BASE DE DATOS II
PARCIAL #1

Modalidad: Trabajo en equipo (máximo 2 estudiantes)
Duración: 2 semanas
Entrega: Moodle

Competencias a evaluar

El estudiante será capaz de:

- Diseñar modelos de datos complejos aplicando cardinalidad correcta.
- Construir diagramas entidad–relación.
- Aplicar normalización hasta 3FN.
- Implementar bases de datos usando Docker.
- Utilizar DDL, DML y CRUD.
- Utilizar VISTAS SQL.
- Justificar decisiones técnicas como DBA.

PARTE I – Diagrama Entidad Relación (10 pts)

Usted ha sido contratado como **Administrador de Base de Datos (DBA)** para diseñar la solución de datos de una plataforma empresarial.

Caso 1: Sistema Empresarial de Servicios Digitales

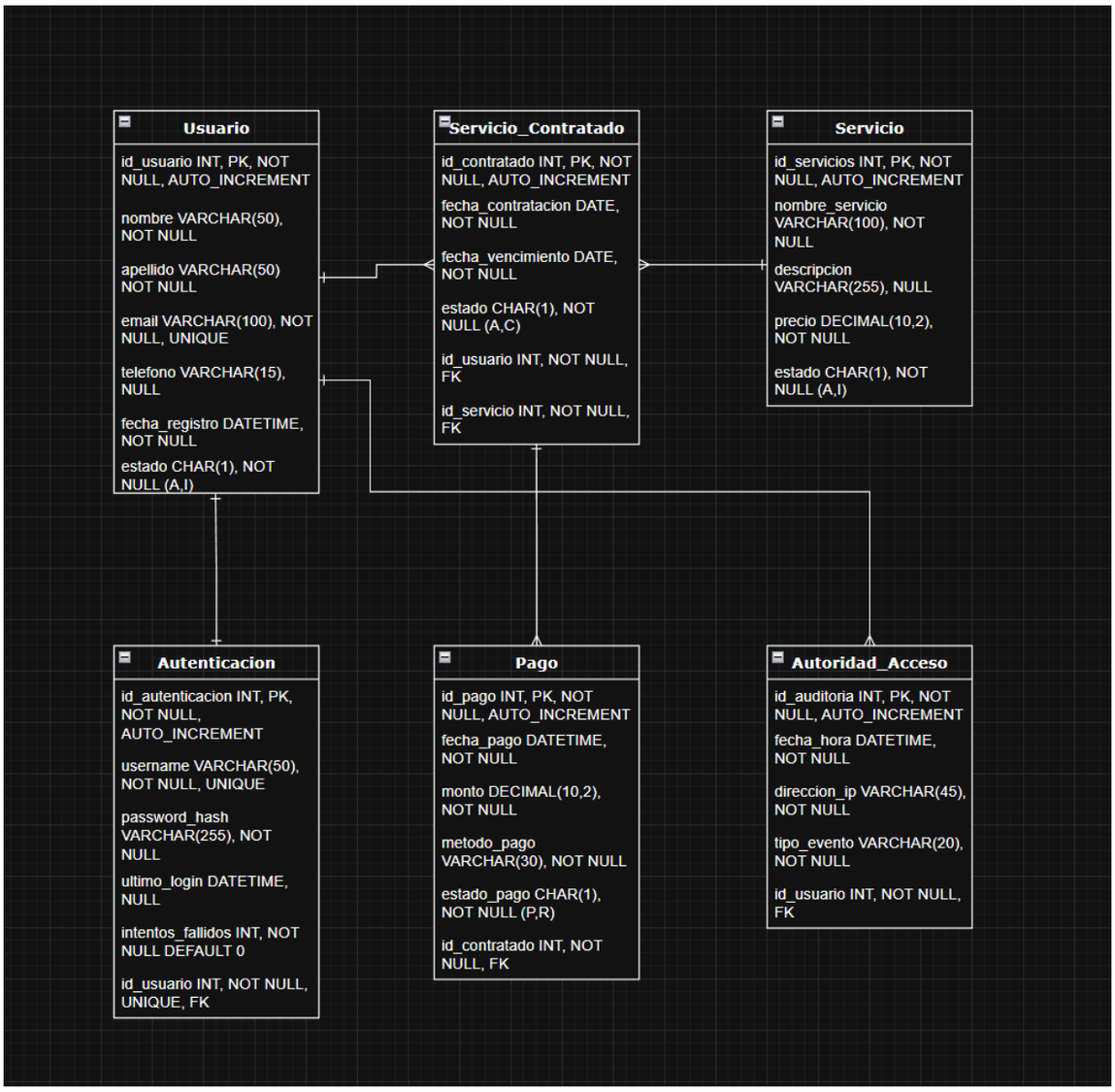
El sistema debe cubrir:

- Gestión de usuarios
- Autenticación
- Servicios contratados
- Pagos
- Auditoría de accesos

Actividades:

- Diseñar un diagrama entidad relación(**DER) completo** que incluya:

  a) Entidades
  b) Atributos
  c) Relaciones
  d) Cardinalidades (1:1, 1:N, N:M)
  e) Entidades asociativas cuando aplique
- El modelo debe reflejar **reglas de negocio reales**



Lo aprendido en la sección: Aprendí que es necesario un buen diseño para el análisis del problema y la identificación clara de las entidades como los atributos, relaciones, cardinalidades y reglas de negocio. El modelo der permite estructura efectivamente y sin problemas el sistema.

PARTE II – Implementación de Base de Datos con Docker (30 pts)

La empresa exige que toda la infraestructura de base de datos sea desplegada mediante Docker, garantizando portabilidad y estandarización.

