

# 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.

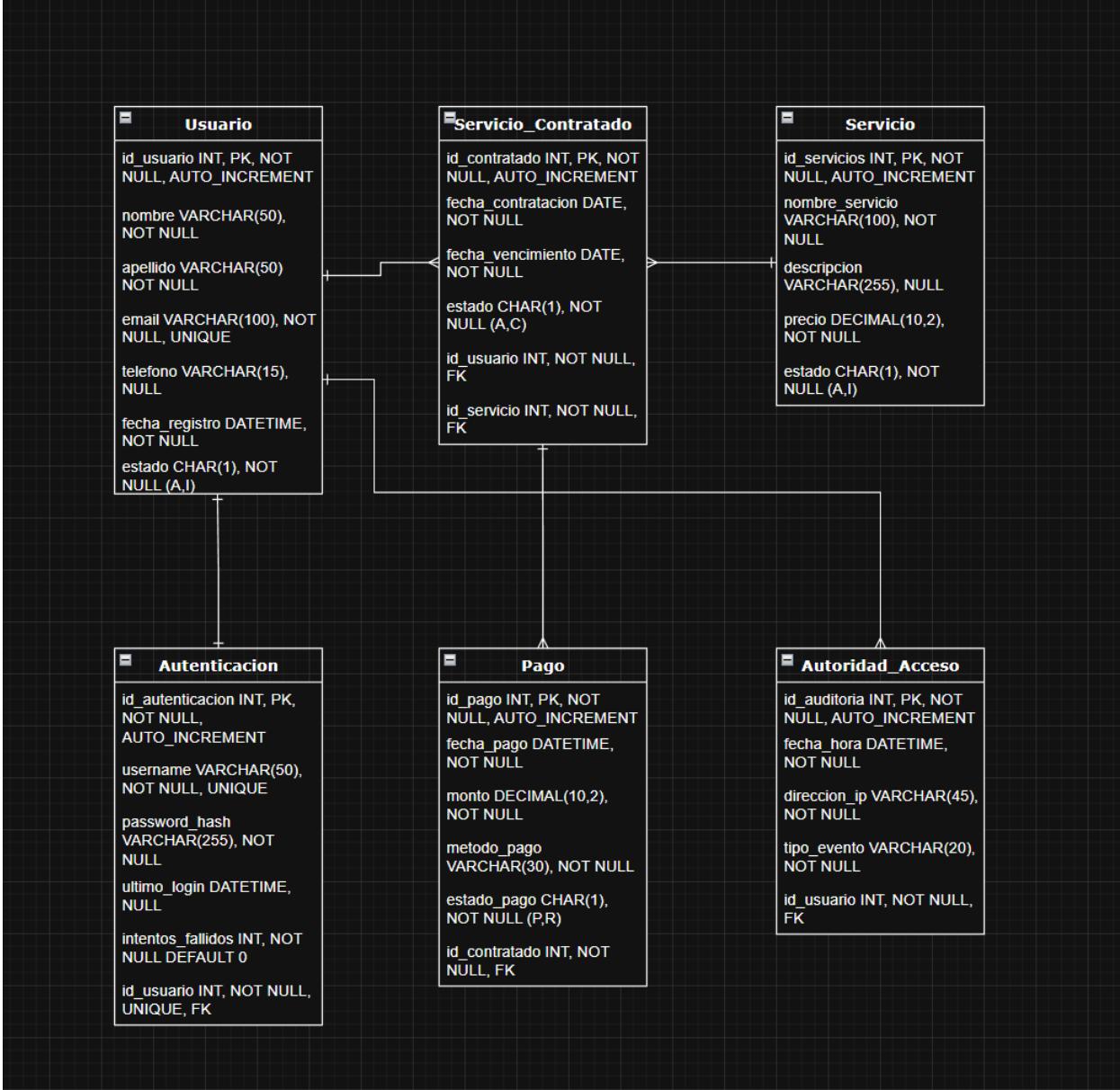
#### Caso 1: Sistema Empresarial de Servicios Digitales

