

```

-- Elimina las tablas si existen
DROP TABLE IF EXISTS usuarios CASCADE;
DROP TABLE IF EXISTS roles CASCADE;

-- Habilita la extensión pgcrypto para generar UUIDs
CREATE EXTENSION IF NOT EXISTS pgcrypto;

-- Tabla de roles
CREATE TABLE roles(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    nombre_rol VARCHAR(50) NOT NULL
);

-- Tabla de usuarios (cada usuario tiene un solo rol)
CREATE TABLE usuarios (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    id_rol UUID NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    correo VARCHAR(100) NOT NULL UNIQUE,
    estado BOOLEAN NOT NULL,
    celular VARCHAR(20),
    fecha_creacion DATE DEFAULT CURRENT_DATE,
    fecha_ultimo_login DATE,
    FOREIGN KEY (id_rol) REFERENCES roles(id)
);

-- Tabla de carreras
CREATE TABLE carreras (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    nombre VARCHAR(100) NOT NULL
);

-- Tabla de materias (relación muchos a uno con carreras)
CREATE TABLE materias (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    nombre VARCHAR(100) NOT NULL,
    id_carrera UUID NOT NULL,
    FOREIGN KEY (id_carrera) REFERENCES carreras(id)
);

-- Tabla de puntaje (ejemplo de relación usuario - materia)
CREATE TABLE puntajes (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    id_usuario UUID NOT NULL,
    id_materia UUID NOT NULL,

```

```
puntaje NUMERIC(10,2) NOT NULL,  
fecha_asignacion DATE DEFAULT CURRENT_DATE,  
FOREIGN KEY (id_usuario) REFERENCES usuarios(id),  
FOREIGN KEY (id_materia) REFERENCES materias(id)  
);
```

```
INSERT INTO roles (id, nombre_rol) VALUES  
(gen_random_uuid(), 'admin');
```

```
INSERT INTO roles (id, nombre_rol) VALUES  
(gen_random_uuid(), 'profesor');
```

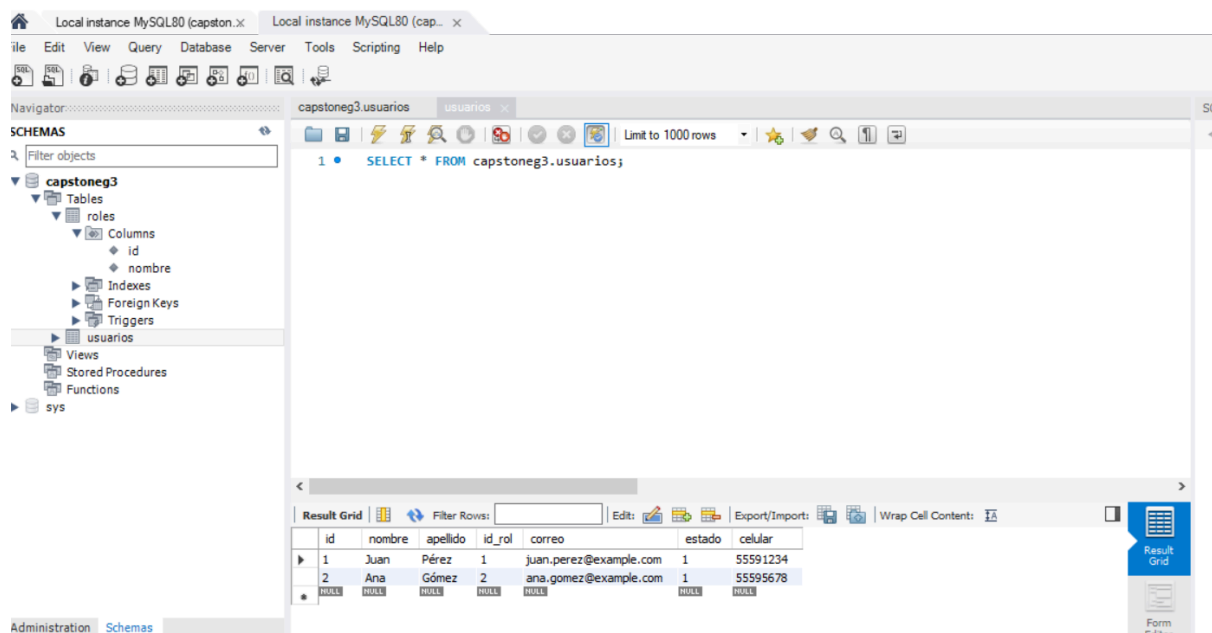
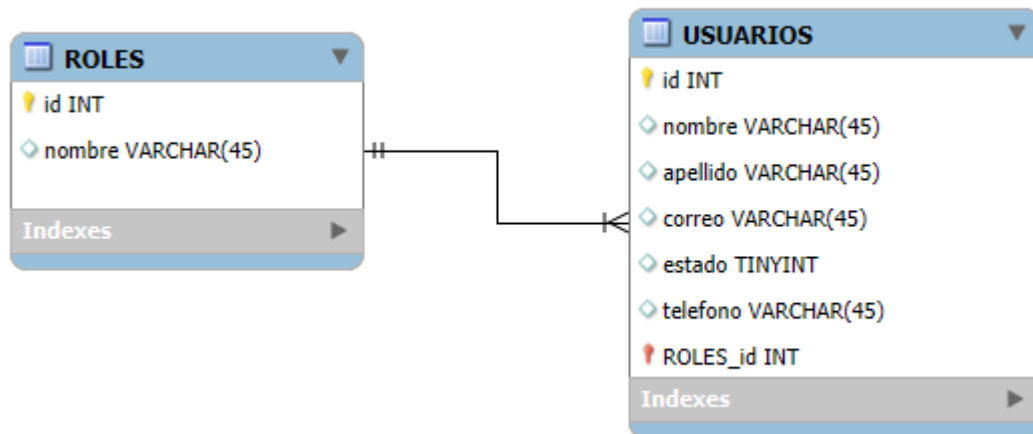
```
INSERT INTO roles (id, nombre_rol) VALUES  
(gen_random_uuid(), 'alumno');
```