Uso de Base de Datos con Docker (10 pts)

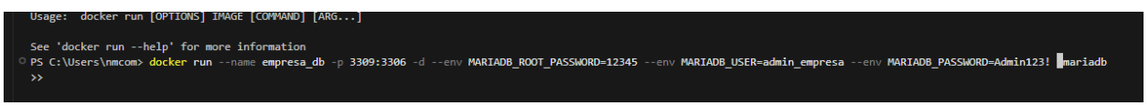
Requerimientos obligatorios

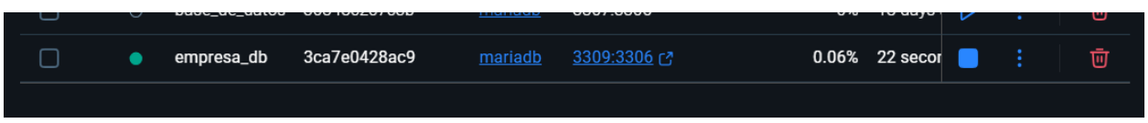
- Crear un contenedor Docker que levante un motor de base de datos relacional (MySQL o MariaDB).
- Utilizar los comandos de docker visto en clase de:
  - Ejecución de contenedor
  - Creación de contenedor
  - La asignación de nombre, variable, puerto e imagen
  - Estado actual del contenedor
  - Visualizar los eventos de logs
  - Conectar al contenedor por CLI
- En la creación del contenedor definir:
  - Nombre del contenedor
  - Usuario
  - Contraseña
  - Puerto expuesto

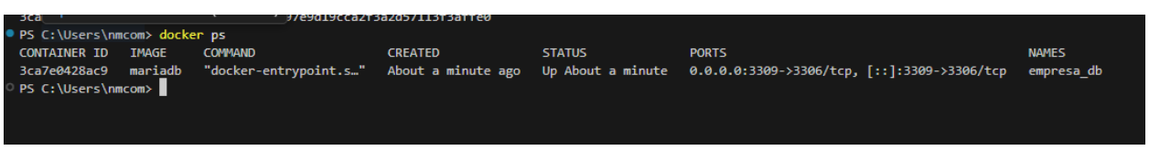
Entregables

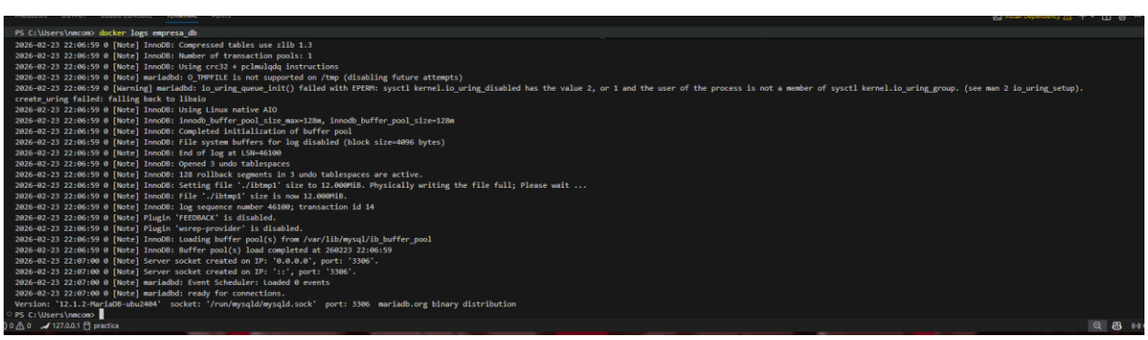
- Comando Docker documentado
- Evidencia de que la base de datos está operativa

<u>NOTA:</u> No se aceptan instalaciones locales fuera de Docker









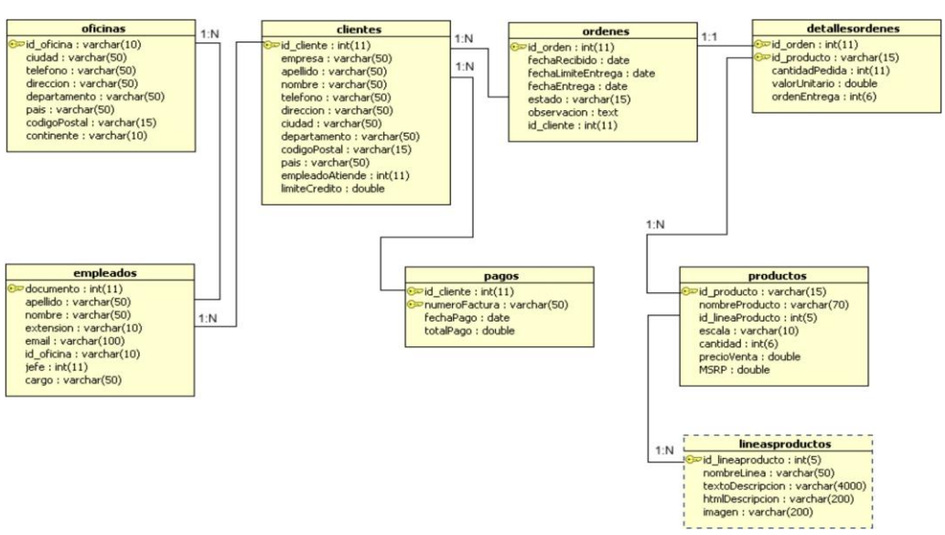
```
PS C:\Users\nmcom> docker exec -it empresa_db mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 12.1.2-MariaDB-ubu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
0 0  127.0.0.1  practica
```

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 7
Server version: 12.1.2-MariaDB-ubu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.006 sec)

MariaDB [(none)]> S
```

Lo aprendido en la sección: el despliegue de la base de datos utilizando Docker me permitió estandarizar el entorno, evitar conflictos que suceden en instalaciones locales, simular un entorno empresarial y garantizar la portabilidad de la base de datos, es importante para la infraestructura como parte del diseño de la base de datos y los sistemas.

Caso 2: Sistema de punto de ventas (20 pts)

Basándonos en el siguiente diagrama entidad relación, realice las siguientes bases de datos.



El diseño base de datos debe tener como mínimo:

- Creación de las estructuras de tablas y relaciones.
- Se deberá agregar por cada tabla, 2 columnas adiciones, qué maneje los diferentes tipos de datos como: tipo lista, tipo si/no y columnas únicas.
- Se debe contar con un mínimo de 10 registros o más por cada tabla.
- Ejecutar con el comando SQL la estructura o descripción de cada tabla.

Tablas creadas:

SistemaDePuntoDeVentas.sql U

C: > Users > nmcom > Desktop > SistemaDePuntoDeVentas.sql > ...

Run | Active: 127.0.0.1 | punto_ventas

```sql
CREATE DATABASE punto_ventas;
```

Run

```sql
USE punto_ventas;

-- Tabla de oficinas
```

Run | Select

```sql
CREATE TABLE oficinas (
    id_oficina VARCHAR(10) PRIMARY KEY,
    ciudad VARCHAR(50),
    telefono VARCHAR(50),
    direccion VARCHAR(50),
    departamento VARCHAR(50),
    pais VARCHAR(50),
    codigoPostal VARCHAR(15),
    continente VARCHAR(10),

    tipo_oficina ENUM('Principal','Sucursal','Regional') NOT NULL,
    activa BOOLEAN NOT NULL,
    codigo_unico VARCHAR(20) UNIQUE
);
```