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**



# 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

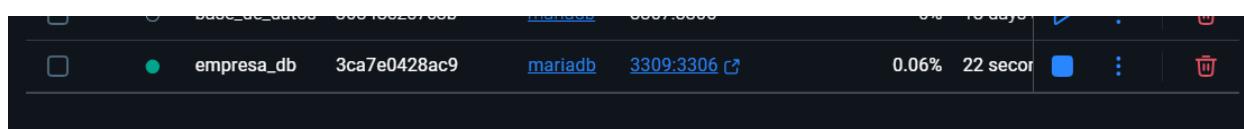
- Crear un contenedor Docker que levante un motor de base de datos relacional (MySQL o MariaDB).
- Utilizar los comandos de docker visto en clase de:
  - Ejecución de contenedor
  - Creación de contenedor
  - La asignación de nombre, variable, puerto e imagen
  - Estado actual del contenedor
  - Visualizar 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

**NOTA:** No se aceptan instalaciones locales fuera de Docker

```
Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
See 'docker run --help' for more information
$ PS C:\Users\nmcom> docker run --name empresa_db -p 3309:3306 -d --env MARIADB_ROOT_PASSWORD=12345 --env MARIADB_USER=admin_empresa --env MARIADB_PASSWORD=Admin123! mariadb
>>
```



```
PS C:\Users\nmcom> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
3ca7e0428ac9        mariadb            "docker-entrypoint.s..."   About a minute ago   Up About a minute   0.0.0.0:3309->3306/tcp, [::]:3309->3306/tcp   empresa_db
$ PS C:\Users\nmcom> █
```

```
PS C:\Users\nmcom> docker logs empresa_db
2026-02-23 22:06:59.0 [Note] InnoDB: Compressed tables use zlib 1.3
2026-02-23 22:06:59.0 [Note] InnoDB: Number of transaction pools: 1
2026-02-23 22:06:59.0 [Note] InnoDB: Using crc32 + pclmuqdq instructions
2026-02-23 22:06:59.0 [Note] mariadb: O_TMPFILE is not supported on /tmp (disabling future attempts)
2026-02-23 22:06:59.0 [Warning] mariadb: io_uring_queue_init() failed with EPERM: sysctl kernel.io_uring_disabled has the value 2, or 1 and the user of the process is not a member of sysctl kernel.io_uring_group. (see man 2 io_uring_setup).
2026-02-23 22:06:59.0 [Note] InnoDB: Using Linux native AIO
2026-02-23 22:06:59.0 [Note] InnoDB: innodb_buffer_pool_size max=128m, innodb_buffer_pool_size=128m
2026-02-23 22:06:59.0 [Note] InnoDB: Completed initialization of buffer pool
2026-02-23 22:06:59.0 [Note] InnoDB: File system buffers for log disabled (block size=4096 bytes)
2026-02-23 22:06:59.0 [Note] InnoDB: End of log at LSN=44100
2026-02-23 22:06:59.0 [Note] InnoDB: Last checkpoint at LSN=44100
2026-02-23 22:06:59.0 [Note] InnoDB: 128 rollback segments in 1 undo tablespaces
2026-02-23 22:06:59.0 [Note] InnoDB: Setting file './ibtmp1' size to 12.000MB. Physically writing the file full; Please wait ...
2026-02-23 22:06:59.0 [Note] InnoDB: File './ibtmp1' size is now 12.000MB.
2026-02-23 22:06:59.0 [Note] InnoDB: log sequence number 46100, transaction id 14
2026-02-23 22:06:59.0 [Note] Plugin 'FEEDBACK' is disabled.
2026-02-23 22:06:59.0 [Note] InnoDB: Buffer pool precreation is enabled.
2026-02-23 22:06:59.0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
2026-02-23 22:06:59.0 [Note] InnoDB: Buffer pool(s) load completed at 2026/02/23 22:06:59
2026-02-23 22:07:00.0 [Note] Server socket created on IP: '0.0.0.0', port: '3306'.
2026-02-23 22:07:00.0 [Note] Server socket created on IP: '::', port: '3306'.
2026-02-23 22:07:00.0 [Note] mariadb: Event Scheduler: Loaded 0 events
2026-02-23 22:07:00.0 [Note] mariadb: ready for connections
2026-02-23 22:07:00.0 [Note] mariadb: listening on port 3306 socket: '/run/mysql/mariadb.sock' port: 3306 mariadb.org binary distribution
$ PS C:\Users\nmcom> █
```

```
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-ubuntu2404 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)]>
⑧ 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-ubuntu2404 mariadb.org binary distribution
```

