

<b>TITULO</b>	<b>Desarrollo Y Aplicación De Una Base De Datos</b>
<b>REF.</b>	<b>UTN-FRM-BD-Trabajo Final-2023</b>

	<b>NOMBRE</b>	<b>FIRMA</b>	<b>FECHA</b>
<b>ESCRITO POR</b>	Yamil Tahan		
	Vincenzo Dallape		
	Isidro Podesta		
	Valentino Turkenicz		
<b>VERIFICADO POR</b>			

---

La Universidad Tecnológica Nacional – Facultad Regional Mendoza tiene los derechos sobre este documento, el cual es confidencial y no será usado para ningún otro propósito, salvo para el que fue suministrado y no será reproducido, copiado o transmitido en todo o en parte sin el permiso de su dueño.

Cualquier otra persona diferente de la persona autorizada, que consiga el documento por haberlo encontrado o por otra causa, deberá enviarlo junto con su nombre y dirección en sobre cerrado a:

Cátedra: Bases de Datos

Dpto de Sistemas

Universidad Tecnológica Nacional – Facultad Regional Mendoza

Rodríguez 273 (5500)

Mendoza ARGENTINA

---

### REGISTRO DE REVISIONES

Revisión de Diseño					
Fecha	29/04/2024				
Autores	Yamil Tahan	Vincenzo Dallape	Isidro Podesta	Valentino Turkenicz	
Documentación Presentada					
Revisión	RFA 001:  RFA 002:				
Profesores	Nombre		Nombre		
	Firma		Firma		

---

**asdiNDICE**

INTRODUCCIÓN.....	5
OBJETO.....	5
ESTADO DEL ARTE.....	5
ABREVIATURAS .....	7
DEFINICIONES .....	7
DESARROLLO.....	8
DESCRIPCIÓN .....	8
RESUMEN DE LA EMPRESA.....	8
RELEVAMIENTO DE DATOS.....	8
DISEÑO CONCEPTUAL - MODELO ENTIDAD/RELACION.....	8
ENTIDADES RELEVADAS.....	8
RELACIONES RELEVADAS.....	12
DIAGRAMA E/R.....	13
DISEÑO LÓGICO CONCEPTUAL – MODELO RELACIONAL.....	14
TABLAS, CLAVES Y RESTRICCIONES.....	14
DIAGRAMA RELACIONAL.....	18
APLICAR TEORÍA DE LA NORMALIZACIÓN.....	19
CONSULTAS EN ALGEBRA RELACIONAL.....	20
DISEÑO LÓGICO ESPECÍFICO - IMPLEMENTACIÓN EN SGBD COMERCIAL	
DESCRIPTIVO GENERAL DE LA IMPLEMENTACIÓN.....	
FORMULARIOS DE ENTRADA.....	
REPORTES DE SALIDA.....	
CONSULTAS EN SQL .....	
PLANIFICACIÓN Y PRESUPUESTO.....	22
PLANIFICACIÓN.....	22
PRESUPUESTO.....	24
CONCLUSIONES.....	
BIBLIOGRAFÍA.....	
ANEXOS.....	

## Desarrollo Y Aplicación De Una Base De Datos

### 1) INTRODUCCIÓN

#### a) OBJETO

El objeto del presente trabajo es evidenciar el aprendizaje obtenido a lo largo del cursado de la materia “Base de datos”. En el mismo se detallará cada estancia del desarrollo y aplicación de una base de datos para una organización, propuesta de manera grupal, la cual consiste en una empresa de distribución de mercaderías varias.

#### b) ESTADO DEL ARTE

##### i) DEFICIENCIAS SISTEMAS ACTUALES

Se llevó a cabo una investigación sobre los distintos SGBD actuales, con tal de dirimir el más adecuado para la aplicación en la problemática elegida. A continuación, se detallarán algunas ventajas y desventajas de algunos de ellos.

