



BASE DE DATOS

75.15-95.05-TA044 - BASE DE DATOS - CÁTEDRA: JOSÉ LUIS CABRERA

JTP: ADRIÁN SERGIO BERNARDI

Trabajo Práctico 1

Trabajo práctico por equipos



23 de junio de 2025

Scopel Nicolas Ignacio
111305

Gustavo Scilingo
110590

Ivan fuschetto
110632

José Patty
109843

Diagrama MER: Entidades, Relaciones y Cardinalidades

Aclaraciones sobre las decisiones tomadas en el MER.

- Las relaciones con el nombre GUARDA hacen referencia a que el usuario agrega a favoritos ya sea una canción, artista o álbum.

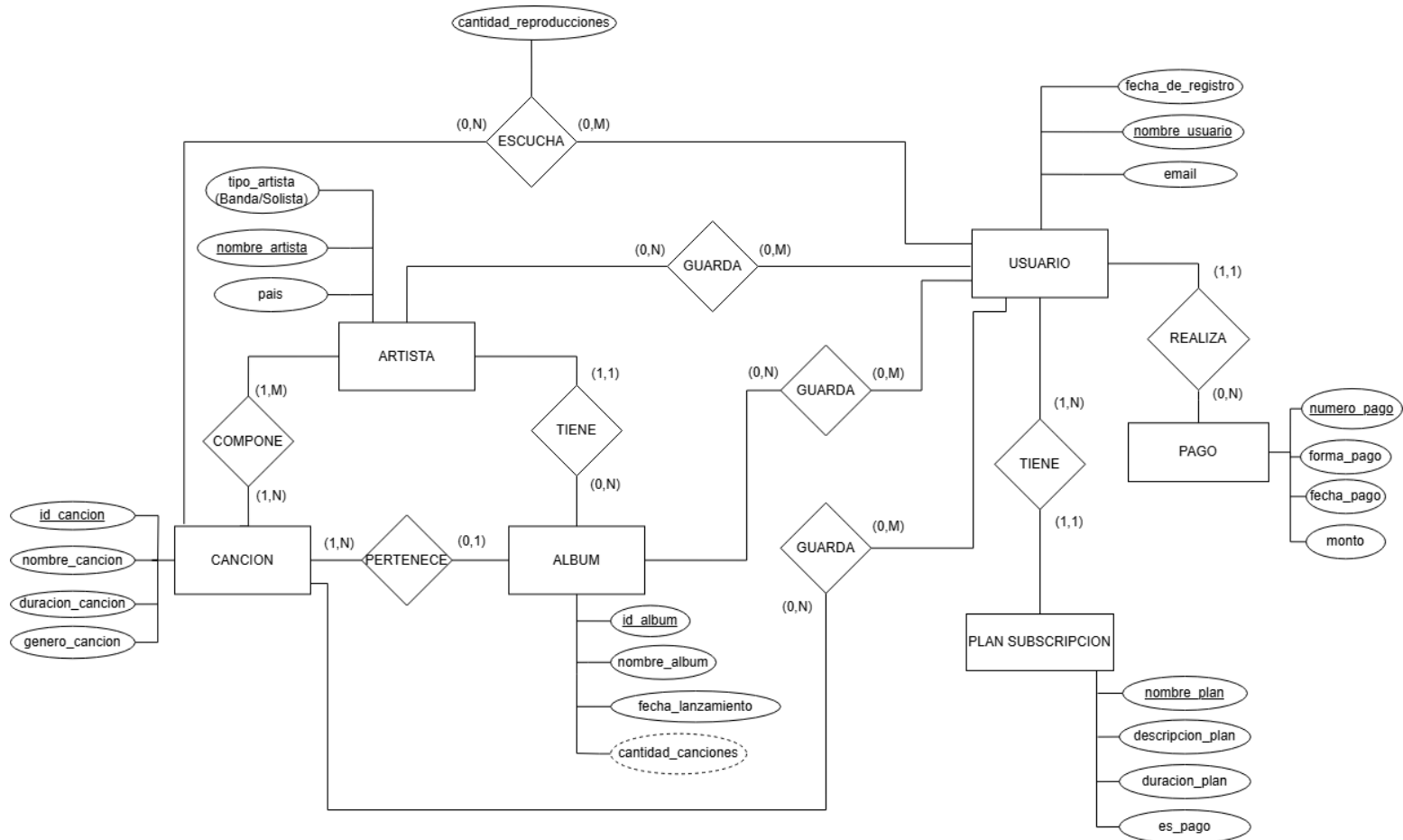


Figura 1: Diagrama MER.

Pasaje del Modelo Entidad-Relación al Modelo Relacional

La transformación del modelo Entidad-Relación (MER) al modelo relacional sigue principios fundamentales: cada entidad se convierte en una tabla, las relaciones N:M se implementan con tablas intermedias que contienen las claves primarias de ambas entidades, y en relaciones 1:N la clave primaria de la entidad del lado “1” se agrega como clave ajena en la tabla del lado “N”.