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

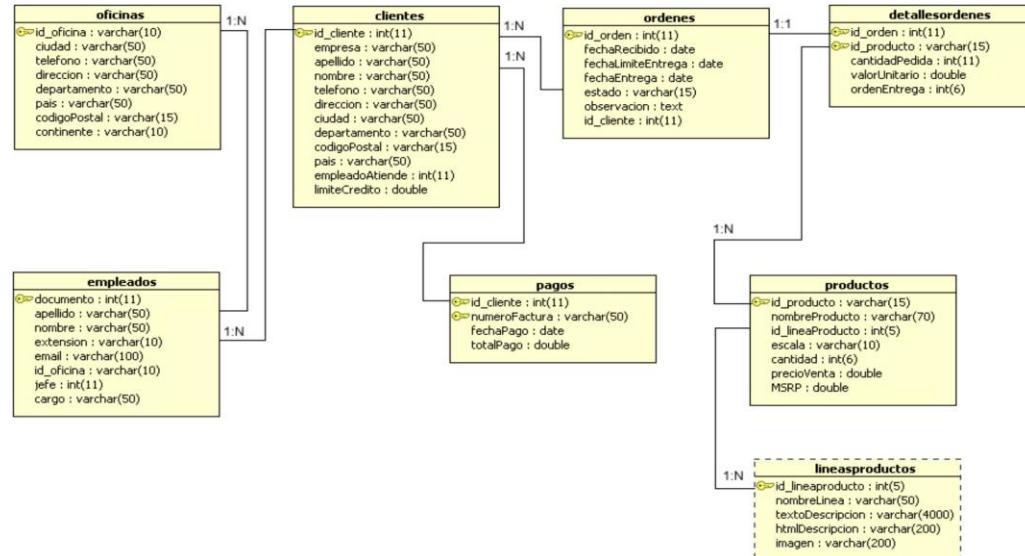
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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

```
MariaDB [(none)]> S
⑧ 0 △ 0 127.0.0.1 practica
```

