

Parcial Bases de Datos - Informe de Práctica

Juan Pablo Escamilla Montilla (2420580-2724)

Fecha de Entrega: 19 de Octubre de 2025

Resumen

Para la resolución de este parcial, tendremos tres apartados principales: Visualización del esquema, Instrucciones utilizadas y Resultado Final. Cada una abarcará y explicará de manera clara y concisa los requisitos solicitados para esta entrega

1. Visualización del Esquema

Para la estructura de la base de datos utilizada por Servicios LiMar, tenemos el siguiente Diagrama Entidad-Relación, con su respectiva representación en el Modelo Relacional:

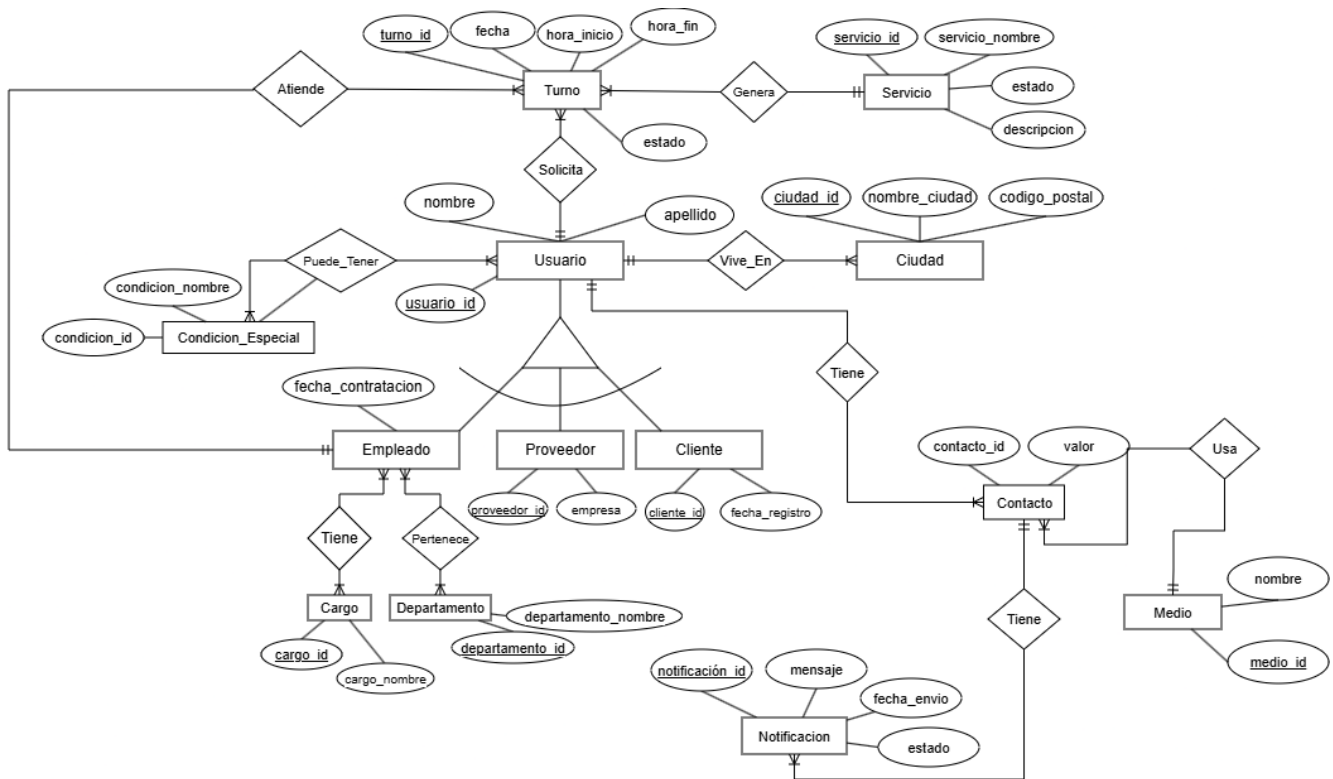


Figura 1: Diagrama Entidad-Relación

```

Ciudad(ciudad_id INT PK, nombre_ciudad STRING, codigo_postal STRING)
Usuario(usuario_id INT PK, nombre STRING, apellido STRING, ciudad_id INT FK)
Cliente(cliente_id INT PK, usuario_id INT FK, fecha_registro TIMESTAMP)
Empleado(empleado_id INT PK, usuario_id INT FK, fecha_contratacion TIMESTAMP)
Proveedor(proveedor_id INT PK, usuario_id INT FK, empresa STRING)
Condicion_Especial(condicion_id INT PK, condicion_nombre STRING)
Usuario_Condicion(usuario_id INT PK FK, condicion_id INT PK FK)
Cargo(cargo_id INT PK, cargo_nombre STRING)
Departamento(departamento_id INT PK, departamento_nombre STRING)
Empleado_Cargo(empleado_id INT PK FK, cargo_id INT PK FK)
Empleado_Departamento(empleado_id INT PK FK, departamento_id INT PK FK)
Servicio(servicio_id INT PK, servicio_nombre STRING, descripcion STRING, estado STRING)
Turno(turno_id INT PK, usuario_id INT FK, servicio_id INT FK, empleado_id INT FK, fecha TIMESTAMP, hora_inicio TIMESTAMP, hora_fin TIMESTAMP, estado STRING)
Medio(medio_id INT PK, nombre STRING)
Contacto(contacto_id INT PK, usuario_id INT FK, medio_id INT FK, valor STRING)
Notificacion(notificacion_id INT PK, contacto_id INT FK, mensaje STRING, fecha_envio TIMESTAMP, estado STRING)