SGBD	Características	Deficiencias
MySQL	<ul style="list-style-type: none"> <li>Facilidad de uso: MySQL es fácil de usar y administrar.</li> <li>Rendimiento: Ofrece un buen rendimiento para consultas y operaciones de bases de datos.</li> <li>Escalabilidad: Puede manejar bases de datos de diferentes tamaños, permitiendo el crecimiento.</li> <li>Compatibilidad: Compatible con la mayoría de lenguajes de programación y sistemas operativos.</li> <li>Costo: Es de código abierto, por lo que no hay costos de licencia.</li> </ul>	<ul style="list-style-type: none"> <li>Funciones avanzadas limitadas: Puede tener limitaciones en funciones avanzadas en comparación con otros SGBD.</li> <li>Integridad referencial: Puede tener ciertas limitaciones en la implementación de reglas de integridad referencial</li> </ul>
PostgreSQL	<ul style="list-style-type: none"> <li>Potente: Ofrece funciones avanzadas y gran flexibilidad.</li> <li>Escalabilidad: Adecuado para bases de datos grandes.</li> <li>Soporte para funciones complejas: Incluye soporte para procedimientos almacenados y tipos de datos complejos.</li> </ul>	<ul style="list-style-type: none"> <li>Curva de aprendizaje: Puede ser más complejo de aprender y administrar que MySQL.</li> <li>Rendimiento: Puede ser más lento que MySQL en ciertas operaciones simples.</li> </ul>
Microsoft SQL Server	<ul style="list-style-type: none"> <li>Integración con productos de Microsoft: Excelente integración con el ecosistema de Microsoft.</li> <li>Herramientas de análisis avanzadas: Incluye herramientas de análisis de datos y business intelligence.</li> </ul>	<ul style="list-style-type: none"> <li>Costo: Puede resultar costoso, especialmente para licencias empresariales.</li> <li>Compatibilidad: Menor compatibilidad con otros sistemas</li> </ul>

	<ul style="list-style-type: none"> <li>○ Funciones de seguridad avanzadas: Ofrece características avanzadas de seguridad.</li> </ul>	operativos fuera de Windows.
Oracle Database	<ul style="list-style-type: none"> <li>○ Funciones avanzadas: Incluye características empresariales y soporte para bases de datos distribuidas.</li> <li>○ Rendimiento: Ofrece un rendimiento robusto para consultas y operaciones.</li> <li>○ Seguridad: Ofrece características avanzadas de seguridad y protección de datos.</li> </ul>	<ul style="list-style-type: none"> <li>○ Costo: Altos costos de licencia y mantenimiento.</li> <li>○ Complejidad: Puede ser complejo de administrar y configurar.</li> </ul>

## ii) ANÁLISIS Y VENTAJAS DEL PROYECTO

En el proyecto se diseñará e implementará una base de datos utilizando el motor PostgreSQL, por los siguientes motivos:

1. **Facilidad de uso y administración:** PostgreSQL es reconocido por su relativa facilidad de uso y administración, lo que lo convierte en una opción atractiva para organizaciones de todos los tamaños. Su interfaz intuitiva y sus herramientas de administración simplifican las tareas relacionadas con la gestión de la base de datos, lo que facilita su manejo para todos los departamentos de la organización.
2. **Escalabilidad y manejo de diferentes tamaños de bases de datos:** PostgreSQL es capaz de manejar bases de datos de diferentes tamaños, desde pequeñas hasta grandes. Esta capacidad es especialmente útil para empresas que experimentan un crecimiento en su cantidad de clientes y volumen de datos con el tiempo. La escalabilidad de PostgreSQL permite a la empresa adaptarse y crecer sin problemas a medida que aumentan sus necesidades de almacenamiento y procesamiento de datos.
3. **Compatibilidad con múltiples lenguajes de programación y sistemas operativos:** PostgreSQL es compatible con la mayoría de los lenguajes de programación y sistemas operativos, lo que garantiza una integración suave con el entorno de desarrollo existente de la empresa. Esta compatibilidad facilita la adopción y el uso de PostgreSQL por parte de los equipos de desarrollo, lo que permite a la empresa aprovechar al máximo sus recursos tecnológicos.
4. **Licencia de código abierto sin costos asociados:** Una de las ventajas clave de PostgreSQL es su modelo de licencia de código abierto, que significa que no hay costos de licencia asociados. Esto puede ser beneficioso para empresas que buscan reducir gastos en software y destinar recursos a otras áreas importantes de su operación. La ausencia de costos de licencia hace que PostgreSQL sea una opción atractiva desde el punto de vista económico para muchas organizaciones.

5. **Gestión eficiente de diversos tipos de datos con integridad referencial y seguridad:** PostgreSQL es capaz de manejar una variedad de datos, incluidos aquellos relacionados con envíos, clientes, empleados y finanzas, mientras mantiene la integridad referencial y la seguridad de los datos. Esto asegura que la empresa pueda administrar sus datos de manera eficiente y confiable, sin comprometer la precisión o la seguridad de la información.
6. **Adecuación a las necesidades sin requerir conocimientos extremadamente avanzados:** Aunque PostgreSQL ofrece una amplia gama de características y capacidades avanzadas, las posibles desventajas asociadas con la complejidad de ciertas funciones avanzadas o referencias no suelen ser un problema en la implementación práctica. Esto significa que la mayoría de las organizaciones pueden aprovechar las capacidades de PostgreSQL sin necesidad de tener un conocimiento extremadamente avanzado en bases de datos, lo que lo convierte en una opción accesible y adecuada para una amplia variedad de casos de uso.