## 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 adicionales, 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

```
C: > Users > nmcom > Desktop > SistemaDePuntoDeVentas.sql > ...
    ▷ Run | ⌂ Active: 127.0.0.1 | ⌂ punto_ventas
1  CREATE DATABASE punto_ventas;
    ▷ Run
2  USE punto_ventas;
3
4  -- Tabla de oficinas
    ▷ Run | ⌂ Select
5  CREATE TABLE oficinas (
6      id_oficina VARCHAR(10) PRIMARY KEY,
7      ciudad VARCHAR(50),
8      telefono VARCHAR(50),
9      direccion VARCHAR(50),
10     departamento VARCHAR(50),
11     pais VARCHAR(50),
12     codigoPostal VARCHAR(15),
13     continente VARCHAR(10),
14
15     tipo_oficina ENUM('Principal','Sucursal','Regional') NOT NULL,
16     activa BOOLEAN NOT NULL,
17     codigo_unico VARCHAR(20) UNIQUE
18 );
19
```

```
19
20  -- Tabla de empleados
    ▷ Run | ⌂ Select
21  CREATE TABLE empleados (
22      documento INT PRIMARY KEY,
23      apellido VARCHAR(50),
24      nombre VARCHAR(50),
25      extension VARCHAR(10),
26      email VARCHAR(100) UNIQUE,
27      id_oficina VARCHAR(10),
28      jefe INT,
29      cargo VARCHAR(50),
30
31      tipo_contrato ENUM('Fijo','Temporal','Servicios') NOT NULL,
32      activo BOOLEAN NOT NULL,
33
34      FOREIGN KEY (id_oficina) REFERENCES oficinas(id_oficina)
35 );
36
```

```
36
37  -- Tabla de clientes
    ▷ Run | ⌂ Select
38  CREATE TABLE clientes (
39      id_cliente INT PRIMARY KEY,
40      empresa VARCHAR(50),
41      apellido VARCHAR(50),
42      nombre VARCHAR(50),
43      telefono VARCHAR(50),
44      direccion VARCHAR(50),
45      ciudad VARCHAR(50),
46      departamento VARCHAR(50),
47      codigoPostal VARCHAR(15),
48      pais VARCHAR(50),
49      empleadoAtiende INT,
50      limiteCredito DOUBLE,
51
52      tipo_cliente ENUM('Minorista','Mayorista','Corporativo') NOT NULL,
53      activo BOOLEAN NOT NULL,
54      email VARCHAR(100) UNIQUE,
55
56      FOREIGN KEY (empleadoAtiende) REFERENCES empleados(documento)
57 );
58
```

```
58
59  -- Tabla de ordenes
60  ▷Run | Select
61  CREATE TABLE ordenes (
62      id_orden INT PRIMARY KEY,
63      fechaRecibido DATE,
64      fechaLimiteEntrega DATE,
65      fechaEntrega DATE,
66      estado VARCHAR(15),
67      observacion TEXT,
68      id_cliente INT,
69
70      prioridad ENUM('Alta','Media','Baja') NOT NULL,
71      facturada BOOLEAN NOT NULL,
72
73      FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)
74 );
75
```

```
74
75  --Tabla de detallesordenes
76  ▷Run | Select
77  CREATE TABLE detallesordenes (
78      id_orden INT,
79      id_producto VARCHAR(15),
80      cantidadPedida INT,
81      valorUnitario DOUBLE,
82      ordenEntrega INT,
83
84      tipo_descuento ENUM('Ninguno','Promocion','ClienteFrecuente') NOT NULL,
85      aplica_impuesto BOOLEAN NOT NULL,
86
87      PRIMARY KEY (id_orden, id_producto),
88      FOREIGN KEY (id_orden) REFERENCES ordenes(id_orden),
89      FOREIGN KEY (id_producto) REFERENCES productos(id_producto)
90 );
```

```
90
91  -- Tabla de lineasproductos
92  ▷Run | Select
93  ✓ CREATE TABLE lineasproductos (
94      id_lineaproducto INT PRIMARY KEY,
95      nombreLinea VARCHAR(50),
96      textoDescripcion VARCHAR(4000),
97      htmlDescripcion VARCHAR(200),
98      imagen VARCHAR(200),
99
100     categoria ENUM('Electrico','Manual','Industrial') NOT NULL,
101     disponible BOOLEAN NOT NULL,
102     codigo_linea VARCHAR(20) UNIQUE
103 );
```

```

104  -- Tabla de productos
105  ▷Run | Select
106  CREATE TABLE productos (
107      id_producto VARCHAR(15) PRIMARY KEY,
108      nombreProducto VARCHAR(70),
109      id_lineaProducto INT,
110      escala VARCHAR(10),
111      cantidad INT,
112      precioventa DOUBLE,
113      MSRP DOUBLE,
114
115      tipo_producto ENUM('Fisico','Digital','Servicio') NOT NULL,
116      activo BOOLEAN NOT NULL,
117      codigo_barra VARCHAR(50) UNIQUE,
118
119      FOREIGN KEY (id_lineaProducto) REFERENCES lineasproductos(id_lineaproducto)
120  );

```

```

120
121  -- Tabla de pagos
122  ▷Run | Select
123  CREATE TABLE pagos (
124      id_pago INT PRIMARY KEY,
125      id_cliente INT,
126      numeroFactura VARCHAR(50) UNIQUE,
127      fechaPago DATE,
128      totalPago DOUBLE,
129
130      metodo_pago ENUM('Efectivo','Tarjeta','Transferencia') NOT NULL,
131      confirmado BOOLEAN NOT NULL,
132
133      FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)
134  );

