

Informe I: Diseño de la Base de Datos

Gestión de Agencias y Grupos de K-pop

Equipo #1

Ronald Provance Valladares	C-312
Agustín Alberto Carbajal Romero	C-312
Daniel Amaranto Mares García	C-312
Dylan Ramsés Cabrera Morales	C-312
Sheila Roque Alemán	C-312

Introducción

En este informe se presenta el diseño y desarrollo de una base de datos relacional para la gestión de agencias y grupos de K-pop. El proyecto forma parte de la evaluación de las asignaturas **Bases de Datos II e Ingeniería de Software**, donde cada asignatura evaluará aspectos específicos.

El objetivo del proyecto es automatizar y optimizar la gestión de agencias, aprendices, artistas, grupos, álbumes, actividades, contratos e ingresos, mejorando la eficiencia y proporcionando herramientas para la toma de decisiones en la industria del entretenimiento coreano.

El sistema permitirá gestionar el ciclo completo de los artistas, desde su etapa de aprendices hasta su debut y carrera profesional, incluyendo la gestión de agendas, conflictos de horarios, ingresos y reportes de éxito.

Requerimientos Funcionales del Sistema

Los requisitos funcionales describen las funciones y características específicas que el sistema debe tener para cumplir con las expectativas del usuario y los objetivos del proyecto. Estos requisitos detallan qué debe hacer el sistema y cómo debe comportarse en diversas situaciones.

Gestión de Agencias

Registrar y consultar información de cada agencia (nombre, ubicación, fecha de fundación)

Gestión de Aprendices

Registrar información personal de los aprendices (nombre completo, edad, fecha de nacimiento, nivel de entrenamiento, estado)

Formación de Grupos

Registrar el proceso de selección de aprendices para conformar grupos, convirtiéndolos en artistas

El sistema debe permitir:

- Gestión de grupos: registrar y actualizar la información de los grupos (nombre, fecha de debut, concepto, estado, número de miembros)
- Gestión de álbumes: registrar datos de cada álbum (título, fecha de lanzamiento, productor, número de canciones, copias vendidas, premios)
- Gestión de actividades: programar y consultar actividades grupales e individuales en un calendario, evitando conflictos de horarios
- Agenda de artistas: permitir a cada artista consultar sus actividades, confirmar asistencia, reportar disponibilidad y enviar solicitudes de ajuste de horario al mánager
- Gestión de contratos: registrar, actualizar y consultar información de contratos (artista, fechas, estado, distribución de ingresos)
- Gestión de ingresos: registrar los ingresos provenientes de actividades grupales e individuales (tipo, monto, fecha, actividad asociada)
- Historial profesional: mantener un registro completo de la trayectoria de cada artista (agencias, grupos, debuts, colaboraciones)
- Control de formación de grupos por artistas: permitir que un aprendiz o solista proponga la creación de un grupo, con aprobación final del director de la agencia
- Gestión de usuarios y roles: diferenciar accesos según rol (mánager, artista, director, administrador) para proteger la información sensible
- Exportación de información: proveer la funcionalidad de exportar reportes y resultados a archivos PDF, con posibilidad de ordenar y filtrar columnas
- Reportes de éxito: generar reportes en forma de tablas o gráficos sobre ventas, premios y posiciones en listas de popularidad
- Detección de conflictos de agenda: identificar artistas con actividades que se solapan en un periodo dado
- Cálculo de ingresos por artista: generar reportes de ingresos grupales e individuales en un periodo específico
- Identificación de historial de cambios de agencia: detectar artistas que cambiaron de agencia al menos dos veces y participaron en más de un grupo
- Identificación de solistas con grupos disueltos: detectar solistas que fueron miembros de grupos disueltos con álbumes exitosos

Requerimientos Informacionales

Para el correcto funcionamiento del sistema de gestión de agencias y grupos de K-pop, es esencial definir y estructurar la información que debe ser almacenada, gestionada y presentada. Estos requerimientos informacionales aseguran que todos los datos relevantes sean capturados de manera precisa y organizada, permitiendo una gestión eficiente y efectiva.



