Tarea 1.3: Normalización de las tablas:

a) Aplicar las reglas de normalización (primera, segunda y tercera forma normal) para asegurar la consistencia y reducir la redundancia de los datos.

   La normalización se aplicó a las tablas del sistema para reducir redundancia, mejorar la integridad de los datos y optimizar las consultas. Se aplicaron las tres primeras formas normales de la siguiente manera:

   Primera Forma Normal (1FN)

   Para que una tabla cumpla con la 1FN, debe: no contener grupos repetitivos ni valores multivaluado, tener un identificador único (clave primaria) y asegurar que todos los valores en una columna sean atómicos (no divisibles).

Ejemplo aplicado: la tabla Recurso (Personal) tenía originalmente un campo Habilidades, donde se almacenaban múltiples habilidades en un solo campo (ejemplo: "JavaScript, SQL, Python"). Esto viola la 1FN.

Solución: Se creó una nueva tabla Habilidad y una tabla intermedia Recurso_Habilidad para normalizar la relación N:M.

Antes (No cumple 1FN):

```
CREATE TABLE Recurso (
   ID_Recurso SERIAL PRIMARY KEY,
   Nombre VARCHAR(50),
   Apellidos VARCHAR(50),
   Rol VARCHAR(50),
   Disponibilidad BOOLEAN,
   Habilidades TEXT  -- Almacena múltiples valores en una sola columna (incorrecto)
);
```

Después de normalizar (Cumple 1FN):

```
CREATE TABLE Habilidad (
   ID_Habilidad SERIAL PRIMARY KEY,
   Nombre VARCHAR(50) UNIQUE
);
CREATE TABLE Recurso_Habilidad (
   ID_Recurso INT REFERENCES Recurso(ID_Recurso),
```

```
    ID_Habilidad INT REFERENCES Habilidad(ID_Habilidad),
    PRIMARY KEY (ID_Recurso, ID_Habilidad)
);
```

Ahora cada habilidad se almacena de forma atómica en una tabla separada, eliminando la redundancia.

Segunda Forma Normal (2FN)

Para que una tabla esté en 2FN, debe: Cumplir con la 1FN y no tener dependencias parciales (es decir, cada atributo debe depender completamente de la clave primaria y no solo de parte de ella en caso de claves compuestas).

Ejemplo aplicado: La tabla Registro_Tiempo tenía las siguientes columnas:

Antes (No cumple 2FN):

```
CREATE TABLE Registro_Tiempo (
    ID_Registro SERIAL PRIMARY KEY,
    Tarea_ID INT REFERENCES Tarea(ID_Tarea),
    Recurso_ID INT REFERENCES Recurso(ID_Recurso),
    Horas INT,
    Fecha DATE,
    Nombre_Recurso VARCHAR(100)  -- Depende solo de Recurso_ID (incorrecto)
);
```

- Problema: Nombre_Recurso depende de Recurso_ID, no de toda la clave primaria.
- Solución: Eliminar Nombre_Recurso, ya que esta información ya está en la tabla Recurso.

Después (Cumple 2FN):

```
CREATE TABLE Registro_Tiempo (
    ID_Registro SERIAL PRIMARY KEY,
    Tarea_ID INT REFERENCES Tarea(ID_Tarea),
    Recurso_ID INT REFERENCES Recurso(ID_Recurso),
    Horas INT,
```

```
    Fecha DATE
);
```

Tercera Forma Normal (3FN)

Para que una tabla esté en 3FN, debe: cumplir con la 2FN y no tener dependencias transitivas (es decir, los atributos no clave deben depender solo de la clave primaria y no de otros atributos no clave).

Ejemplo aplicado: La tabla Cliente tenía un campo Razon_Social que dependía de Nombre, lo que representa una dependencia transitiva.

Antes (No cumple 3FN)

```
CREATE TABLE Cliente (
    ID_Cliente SERIAL PRIMARY KEY,
    Nombre VARCHAR(100),
    Razon_Social VARCHAR(150),  -- Depende de Nombre (incorrecto)
    Contacto VARCHAR(100),
    Telefono VARCHAR(20),
    Correo_Electronico VARCHAR(100)
);
```

Solución: Crear una nueva tabla Razon_Social y referenciarla desde Cliente.

```
Después (Cumple 3FN):
CREATE TABLE Razon_Social (
    ID_Razon SERIAL PRIMARY KEY,
    Descripcion VARCHAR(150) UNIQUE
);
CREATE TABLE Cliente (
    ID_Cliente SERIAL PRIMARY KEY,
    Nombre VARCHAR(100),
    ID_Razon INT REFERENCES Razon_Social(ID_Razon),
    Contacto VARCHAR(100),
    Telefono VARCHAR(20),
    Correo_Electronico VARCHAR(100)
);
```

Aplicando las tres formas normales, logre:

- Eliminar las redundancias al separar datos en tablas especializadas.
- Mejorar la integridad de los datos asegurando dependencias claras.
- Optimizar las consultas al reducir el almacenamiento innecesario.