```

## Selects

SistemaDePuntoDeVentas.sql U    oficinas X

Properties DATA Log ER Monitor (1) (1) punto\_ventas

SELECT \* FROM oficinas LIMIT 100

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

SistemaDePuntoDeVentas.sql U    empleados X

Properties DATA Log ER Monitor (1) (1) punto\_ventas

SELECT \* FROM empleados LIMIT 100

	Q	Search Results	dокументo	apellido	nombre	extension	email	id_oficina	jefe	cargo	tipo_contrato	activo
>	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

	id_cliente	empresa	apellido	nombre	telefono	direccion	ciudad	departamento	codigoPostal	pais	empleadoAtende	limiteCredito	tipo_cliente	activo	email
>	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	Chiriquí	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	Penonomé	Cocle	0701	Panama	9	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	Aguaduque	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

	id_orden	fechaRecibido	fechaLimiteEntrega	fechaEntrega	estado	observacion	id_cliente	prioridad	facturada
>	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

	id_orden	id_producto	cantidadPedida	valorUnitario	ordenEntrega	tipo_descuento	aplica_impuesto
>	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

	id_lineaproducto	nombreLinea	textoDescripcion	htmlDescripcion	imagen	categoria	disponible	codigo_linea
>	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

	id_producto	nombreProducto	id_lineaproducto	escala	cantidad	precioventa	MSRP	tipo_producto	activo	codigo_barra
>	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	Tecnologia	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

	id_pago	id_cliente	numeroFactura	fechaPago	totalPago	metodo_pago	confirmado
1	1	1	F001	2026-01-08	1600	Tarjeta	1
2	2	2	F002	2026-01-09	100	Efectivo	1
3	3	3	F003	2026-01-10	150	Transferencia	1
4	4	4	F004	2026-01-11	360	Tarjeta	1
5	5	5	F005	2026-01-12	500	Efectivo	1
6	6	6	F006	2026-01-13	400	Transferencia	1
7	7	7	F007	2026-01-14	120	Tarjeta	1
8	8	8	F008	2026-01-15	90	Efectivo	1
9	9	9	F009	2026-01-16	150	Transferencia	1
10	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 separó los datos en entidades individuales (usuario, servicios, pago, auditoría y autenticación), básicamente cada tabla representa una sola entidad específica y cada atributo contiene su único valor.
- **2FN:** En el sistema tenía 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 está 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 garantizó que cada uno 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 redujeron las redundancias de datos, se aseguró que 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

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

```
    ▷Run | ⚡ Active: 127.0.0.1 | ⏺ punto_ventas
1  CREATE DATABASE empresa_servicios;
    ▷Run
2  USE empresa_servicios;
3
4  -- Tabla de usuarios
    ▷Run | ⏺ Select
5  CREATE TABLE usuario (
6      id_usuario INT AUTO_INCREMENT PRIMARY KEY,
7      nombre VARCHAR(50) NOT NULL,
8      apellido VARCHAR(50) NOT NULL,
9      email VARCHAR(100) NOT NULL UNIQUE,
10     telefono VARCHAR(15),
11     fecha_registro DATETIME NOT NULL,
12     estado ENUM('Activo','Inactivo') NOT NULL,
13     verificado BOOLEAN NOT NULL,
14     codigo_cliente VARCHAR(20) UNIQUE
15 );
16
```

```
6
7  -- Tabla de autenticacion
    ▷Run | ⏺ Select
8  CREATE TABLE autenticacion (
9      id_autenticacion INT AUTO_INCREMENT PRIMARY KEY,
0      username VARCHAR(50) NOT NULL UNIQUE,
1      password_hash VARCHAR(255) NOT NULL,
2      ultimo_login DATETIME,
3      intentos_fallidos INT DEFAULT 0 CHECK (intentos_fallidos >= 0),
4      tipo_usuario ENUM('Admin','Cliente') NOT NULL,
5      bloqueado BOOLEAN NOT NULL,
6      id_usuario INT NOT NULL UNIQUE,
7      FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario)
8 );
9
```

```
9
0  -- Tabla de servicio
    ▷Run | ⏺ Select
1  CREATE TABLE servicio (
2      id_servicio INT AUTO_INCREMENT PRIMARY KEY,
3      nombre_servicio VARCHAR(100) NOT NULL UNIQUE,
4      descripcion VARCHAR(255),
5      precio DECIMAL(10,2) NOT NULL CHECK (precio >= 0),
6      categoria ENUM('Streaming','Cloud','Soporte','Software') NOT NULL,
7      activo BOOLEAN NOT NULL,
8      codigo_servicio VARCHAR(20) UNIQUE
9 );
0
```

```

40
41    -- Tabla servicio_contratado
42    ▷Run | Select
43    CREATE TABLE servicio_contratado (
44        id_contrato INT AUTO_INCREMENT PRIMARY KEY,
45        fecha_contratacion DATE NOT NULL,
46        fecha_vencimiento DATE NOT NULL,
47        estado ENUM('Activo','Cancelado','Suspendido') NOT NULL,
48        renovacionAutomatica BOOLEAN NOT NULL,
49        numero_contrato VARCHAR(30) UNIQUE,
50        id_usuario INT NOT NULL,
51        id_servicio INT NOT NULL,
52        FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario),
53        FOREIGN KEY (id_servicio) REFERENCES servicio(id_servicio)
54    );