#### c) ABREVIATURAS

SGDB – Sistema de Gestión de Bases de Datos

#### d) DEFINICIONES

---

## 2) DESARROLLO

### a) DESCRIPCIÓN

#### i) RESUMEN DE LA ORGANIZACIÓN

DeliverMaxxing es una organización dedicada a la distribución de mercaderías varias, desde proveedores hacia sus clientes. Para llevar a cabo esta tarea, cuenta con una flota de 20 camiones con sus respectivos choferes, los cuales cobran por cada viaje realizado. Los empleados se dividen en tres áreas:

- Atención al cliente: Se encarga de tomar los pedidos de los proveedores.
- Logística: Recibe los pedidos y planifica los viajes.
- Finanzas: Se encarga de los cobros, pago de salarios, y emisión de informes de gastos.

#### ii) RELEVAMIENTO DE DATOS

De acuerdo al relevamiento realizado, los datos a tener en cuenta son:

- Proveedores: Código de identificación, razón social, dirección, CUIT, teléfono.
- Camiones: Patente, marca, modelo, antigüedad, consumo promedio y conductor asignado.
- Los empleados: DNI, nombre, edad, dirección, teléfono, área.
- Choferes: Son una clase especial de empleado, no pertenecen a ninguna de las tres áreas. Además de los otros atributos de empleados, poseen código de chofer indicador de disponibilidad.
- Pedidos: Código, detalle, fecha de pedido, fecha de entrega, condición de pago, dirección de entrega.
- Viajes: Código, dirección de retiro (dirección del proveedor), pedidos, fecha de inicio, fecha de fin, chofer.
- Área: Descripción (Atención al cliente, logística o finanzas), teléfono, empleados.

### b) DISEÑO CONCEPTUAL - MODELO ENTIDAD/RELACION

#### i) ENTIDADES RELEVADAS

Las entidades relevadas son las mismas que fueron listadas en el inciso 2.a.ii. A continuación, se presenta el código para la creación de cada una de las tablas representativas de estas entidades con sus respectivos atributos.



---

Con el siguiente código se crea la tabla Area

```
CREATE TABLE Areas (  
    AreaID int PRIMARY KEY,  
    Descripcion VARCHAR(100),  
    Telefono int,  
);
```

Esta tabla tiene **AreaID** como clave primaria.

Con el siguiente código se crea la tabla Empleados

```
CREATE TABLE Empleados (  
    DNI VARCHAR(20) PRIMARY KEY,  
    Nombre VARCHAR(100),  
    Edad INT,  
    Direccion VARCHAR(200),  
    Telefono INT,  
    AreaID INT,  
    FOREIGN KEY (AreaID) REFERENCES Areas(AreaID)  
);
```

Esta tabla tiene como clave primaria **DNI** y como clave foránea **AreaID** de áreas.

---

Con el siguiente código se crea la tabla Choferes

```
CREATE TABLE choferes (  
  
    codchofer INT PRIMARY KEY,  
  
    dni INT,  
  
    disponibilidad BOOLEAN,  
  
    FOREIGN KEY (dni) REFERENCES empleados(dni),  
  
);
```

Esta tabla tiene como clave primaria **codchofer** y como clave foránea **DNI** de empleados.

Con el siguiente código se crea la tabla Camiones

```
CREATE TABLE Camiones (  
  
    Patente VARCHAR(20) PRIMARY KEY,  
  
    Marca VARCHAR(50),  
  
    Modelo VARCHAR(50),  
  
    Antigüedad INT,  
  
    ConsumoPromedio DECIMAL(10, 2),  
  
    ConductorAsignado int,  
  
    FOREIGN KEY (ConductorAsignado) REFERENCES Choferes(codchofer)  
  
);
```

Esta tabla tiene como clave primaria **Patente** y como clave foránea **ConductorAsignado** de codchofer.

---

Con el siguiente código se crea la tabla Viajes

```
CREATE TABLE Viajes (  
    ViajeID INT PRIMARY KEY,  
    DireccionRetiro VARCHAR(200),  
    FechaInicio DATE,  
    FechaFin DATE,  
    Chofer INT,  
    FOREIGN KEY (Chofer) REFERENCES Choferes(codchofer)  
);
```

Esta tabla tiene como clave primaria **ViajeID** y como clave foránea **Chofer** de codchofer.