Agencias

Nombre, ubicación, fecha de fundación, artistas y grupos bajo contrato



Aprendices

Nombre completo, edad, fecha de nacimiento, fecha de ingreso a la agencia, nivel de entrenamiento, historial de evaluaciones y estado actual (en entrenamiento, en proceso de selección, transferido)



Artistas

Nombre artístico, nombre real, fecha de nacimiento, rol dentro del grupo (líder, vocalista, rapero, bailarín, etc.), fecha de debut, estado actual (activo, en pausa, inactivo), agencia a la que pertenecen



Grupos

Nombre, fecha de debut, concepto, cantidad de miembros, estado (activo, en pausa, disuelto), agencia representante



Álbumes

Título, fecha de lanzamiento, número de canciones, productor principal, cantidad de copias vendidas, premios recibidos, posiciones en listas de popularidad Billboard



Actividades

Tipo (grupal o individual), lugar, fecha, hora, responsables y estado

A continuación, se detallan los principales tipos de información que el sistema debe manejar:

- **Contratos:** artista asociado, fecha de inicio, fecha de finalización, condiciones principales, estado (activo, en renovación, finalizado, rescindido), porcentaje de distribución de ingresos
- **Ingresos:** tipo, monto, fecha, actividad asociada y responsable de gestión
- **Historial profesional:** agencias pasadas, grupos en los que ha participado, debuts, colaboraciones y actividades de cada artista
- **Usuarios:** roles y permisos asociados (mánager, artista, director, administrador)
- **Reportes de éxito:** evolución de ventas de álbumes, premios obtenidos, posiciones en rankings musicales

Modelo Conceptual de la Base de Datos

Con base en los requerimientos y especificaciones del problema, se propone un modelo conceptual que busca cumplir con las necesidades de la industria del K-pop. Este modelo está diseñado para **centralizar la información, asignar responsabilidades claras y generar reportes detallados**.

Aunque se estructura alrededor de entidades clave y sus relaciones, la propuesta está sujeta a ajustes y mejoras conforme se avance en el desarrollo y se identifiquen nuevas necesidades.

Abreviaturas Utilizadas

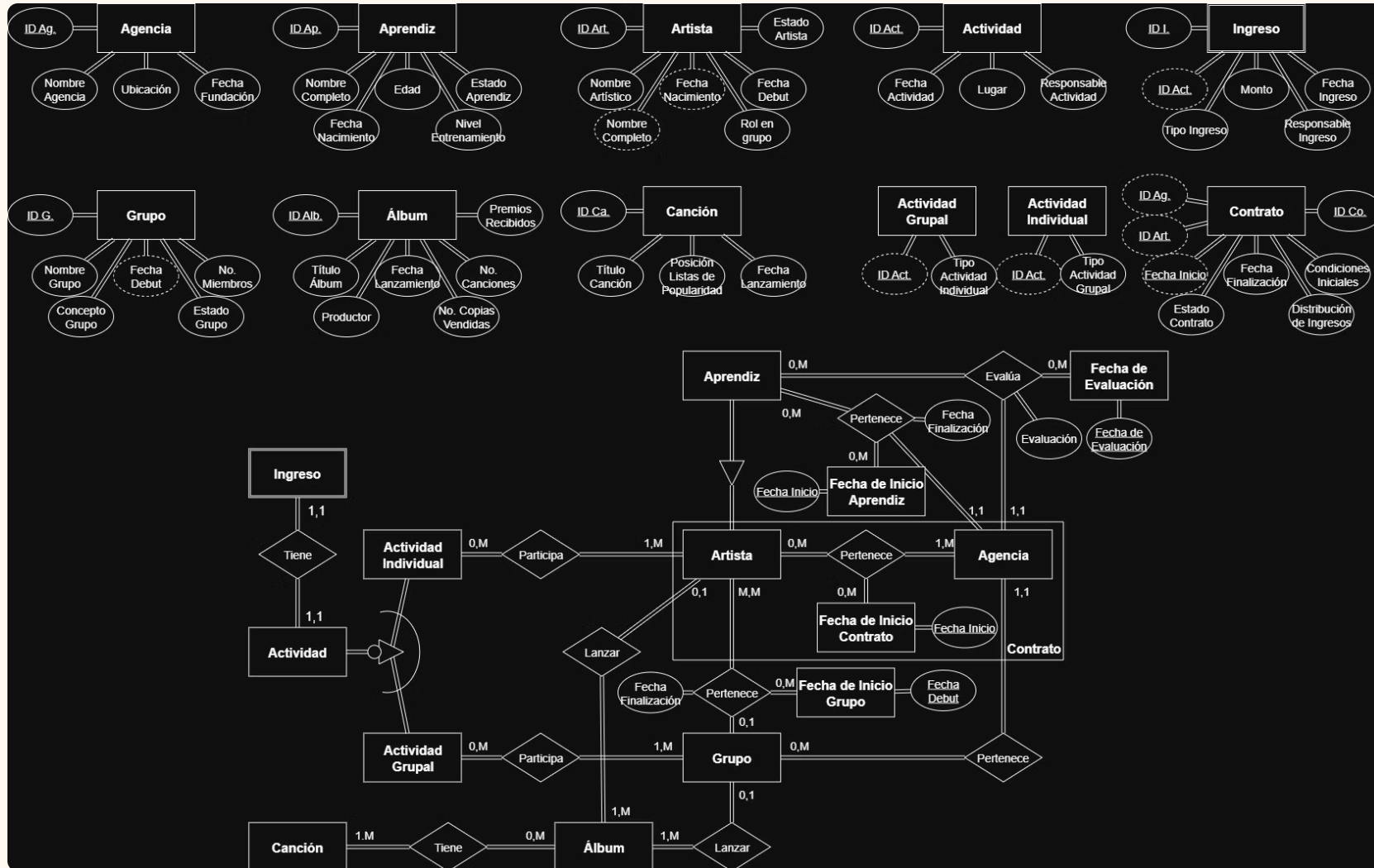
- **ID_Ag:** ID de Agencia
- **ID_Ap:** ID de Aprendiz
- **ID_Art:** ID de Artista
- **ID_G:** ID de Grupo
- **ID_Alb:** ID de Álbum
- **ID_Ca:** ID de Canción
- **ID_Act:** ID de Actividad
- **ID_Act_I:** ID de Actividad Individual
- **ID_Act_G:** ID de Actividad Grupal
- **ID_I:** ID de Ingreso
- **ID_Co:** ID de Contrato
- **FechaDeb:** Fecha de Debut
- **FechaLanz:** Fecha de Lanzamiento
- **No_Miembros:** Número de Miembros



