

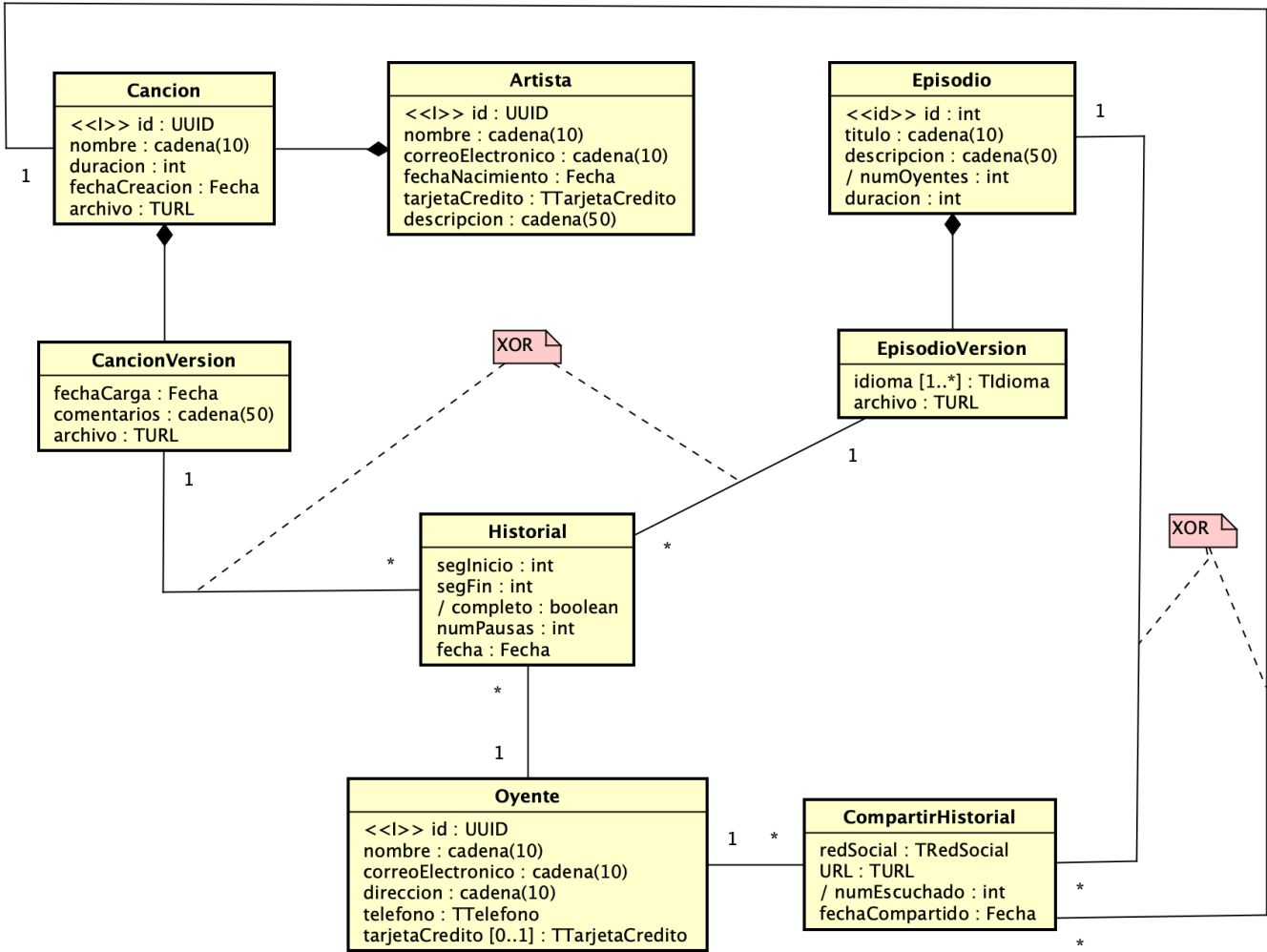
En este ciclo de trabajo vamos a implementar la siguiente área.

HISTORIAL

Así mismo un artista puede subir varias versiones de una canción, para solucionar problemas de edición, de cada versión se debe conocer la fecha de carga, comentarios por parte del artista explicando los cambios y el archivo de la canción. Los episodios pueden tener muchas versiones, cada versión corresponderá a un idioma diferente, con lo cual nos interesa saber de esa versión en que idioma se encuentra así como un archivo con el texto del episodio.

