión de los mantenimientos, lo que es crucial para

### 3. Campo `costo_mantenimiento` ( `NUMERIC(10, 2)`):

- **Razonamiento:** El campo `costo_mantenimiento` almacena el costo asociado a cada mantenimiento. Se utiliza `NUMERIC(10, 2)` para garantizar precisión en operaciones monetarias, evitando errores de redondeo.
- **Beneficio:** Garantiza que los costos de mantenimiento sean precisos y consistentes, lo que es crucial para la gestión financiera del negocio. También permite generar informes detallados sobre los costos totales de mantenimiento.

## Tabla Logística de Entrega y Recogida

### 1. Clave primaria ( `logística_id`):

- **Razonamiento:** La clave primaria `logística_id` utiliza un campo `SERIAL` para garantizar la unicidad de cada registro de logística en el sistema. Este identificador es esencial para gestionar la entrega y recogida de equipos, asegurando que cada operación logística esté correctamente documentada y rastreable.
- **Beneficio:** Facilita la identificación y gestión de las operaciones logísticas, lo que es crucial para la coordinación eficiente de entregas y recogidas. Además, permite generar informes detallados sobre la logística, lo que ayuda a optimizar los tiempos y costos asociados a estas operaciones.

### 2. Campo `fecha_hora_entrega` ( `TIMESTAMP`):

- **Razonamiento:** El campo `fecha_hora_entrega` almacena la fecha y hora exacta de la entrega del equipo. Se utiliza `TIMESTAMP` para garantizar precisión en el tiempo,
- **Beneficio:** Permite gestionar el tiempo de entrega de manera precisa, lo que es crucial para cumplir con los plazos acordados con los clientes. También facilita la generación de informes sobre los tiempos
- **Consideraciones adicionales:** Para mejorar la usabilidad, se podría implementar una validación adicional para asegurar que las fechas y horas ingresadas sean realistas y coherentes con el contrato de alquiler. Además, dado que este campo es crítico para la planificación logística, se podría crear un índice en `fecha_hora_entrega` para optimizar las consultas relacionadas con este atributo, mejorando el rendimiento del sistema.

### 3. Relación con Contrato de Alquiler ( `contrato_id`):

- **Razonamiento:** La relación de 1 a 1 entre **Logística de Entrega y Recogida** y **Contrato de Alquiler** refleja la naturaleza del negocio, donde cada operación logística está asociada a un contrato específico. Esta relación se establece mediante una clave foránea en la tabla **Logística de Entrega y Recogida** que referencia a `contrato_id`.
- **Beneficio:** Permite gestionar la logística de entrega y recogida de manera eficiente, manteniendo la integridad referencial. También facilita la

generación de informes sobre las operaciones logísticas asociadas a un contrato específico.

## Tabla Facturación

### 1. Clave primaria ( **factura\_id**):

- **Razonamiento:** La clave primaria **factura\_id** utiliza un campo SERIAL para garantizar la unicidad de cada factura registrada en el sistema. Este identificador es esencial para gestionar las facturas generadas por los contratos de alquiler, asegurando que cada transacción esté correctamente documentada y rastreable.
- **Beneficio:** Facilita la identificación y gestión de las facturas, lo que es crucial para la gestión financiera del negocio. También permite generar informes detallados sobre los ingresos generados por los contratos de alquiler.

### 2. Campo importe ( **NUMERIC(10, 2)**):

- **Razonamiento:** El campo **importe** almacena el monto total de la factura. Se utiliza `NUMERIC(10, 2)` para garantizar precisión en operaciones monetarias, evitando errores de redondeo.
- **Beneficio:** Garantiza que los valores monetarios sean precisos y consistentes, lo que es crucial para la gestión financiera del negocio. También permite generar informes detallados sobre los ingresos total

### 3. Relación con Contrato de Alquiler ( **contrato\_id**):

- **Razonamiento:** La relación de Muchos a 1 entre **Facturación** y **Contrato de Alquiler** refleja la naturaleza del negocio, donde una factura puede estar asociada a un contrato específico. Esta relación se establece mediante una clave foránea en la tabla **Facturación** que referencia a **contrato\_id**.
- **Beneficio:** Permite gestionar múltiples facturas asociadas a un mismo contrato, manteniendo la integridad referencial. También facilita la generación de informes sobre los ingresos generados por un contrato específico.