Conclusiones Inferidas:

- Un artista puede debutar en múltiples grupos a lo largo de su carrera, por lo que se debe mantener un historial de membresía en grupos
- Las actividades grupales e individuales deben gestionarse de manera centralizada para evitar conflictos de horarios
- Los ingresos deben asociarse a actividades específicas para permitir un cálculo preciso por artista
- El estado de los aprendices y artistas puede cambiar con el tiempo, por lo que se requiere un seguimiento histórico

MERX



Esquema Relacional y Restricciones de Integridad

Las restricciones de integridad son reglas que aseguran la exactitud y consistencia de los datos en una base de datos. En el esquema MERX, estas restricciones son fundamentales para mantener la integridad de la información.

1. **Agencia**
 - PK:* ID_Ag int NOT NULL UNIQUE
 - Nombre char(50) NOT NULL
 - Ubicación char(50) NOT NULL
 - FechaFundación date NOT NULL
 - Check (FechaFundación < FechaActual)
2. **Aprendiz**
 - PK:* ID_Ap int NOT NULL UNIQUE
 - NombreCompleto char(50) NOT NULL
 - FechaNacimiento date NOT NULL
 - Edad int NOT NULL
 - NivelEntrenamiento int NOT NULL
 - EstadoAprendiz char(50) NOT NULL
 - Check (NivelEntrenamiento > 0)
 - Check (Edad >= 15)
3. **Artista**
 - PK:* ID_Art int NOT NULL UNIQUE
 - NombreArtístico char(50) NOT NULL
 - FechaDebut date NOT NULL
 - RolGrupo char(50) NOT NULL
 - EstadoArtista char(50) NOT NULL
 - FK:* ID_Art REFERENCES Aprendiz(ID_Ap)
 - Check (FechaDebut > FechaFundación(Agencia))
4. **Actividad**
 - PK:* ID_Act int NOT NULL UNIQUE
 - Fecha date NOT NULL
 - Lugar char(50) NOT NULL
 - Responsable char(50) NOT NULL
 - Check (Fecha > FechaActual)
5. **ActividadGrupal**
 - PK:* ID_Act_G int NOT NULL UNIQUE
 - Tipo char(50) NOT NULL
 - FK:* ID_Act_G REFERENCES Actividad(ID_Act)
6. **ActividadIndividual**
 - PK:* ID_Act_I int NOT NULL UNIQUE
 - Tipo char(50) NOT NULL
 - FK:* ID_Act_I REFERENCES Actividad(ID_Act)
7. **Ingreso**
 - PK:* ID_I int NOT NULL UNIQUE
 - FK:* ID_Act int NOT NULL REFERENCES Actividad(ID_Act)
 - Fecha date NOT NULL
 - TipoIngreso char(50) NOT NULL
 - Responsable char(50) NOT NULL
 - Monto decimal(10,2) NOT NULL
 - Check (Monto > 0)
 - Check (Fecha > Fecha(Actividad))
8. **Grupo**
 - PK:* ID_G int NOT NULL UNIQUE
 - FK:* ID_Ag int NOT NULL REFERENCES Agencia(ID_Ag)
 - NombreGrupo char(50) NOT NULL
 - FechaDebut date NOT NULL
 - EstadoGrupo char(50) NOT NULL
 - ConceptoGrupo char(50) NOT NULL
 - NoMiembros int NOT NULL
 - Check (NoMiembros > 0)
 - Check (FechaDebut > FechaFundación(Agencia))
9. **Álbum**