Explicación diagrama:

1. Relación entre usuarios y roles (relación muchos a uno) → Un usuario tiene solo un rol. Un rol puede ser asignado a muchos usuarios.
2. Relación materias y usuarios → (relación muchos a uno) → Varias materias pueden pertenecer a una misma carrera.
3. Relación entre puntajes, usuarios y materias (relación muchos a muchos ? por confirmar con el profe) → Un usuario puede tener puntajes en muchas materias, y una materia puede tener puntajes de muchos usuarios. La tabla puntajes actúa como una tabla intermedia o de unión para conectar a los usuarios con las materias

Comando para crear la bbdd que ocupé en el pc de la pega →

```
docker run --name capstone-g3 \  
-e POSTGRES_PASSWORD=capstone-g3 \  
-e POSTGRES_USER=capstone-g3 \  
-e POSTGRES_DB=capstone-g3 \  
-p 5432:5432 \  
-v pg_data:/var/lib/postgresql/data \  
-d postgres
```



1.Características de la MV (Google Cloud Compute Engine)

- **Proveedor:** Google Cloud Platform (GCP)
- **Servicio:** Compute Engine
- **Nombre de la Instancia:** server-capstone-g3
- **Tipo de Máquina:** e2-micro. Esta instancia de nivel gratuito está configurada con **2 vCPUs** y **1 GB de RAM**, ideal para entornos de desarrollo y pruebas con baja carga de trabajo.
- **Sistema Operativo:** Distribución de Linux (Debian/Ubuntu).
- **Región y Zona:** us-central1-a
- **Disco de Arranque:** Un disco persistente con una capacidad de hasta 30 GB.
- **Dirección IP Externa:** **35.223.70.199**. Esta dirección IP es estática y sirve como punto de acceso público para todos los servicios del servidor.

2. Configuración de la Red y Firewall

Las reglas del firewall de GCP se han configurado para permitir el tráfico entrante a los servicios esenciales del servidor:

- **Puerto 80 (HTTP):** Abierto para el servidor web Nginx. Permite el acceso al HTML de prueba creado.
- **Puerto 5432 (PostgreSQL):** Abierto para conexiones entrantes desde cualquier dirección IPv4 (0.0.0.0/0). Esta configuración es adecuada para desarrollo, pero se recomienda restringirla en entornos de producción.