## Tabla Seguimiento de Ubicación

### 1. Clave primaria ( **seguimiento\_id**):

- **Razonamiento:** La clave primaria `seguimiento_id` utiliza un campo `SERIAL` para garantizar la unicidad de cada registro de seguimiento en el sistema. Este identificador es esencial para gestionar el historial de ubicaciones de los equipos y mantener un rastro detallado de sus movimientos
- **Beneficio:** Facilita la identificación y gestión de los registros de seguimiento, lo que es fundamental para la optimización de la logística y la toma de decisiones estratégicas. Además, permite generar informes detallados sobre los movimientos de los equipos, lo

## 2. Campo `ubicacion_actual ( VARCHAR(100))`:

- **Razonamiento:** El campo `ubicacion_actual` almacena la ubicación física del equipo (por ejemplo, "Almacén Central", "Obra Calle Mayor"). Se utiliza `VARCHAR(100)` para permitir descripciones detalladas de las ubicaciones.
- **Beneficio:** Permite rastrear la ubicación exacta del equipo, lo que es crucial para la logística y la seguridad. También facilita la generación de informes sobre los movimientos de los equipos, lo que ayuda a optimizar la planificación de entregas y recog

## 3. Campo `fecha_hora_actualizacion ( TIMESTAMP)`:

- **Razonamiento:** El campo `fecha_hora_actualizacion` almacena la fecha y hora exacta de la última actualización de `TIMESTAMP` para garantizar precisión en el tiempo, lo que es esencial para el seguimiento cronológico de los movimientos de los equipos.
- **Beneficio:** Proporciona un historial cronológico preciso del movimiento de los equipos, lo que es crucial para la planificación logística y la seguridad. También facilita la generación de informes sobre los tiempos de entrega y recogida, lo que ayuda a optimizar los procesos operativos.

## Tabla Comunicación

### 1. Clave primaria ( `comunicacion_id`):

- **Razonamiento:** La clave primaria `comunicacion_id` utiliza un campo `SERIAL` para garantizar la unicidad de cada comunicación registrada en el sistema. Este identificador es esencial para gestionar las interacciones con los clientes y mantener un historial detallado de todas las comunicaciones realizadas.
- **Beneficio:** Facilita la identificación y gestión de las comunicaciones, lo que es crucial para la atención al cliente y la auditoría de interacciones. Además, permite generar informes detallados sobre las comunicaciones realizadas, lo que ayuda a identificar patrones y tendencias en la atención al cliente.

## 2. Campo `tipo_comunicacion` ( `VARCHAR(50)`):

- **Razonamiento:** El campo `tipo_comunicacion` almacena el tipo de comunicación realizada (email, SMS, llamada). Se utiliza `VARCHAR(50)` para permitir flexibilidad en la clasificación de los tipos de comunicación.
- **Beneficio:** Facilita la clasificación y gestión de las comunicaciones, lo que es crucial para la atención al cliente y la generación de informes. También permite analizar el rendimiento de diferentes canales de comunicación, lo que ayuda a optimizar la estrategia.

## 3. Campo `mensaje` ( `TEXT`):

- **Razonamiento:** El campo `mensaje` almacena el contenido completo del mensaje enviado al cliente. Se utiliza `TEXT` para garantizar que no haya truncamiento de información crítica en las comunicaciones.
- **Beneficio:** Garantiza que toda la información relevante esté disponible para futuras referencias, lo que es crucial para la atención al cliente y la resolución de disputas. También facilita la generación de informes sobre el contenido.