 - PK:* ID_Alb int NOT NULL UNIQUE
 - FK:* ID_Art int NULL REFERENCES Artista(ID_Art)
 - FK:* ID_G int NULL REFERENCES Grupo(ID_G)
 - Título char(50) NOT NULL
 - FechaLanzamiento date NOT NULL
 - Productor char(50) NOT NULL
 - PremiosRecibidos char(50) NOT NULL
 - NoCanciones int NOT NULL
 - NoCopiasVendidas int NOT NULL
 - Check (NoCanciones > 0)
 - Check (NoCopiasVendidas > 0)
 - Check (FechaLanzamiento > FechaDebut(Artista) OR FechaDebut(Grupo))
10. **Canción**
 - PK:* ID_Ca int NOT NULL UNIQUE
 - Título char(50) NOT NULL
 - PosiciónListasPopularidad int NOT NULL
 - FechaLanzamiento date NOT NULL
 - Check (FechaLanzamiento > FechaActual)
11. **Contrato**
 - PK:* ID_Co int NOT NULL UNIQUE
 - FK:* ID_Ag int NOT NULL REFERENCES Agencia(ID_Ag)
 - FK:* ID_Art int NOT NULL REFERENCES Artista(ID_Art)
 - FechaInicio date NOT NULL
 - FechaFinalización date NOT NULL
 - Estado char(50) NOT NULL
 - CondicionesIniciales char(50) NOT NULL
 - DistribuciónIngresos char(50) NOT NULL
 - Check (FechaInicio >= FechaActual)
 - Check (FechaFinalización > FechaInicio)
12. **ArtistaEnGrupo**
 - PK:* ID_Art int NOT NULL REFERENCES Artista(ID_Art)
 - PK:* ID_G int NOT NULL REFERENCES Grupo(ID_G)
 - PK:* FechaInicio date NOT NULL
 - FechaFinalización date NULL
 - Rol char(50) NOT NULL
 - Check (FechaFinalización > FechaInicio)
13. **AprendizEnAgencia**
 - PK:* ID_Ap int NOT NULL REFERENCES Aprendiz(ID_Ap)
 - PK:* ID_Ag int NOT NULL REFERENCES Agencia(ID_Ag)
 - PK:* FechaInicio date NOT NULL
 - FechaFinalización date NULL
 - Check (FechaFinalización > FechaInicio)
14. **EvaluaciónAprendiz**
 - PK:* ID_Ap int NOT NULL REFERENCES Aprendiz(ID_Ap)
 - PK:* ID_Ag int NOT NULL REFERENCES Agencia(ID_Ag)
 - PK:* FechaEvaluación date NOT NULL
 - Evaluación int NOT NULL
15. **ArtistaEnActividadIndividual**
 - PK:* ID_Art int NOT NULL REFERENCES Artista(ID_Art)
 - PK:* ID_Act_I int NOT NULL REFERENCES ActividadIndividual(ID_Act_I)
16. **GrupoEnActividadGrupal**
 - PK:* ID_G int NOT NULL REFERENCES Grupo(ID_G)
 - PK:* ID_Act_G int NOT NULL REFERENCES ActividadGrupal(ID_Act_G)
17. **CanciónEnÁlbum**
 - PK:* ID_Ca int NOT NULL REFERENCES Canción(ID_Ca)
 - PK:* ID_Alb int NOT NULL REFERENCES Álbum(ID_Alb)
18. **ArtistaEnÁlbum**
 - PK:* ID_Art int NOT NULL REFERENCES Artista(ID_Art)
 - PK:* ID_Alb int NOT NULL REFERENCES Álbum(ID_Alb)
19. **GrupoEnÁlbum**
 - PK:* ID_G int NOT NULL REFERENCES Grupo(ID_G)
 - PK:* ID_Alb int NOT NULL REFERENCES Álbum(ID_Alb)
20. **AgenciaEnGrupo**
 - PK:* ID_Ag int NOT NULL REFERENCES Agencia(ID_Ag)
 - PK:* ID_G int NOT NULL REFERENCES Grupo(ID_G)