3. Configuración de la Base de Datos (PostgreSQL)

- **Motor:** PostgreSQL
- **Versión:** 15.13

Método de Inicio: El servicio no se gestiona con systemd debido a un error en el script de inicio. Se debe iniciar manualmente con el siguiente comando SSH:

Bash

```
sudo -u postgres /usr/lib/postgresql/15/bin/pg_ctl start -D /var/lib/postgresql/15/main
```

-
- **Usuario:** postgres
- **Contraseña:** admin (configurada para pruebas). Se recomienda una contraseña fuerte en producción.
- **Configuración de Conexión Remota:**
 - `postgresql.conf`: La directiva `listen_addresses` se ha establecido en `'*'` para aceptar conexiones de cualquier IP.
 - `pg_hba.conf`: Se ha añadido una entrada de host (`host all all 0.0.0.0/0 md5`) que permite conexiones de todos los usuarios a todas las bases de datos con autenticación de contraseña (md5).
- **Estructura de la Base de Datos:**
 - **Tablas:** roles y usuarios
 - **Relaciones:** La tabla usuarios contiene una clave foránea (`id_rol`) que hace referencia a la clave primaria de la tabla roles, estableciendo una relación de muchos a uno.

4. Instrucciones para la Conexión

Para conectarse a la base de datos desde un cliente externo (como `psql`, DBeaver o pgAdmin), se deben usar los siguientes parámetros:

- **Host/Dirección IP:** 35.223.70.199
- **Puerto:** 5432
- **Usuario:** postgres
- **Contraseña:** admin

El comando de ejemplo para la conexión desde la línea de comandos de Windows es:

```
"C:\Program Files\PostgreSQL\15\bin\psql.exe" -h 35.223.70.199 -p 5432 -U postgres
```

```
+++++
```

Versión 2 de la base de datos a la espera que el profe la destruya
jeje

```
-- Extensión para generar UUIDs
```

```
CREATE EXTENSION IF NOT EXISTS "uuid-oss";
```

```
--- Tablas de Ubicación e Institución
```

```
CREATE TABLE paises (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL UNIQUE  
);
```

```
CREATE TABLE regiones (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL UNIQUE,  
  id_pais UUID NOT NULL,  
  FOREIGN KEY (id_pais) REFERENCES paises(id)  
);
```

```
CREATE TABLE comunas (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL UNIQUE,  
  id_region UUID NOT NULL,  
  FOREIGN KEY (id_region) REFERENCES regiones(id)  
);
```

```
CREATE TABLE instituciones (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(255) NOT NULL UNIQUE,  
  id_comuna UUID,  
  FOREIGN KEY (id_comuna) REFERENCES comunas(id)  
);
```

```
--
```

-- Tablas de Roles y Contenido

--

```
CREATE TABLE roles (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(50) NOT NULL UNIQUE  
);
```

```
CREATE TABLE carreras (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL,  
  id_institucion UUID NOT NULL,  
  FOREIGN KEY (id_institucion) REFERENCES instituciones(id)  
);
```

```
CREATE TABLE semestres (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(50) NOT NULL UNIQUE  
);
```

```
CREATE TABLE asignaturas (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL,  
  id_carrera UUID NOT NULL,  
  FOREIGN KEY (id_carrera) REFERENCES carreras(id)  
);
```

-- Tabla intermedia para la relación N:M entre Asignatura y Semestre

```
CREATE TABLE asignaturas_semestre (  
  id_asignatura UUID NOT NULL,  
  id_semestre UUID NOT NULL,  
  PRIMARY KEY (id_asignatura, id_semestre),  
  FOREIGN KEY (id_asignatura) REFERENCES asignaturas(id),  
  FOREIGN KEY (id_semestre) REFERENCES semestres(id)  
);
```