```sql

-- Tabla de empleados
```

Run | Select

```sql
CREATE TABLE empleados (
    documento INT PRIMARY KEY,
    apellido VARCHAR(50),
    nombre VARCHAR(50),
    extension VARCHAR(10),
    email VARCHAR(100) UNIQUE,
    id_oficina VARCHAR(10),
    jefe INT,
    cargo VARCHAR(50),

    tipo_contrato ENUM('Fijo','Temporal','Servicios') NOT NULL,
    activo BOOLEAN NOT NULL,

    FOREIGN KEY (id_oficina) REFERENCES oficinas(id_oficina)
);
```

```sql
-- Tabla de clientes
CREATE TABLE clientes (
    id_cliente INT PRIMARY KEY,
    empresa VARCHAR(50),
    apellido VARCHAR(50),
    nombre VARCHAR(50),
    telefono VARCHAR(50),
    direccion VARCHAR(50),
    ciudad VARCHAR(50),
    departamento VARCHAR(50),
    codigoPostal VARCHAR(15),
    pais VARCHAR(50),
    empleadoAtiende INT,
    limiteCredito DOUBLE,

    tipo_cliente ENUM('Minorista','Mayorista','Corporativo') NOT NULL,
    activo BOOLEAN NOT NULL,
    email VARCHAR(100) UNIQUE,

    FOREIGN KEY (empleadoAtiende) REFERENCES empleados(documento)
);
```

```sql
-- Tabla de ordenes
CREATE TABLE ordenes (
    id_orden INT PRIMARY KEY,
    fechaRecibido DATE,
    fechaLimiteEntrega DATE,
    fechaEntrega DATE,
    estado VARCHAR(15),
    observacion TEXT,
    id_cliente INT,

    prioridad ENUM('Alta','Media','Baja') NOT NULL,
    facturada BOOLEAN NOT NULL,

    FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)
);
```

```sql
--Tabla de detallesordenes
CREATE TABLE detallesordenes (
    id_orden INT,
    id_producto VARCHAR(15),
    cantidadPedida INT,
    valorUnitario DOUBLE,
    ordenEntrega INT,

    tipo_descuento ENUM('Ninguno','Promocion','ClienteFrecuente') NOT NULL,
    aplica_impuesto BOOLEAN NOT NULL,

    PRIMARY KEY (id_orden, id_producto),
    FOREIGN KEY (id_orden) REFERENCES ordenes(id_orden),
    FOREIGN KEY (id_producto) REFERENCES productos(id_producto)
);
```

```sql
-- Tabla de lineasproductos
CREATE TABLE lineasproductos (
    id_lineaproducto INT PRIMARY KEY,
    nombreLinea VARCHAR(50),
    textoDescripcion VARCHAR(4000),
    htmlDescripcion VARCHAR(200),
    imagen VARCHAR(200),

    categoria ENUM('Electrico','Manual','Industrial') NOT NULL,
    disponible BOOLEAN NOT NULL,
    codigo_linea VARCHAR(20) UNIQUE
);
```

```sql
-- Tabla de productos
CREATE TABLE productos (
    id_producto VARCHAR(15) PRIMARY KEY,
    nombreProducto VARCHAR(70),
    id_lineaProducto INT,
    escala VARCHAR(10),
    cantidad INT,
    precioventa DOUBLE,
    MSRP DOUBLE,

    tipo_producto ENUM('Fisico','Digital','Servicio') NOT NULL,
    activo BOOLEAN NOT NULL,
    codigo_barra VARCHAR(50) UNIQUE,

    FOREIGN KEY (id_lineaProducto) REFERENCES lineasproductos(id_lineaproducto)
);
```

```sql
-- Tabla de pagos
CREATE TABLE pagos (
    id_pago INT PRIMARY KEY,
    id_cliente INT,
    numeroFactura VARCHAR(50) UNIQUE,
    fechaPago DATE,
    totalPago DOUBLE,

    metodo_pago ENUM('Efectivo','Tarjeta','Transferencia') NOT NULL,
    confirmado BOOLEAN NOT NULL,

    FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)
);
```

Selects

SELECT * FROM oficinas LIMIT 100

id_oficina varchar(10)	ciudad varchar(50)	telefono varchar(50)	direccion varchar(50)	departamento varchar(50)	pais varchar(50)	codigoPostal varchar(15)	continente varchar(10)	* tipo_oficina enum('Principal','Sucur	* activa tinyint(1)	codigo_unico varchar(20)
OF1	Panama	2222-1111	Via España	Panama	Panama	0801	America	Principal	1	OF-001
OF10	Arraijan	2222-0000	Centro	Panama Oeste	Panama	1201	America	Sucursal	1	OF-010
OF2	Colon	2222-2222	Zona Libre	Colon	Panama	0301	America	Sucursal	1	OF-002
OF3	David	2222-3333	Centro	Chiriqui	Panama	0401	America	Regional	1	OF-003
OF4	Santiago	2222-4444	Central	Veraguas	Panama	0501	America	Sucursal	1	OF-004
OF5	Chitre	2222-5555	Principal	Herrera	Panama	0601	America	Sucursal	1	OF-005
OF6	Penonome	2222-6666	Centro	Cocle	Panama	0701	America	Regional	1	OF-006
OF7	Bocas	2222-7777	Isla	Bocas	Panama	0901	America	Sucursal	1	OF-007
OF8	Las Tablas	2222-8888	Centro	Los Santos	Panama	1001	America	Sucursal	1	OF-008
OF9	Aguadulce	2222-9999	Principal	Cocle	Panama	1101	America	Regional	1	OF-009

SistemaDePuntoDeVentas.sql U | empleados

Properties | DATA | Log | ER | Monitor | punto_ventas