Universo y Dependencias Funcionales

Consideremos el esquema relacional $R = \{U, F\}$ que se corresponde con el proyecto de la empresa:

U = {ID_Ag, NombreAgencia, FechaFundación, Ubicación, ID_Ap, NombreCompleto, FechaNacimiento, Edad, NivelEntrenamiento, EstadoAprendiz, ID_Art, NombreArtístico, FechaDebut, RolGrupo, EstadoArtista, ID_G, NombreGrupo, ConceptoGrupo, EstadoGrupo, NoMiembros, ID_Alb, TítuloÁlbum, FechaLanzamiento, Productor, NoCopiasVendidas, NoCanciones, PremiosRecibidos, ID_Ca, TítuloCanción, PosiciónListasPopularidad, ID_Act, FechaActividad, Lugar, ResponsableActividad, ID_Act_I, TipoActividadIndividual, ID_Act_G, TipoActividadGrupal, ID_I, TipoIngreso, FechaIngreso, Monto, ResponsableIngreso, ID_Co, FechaInicioContrato, FechaFinalización, EstadoContrato, DistribuciónIngresos, CondicionesIniciales, FechaInicioAprendiz, FechaFinalizaciónAprendiz, FechaEvaluación, Evaluación, FechaInicioGrupo, FechaFinalizaciónGrupo}

Dependencias Funcionales

F = {

NombreAgencia, FechaFundación \rightarrow ID_Ag,

ID_Ag \rightarrow NombreAgencia, FechaFundación, Ubicación,

NombreCompleto, FechaNacimiento \rightarrow ID_Ap,

ID_Ap \rightarrow NombreCompleto, FechaNacimiento, Edad, NivelEntrenamiento, EstadoAprendiz,

FechaNacimiento \rightarrow Edad,

ID_Ap, FechaDebut \rightarrow ID_Art,

ID_Art \rightarrow NombreCompleto, FechaNacimiento, FechaDebut, NombreArtístico, RolGrupo, EstadoArtista,

NombreGrupo, FechaDebut \rightarrow ID_G,

ID_G \rightarrow NombreGrupo, FechaDebut, ConceptoGrupo, EstadoGrupo, NoMiembros,

FechaLanzamiento, TítuloÁlbum, Productor \rightarrow ID_Alb,

ID_Alb \rightarrow TítuloÁlbum, FechaLanzamiento, Productor, NoCopiasVendidas, NoCanciones, PremiosRecibidos,

FechaLanzamiento, TítuloCanción \rightarrow ID_Ca,

ID_Ca \rightarrow TítuloCanción, FechaLanzamiento, PosiciónListasPopularidad,

FechaActividad, Lugar \rightarrow ID_Act,

ID_Act \rightarrow FechaActividad, Lugar, ResponsableActividad,

ID_Act, TipoActividadIndividual \rightarrow ID_Act_I,

ID_Act_I \rightarrow TipoActividadIndividual,

ID_Act, TipoActividadGrupal \rightarrow ID_Act_G,

ID_Act_G \rightarrow TipoActividadGrupal,

ID_Act, TipoIngreso, FechaIngreso \rightarrow ID_I,

ID_I \rightarrow ID_Act, TipoIngreso, FechaIngreso, Monto, ResponsableIngreso,

ID_Ag, ID_Art, FechaInicio \rightarrow ID_Co,

ID_Co \rightarrow ID_Ag, ID_Art, FechaInicio, EstadoContrato, FechaFinalización, DistribuciónIngresos, CondicionesIniciales,