A continuación se describen las tablas resultantes para cada entidad y relación, indicando sus claves candidatas, clave primaria y claves foráneas.

1. Entidades y tablas básicas

1. **planes_de_subscripción**: {nombre_plan, descripcion_plan, duracion_plan, es_pago, precio}

- Entidad MER: Planes de Subscripción.
- Atributos: **nombre_plan**, **descripcion_plan**, **duracion_plan**, **es_pago**, **precio**.
- Claves candidatas: **nombre_plan**.
- Clave primaria: **nombre_plan**.

Justificación: Cada entidad se convierte en una tabla.

En este caso, **nombre_plan** identifica de forma única cada plan (clave candidata), por lo que se elige como clave primaria. El resto de atributos de la entidad pasan a columnas de la tabla. Se agrega una restricción **CHECK** para garantizar que los planes de pago tengan un precio válido mínimo y los planes gratuitos tengan precio 0, siguiendo la regla de negocio.

2. **usuarios**: {nombre_usuario, email, fecha_de_registro, nombre_plan }

- Entidad MER: Usuarios.
- Atributos: **nombre_usuario**, **email**, **fecha_de_registro**, **nombre_plan**.
- Claves candidatas: **nombre_usuario**, **email**.
- Clave primaria: **nombre_usuario**.
- Clave foránea: **nombre_plan** → **planes_subscripcion(nombre_plan)**

Justificación: La clave primaria de **usuarios** es **nombre_usuario**, que es mínima y única. Se agrega una restricción **UNIQUE** a **email** para que también sea única y además, es necesario que cada usuario tenga un email registrado. La relación 1:N entre Planes de Subscripción y Usuarios se implementa propagando la clave primaria de **planes_subscripcion(nombre_plan)** como clave foránea en **usuarios**. Por conveniencia, se define **ON DELETE SET DEFAULT** para que, si un plan se elimina, los usuarios queden asociados a un plan por defecto (por ejemplo, el plan gratuito). La acción **ON UPDATE CASCADE** mantiene la integridad si cambia el nombre de un plan.

3. **pagos**: {numero_pago, nombre_usuario, nombre_plan, forma_pago, fecha_pago, precio_plan }

- Entidad MER: Pagos
- Atributos: **numero_pago**, **nombre_usuario**, **nombre_plan**, **forma_pago**, **fecha_pago**, **precio_plan**.
- Clave primaria: **numero_pago**.
- Claves candidatas: **numero_pago**.
- Claves foráneas: **nombre_usuario** → **usuarios(nombre_usuario)**.
- Tabla relacional: **pagos**.

Justificación: Esta tabla representa un historial de transacciones entre usuarios y planes, por lo que se decidió utilizar una clave primaria autogenerada **numero_pago** que identifica de forma única cada registro. Aunque inicialmente **nombre_plan** era una clave foránea hacia la tabla **planes_subscripcion**, se optó por quitar dicha restricción para evitar efectos colaterales si un plan es eliminado del sistema. Dado que la tabla **pagos** registra eventos pasados, su contenido no debe verse afectado por actualizaciones o borrados de otras tablas; preservar su integridad histórica tiene prioridad. La única clave foránea mantenida es **nombre_usuario**, ya que sí se espera que un usuario no pueda existir si no está registrado.

4. **artistas:**{nombre_artista, tipo_artista, pais }

- Entidad MER: Artistas.
- Atributos: **nombre_artista**, **tipo_artista**, **pais**.
- Claves candidatas: **nombre_artista**.
- Clave primaria: **nombre_artista**.

Justificación: Aquí usamos **nombre_artista** como clave primaria natural, suponiendo que es único en la colección de artistas. Se añade una restricción **CHECK** para limitar **tipo_artista** a valores válidos (solista o banda). Este diseño corresponde a la entidad Artistas sin necesidad de claves subrogada.

5. **canciones:**{id_cancion, nombre_cancion, duracion_cancion, genero_cancion }

- Entidad MER: Canciones.
- Atributos: **id_cancion**, **nombre_cancion**, **duracion_cancion**, **genero_cancion**.
- Clave primaria: **id_cancion**.
- Claves candidatas: **id_cancion**.

Justificación: Se emplea un identificador numérico **id_cancion** como clave primaria (clave subrogada), ya que el nombre de la canción podría no ser único. Esto simplifica las referencias desde otras tablas y cumple con la definición de clave primaria. El campo **duracion_cancion** tiene un **CHECK** para asegurar duración positiva, y **genero_cancion** es un **ENUM** para restringir géneros válidos.

6. **albumes:**{id_album, nombre_album, fecha_lanzamiento, nombre_artista }

- Entidad MER: Álbumes.
- Atributos: **id_album**, **nombre_album**, **fecha_lanzamiento**, **nombre_artista**.
- Clave primaria: **id_album**.
- Claves candidatas: **id_album**.
- Clave foránea: **nombre_artista** → **artistas(nombre_artista)**.