```
SELECT * FROM empleados LIMIT 100
```

Search Results | Export | Cost: 4ms | 1 | Total 10

documer int(11)	apellido varchar(50)	nombre varchar(50)	extension varchar(10)	email varchar(100)	id_oficina varchar(10)	jefe int(11)	cargo varchar(50)	* tipo_contrato enum('Fijo','Temporal',	* activo tinyint(1)
1	Perez	Juan	101	juan@empresa.com	OF1	(NULL)	Gerente	Fijo	1
2	Lopez	Ana	102	ana@empresa.com	OF2	1	Supervisor	Fijo	1
3	Gomez	Luis	103	luis@empresa.com	OF3	1	Vendedor	Temporal	1
4	Diaz	Maria	104	maria@empresa.com	OF4	2	Vendedor	Servicios	1
5	Torres	Carlos	105	carlos@empresa.com	OF5	2	Vendedor	Fijo	1
6	Ruiz	Laura	106	laura@empresa.com	OF6	1	Supervisor	Fijo	1
7	Mendez	Jose	107	jose@empresa.com	OF7	6	Vendedor	Temporal	1
8	Castro	Elena	108	elena@empresa.com	OF8	6	Vendedor	Servicios	1
9	Vega	Mario	109	mario@empresa.com	OF9	1	Supervisor	Fijo	1
10	Rojas	Sofia	110	sofia@empresa.com	OF10	9	Vendedor	Temporal	1

SistemaDePuntoDeVentas.sql U | clientes

Properties | DATA | Log | ER | Monitor | punto_ventas

```
SELECT * FROM clientes LIMIT 100
```

Search Results | Export | Cost: 13ms | 1 | Total 10

id_cliente int(11)	empresa varchar(50)	apellido varchar(50)	nombre varchar(50)	telefono varchar(50)	direccion varchar(50)	ciudad varchar(50)	departamento varchar(50)	codigoPostal varchar(15)	pais varchar(50)	empleadoAtiende int(11)	limiteCredito double	* tipo_cliente enum('Minorista','May	* activo tinyint(1)	email varchar(100)
1	TechCorp	Santos	Miguel	6000-0001	Dir1	Panama	Panama	0801	Panama	3	5000	Corporativo	1	c1@mail.com
2	Comercial SA	Diaz	Pedro	6000-0002	Dir2	Colon	Colon	0301	Panama	4	3000	Mayorista	1	c2@mail.com
3	Retail PTY	Lopez	Ana	6000-0003	Dir3	David	Chiriqui	0401	Panama	5	2000	Minorista	1	c3@mail.com
4	Distribuidora X	Perez	Luis	6000-0004	Dir4	Santiago	Veraguas	0501	Panama	7	4000	Mayorista	1	c4@mail.com
5	Global Inc	Martinez	Rosa	6000-0005	Dir5	Chitre	Herrera	0601	Panama	8	3500	Corporativo	1	c5@mail.com
6	Empresa 6	Ruiz	Mario	6000-0006	Dir6	Penonome	Cocle	0701	Panama	3	2500	Minorista	1	c6@mail.com
7	Empresa 7	Gomez	Laura	6000-0007	Dir7	Bocas	Bocas	0901	Panama	4	1500	Mayorista	1	c7@mail.com
8	Empresa 8	Torres	Jose	6000-0008	Dir8	Las Tablas	Los Santos	1001	Panama	5	2800	Minorista	1	c8@mail.com
9	Empresa 9	Castro	Elena	6000-0009	Dir9	Aguadulce	Cocle	1101	Panama	7	3200	Corporativo	1	c9@mail.com
10	Empresa 10	Rojas	Sofia	6000-0010	Dir10	Arraijan	Panama Oeste	1201	Panama	8	4100	Mayorista	1	c10@mail.com

SistemaDePuntoDeVentas.sql U | ordenes

Properties | DATA | Log | ER | Monitor | punto_ventas

```
SELECT * FROM ordenes LIMIT 100
```

Search Results | Export | Cost: 5ms | 1 | Total 10

id_orde int(11)	fechaRecibido date	fechaLimiteEntrega date	fechaEntrega date	estado varchar(15)	observacion text	id_cliente int(11)	* prioridad enum('Alta','Media','Ba	* facturada tinyint(1)
1	2026-01-01	2026-01-10	2026-01-08	Entregado	OK	1	Alta	1
2	2026-01-02	2026-01-11	2026-01-09	Entregado	OK	2	Media	1
3	2026-01-03	2026-01-12	(NULL)	Pendiente		3	Baja	0
4	2026-01-04	2026-01-13	(NULL)	Pendiente		4	Alta	0
5	2026-01-05	2026-01-14	2026-01-12	Entregado	OK	5	Media	1
6	2026-01-06	2026-01-15	(NULL)	Pendiente		6	Alta	0
7	2026-01-07	2026-01-16	2026-01-14	Entregado	OK	7	Media	1
8	2026-01-08	2026-01-17	(NULL)	Pendiente		8	Baja	0
9	2026-01-09	2026-01-18	2026-01-16	Entregado	OK	9	Alta	1
10	2026-01-10	2026-01-19	(NULL)	Pendiente		10	Media	0

SistemaDePuntoDeVentas.sql U | detallesordenes

Properties | DATA | Log | ER | Monitor | punto_ventas

```
SELECT * FROM detallesordenes LIMIT 100
```

Search Results | Export | Cost: 5ms | 1 | Total 10

id_orde int(11)	id_producto varchar(15)	cantidadPedida int(11)	valorUnitario double	ordenEntrega int(11)	* tipo_descuento enum('Ninguno','Prom	* aplica_impuesto tinyint(1)
1	P1	2	800	1	Ninguno	1
2	P2	5	20	1	Promocion	1
3	P3	1	150	1	Ninguno	1
4	P4	3	120	1	ClienteFrecuente	1
5	P5	1	500	1	Ninguno	0
6	P6	2	200	1	Promocion	1
7	P7	4	30	1	Ninguno	1
8	P8	1	90	1	ClienteFrecuente	1
9	P9	10	15	1	Ninguno	1
10	P10	1	1000	1	Promocion	0

SistemaDePuntoDeVentas.sql U | lineasproductos

Properties | DATA | Log | ER | Monitor | punto_ventas

```
SELECT * FROM lineasproductos LIMIT 100
```

Search Results | Export | Cost: 9ms | 1 | Total 10

id_lineaproducto int(11)	nombreLinea varchar(50)	textoDescripcion varchar(4000)	htmlDescripcion varchar(200)	imagen varchar(200)	* categoria enum('Electrico','Manu	* disponible tinyint(1)	codigo_linea varchar(20)
1	Electronica	Desc1	html1	img1	Electrico	1	L-001
2	Hogar	Desc2	html2	img2	Manual	1	L-002
3	Industrial	Desc3	html3	img3	Industrial	1	L-003
4	Oficina	Desc4	html4	img4	Manual	1	L-004
5	Tecnologia	Desc5	html5	img5	Electrico	1	L-005
6	Automotriz	Desc6	html6	img6	Industrial	1	L-006
7	Salud	Desc7	html7	img7	Manual	1	L-007
8	Construccion	Desc8	html8	img8	Industrial	1	L-008
9	Deportes	Desc9	html9	img9	Manual	1	L-009
10	Servicios	Desc10	html10	img10	Electrico	1	L-010

SistemaDePuntoDeVentas.sql U | productos

Properties | DATA | Log | ER | Monitor | punto_ventas