--

-- Tablas de Usuarios

--

```
CREATE TABLE usuarios (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL,  
  apellido VARCHAR(100) NOT NULL,  
  id_rol UUID NOT NULL,  
  id_carrera UUID NOT NULL,  
  password_hash VARCHAR(255) NOT NULL,  
  correo VARCHAR(100) NOT NULL UNIQUE,  
  estado BOOLEAN NOT NULL DEFAULT TRUE,  
  celular VARCHAR(20),  
  fecha_creacion TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
```

```

    fecha_ultimo_login TIMESTAMP,
    FOREIGN KEY (id_rol) REFERENCES roles(id),
    FOREIGN KEY (id_carrera) REFERENCES carreras(id)
);
--
-- Tablas de Carga y Auditoría
--
CREATE TABLE estados_carga (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    nombre_estado VARCHAR(50) NOT NULL UNIQUE
);
CREATE TABLE tipos_carga (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    nombre_tipo VARCHAR(100) NOT NULL UNIQUE
);
CREATE TABLE cargas (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    id_usuario_carga UUID NOT NULL,
    fecha_hora_carga TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    nombre_archivo VARCHAR(255) NOT NULL,
    id_estado UUID NOT NULL,
    detalle_error TEXT,
    id_tipo_carga UUID NOT NULL,
    FOREIGN KEY (id_usuario_carga) REFERENCES usuarios(id),
    FOREIGN KEY (id_estado) REFERENCES estados_carga(id),
    FOREIGN KEY (id_tipo_carga) REFERENCES tipos_carga(id)
);
--
-- Tablas de Gamificación
--
CREATE TABLE preguntas (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    texto TEXT NOT NULL,
    respuesta_correcta TEXT NOT NULL,
    tema VARCHAR(100),
    id_asignatura UUID NOT NULL,
    FOREIGN KEY (id_asignatura) REFERENCES asignaturas(id)
);
CREATE TABLE juegos (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    id_usuario UUID NOT NULL,
    id_asignatura UUID NOT NULL,
    fecha_inicio TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    fecha_fin TIMESTAMP,

```

```

    puntaje NUMERIC(10, 2),
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id),
    FOREIGN KEY (id_asignatura) REFERENCES asignaturas(id)
);
CREATE TABLE puntajes (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    id_usuario UUID NOT NULL,
    id_asignatura UUID NOT NULL,
    puntaje NUMERIC(10, 2) NOT NULL,
    fecha_asignacion DATE NOT NULL DEFAULT CURRENT_DATE,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id),
    FOREIGN KEY (id_asignatura) REFERENCES asignaturas(id)
);
CREATE TABLE ranking (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    id_usuario UUID NOT NULL,
    id_asignatura UUID NOT NULL,
    puntaje_total NUMERIC(10, 2) NOT NULL,
    posicion INT NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id),
    FOREIGN KEY (id_asignatura) REFERENCES asignaturas(id)
);
CREATE TABLE metricas (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    id_usuario UUID NOT NULL,
    id_pregunta UUID NOT NULL,
    respuesta_correcta BOOLEAN NOT NULL,
    tiempo_respuesta_ms INT,
    fecha_hora TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id),
    FOREIGN KEY (id_pregunta) REFERENCES preguntas(id)
);

```

Script Actualizado 23-10-2025

-- Extensión para generar UUIDs

CREATE EXTENSION IF NOT EXISTS "uuid-oss";

-- Se elimina la tabla ranking ya que su cálculo se realizará por el Scoring Service

DROP TABLE IF EXISTS ranking **CASCADE**;

--- Tablas de Ubicación e Institución

```
CREATE TABLE paises (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL UNIQUE  
);  
CREATE TABLE regiones (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL UNIQUE,  
  id_pais UUID NOT NULL,  
  FOREIGN KEY (id_pais) REFERENCES paises(id)  
);  
CREATE TABLE comunas (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL UNIQUE,  
  id_region UUID NOT NULL,  
  FOREIGN KEY (id_region) REFERENCES regiones(id)  
);  
CREATE TABLE instituciones (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(255) NOT NULL UNIQUE,  
  id_comuna UUID,  
  FOREIGN KEY (id_comuna) REFERENCES comunas(id)  
);
```

--- Tablas de Roles y Contenido

```
CREATE TABLE roles (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(50) NOT NULL UNIQUE  
);  
CREATE TABLE carreras (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL,  
  id_institucion UUID NOT NULL,  
  FOREIGN KEY (id_institucion) REFERENCES instituciones(id),  
  UNIQUE (nombre, id_institucion)  
);  
CREATE TABLE semestres (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(50) NOT NULL UNIQUE  
);  
CREATE TABLE asignaturas (  
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),  
  nombre VARCHAR(100) NOT NULL,  
  id_carrera UUID NOT NULL,  
  FOREIGN KEY (id_carrera) REFERENCES carreras(id)  
);  
-- Tabla intermedia para la relación N:M entre Asignatura y Semestre  
CREATE TABLE asignaturas_semestre (  
  id_asignatura UUID NOT NULL,  
  id_semestre UUID NOT NULL,
```

```

PRIMARY KEY (id_asignatura, id_semestre),
FOREIGN KEY (id_asignatura) REFERENCES asignaturas(id),
FOREIGN KEY (id_semestre) REFERENCES semestres(id)
);