Justificación: Usamos **id_album** como clave primaria subrogada por las mismas razones. La relación 1:N entre Artistas y Álbumes se implementa mediante la clave ajena **nombre_artista** en **albumes**. Esto significa que cada álbum referencia al artista correspondiente. Se podría agregar una restricción **UNIQUE** sobre (**nombre_album**, **nombre_artista**) para evitar duplicados, pero no es estrictamente necesario si ya usamos el **id**.

7. **albumes_canciones:**{ id_album, id_cancion }

Cancion pertenece a Álbum (una canción puede pertenecer a un álbum y un álbum contiene varias canciones). Se crea la tabla puente **albumes_canciones** con:

- Clave primaria compuesta: (**id_album**, **id_cancion**).
- Claves foráneas: **id_album** → **albumes(id_album)**, **id_cancion** → **canciones(id_cancion)**.

Justificación: La relación (1,N)–(0,1) entre **CANCION** y **ALBUM** requiere una tabla intermedia porque un álbum puede tener muchas canciones, y una canción puede no pertenecer a ningún álbum o pertenecer a uno solo.

Por eso, la tabla **albumes_canciones** registra cada asociación posible sin romper la cardinalidad ni introducir nulos, y permite agregar más datos si fuera necesario (como el orden en el álbum).

8. **albumes_canciones**{ id_album, id_cancion }

Relación MER: Cancion Pertenece Album

- Significado: un álbum puede contener muchas canciones, y cada canción puede no pertenecer a ningún álbum o pertenecer a uno solo.
- Clave primaria compuesta: (**id_album, id_cancion**)
- Claves foráneas:
 - **id_album** → **albumes(id_album)**
 - **id_cancion** → **canciones(id_cancion)**

Justificación: Se modela como tabla intermedia para reflejar la asociación. La clave primaria compuesta evita duplicar la misma pareja álbum–canción, y las claves foráneas aseguran que solo se enlacen álbumes y canciones válidos. La cardinalidad (1,N)–(0,1) se respeta porque el diseño no obliga a que una canción esté siempre en un álbum y permite varios enlaces desde un álbum.

9. **canciones_artistas**:{ id_cancion, nombre_artista }

- Relación: Canciones–Artistas (una canción puede ser escrita por uno o más artistas, y un artista puede haber compuesto varias canciones).
- Clave primaria compuesta: (**id_cancion, nombre_artista**).
- Claves foráneas:
 - **id_cancion** → **canciones(id_cancion)**
 - **nombre_artista** → **artistas(nombre_artista)**

Justificación: Se implementa una tabla intermedia para modelar la relación (1,N) – (1,M) entre canciones y artistas, ya que una canción puede estar asociada a uno o más artistas, y un artista puede participar en múltiples canciones. El uso de una clave primaria compuesta garantiza la unicidad de cada par canción–artista, mientras que las claves foráneas con opciones ON DELETE CASCADE y ON UPDATE CASCADE aseguran la integridad referencial ante cambios o eliminaciones en las entidades relacionadas.

10. **reproducciones_usuarios**: { nombre_usuario, id_cancion, cantidad_reproducciones }

- Relación MER: Usuarios escucha Canciones (registro de reproducciones con contador).
- Atributos: **nombre_usuario, id_cancion, cantidad_reproducciones**.
- Cardinalidad: Usuarios (0,N) - Canciones (0,N) cada par usuario–canción puede tener múltiples reproducciones.
- Clave primaria compuesta: (**nombre_usuario, id_cancion**)
- Claves foráneas:
 - **nombre_usuario** → **usuarios(nombre_usuario)**
 - **id_cancion** → **canciones(id_cancion)**

Justificación: Se incluye el atributo **cantidad_reproducciones** para modelar el número de veces que un usuario ha reproducido una misma canción sin generar múltiples registros. La clave primaria compuesta garantiza la unicidad del par usuario–canción, y el campo **cantidad_reproducciones** almacena el contador acumulado de reproducciones. Las claves foráneas aseguran la integridad referencial con las tablas **usuarios** y **canciones**.

11. Guardados (favoritos):

- **artistas_guardados** { nombre_usuario, nombre_artista }
- **canciones_guardadas** { nombre_usuario, id_cancion }
- **albumes_guardados** { nombre_usuario, id_album }
- Los usuarios pueden guardar en favoritos artistas, canciones y álbumes.
- **artistas_guardados(nombre_usuario, nombre_artista)**: artistas guardados por usuario.
- **canciones_guardadas(nombre_usuario, id_cancion)**: canciones guardadas por usuario.
- **albumes_guardados(nombre_usuario, id_album)**: álbumes guardados por usuario.

En cada tabla, la clave primaria es la combinación (nombre_usuario, recurso) y las columnas referencian a las tablas correspondientes (usuarios, artistas/canciones/álbumes).

Justificación: Las relaciones guarda entre usuarios y recursos (artistas, canciones y álbumes) son de tipo (0,N)–(0,M), por lo que se modelan con tablas intermedias. Cada una tiene una clave primaria compuesta que evita duplicados, y claves foráneas que aseguran integridad referencial.