Estructura final de la base de datos (cada una de las 12 tablas)

```
CREATE TABLE "public"."cliente" (
  "id_cliente" int4 NOT NULL DEFAULT nextval('cliente_id_cliente_seq'::regclass),
  "nombre" varchar(200) COLLATE "pg_catalog"."default" NOT NULL,
  "id_razon" int4
);

CREATE TABLE "public"."clientes" (
  "id_cliente" int4 NOT NULL DEFAULT nextval('clientes_id_cliente_seq'::regclass),
  "nombre" varchar(255) COLLATE "pg_catalog"."default" NOT NULL,
  "razon_social" varchar(255) COLLATE "pg_catalog"."default",
  "contacto_principal" varchar(255) COLLATE "pg_catalog"."default" NOT NULL,
  "telefono" varchar(20) COLLATE "pg_catalog"."default",
  "correo" varchar(255) COLLATE "pg_catalog"."default" NOT NULL
);

CREATE TABLE "public"."comunicaciones" (
  "id_mensaje" int4 NOT NULL DEFAULT nextval('comunicaciones_id_mensaje_seq'::regclass),
  "id_remitente" int4 NOT NULL,
  "id_destinatario" int4 NOT NULL,
  "mensaje" text COLLATE "pg_catalog"."default" NOT NULL,
  "fecha_envio" timestamp(6) DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE "public"."documentos" (
  "id_documento" int4 NOT NULL DEFAULT nextval('documentos_id_documento_seq'::regclass),
  "nombre" varchar(255) COLLATE "pg_catalog"."default" NOT NULL,
  "id_proyecto" int4 NOT NULL,
  "version" int4 DEFAULT 1,
  "url" text COLLATE "pg_catalog"."default" NOT NULL
);
```

```
CREATE TABLE "public"."proyecto_recurso" (
  "id_proyecto" int4 NOT NULL,
  "id_recurso" int4 NOT NULL
);

CREATE TABLE "public"."proyectos" (
  "id_proyecto" int4 NOT NULL DEFAULT nextval('proyectos_id_proyecto_seq'::regclass),
  "nombre" varchar(255) COLLATE "pg_catalog"."default" NOT NULL,
  "descripcion" text COLLATE "pg_catalog"."default",
  "id_cliente" int4 NOT NULL,
  "fecha_inicio" date NOT NULL,
  "fecha_fin" date,
  "presupuesto" numeric(15,2),
  "estado" varchar(50) COLLATE "pg_catalog"."default"
);

CREATE TABLE "public"."razon_social" (
  "id_razon" int4 NOT NULL DEFAULT nextval('razon_social_id_razon_seq'::regclass),
  "descripcion" varchar(150) COLLATE "pg_catalog"."default" NOT NULL
);

CREATE TABLE "public"."recurso" (
  "id_recurso" int4 NOT NULL DEFAULT nextval('recurso_id_recurso_seq'::regclass),
  "nombre" varchar(100) COLLATE "pg_catalog"."default" NOT NULL,
  "apellido" varchar(100) COLLATE "pg_catalog"."default" NOT NULL,
  "rol" varchar(50) COLLATE "pg_catalog"."default" NOT NULL
);

CREATE TABLE "public"."recursos" (
  "id_recurso" int4 NOT NULL DEFAULT nextval('recursos_id_recurso_seq'::regclass),
  "nombre" varchar(255) COLLATE "pg_catalog"."default" NOT NULL,
  "apellidos" varchar(255) COLLATE "pg_catalog"."default" NOT NULL,
  "rol" varchar(100) COLLATE "pg_catalog"."default" NOT NULL,
  "disponibilidad" bool DEFAULT true,
  "habilidades" text COLLATE "pg_catalog"."default"
);

CREATE TABLE "public"."registro_tiempo" (
  "id_registro" int4 NOT NULL DEFAULT nextval('registro_tiempo_id_registro_seq'::regclass),
```

```
  "id_tarea" int4,
  "id_recurso" int4,
  "horas" int4 NOT NULL,
  "fecha" date NOT NULL DEFAULT CURRENT_DATE
);

CREATE TABLE "public"."seguimiento_horas" (
  "id_registro" int4 NOT NULL DEFAULT nextval('seguimiento_horas_id_registro_seq'::regclass),
  "id_recurso" int4 NOT NULL,
  "id_tarea" int4 NOT NULL,
  "horas_trabajadas" numeric(5,2) NOT NULL,
  "fecha_registro" date NOT NULL DEFAULT CURRENT_DATE
);

CREATE TABLE "public"."tareas" (
  "id_tarea" int4 NOT NULL DEFAULT nextval('tareas_id_tarea_seq'::regclass),
  "nombre" varchar(255) COLLATE "pg_catalog"."default" NOT NULL,
  "descripcion" text COLLATE "pg_catalog"."default",
  "id_proyecto" int4 NOT NULL,
  "id_responsable" int4,
  "fecha_inicio" date NOT NULL,
  "fecha_fin" date,
  "estado" varchar(50) COLLATE "pg_catalog"."default",
  "prioridad" varchar(20) COLLATE "pg_catalog"."default"
);
```