ID_Ap, ID_Ag, FechaInicio \rightarrow FechaFinalización,

ID_Ap, ID_Ag, FechaEvaluación \rightarrow Evaluación,

ID_Art, ID_G, FechaInicio \rightarrow FechaFinalización,

ID_Ag, ID_G \rightarrow ID_Ag, ID_G,

ID_Art, ID_Alb \rightarrow ID_Art, ID_Alb,

ID_G, ID_Alb \rightarrow ID_G, ID_Alb,

ID_Alb, ID_Ca \rightarrow ID_Alb, ID_Ca,

ID_Art, ID_Act_I \rightarrow ID_Art, ID_Act_I,

ID_G, ID_Act_G \rightarrow ID_G, ID_Act_G

}

R = {U, F}

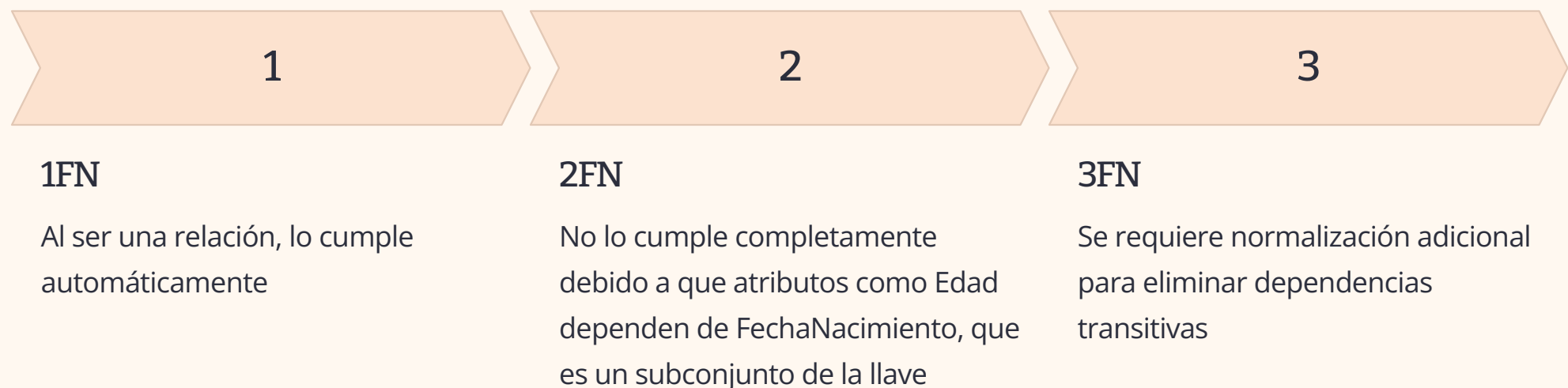
Llave: {ID_Ag, ID_Ap, ID_Art, ID_G, ID_Alb, ID_Ca, ID_Act, ID_Co}


Cubrimiento Minimal y Análisis del Diseño

Obtengamos el cubrimiento minimal del conjunto de dependencias funcionales F:

F' = {
NombreAgencia, FechaFundación → ID_Ag,
ID_Ag → NombreAgencia, FechaFundación, Ubicación,
NombreCompleto, FechaNacimiento → ID_Ap,
ID_Ap → NombreCompleto, FechaNacimiento, Edad, NivelEntrenamiento, EstadoAprendiz,
FechaNacimiento → Edad,
ID_Ap, FechaDebut → ID_Art,
ID_Art → NombreCompleto, FechaNacimiento, FechaDebut, NombreArtístico, RolGrupo, EstadoArtista,
... [continúa con todas las dependencias funcionales]
}

Análisis del Diseño



 **Conclusión del Análisis:** El esquema requiere un proceso de normalización para cumplir completamente con las formas normales superiores y garantizar la integridad de los datos.

Proceso de Normalización

Se aplica un proceso sistemático de normalización para optimizar el diseño de la base de datos y eliminar redundancias.