```
SELECT * FROM productos LIMIT 100
```

Search Results | Export | Cost: 4ms | 1 | Total 10

id_producto varchar(15)	nombreProducto varchar(70)	id_lineaProducto int(11)	escala varchar(10	cantidad int(11)	precioventa double	MSRP double	* tipo_producto enum('Fisico','Digital',	* activo tinyint(1)	codigo_barra varchar(50)
P1	Laptop	1	Alta	50	800	900	Fisico	1	CB001
P10	Consultoria	10	N/A	0	1000	1200	Servicio	1	CB010
P2	Mouse	1	Media	100	20	25	Fisico	1	CB002
P3	Impresora	4	Alta	30	150	180	Fisico	1	CB003
P4	Taladro	8	Media	40	120	150	Fisico	1	CB004
P5	Servicio IT	10	N/A	0	500	600	Servicio	1	CB005
P6	Monitor	5	Alta	60	200	250	Fisico	1	CB006
P7	Teclado	1	Media	80	30	40	Fisico	1	CB007
P8	Silla Oficina	4	Alta	20	90	110	Fisico	1	CB008
P9	Aceite Motor	6	Media	200	15	20	Fisico	1	CB009

SistemaDePuntoDeVentas.sql U | pagos

Properties | DATA | Log | ER | Monitor | punto_ventas

```
SELECT * FROM pagos LIMIT 100
```

Search Results | Export | Cost: 8ms | 1 | Total 10

id_pag int(11)	id_cliente int(11)	numeroFactura varchar(50)	fechaPago date	totalPago double	* metodo_pago enum('Efectivo','Tarjeta	* confirmado tinyint(1)
1	1	F001	2026-01-08	1600	Tarjeta	1
2	2	F002	2026-01-09	100	Efectivo	1
3	3	F003	2026-01-10	150	Transferencia	1
4	4	F004	2026-01-11	360	Tarjeta	1
5	5	F005	2026-01-12	500	Efectivo	1
6	6	F006	2026-01-13	400	Transferencia	1
7	7	F007	2026-01-14	120	Tarjeta	1
8	8	F008	2026-01-15	90	Efectivo	1
9	9	F009	2026-01-16	150	Transferencia	1
10	10	F010	2026-01-17	1000	Tarjeta	1

Los puntos **(III y IV)** se basará bajo el contexto del **caso 1**
"Sistema Empresarial de Servicios Digitales"

PARTE III - Normalización y diseño lógico (20 puntos)

Normalización de la Base de Datos (10 pts)

- Aplicar normalización hasta Tercera Forma Normal (3FN).
- **1FN: Se separo los datos en entidades individuales (usuario, servicios, pago, auditoría y autentificación), básicamente cada tabla representa una sola entidad especifica y cada atributo contiene su único valor.**
- **2FN: En el sistema tenia una relación muchos a muchos entre el usuario y el servicios, para resolverlo cree una entidad llamada Servicio_Contratado, y la tabla almacena su propia información del contrato (fecha_contratacion, fecha_vencimiento. estado.)**
- **3FN: Luego de todo esto en el diseño final hice en base a la 3FN: cada tabla depende únicamente de su clave primaria, no existen atributos que dependen de otros que no sean atributos clave, y la información esta separada por responsabilidad.**
- Explicar con sus propias palabras:
    - Qué problemas se resolvieron en cada etapa
    - **1FN: Se eliminaron los grupos que se repetían y también se garantizo que cada una de los campos tenga un único valor.**
    - **2FN: Se resolvió el problema de la relación de muchos a muchos, cree la entidad Servicio_Contratado y también se eliminaron las dependencias parciales.**
    - **3FN: Se redujo las redundancias de datos, se aseguro cada atributo dependiente únicamente de la clave primaria y se eliminaron dependencias transitivas.**
    - Por qué el diseño final es óptimo
    - **Porque se redujeron las redundancias de datos, evitamos la inconsistencia de actualizaciones, elimine anomalías de inserción y eliminación, mejore la integridad referencial, separe las responsabilidades de identidades y cumple con la tercera forma normal.**

NOTA: Respuestas copiadas de internet o sin justificación técnica no suman puntos.

Implementación SQL – DDL (10 pts)

- Crear todas las tablas del modelo usando SQL.
- Cada tabla debe incluir:
    - Clave primaria
    - Claves foráneas
    - Restricciones (NOT NULL, UNIQUE, CHECK)
- Por cada tabla:
    - 1 campo tipo lista (ENUM o tabla catálogo)
    - 1 campo tipo booleano
    - 1 restricción UNIQUE

Lo aprendido de esta sección: Aplicar normalizaciones hasta la tercera forma normal reduce redundancia de datos, eliminar dependencias parciales, previenen anomalías de inserción, actualización y eliminación, además mejora la integridad y consistencia de la información.
La integridad referencial y restricciones nos permite implementar las claves foráneas, primarias, not null, unique etc. Permite entender como el motor de base de datos garantiza coherencia de los datos automáticamente.

C: > Users > nmcom > Desktop > EmpresaServicios.sql > ...

▷Run | Active: 127.0.0.1 | punto_ventas

```sql
CREATE DATABASE empresa_servicios;
```

▷Run

```sql
USE empresa_servicios;

-- Tabla de usuarios
```

▷Run | Select

```sql
CREATE TABLE usuario (
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    telefono VARCHAR(15),
    fecha_registro DATETIME NOT NULL,
    estado ENUM('Activo','Inactivo') NOT NULL,
    verificado BOOLEAN NOT NULL,
    codigo_cliente VARCHAR(20) UNIQUE
);
```

```sql

-- Tabla de autenticacion
```

▷Run | Select

```sql
CREATE TABLE autenticacion (
    id_autenticacion INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL,
    ultimo_login DATETIME,
    intentos_fallidos INT DEFAULT 0 CHECK (intentos_fallidos >= 0),
    tipo_usuario ENUM('Admin','Cliente') NOT NULL,
    bloqueado BOOLEAN NOT NULL,
    id_usuario INT NOT NULL UNIQUE,
    FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario)
);