Con el siguiente código se crea la tabla Pedidos

```
CREATE TABLE Pedidos (  
    PedidoID int PRIMARY KEY,  
    Detalle TEXT,  
    FechaPedido DATE,  
    FechaEntrega DATE,  
    CondicionPago VARCHAR(50),  
    DireccionEntrega VARCHAR(200),  
    ProveedorID INT,  
    ViajeID INT,  
    FOREIGN KEY (ProveedorID) REFERENCES Proveedores(ProveedorID),  
    FOREIGN KEY (ViajeID) REFERENCES Viajes(ViajeID)  
);
```

Esta tabla tiene como clave primaria **PedidoID** y como claves foráneas **ProveedorID** de proveedores y **ViajeID** de viajes.

---

Con el siguiente código se crea la tabla de proveedores

```
CREATE TABLE Proveedores (  
  
    ProveedorID INT PRIMARY KEY,  
  
    RazonSocial VARCHAR(100),  
  
    Direccion VARCHAR(200),  
  
    CUIT BIGINT,  
  
    Telefono BIGINT,  
  
);
```

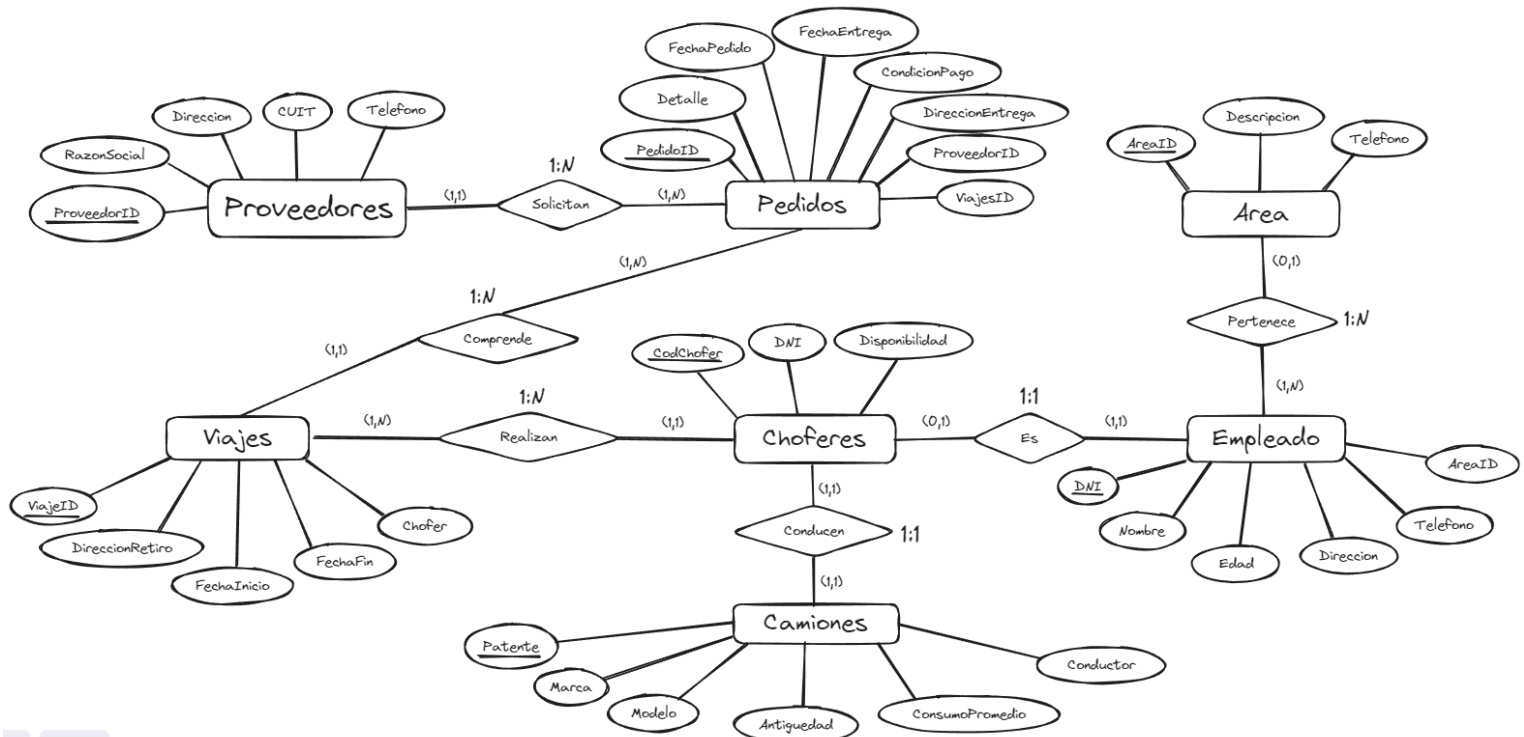
Esta tabla tiene como clave primaria **ProveedorID** y no tiene clave foránea