## Tabla Informes y Estadísticas

### 1. Clave primaria ( `informe_id`):

- **Razonamiento:** La clave primaria `informe_id` utiliza un campo SERIAL para garantizar la unicidad de cada informe generado en el sistema. Este identificador es esencial para gestionar los informes y estadísticas generados por el sistema y
- **Beneficio:** Facilita la identificación y gestión de los informes, lo que es crucial para la toma de decisiones estratégicas. Además, permite generar informes detallados sobre los resultados de los análisis, lo que ayuda a identificar patrones y tendencias en el negocio.
- **Consideraciones adicionales:** Dado que los informes son una parte crítica del negocio, la clave primaria debe ser robusta y escalable. El uso de SERIAL asegura que el sistema pueda manejar un gran volumen de registros sin problemas. Además, se podría implementar un índice `informe_id`, mejorando aún más el rendimiento del sistema.

## 2. Campo `datos_informe` ( JSONB):

- **Razonamiento:** El campo `datos_informe` almacena los datos del informe en formato JSONB, lo que permite flexibilidad en la estructura de los datos. Este campo es esencial para adaptarse a diferentes tipos de informes sin necesidad de modificar la estructura de la tabla.
- **Beneficio:** Permite generar informes detallados y personalizados según las necesidades del negocio, lo que es crucial para la toma de decisiones estratégicas. También facilita la integración con sistemas externos, lo que ayuda a optimizar los procesos operativos.

## 3. Relación con Contrato de Alquiler ( `contrato_id`):

- **Razonamiento:** La relación de Muchos a 1 entre **Informes y Estadísticas** y **Contrato de Alquiler** refleja la naturaleza del negocio, donde muchos informes están asociados a contratos específicos. Esta relación se establece mediante una clave foránea en la tabla **Informes y Estadísticas** que referencia a `contrato_id`.
- **Beneficio:** Permite generar informes detallados basados en contratos específicos, lo que es crucial para la toma de decisiones estratégicas. También facilita la generación de informes segmentados por contrato, lo que ayuda a identificar áreas de mejora en la gestión de alquileres.

## Tabla Funcionalidades Avanzadas

### 1. Clave primaria ( funcionalidad\_id):

- **Razonamiento:** La clave primaria funcionalidad\_id utiliza un campo SERIAL para garantizar la unicidad de cada funcionalidad registrada en el sistema. Este identificador es esencial para gestionar las funcionalidades avanzadas del sistema y mantener un historial detallado de cada configuración realizada.
- **Beneficio:** Facilita la identificación y gestión de las funcionalidades, lo que es crucial para la integración con sistemas externos y el control de acceso a los equipos. Además, permite generar informes detallados sobre las configuraciones realizadas, lo que ayuda a optimizar los procesos operativos.

### 2. Campo configuracion ( JSONB):

- **Razonamiento:** El campo configuracion almacena las configuraciones específicas de cada funcionalidad en formato JSONB, lo que permite flexibilidad en la estructura de los datos. Este campo es esencial para adaptarse a diferentes configuraciones de funcionalidades avanzadas.
- **Beneficio:** Permite gestionar funcionalidades avanzadas como la integración con sistemas de gestión de almacenes (WMS), seguros y control de acceso. También facilita la integración con sistemas externos, lo que ayuda a optimizar los procesos operativos.

### 3. Relación con Equipo (equipo\_id):

- **Razonamiento:** La relación de Muchos a 1 entre **Funcionalidades Avanzadas** y **Equipo** refleja la naturaleza del negocio, donde muchas funcionalidades están asociadas a equipos específicos. Esta relación se establece mediante una clave foránea en la tabla **Funcionalidades Avanzadas** que referencia a equipo\_id.
- **Beneficio:** Permite gestionar funcionalidades avanzadas asociadas a equipos específicos, lo que es crucial para la integración con sistemas externos y el control de acceso a los equipos. También facilita la generación

de informes detallados sobre las configuraciones realizadas, lo que ayuda a optimizar los procesos operativos.



## **Conclusión**

El diseño de la base de datos para "Construcción Rentable" sigue las mejores prácticas de normalización y diseño relacional. Se han identificado las entidades clave, sus atributos y las relaciones entre ellas, lo que permite un sistema eficiente y escalable. La implementación en PostgreSQL garantiza la integridad de los datos y facilita la gestión de la información de la empresa.