-- Tabla de servicio
```

▷Run | Select

```sql
CREATE TABLE servicio (
    id_servicio INT AUTO_INCREMENT PRIMARY KEY,
    nombre_servicio VARCHAR(100) NOT NULL UNIQUE,
    descripcion VARCHAR(255),
    precio DECIMAL(10,2) NOT NULL CHECK (precio >= 0),
    categoria ENUM('Streaming','Cloud','Soporte','Software') NOT NULL,
    activo BOOLEAN NOT NULL,
    codigo_servicio VARCHAR(20) UNIQUE
);
```

```sql
-- Tabla servicio_contratado
CREATE TABLE servicio_contratado (
    id_contrato INT AUTO_INCREMENT PRIMARY KEY,
    fecha_contratacion DATE NOT NULL,
    fecha_vencimiento DATE NOT NULL,
    estado ENUM('Activo','Cancelado','Suspendido') NOT NULL,
    renovacion_automatica BOOLEAN NOT NULL,
    numero_contrato VARCHAR(30) UNIQUE,
    id_usuario INT NOT NULL,
    id_servicio INT NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario),
    FOREIGN KEY (id_servicio) REFERENCES servicio(id_servicio)
);
```

```sql
-- Tabla de pago
CREATE TABLE pago (
    id_pago INT AUTO_INCREMENT PRIMARY KEY,
    fecha_pago DATETIME NOT NULL,
    monto DECIMAL(10,2) NOT NULL CHECK (monto > 0),
    metodo_pago ENUM('Tarjeta','Transferencia','PayPal') NOT NULL,
    confirmado BOOLEAN NOT NULL,
    numero_factura VARCHAR(30) UNIQUE,
    id_contrato INT NOT NULL,
    FOREIGN KEY (id_contrato) REFERENCES servicio_contratado(id_contrato)
);
```

```sql
--Tabla de autoridad_acceso
CREATE TABLE auditoria_acceso (
    id_auditoria INT AUTO_INCREMENT PRIMARY KEY,
    fecha_hora DATETIME NOT NULL,
    direccion_ip VARCHAR(45) NOT NULL,
    tipo_evento ENUM('Login','Logout','IntentoFallido') NOT NULL,
    exitoso BOOLEAN NOT NULL,
    codigo_evento VARCHAR(30) UNIQUE,
    id_usuario INT NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario)
);  26ms
```

PARTE IV – DML, CRUD y Vistas (40 puntos)

Inserción de Datos – DML (10 pts)

- Insertar **mínimo 10 registros por tabla**.
- Los datos deben ser:
  - Realistas
  - Coherentes
  - Relacionados correctamente

EmpresaServicios.sql u | servicio_contratado

Properties | DATA | Log | ER | Monitor | empresa_servicios

```
SELECT * FROM servicio_contratado LIMIT 100
```

Search Results | Export | Cost: 6ms | 1 | Total 10

id_contrato int(11)	fecha_contratacion date	fecha_vencimiento date	estado enum('Activo','Cancela	renovacion_automat tinyint(1)	numero_contrato varchar(30)	id_usuari int(11)	id_servicio int(11)
1	2026-01-01	2026-12-31	Activo	1	CON-001	1	1
2	2026-01-02	2026-12-31	Activo	1	CON-002	2	2
3	2026-01-03	2026-12-31	Activo	0	CON-003	3	3
4	2026-01-04	2026-12-31	Suspendido	0	CON-004	4	4
5	2026-01-05	2026-12-31	Activo	1	CON-005	5	5
6	2026-01-06	2026-12-31	Activo	1	CON-006	6	6
7	2026-01-07	2026-12-31	Activo	0	CON-007	7	7
8	2026-01-08	2026-12-31	Cancelado	0	CON-008	8	8
9	2026-01-09	2026-12-31	Activo	1	CON-009	9	9
10	2026-01-10	2026-12-31	Activo	1	CON-010	10	10

EmpresaServicios.sql u | usuario

Properties | DATA | Log | ER | Monitor | empresa_servicios

```
SELECT * FROM usuario LIMIT 100
```

Search Results | Export | Cost: 5ms | 1 | Total 10

id_usuario int(11)	nombre varchar(50)	apellido varchar(50)	email varchar(100)	telefono varchar(15)	fecha_registro datetime	estado enum('Activo','Inactivo	verificado tinyint(1)	codigo_cliente varchar(20)
1	Carlos	Poveda	carlos@empresa.com	6000-1001	2026-02-24 23:56:50	Activo	1	CLI-001
2	Ana	Gomez	ana@empresa.com	6000-1002	2026-02-24 23:56:50	Activo	1	CLI-002
3	Luis	Martinez	luis@empresa.com	6000-1003	2026-02-24 23:56:50	Activo	0	CLI-003
4	Maria	Lopez	maria@empresa.com	6000-1004	2026-02-24 23:56:50	Activo	1	CLI-004
5	Pedro	Sanchez	pedro@empresa.com	6000-1005	2026-02-24 23:56:50	Inactivo	0	CLI-005
6	Laura	Diaz	laura@empresa.com	6000-1006	2026-02-24 23:56:50	Activo	1	CLI-006
7	Jorge	Ruiz	jorge@empresa.com	6000-1007	2026-02-24 23:56:50	Activo	1	CLI-007
8	Elena	Torres	elena@empresa.com	6000-1008	2026-02-24 23:56:50	Activo	0	CLI-008
9	Miguel	Castro	miguel@empresa.com	6000-1009	2026-02-24 23:56:50	Activo	1	CLI-009
10	Sofia	Rojas	sofia@empresa.com	6000-1010	2026-02-24 23:56:50	Activo	1	CLI-010

EmpresaServicios.sql u | auditoria_acceso

Properties | DATA | Log | ER | Monitor | empresa_servicios