## ii) RELACIONES RELEVADAS

Las relaciones encontradas correspondientes para la organización son descritas en la siguiente lista:

- Empleado-Area (1-N): Cada empleado pertenece a un área. A un área le pueden pertenecer varios empleados.
- Choferes-Empleados (1-1): Los choferes son un tipo de empleado especial, a los cuales no se les asigna un área, y se le agregan atributos especiales. Estos determinan su código de identificación dentro de la empresa y su disponibilidad para realizar un viaje. Un chofer es un empleado cuando se cumplen estos requisitos. Un empleado puede no ser chofer.
- Camiones-Choferes (1-1): A cada camión le es asignado un chofer. Los choferes solo pueden conducir el camión al cual fueron asignados.
- Choferes-Viajes (1-N): Un chofer puede realizar varios viajes. Un viaje es realizado solo por un chofer.
- Viajes-Pedidos (1-N): A un viaje lo comprenden varios pedidos. Un pedido es comprendido en un viaje.
- Proveedores-Pedidos (1-N): Los proveedores pueden solicitar varios pedidos. Un pedido es solicitado únicamente por un proveedor.

iii) DIAGRAMA E/R



### c) DISEÑO LÓGICO CONCEPTUAL – MODELO RELACIONAL

#### i) TABLAS, CLAVES Y RESTRICCIONES

Entidades a Tablas: Cada entidad en el MER se convierte en una tabla en el MR. Los atributos de la entidad se convierten en columnas de la tabla correspondiente:

Area		
Tipo de Clave	Nombre de Columna	Tipo de Dato
PK	AreaID	Int
	Descripcion	String
	Telefono	Int

Empleado		
Tipo de Clave	Nombre de Columna	Tipo de Dato
PK	DNI	String
	Nombre	String
	Edad	Int
	Direccion	String
	Telefono	Int
FK	AreaID	Int

Choferes		
Tipo de Clave	Nombre de Columna	Tipo de Dato
PK	CodChofer	Int
	DNI	Int
	Disponibilidad	Boolean
FK	DNI	Int

Camiones		
Tipo de Clave	Nombre de Columna	Tipo de Dato
PK	Patente	String
	Marca	String
	Modelo	String
	Antigüedad	Int
	ConsumoPromedio	Double
FK	ConductorAsignado	Int

Viajes		
Tipo de Clave	Nombre de Columna	Tipo de Dato
PK	ViajeID	Int
	DireccionRetiro	String
	FechaInicio	Date
	FechaFin	Date
FK	Chofer	Int

Pedidos		
Tipo de Clave	Nombre de Columna	Tipo de Dato
PK	PedidosID	Int
	Detalles	String
	FechaPedido	Date
	FechaEntrega	Date
	CondicionPago	String
	DireccionEntregas	String
FK	ProveedorID	Int



FK	ViajeID	Int
----	---------	-----

Proveedores		
Tipo de Clave	Nombre de Columna	Tipo de Dato
PK	ProveedorID	Int
	RazonSocial	String
	Direccion	String
	CUIT	Int
	Telefono	Int