```

--- Tablas de Usuarios

```

CREATE TABLE usuarios (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  nombre VARCHAR(100) NOT NULL,
  apellido VARCHAR(100) NOT NULL,
  RUT VARCHAR(9) NOT NULL UNIQUE, -- Añadido UNIQUE para el RUT
  id_rol UUID NOT NULL,
  id_carrera UUID NOT NULL,
  password_hash VARCHAR(255) NOT NULL,
  correo VARCHAR(100) NOT NULL UNIQUE,
  estado BOOLEAN NOT NULL DEFAULT TRUE,
  celular VARCHAR(20),
  fecha_creacion TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  fecha_ultimo_login TIMESTAMP,
  FOREIGN KEY (id_rol) REFERENCES roles(id),
  FOREIGN KEY (id_carrera) REFERENCES carreras(id)
);

```

--- Tablas de Carga y Auditoría

```

CREATE TABLE estados_carga (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  nombre_estado VARCHAR(50) NOT NULL UNIQUE
);
CREATE TABLE tipos_carga (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  nombre_tipo VARCHAR(100) NOT NULL UNIQUE
);
CREATE TABLE cargas (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  id_usuario_carga UUID NOT NULL,
  fecha_hora_carga TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  nombre_archivo VARCHAR(255) NOT NULL,
  id_estado UUID NOT NULL,
  detalle_error TEXT,
  id_tipo_carga UUID NOT NULL,
  FOREIGN KEY (id_usuario_carga) REFERENCES usuarios(id),
  FOREIGN KEY (id_estado) REFERENCES estados_carga(id),
  FOREIGN KEY (id_tipo_carga) REFERENCES tipos_carga(id)
);

```

--- Tablas de Gamificación

```

CREATE TABLE temas (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  nombre VARCHAR(100) NOT NULL,
  id_asignatura UUID NOT NULL,
  FOREIGN KEY (id_asignatura) REFERENCES asignaturas(id),
  UNIQUE (nombre, id_asignatura)
);

```

```
);
CREATE TABLE preguntas (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  texto TEXT NOT NULL,
  respuesta_correcta TEXT NOT NULL,
  id_asignatura UUID NOT NULL,
  FOREIGN KEY (id_asignatura) REFERENCES asignaturas(id)
);
-- MODIFICACIONES A LA TABLA 'preguntas' para enlazar con 'temas'
ALTER TABLE preguntas
DROP COLUMN tema;
ALTER TABLE preguntas
ADD COLUMN id_tema UUID NOT NULL,
ADD FOREIGN KEY (id_tema) REFERENCES temas(id);
```

```
CREATE TABLE conceptos (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  palabra_concepto VARCHAR(255) NOT NULL,
  id_tema UUID NOT NULL,
  FOREIGN KEY (id_tema) REFERENCES temas(id)
);
CREATE TABLE juegos (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  id_usuario UUID NOT NULL,
  id_asignatura UUID NOT NULL,
  fecha_inicio TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  fecha_fin TIMESTAMP,
  puntaje NUMERIC(10, 2),
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id),
  FOREIGN KEY (id_asignatura) REFERENCES asignaturas(id)
);
```

X

-- Modificación de la tabla 'juegos'

```
ALTER TABLE juegos
ADD COLUMN nombre_juego VARCHAR(50),
ADD COLUMN intentos_restantes INT,
ADD COLUMN estado_partida VARCHAR(50) NOT NULL DEFAULT 'En curso';
```

```
CREATE TABLE puntajes (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  id_usuario UUID NOT NULL,
  id_asignatura UUID NOT NULL,
  puntaje NUMERIC(10, 2) NOT NULL,
  fecha_asignacion DATE NOT NULL DEFAULT CURRENT_DATE,
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id),
  FOREIGN KEY (id_asignatura) REFERENCES asignaturas(id)
);
CREATE TABLE metricas (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  id_usuario UUID NOT NULL,
  id_pregunta UUID NOT NULL,
```

```
respuesta_correcta BOOLEAN NOT NULL,  
tiempo_respuesta_ms INT,  
fecha_hora TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
FOREIGN KEY (id_usuario) REFERENCES usuarios(id),  
FOREIGN KEY (id_pregunta) REFERENCES preguntas(id)  
);
```