```
SELECT * FROM auditoria_acceso LIMIT 100
```

Search Results | Export | Cost: 7ms | 1 | Total 10

id_auditoria int(11)	fecha_hora datetime	direccion_ip varchar(45)	tipo_evento enum('Login','Logout',	exitoso tinyint(1)	codigo_evento varchar(30)	id_usuari int(11)
1	2026-02-24 23:59:40	192.168.1.1	Login	1	EV-001	1
2	2026-02-24 23:59:40	192.168.1.2	Login	1	EV-002	2
3	2026-02-24 23:59:40	192.168.1.3	IntentoFallido	0	EV-003	3
4	2026-02-24 23:59:40	192.168.1.4	Login	1	EV-004	4
5	2026-02-24 23:59:40	192.168.1.5	Logout	1	EV-005	5
6	2026-02-24 23:59:40	192.168.1.6	Login	1	EV-006	6
7	2026-02-24 23:59:40	192.168.1.7	IntentoFallido	0	EV-007	7
8	2026-02-24 23:59:40	192.168.1.8	Login	1	EV-008	8
9	2026-02-24 23:59:40	192.168.1.9	Logout	1	EV-009	9
10	2026-02-24 23:59:40	192.168.1.10	Login	1	EV-010	10

EmpresaServicios.sql u | autenticacion

Properties | DATA | Log | ER | Monitor | empresa_servicios

```
SELECT * FROM autenticacion LIMIT 100
```

Search Results | Export | Cost: 9ms | 1 | Total 10

id_autenticacion int(11)	usernam varchar(50)	password_hash varchar(255)	ultimo_login datetime	intentos_fallidos int(11)	tipo_usuario enum('Admin','Cliente	bloqueado tinyint(1)	id_usuari int(11)
1	carlos_admin	hash1	2026-02-24 23:59:05	0	Admin	0	1
2	ana_user	hash2	2026-02-24 23:59:05	0	Cliente	0	2
3	luis_user	hash3	2026-02-24 23:59:05	1	Cliente	0	3
4	maria_user	hash4	2026-02-24 23:59:05	0	Cliente	0	4
5	pedro_user	hash5	2026-02-24 23:59:05	2	Cliente	0	5
6	laura_user	hash6	2026-02-24 23:59:05	0	Cliente	0	6
7	jorge_user	hash7	2026-02-24 23:59:05	0	Cliente	0	7
8	elena_user	hash8	2026-02-24 23:59:05	1	Cliente	0	8
9	miguel_user	hash9	2026-02-24 23:59:05	0	Cliente	0	9
10	sofia_user	hash10	2026-02-24 23:59:05	0	Cliente	0	10

EmpresaServicios.sql U | pago

Properties | DATA | Log | ER | Monitor | empresa_servicios

```
SELECT * FROM pago LIMIT 100
```

Search Results | Export | Cost: 9ms | 1 | Total 10

id_pag int(11)	fecha_pago datetime	monto decimal(10,2)	metodo_pago enum('Tarjeta','Transfe	confirmado tinyint(1)	numero_factura varchar(30)	id_contrat(int(11)
1	2026-02-24 23:59:35	15.99	Tarjeta	1	FAC-001	1
2	2026-02-24 23:59:35	9.99	Transferencia	1	FAC-002	2
3	2026-02-24 23:59:35	19.99	Tarjeta	1	FAC-003	3
4	2026-02-24 23:59:35	29.99	PayPal	1	FAC-004	4
5	2026-02-24 23:59:35	12.50	Tarjeta	1	FAC-005	5
6	2026-02-24 23:59:35	99.99	Transferencia	1	FAC-006	6
7	2026-02-24 23:59:35	8.99	Tarjeta	1	FAC-007	7
8	2026-02-24 23:59:35	14.99	PayPal	1	FAC-008	8
9	2026-02-24 23:59:35	45.00	Tarjeta	1	FAC-009	9
10	2026-02-24 23:59:35	11.00	Transferencia	1	FAC-010	10

EmpresaServicios.sql U | servicio

Properties | DATA | Log | ER | Monitor | empresa_servicios

```
SELECT * FROM servicio LIMIT 100
```

Search Results | Export | Cost: 5ms | 1 | Total 10

id_servicio int(11)	nombre_servicio varchar(100)	descripcion varchar(255)	precio decimal(10,2)	categoria enum('Streaming','Clou	activo tinyint(1)	codigo_servicio varchar(20)
1	Streaming Pro	Plataforma streaming HD	15.99	Streaming	1	SER-001
2	Cloud Basic	Almacenamiento 100GB	9.99	Cloud	1	SER-002
3	Cloud Premium	Almacenamiento 1TB	19.99	Cloud	1	SER-003
4	Soporte 24/7	Soporte tecnico permanent	29.99	Soporte	1	SER-004
5	Antivirus	Proteccion empresarial	12.50	Software	1	SER-005
6	ERP Empresarial	Sistema ERP completo	99.99	Software	1	SER-006
7	Backup Online	Respaldo automatico	8.99	Cloud	1	SER-007
8	Hosting Web	Hosting empresarial	14.99	Cloud	1	SER-008
9	Licencia Office	Licencia anual office	45.00	Software	1	SER-009
10	VPN Empresarial	Conexion segura remota	11.00	Software	1	SER-010

Operaciones CRUD (15 pts)

Seleccionar **una entidad principal** del sistema y realizar:

- CREATE → Inserción
- READ → Consulta filtrada
- UPDATE → Actualización controlada
  DELETE → Eliminación lógica o física (justificar)

Se deben entregar los **scripts SQL** y explicar el propósito de cada operación.

```sql

--Opreaciones CRUD

-- CREATE Registra un nuevo usuario en la tabla de usuarios
▷Run
INSERT INTO usuario (nombre,apellido,email,telefono,fecha_registro,estado,verificado,codigo_cliente)
VALUES ('Nuevo','Cliente','nuevo@empresa.com','6000-2020',NOW(),'Activo',1,'CLI-011');

-- READ Consulta usuarios activos, por ejemplo para analisis comerical
▷Run | +Tab | JSON
SELECT nombre, apellido, email
FROM usuario
WHERE estado = 'Activo';

-- UPDATE desactiva usuario sin eliminarlo, por ejemplo para mantener historial de contratos
▷Run | Select
UPDATE usuario
SET estado = 'Inactivo'
WHERE id_usuario = 3;