```

Figura 2: Modelo Relacional

A continuación se explicarán las relaciones más relevantes que se decidieron mantener para esta base de datos:

1.1. Relaciones de Usuario

1.1.1. Ciudad/Usuario (1:N)

Una ciudad tiene muchos usuarios. Un usuario vive en una sola ciudad.

1.1.2. Usuario/Turno (1:N)

Un usuario puede solicitar muchos turnos. Cada turno es solicitado por un solo usuario.

1.1.3. Usuario/Contacto (1:N)

Un usuario tiene múltiples formas de ser contactado. Cada contacto pertenece a un solo usuario.

1.1.4. Usuario/Condición Especial (N:M)

Un usuario puede tener múltiples condiciones especiales. Una condición especial se puede aplicar a varios usuarios.

1.2. Relaciones de Empleado

1.2.1. Empleado/Turno (1:N)

Un empleado atiende muchos turnos. Cada turno es atendido por un solo empleado.

1.2.2. Empleado/Cargo (N:M)

Un empleado puede tener muchos cargos. Un cargo puede ser ocupado por muchos empleados.

1.2.3. Empleado/Departamento (N:M)

Un empleado puede pertenecer a varios departamentos. Un departamento tiene varios empleados.

1.3. Relaciones de Turno

1.3.1. Usuario/Turno (1:N)

Un usuario puede solicitar muchos turnos. Un turno es otorgado a un solo usuario.

1.3.2. Servicio/Turno (1:N)

Un servicio puede tener muchos turnos. Un turno consta de un solo servicio

1.3.3. Empleado/Turno (1:N)

Un empleado puede atender muchos turnos. Un turno es atendido por un solo empleado.

1.4. Relaciones con Notificaciones

1.4.1. Medio de Notificación/Contacto

Un medio de notificación define muchos contactos. Cada contacto usa solo un medio.

1.4.2. Contacto/Notificación

Un contacto específico recibe multiples notificaciones. Cada notificación se envía a un solo contacto.

2. Instrucciones utilizadas

Como se solicitó en el enunciado de la entrega, estaremos utilizando Docker para poder desplegar los contenedores de PostgreSQL y PGAdmin. En el caso del contenedor de PostgreSQL, el comando que se usó para desplegarlo fue:

```
docker run --rm --name parcial_bd_practica -e POSTGRES_USER=ulimar -e  
POSTGRES_PASSWORD=ex4men_db -p 5432:5432 postgres:12
```

Donde el nombre del contenedor es "**parcial_bd_practica**", el usuario de Postgres se definió como **ulimar**, y la contraseña como **ex4men_db**.

Por su parte, el comando utilizado para desplegar el contenedor de PGAdmin fue:

```
docker run --rm -p 5050:80 --link parcial_bd_practica:ex4men_db -e "  
PGADMIN_DEFAULT_EMAIL=usuario@servlimar.com" -e "PGADMIN_DEFAULT_PASSWORD=limar  
#123" -d dpape/pgadmin4
```

En este caso tenemos que las credenciales para ingresar a PGAdmin serán **usuario@servlimar.com** como el usuario, y **limar#123** como contraseña.

Además de eso, también se usaron los archivos **instrucciones_ddl.sql** y **instrucciones_dml.sql** para poder crear las tablas y los registros que estarán en ellas. Dichos archivos se pueden encontrar dentro del [repositorio de GitHub](#)

3. Resultado Final

Como todos los pasos se realizaron correctamente, podemos ver que, en este caso, en la tabla de usuarios, los registros se almacenan correctamente, cosa que pasa con las demás tablas.

	usuario_id [PK] integer 	nombre character varying (100) 	apellido character varying (100) 	ciudad_id integer 
1	1	Juan	Pérez	1
2	2	María	González	1
3	3	Carlos	Rodríguez	2
4	4	Ana	Martínez	3
5	5	Luis	López	1
6	6	Laura	Hernández	4
7	7	Pedro	García	5
8	8	Carmen	Díaz	2
9	9	Jorge	Ramírez	6
10	10	Sofía	Torres	3

Figura 3: Registros de la tabla Usuario