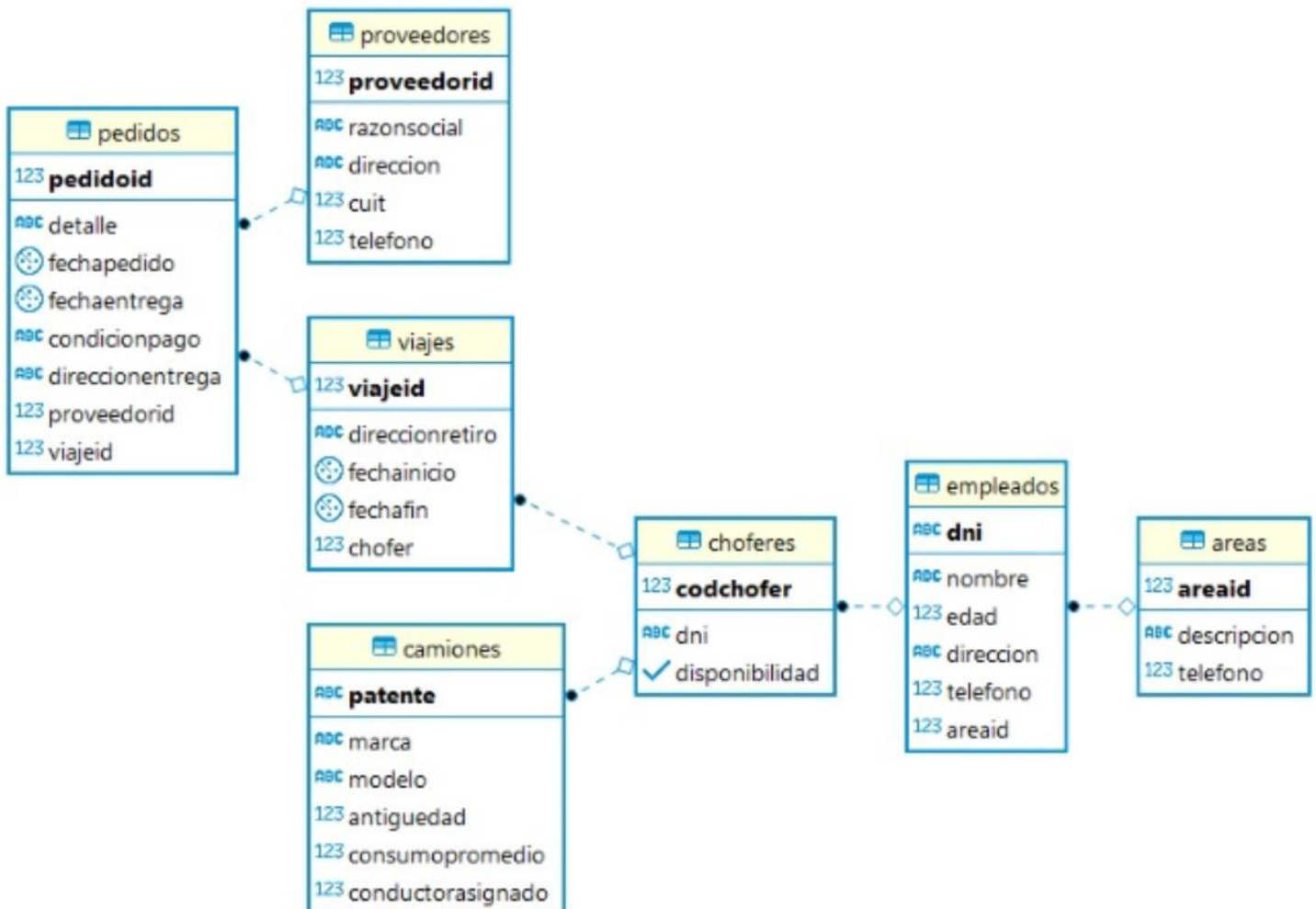
Relaciones (1:N):

- Proveedores - Solicitan - Pedidos: Añadir una columna ProveedorID en la tabla Pedidos como llave foránea.
- Pedidos - Comprende - Viajes: Añadir una columna ViajeID en la tabla Pedidos como llave foránea.
- Area - Pertenece - Empleado: Añadir una columna AreaID en la tabla Empleado como llave foránea.
- Viajes - Realizan - Choferes: Añadir una columna ChoferID en la tabla Viajes como llave foránea.

Relaciones (1:1):

- Empleado - Es - Choferes: Se añade DNI de empleado como llave foránea en la tabla Choferes.
- Choferes - Conducen - Camiones: Añadir una columna codchofer en la tabla Camiones como llave foránea.

ii) DIAGRAMA RELACIONAL



---

### iii) APLICAR TEORÍA DE LA NORMALIZACIÓN

- **Primera Forma Normal (1NF)**

La Primera Forma Normal establece que todos los atributos deben contener solo valores atómicos, y cada campo debe contener un único valor. En el código proporcionado, todas las tablas cumplen con esta regla ya que:

Cada campo contiene solo un valor.

No hay listas o conjuntos de valores en un solo campo.

Ejemplo: En la tabla Empleados, el campo Nombre contiene un solo nombre para cada empleado.

- **Segunda Forma Normal (2NF)**

La Segunda Forma Normal requiere que la tabla esté en 1NF y que todos los atributos no clave sean dependientes completamente de la clave primaria. Esto implica eliminar dependencias parciales.

En las tablas proporcionadas, cada tabla tiene una clave primaria y todos los atributos no clave son completamente dependientes de la clave primaria:

En la tabla Empleados, todos los atributos (Nombre, Edad, Direccion, Telefono, AreaID) dependen completamente de DNI, que es la clave primaria.