--DELETE no elimina fisicamente un usuario, sino que lo marca como inactivo para preservar integridad referencial
▷Run | Select
UPDATE usuario
SET estado = 'Inactivo'
WHERE id_usuario = 5;
```

- **CREATE Registra un nuevo usuario en la tabla de usuarios**
- **READ Consulta usuarios activos, por ejemplo para analisis comerical**
- **UPDATE desactiva usuario sin eliminarlo, por ejemplo para mantener historial de contratos**
- **DELETE no elimina fisicamente un usuario, sino que lo marca como inactivo para preservar integridad referencial**

Lo aprendido en esta sección: El DML y las operaciones CRUD nos permite la eliminaciónde lógica preservar historial y auditoria, las consultas filtradas permiten análisis específicos,las actualizaciones deben ser controladas para no afectar integridad

Creación y Uso de VISTAS SQL (15 pts)

Requerimientos obligatorios

Crear **mínimo 3 VISTAS**, cumpliendo:

1. **Vista de negocio**
   - Consolidación de datos relevantes para toma de decisiones
   - Ejemplo: usuarios + servicios + pagos

2. **Vista de seguridad**
   - Oculta datos sensibles (contraseñas, tokens, etc.)

3. **Vista de auditoría**
   - Muestra accesos, fechas, estado de usuarios o transacciones

```sql

-- VISTAS SQLS

-- Vista de negocio / utilidad: analiza el ingeso por usuario y servicio
CREATE VIEW vista_negocio AS
SELECT
u.nombre,
u.apellido,
s.nombre_servicio,
p.monto,
p.fecha_pago
FROM usuario u
JOIN servicio_contratado sc ON u.id_usuario = sc.id_usuario
JOIN servicio s ON sc.id_servicio = s.id_servicio
JOIN pago p ON sc.id_contrato = p.id_contrato;

--Vista de seguridad  / utilidad: muestra la informacion sin exponer la password_hash
CREATE VIEW vista_seguridad AS
SELECT
u.id_usuario,
u.nombre,
u.email,
a.username,
a.bloqueado
FROM usuario u
JOIN autenticacion a ON u.id_usuario = a.id_usuario;

-- Vista de auditoria / utilidad: monitorea los eventos sospechosos y tambien los accesos
CREATE VIEW vista_auditoria AS
SELECT
u.nombre,
a.fecha_hora,
a.tipo_evento,
a.exitoso
FROM auditoria_acceso a
JOIN usuario u ON a.id_usuario = u.id_usuario;
```

- **Vista de negocio / utilidad: analiza el ingeso por usuario y servicio**
- **Vista de seguridad / utilidad: muestra la informacion sin exponer la password_hash**
- **Vista de auditoria / utilidad: monitorea los eventos sospechosos y tambien los accesos**

EmpresaServicios.sql U | vista_auditoria

Properties | DATA | Log | ER | Monitor | empresa_servicios

```
SELECT * FROM vista_auditoria LIMIT 100
```

Search Results | Export | Cost: 9ms | 1 | Total 10

* nombre varchar(50)	* fecha_hora datetime	* tipo_evento enum('Login','Logout',	* exitoso tinyint(1)
Carlos	2026-02-24 23:59:40	Login	1
Ana	2026-02-24 23:59:40	Login	1
Luis	2026-02-24 23:59:40	IntentoFallido	0
Maria	2026-02-24 23:59:40	Login	1
Pedro	2026-02-24 23:59:40	Logout	1
Laura	2026-02-24 23:59:40	Login	1
Jorge	2026-02-24 23:59:40	IntentoFallido	0
Elena	2026-02-24 23:59:40	Login	1
Miguel	2026-02-24 23:59:40	Logout	1
Sofia	2026-02-24 23:59:40	Login	1

EmpresaServicios.sql U | vista_negocio

Properties | DATA | Log | ER | Monitor | empresa_servicios

```
SELECT * FROM vista_negocio LIMIT 100
```

Search Results | Export | Cost: 10ms | 1 | Total 10

* nombre varchar(50)	* apellido varchar(50)	* nombre_servicio varchar(100)	* monto decimal(10,2)	* fecha_pago datetime
Carlos	Poveda	Streaming Pro	15.99	2026-02-24 23:59:35
Ana	Gomez	Cloud Basic	9.99	2026-02-24 23:59:35
Luis	Martinez	Cloud Premium	19.99	2026-02-24 23:59:35
Maria	Lopez	Soporte 24/7	29.99	2026-02-24 23:59:35
Pedro	Sanchez	Antivirus	12.50	2026-02-24 23:59:35
Laura	Diaz	ERP Empresarial	99.99	2026-02-24 23:59:35
Jorge	Ruiz	Backup Online	8.99	2026-02-24 23:59:35
Elena	Torres	Hosting Web	14.99	2026-02-24 23:59:35
Miguel	Castro	Licencia Office	45.00	2026-02-24 23:59:35
Sofia	Rojas	VPN Empresarial	11.00	2026-02-24 23:59:35

EmpresaServicios.sql U | vista_seguridad

Properties | DATA | Log | ER | Monitor | empresa_servicios

```
SELECT * FROM vista_seguridad LIMIT 100
```

Search Results | Export | Cost: 11ms | 1 | Total 10

* id_usuario int(11)	* nombre varchar(50)	* email varchar(100)	* usernam varchar(50)	* bloqueado tinyint(1)
1	Carlos	carlos@empresa.com	carlos_admin	0
2	Ana	ana@empresa.com	ana_user	0
3	Luis	luis@empresa.com	luis_user	0
4	Maria	maria@empresa.com	maria_user	0
5	Pedro	pedro@empresa.com	pedro_user	0
6	Laura	laura@empresa.com	laura_user	0
7	Jorge	jorge@empresa.com	jorge_user	0
8	Elena	elena@empresa.com	elena_user	0
9	Miguel	miguel@empresa.com	miguel_user	0
10	Sofia	sofia@empresa.com	sofia_user	0

Lo aprendido en esta sección: La creación de vistas me ayudo a comprender que podemos consolidar datos para análisis de negocios, también es posible proteger información sensible, facilitar el monitoreo mediante vistas de auditoría. Y mejorar la seguridad y la organización de las informaciones clave como herramienta para los entornos empresariales.

Evaluación

- Correcta construcción de la vista
- Uso real (SELECT sobre la vista)
- Justificación de su utilidad

Lineamiento e información a entregar

1. Desarrollar cada uno de los puntos
2. Crear el repositorio de carpeta ***"PARCIAL_01"***
3. Subir la copia de toda la base de datos
4. Capturas de cada de los procesos (Sentencia SQL y resultado)
5. Documentar todo el proceso y agregar las lecciones aprendidas.

Se puede realizar individual o en equipo con **un máximo de 2 integrantes,** con la condición qué por cada integrante adicional se deben agregar 10 registros en cada módulo, en caso de no cumplir con esta cláusula, se le baja automáticamente **-20 puntos de la nota final.**

Se debe subir a la plataforma el link o todos los documentos.

¡Buena Suerte!