El historial se encarga de almacenar todo el detalle del contenido escuchado por los oyentes, (canciones y episodios), cada vez que un cliente escucha alguno de estos recursos en la base de datos quedará un registro de el recurso que escucho, su version especifica, segundo de inicio, segundo de fin, si se escucho de forma completa o no y el número de veces que se pauso el recurso. Es importante no olvidar que tanto una canción como un episodio pueden ser escuchadas múltiples veces por el mismo oyente, cada vez que ocurra una reproducción debe quedar un registro histórico de dicha interacción.

Si el oyente comparte la canción o el episodio en las redes sociales (Instagram, Facebook, TikTok) se deberá crear un registro histórico de esta interacción de ello nos interesa conocer la fecha en la que se compartió el episodio o la canción, una URL autogenerada para hacer seguimiento de la publicidad, el número de veces que el oyente escucho el recurso.



(30%) MODELO RELACIONAL (Modelo mini)

Transforme el modelo conceptual extendido anterior en un modelo relacional usando el estándar mínimo. Marque cada CRUD en un color diferente.

## (20%) MODELO RELACIONAL. INTEGRIDAD DECLARATIVA. (SQL Estándar)

Seleccione e implemente los mecanismos necesarios para las siguientes restricciones:

1. La fecha de los historiales es automática dada por el sistema.
2. El enlace de los archivos es una URL que termina con extensión .mp3
3. Segundo de Inicio y de Fin no pueden ser menores a 0.
4. Las redes sociales son Instagram, Facebook y TikTok.
5. El id de las canciones y episodios es automático incremental comenzando en 100.

## (20%) MODELO RELACIONAL. INTEGRIDAD PROCEDIMENTAL. (SQL Estándar)

Seleccione y explique los mecanismos necesarios para cumplir con las siguientes reglas de negocio:

**(SOLO DISEÑO. NO LAS IMPLEMENTE)**

**Mantener Historial y Compartir Historial**

- Adicionar
  - Cada vez que se crea un nuevo registro de historial asociado a un episodio de podcast, se debe actualizar el número de oyentes del mismo episodio como la cuenta distinta de todos los oyentes de ese episodio en el historial.
  - Si la duración de una canción o un episodio es equivalente a el valor absoluto de la resta del segundo de fin y el segundo de inicio del registro de historial, entonces el recurso se escuchó de forma completa.
  - Automatizar la mayor cantidad de campos de insertar en CompartirHistorial dado que: La red social por defecto es Instagram, el número de veces que se ha escuchado el recurso se debe calcular de la tabla Historial, la URL es de la forma `https://spotify.com/share/{idOyente}/{redSocial}/{timestamp}`
- Modificar
  - No es posible actualizar los Historial
- Eliminar
  - Al eliminar un oyente todo su historial debe desaparecer con el.
  - No se pueden eliminar episodios con más de 50 oyentes.

## (30%) CONSTRUCCIÓN (SQL ORACLE)

1. Implemente la estructura de las tablas diseñadas en el punto 1. (sin restricciones de integridad externa).  
**SOLO SE PIDEN LAS RESTRICCIONES PROPIAS DE LA DEFINICIÓN DE TABLAS**
2. Implemente todas las restricciones de integridad declarativa correspondientes al CRUD **Mantener Historial y Compartir Historial**  
**ÚNICAMENTE LAS RESTRICCIONES DE ESE CRUD Y LAS NECESARIAS PARA ELLAS.**
3. Implemente los mecanismos diseñados para cumplir las reglas de negocio del caso de uso **Mantener Historial y Compartir Historial**

**BONO (+.2)**

¿Cómo mejoraría el modelo conceptual extendido? ¿Por qué?

## Sintaxis

Tipo	CREATE DOMAIN ..... AS ..... CONSTRAINT CHECK .....
Atributo y tupla	ALTER TABLE ..... ADD CONSTRAINT ..... CHECK .....
Tabla y base de datos	CREATE ASSERTION ..... CHECK .....
Acciones	CONSTRAINT ..... FOREIGN KEY ..... REFERENCES ..... [ON DELETE (RESTRICT   CASCADE   SET NULL)] [ON UPDATE (RESTRICT   CASCADE   SET NULL)]
Disparadores	CREATE TRIGGER nombre [BEFORE   AFTER   INSTEAD OF ] [DELETE INSERT UPDATE [OF nombre columnas]] ON nombre Tabla [FOR EACH ROW [WHEN (condición)]] DECLARE -- Definición de variables BEGIN -- Instrucciones del disparador END nombre;  En la tercer línea puede incluirse más de una acción uniéndose con un OR