En la tabla Pedidos, todos los atributos (Detalle, FechaPedido, FechaEntrega, CondicionPago, DireccionEntrega, ProveedorID, ViajeID) dependen completamente de PedidoID.

- **Tercera Forma Normal (3NF)**

La Tercera Forma Normal requiere que la tabla esté en 2NF y que no haya dependencias transitivas. Esto significa que los atributos no clave no deben depender de otros atributos no clave.

En las tablas proporcionadas:

En la tabla Empleados, todos los atributos dependen directamente de DNI.

En la tabla Pedidos, todos los atributos dependen directamente de PedidoID y no de otros atributos no clave como ProveedorID o ViajeID.

- **Forma Normal de Boyce-Codd (BCNF)**

La Forma Normal de Boyce-Codd es una versión más estricta de la 3NF. Establece que para cada dependencia funcional  $A \rightarrow B$ , A debe ser una superclave.

En las tablas proporcionadas, todas las dependencias funcionales cumplen con esta regla porque:

Las claves primarias son superclaves, y todos los atributos no clave dependen completamente de ellas.

Por lo que sigue las cuatro primeras reglas de normalización:

1NF: Todos los campos contienen valores atómicos.

2NF: Todos los atributos no clave dependen completamente de la clave primaria.

3NF: No hay dependencias transitivas entre atributos no clave.

BCNF: Cada dependencia funcional tiene una superclave en el lado izquierdo.

#### iv) CONSULTAS EN ÁLGEBRA RELACIONAL

Algunos ejemplos de consultas en álgebra relacional resultan:

Listar todos los empleados con su nombre, edad, dirección, teléfono y área a la que pertenecen:

$\pi_{\text{Nombre, Edad, Direccion, Telefono, Descripcion}}(\sigma_{\text{E.AreaID} = \text{A.AreaID}}(\text{Empleados E} \bowtie \text{Areas A}))$

Listar todos los choferes con su código de chofer, DNI, disponibilidad y nombre:

$\pi_{\text{codchofer, C.DNI, disponibilidad, Nombre}}(\sigma_{\text{C.DNI} = \text{E.DNI}}(\text{Choferes C} \bowtie \text{Empleados E}))$

Listar todos los camiones con su patente, marca, modelo, antigüedad, consumo promedio y nombre del conductor asignado (si lo tiene):

$\pi_{\text{Patente, Marca, Modelo, Antigüedad, ConsumoPromedio, Nombre}}(\sigma_{\text{C.ConductorAsignado} = \text{CH.codchofer} \wedge \text{CH.DNI} = \text{E.DNI}}(\text{Camiones C} \bowtie \text{Choferes CH} \bowtie \text{Empleados E}))$

Listar todos los viajes con su ID, dirección de retiro, fecha de inicio, fecha de fin y nombre del chofer:

$\pi_{\text{ViajeID, DireccionRetiro, FechaInicio, FechaFin, E.Nombre}}(\sigma_{\text{V.Chofer} = \text{CH.codchofer} \wedge \text{CH.DNI} = \text{E.DNI}}(\text{Viajes V} \bowtie \text{Choferes CH} \bowtie \text{Empleados E}))$

Listar todos los pedidos con su ID, detalle, fecha de pedido, fecha de entrega, condición de pago, dirección de entrega y nombre del proveedor:

---

$\pi_{\text{PedidoID}, \text{Detalle}, \text{FechaPedido}, \text{FechaEntrega}, \text{CondicionPago}, \text{DireccionEntrega}, \text{RazonSocial}}(\sigma_{\text{P.ProveedorID} = \text{PR.ProveedorID}} (\text{Pedidos P} \bowtie \text{Proveedores PR}))$

---

### 3) PLANIFICACIÓN Y PRESUPUESTO

En este apartado se explican las posibles etapas del proyecto, los tiempos previstos para cada una de ellas y el presupuesto resultante. Conociendo que la mayoría de proyectos que no cumplen con las previsiones suele ser por desviaciones en presupuesto y en tiempo, antes de empezar el proyecto, se planificó y se presupuestó.

Es evidente que una planificación previa no asegura que se vaya a cumplir todo lo que está previsto, pero tener un plan de acción por escrito que permita disminuir el riesgo de tener desviaciones entre lo que se ha previsto y lo que finalmente se ha obtenido. Es por este motivo que se decidió planificar todas las etapas del proyecto y usar el presupuesto como una herramienta para gestionar la planificación.