```

```

4
5    -- Tabla de pago
5    ▷Run | Select
6    CREATE TABLE pago (
7        id_pago INT AUTO_INCREMENT PRIMARY KEY,
8        fecha_pago DATETIME NOT NULL,
9        monto DECIMAL(10,2) NOT NULL CHECK (monto > 0),
10       metodo_pago ENUM('Tarjeta','Transferencia','PayPal') NOT NULL,
11       confirmado BOOLEAN NOT NULL,
12       numero_factura VARCHAR(30) UNIQUE,
13       id_contrato INT NOT NULL,
14       FOREIGN KEY (id_contrato) REFERENCES servicio_contratado(id_contrato)
15   );
16

```

```

--Tabla de autoridad_acceso
▷Run | Select
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

**Properties** **DATA** **Log** **ER** **Monitor** **{} { } { }** **empresa\_servicios**

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

Q	Search Results	Q	id_contrato	fecha_contratacion	fecha_vencimiento	estado	renovacion_automatica	numero_contrato	id_usuario	id_servicio
>	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		

**Properties** **DATA** **Log** **ER** **Monitor** **{} { } { }** **empresa\_servicios**

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

Q	Search Results	Q	id_usuario	nombre	apellido	email	telefono	fecha_registro	estado	verificado	codigo_cliente
>	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		

**Properties** **DATA** **Log** **ER** **Monitor** **{} { } { }** **empresa\_servicios**

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

Q	Search Results	Q	id_auditoria	fecha_hora	direccion_ip	tipo_evento	exito	codigo_evento	id_usuario
>	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		

**Properties** **DATA** **Log** **ER** **Monitor** **{} { } { }** **empresa\_servicios**

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

Q	Search Results	Q	id_autenticacion	username	password_hash	ultimo_login	intentos_fallidos	tipo_usuario	bloqueado	id_usuario
>	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

Properties DATA Log ER Monitor Empresa\_servicios

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

	Search Results	Export	Cost: 9ms	1	Total 10		
	id_pago	fecha_pago	monto	metodo_pago	confirmado	numero_factura	id_contrato
>	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

Properties DATA Log ER Monitor Empresa\_servicios

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

	Search Results	Export	Cost: 5ms	1	Total 10		
	id_servicio	nombre_servicio	descripcion	precio	categoria	activo	codigo_servicio
>	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 permanente	29.99	Soporte	1	SER-004
>	5	Antivirus	Protección empresarial	12.50	Software	1	SER-005
>	6	ERP Empresarial	Sistema ERP completo	99.99	Software	1	SER-006
>	7	Backup Online	Respaldo automático	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	Conexión 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.

```

158
159 --Opreaciones CRUD
160
161 -- CREATE Registra un nuevo usuario en la tabla de usuarios
162   ▷Run
163 INSERT INTO usuario (nombre, apellido, email, telefono, fecha_registro, estado, verificado, codigo_cliente)
164 VALUES ('Nuevo', 'Cliente', 'nuevo@empresa.com', '6000-2020', NOW(), 'Activo', 1, 'CLI-011');
165
166 -- READ Consulta usuarios activos, por ejemplo para análisis comercial
167   ▷Run | +Tab | JSON
168 ✓ SELECT nombre, apellido, email
169   FROM usuario
170   WHERE estado = 'Activo';
171
172 -- UPDATE desactiva usuario sin eliminarlo, por ejemplo para mantener historial de contratos
173   ▷Run | □Select
174 ✓ UPDATE usuario
175   SET estado = 'Inactivo'
176   WHERE id_usuario = 3;
177
178 --DELETE no elimina físicamente un usuario, sino que lo marca como inactivo para preservar integridad referencial
179   ▷Run | □Select
180 ✓ UPDATE usuario
181   SET estado = 'Inactivo'
182   WHERE id_usuario = 5;

```

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

# Creación y Uso de VISTAS SQL (15 pts)

## Requerimientos obligatorios

Crear **mínimo 3 VISTAS**, cumpliendo:

### 1. Vista de negocio

- o Consolidación de datos relevantes para toma de decisiones
- o Ejemplo: usuarios + servicios + pagos

### 2. Vista de seguridad

- o Oculta datos sensibles (contraseñas, tokens, etc.)

### 3. Vista de auditoría

- o Muestra accesos, fechas, estado de usuarios o transacciones

```
179
180    -- VISTAS SQLS
181
182    -- Vista de negocio / utilidad: analiza el ingreso por usuario y servicio
183    ▷Run | ▷Select
183    CREATE VIEW vista_negocio AS
184        SELECT
185            u.nombre,
186            u.apellido,
187            s.nombre_servicio,
188            p.monto,
189            p.fecha_pago
190        FROM usuario u
191        JOIN servicio_contratado sc ON u.id_usuario = sc.id_usuario
192        JOIN servicio s ON sc.id_servicio = s.id_servicio
193        JOIN pago p ON sc.id_contrato = p.id_contrato;
194
195    --Vista de seguridad / utilidad: muestra la informacion sin exponer la password_hash
196    ▷Run | ▷Select
196    CREATE VIEW vista_seguridad AS
197        SELECT
198            u.id_usuario,
199            u.nombre,
200            u.email,
201            a.username,
202            a.bloqueado
203        FROM usuario u
204        JOIN autenticacion a ON u.id_usuario = a.id_usuario;
205
206    -- Vista de auditoria / utilidad: monitorea los eventos sospechosos y tambien los accesos
207    ▷Run | ▷Select
207    CREATE VIEW vista_auditoria AS
208        SELECT
209            u.nombre,
210            a.fecha_hora,
211            a.tipo_evento,
212            a.exitoso
213        FROM auditoria_acceso a
214        JOIN usuario u ON a.id_usuario = u.id_usuario;
```

- **Vista de negocio / utilidad: analiza el ingreso 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** **vista\_auditoria**

Properties DATA Log ER Monitor { } ⏲ empresa\_servicios

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

Cost: 9ms < 1 > Total 10

	* nombre	* fecha_hora	* tipo_evento	* exitoso
	varchar(50)	datetime	enum('Login','Logout')	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** **vista\_negocio**

Properties DATA Log ER Monitor { } ⏲ empresa\_servicios

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

Cost: 10ms < 1 > Total 10

	* nombre	* apellido	* nombre_servicio	* monto	* fecha_pago
	varchar(50)	varchar(50)	varchar(100)	decimal(10,2)	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** **vista\_seguridad**

Properties DATA Log ER Monitor { } ⏲ empresa\_servicios

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

Cost: 11ms < 1 > Total 10

	* id_usuario	* nombre	* email	* username	* bloqueado
	int(11)	varchar(50)	varchar(100)	varchar(50)	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

## 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!*