#### a- PLANIFICACIÓN:

Para realizar la planificación y llevar el seguimiento de la evolución del proyecto se utilizó un software de seguimiento de proyectos, como Microsoft Project u otro.

Desde un principio el proyecto se dividió en cuatro etapas:

- Análisis de los requerimientos del gestor de BBDD para diferenciarse de los sistemas existentes: En esta parte del proyecto se estudiaron los sistemas existentes y se evaluaron sus fortalezas y debilidades. Gracias a este estudio, a partir de las debilidades encontradas, se determinaron las características que debía tener nuestra herramienta.
- Diseño de la BBDD relacional: Durante esta etapa se crearon los diseños conceptuales y lógicos de la base de datos. Las horas dedicadas incluyen también las horas de aprendizaje.
- Implementación del gestor de BBDD mediante el SGBD elegido: Este fue el proceso más largo de todo el proyecto y es la parte en la que se transformó el diseño lógico en una aplicación informática. Durante esta etapa se combinó el trabajo de aprendizaje, el trabajo de diseño y el trabajo de programación.
- Memoria y presentación: Esta etapa consistió en redactar la memoria y preparar la presentación

<b>TAREAS PROGRAMADAS</b>	<b>DURACIÓN (horas)</b>	<b>COSTE (\$/h)</b>	<b>COSTE TOTAL (U\$D)</b>
<b>Análisis de los requerimientos del gestor de BBDD para diferenciarse de los sistemas existentes</b>	35	50	1750
<i>Conocer los actuales sistemas y sus características</i>	10	50	500
<i>Identificación y análisis de los puntos donde podemos aportar valor añadido.</i>	15	50	750
<i>Análisis de las características diferenciales</i>	10	50	500
<b>Diseño de la BBDD Relacional</b>	64	60	3840
<i>Diseño Conceptual: El Modelo E-R</i>	20	60	1,200
<i>Descripción de datos</i>	8	60	480
<i>Descripción de relaciones</i>	8	60	480
<i>Descripción de restricciones</i>	8	60	480
<i>Diseño lógico: Transformación del Modelo E-R al modelo relacional</i>	20	60	1,200
<b>Implementación del Gestor de BBDD mediante SGBD</b>	177	65,6	11.610
<i>Implementación de las tablas mediante SGBD postgresQL</i>	25	65	1,625
<i>Crear los formularios de la BBDD</i>	15	65	975
<i>Diseño e Implementación de los formularios .....</i>	20	65	1,300
<i>Análisis de la información relevante para el usuario</i>	12	55	660
<i>Implementación de las consultas mediante SGBD PostgreSQL</i>	15	65	975
<i>Diseño e Implementación de los informes mediante Holistics</i>	20	70	1,400
<i>Diseñar e implementar el panel de control</i>	25	70	1,750
<i>Diseñar e implementar los cuadros de diálogo</i>	15	65	975
<i>Diseñar e implementar los menús personalizados</i>	15	65	975
<i>Diseñar e implementar acciones personalizadas</i>	15	65	975
<b>Redactar la memoria</b>	10	50	500

<b>Preparar presentación</b>	5	50	250
<b>TOTAL PROYECTO</b>	331		19,925

#### b- PRESUPUESTO

En este proyecto el presupuesto además de definir los costes del proyecto ha sido la herramienta que ha permitido gestionar la planificación y evitar de este modo las desviaciones en tiempo y como consecuencia directa en costes.

Evidentemente durante la realización de proyecto si han surgido algunas desviaciones en cuanto al tiempo de dedicación en algunas tareas o en cuanto a las fechas concretas de realización de algunas de éstas como se comentarán a continuación.

Debido a las características de este proyecto en el presupuesto se requiere detallar las horas dedicadas a cada tarea y el coste por hora.

Para determinar el coste unitario de las horas se ha tomado como referencia el precio de las horas presupuestadas según las capacidades requeridas de programadores, arquitectos de sistemas, consultores y diseñadores.

A continuación, se detallan los costes por hora de las capacidades requeridas:

Especialidades probables requeridas	Coste unitario
Analista:	U\$D 1750
Diseñador de Base de Datos:	U\$D 3840
Programador:	U\$D 11.610
Presentador:	U\$D 250

Una vez definidos los costes unitarios por hora y especificado el criterio seguido para decidir éstos a continuación se presenta el presupuesto definitivo

En este apartado se pretende realizar una breve reflexión sobre el coste del proyecto relacionándolo con los beneficios y sobre todo con la finalidad de éste.

El coste del proyecto asciende a U\$D 19,925.