1FN - Primera Forma Normal El esquema cumple con la primera forma normal (1FN) al tener atributos atómicos y no tener grupos repetitivos.	2FN - Segunda Forma Normal Se aplica la segunda forma normal (2FN) para eliminar dependencias parciales. Las dependencias funcionales se ajustan para que todos los atributos no primos dependan completamente de la clave primaria.	3FN - Tercera Forma Normal Se aplica la tercera forma normal (3FN) para eliminar dependencias transitivas. El esquema resultante cumple con 3FN.
---	--	--

Partición del Esquema

La partición del esquema en relaciones más pequeñas se define como sigue:

<ul style="list-style-type: none">R1: U 1 = {ID_Ag, NombreAgencia, FechaFundación, Ubicación} F 1 = {NombreAgencia, FechaFundación → ID_Ag; ID_Ag → NombreAgencia, FechaFundación, Ubicación}R2: U 2 = {ID_Ap, NombreCompleto, FechaNacimiento, Edad, NivelEntrenamiento, EstadoAprendiz} F 2 = {NombreCompleto, FechaNacimiento → ID_Ap; ID_Ap → NombreCompleto, FechaNacimiento, Edad, NivelEntrenamiento, EstadoAprendiz; FechaNacimiento → Edad}R3: U 3 = {ID_Ap, ID_Art, NombreCompleto, FechaNacimiento, FechaDebut, NombreArtístico, RolGrupo, EstadoArtista} F 3 = {ID_Ap, FechaDebut → ID_Art; ID_Art → NombreCompleto, FechaNacimiento, FechaDebut, NombreArtístico, RolGrupo, EstadoArtista}R4: U 4 = {ID_G, NombreGrupo, FechaDebut, ConceptoGrupo, EstadoGrupo, NoMiembros} F 4 = {NombreGrupo, FechaDebut → ID_G; ID_G → NombreGrupo, FechaDebut, ConceptoGrupo, EstadoGrupo, NoMiembros}R5: U 5 = {ID_Alb, TítuloÁlbum, FechaLanzamiento, Productor, NoCopiasVendidas, NoCanciones, PremiosRecibidos} F 5 = {FechaLanzamiento, TítuloÁlbum, Productor → ID_Alb; ID_Alb → TítuloÁlbum, FechaLanzamiento, Productor, NoCopiasVendidas, NoCanciones, PremiosRecibidos}R6: U 6 = {ID_Ca, TítuloCanción, FechaLanzamiento, PosiciónListasPopularidad} F 6 = {FechaLanzamiento, TítuloCanción → ID_Ca; ID_Ca → TítuloCanción, FechaLanzamiento, PosiciónListasPopularidad}R7: U 7 = {ID_Act, FechaActividad, Lugar, ResponsableActividad} F 7 = {FechaActividad, Lugar → ID_Act; ID_Act → FechaActividad, Lugar, ResponsableActividad}R8: U 8 = {ID_Act, ID_Act_I, TipoActivadadIndividual} F 8 = {ID_Act, TipoActivadadIndividual → ID_Act_I; ID_Act_I → TipoActivadadIndividual}R9: U 9 = {ID_Act, ID_Act_G, TipoActivadadGrupal} F 9 = {ID_Act, TipoActivadadGrupal → ID_Act_G; ID_Act_G → TipoActivadadGrupal})R10: U 10 = {ID_I, ID_Act, TipoIngreso, FechaIngreso, Monto, ResponsableIngreso} F 10 = {ID_Act, TipoIngreso, FechaIngreso → ID_I; ID_I → ID_Act, TipoIngreso, FechaIngreso, Monto, ResponsableIngreso}	<ul style="list-style-type: none">R11: U 11= {ID_Co, ID_Ag, ID_Art, FechaInicio, EstadoContrato, FechaFinalización, DistribuciónIngresos, CondicionesIniciales} F 11 = {ID_Ag, ID_Art, FechaInicio → ID_Co; ID_Co → ID_Ag, ID_Art, FechaInicio, EstadoContrato, FechaFinalización, DistribuciónIngresos, CondicionesIniciales}R12: U 12 = {ID_Ap, ID_Ag, FechaInicio, FechaFinalización} F 12= {ID_Ap, ID_Ag, FechaInicio → FechaFinalización}R13: U 13= {ID_Ap, ID_Ag, FechaEvaluación, Evaluación} F 13= {ID_Ap, ID_Ag, FechaEvaluación → Evaluación}R14: U 14= {ID_Art, ID_G, FechaInicio, FechaFinalización} F 14= {ID_Art, ID_G, FechaInicio → FechaFinalización}R15: U 15= {ID_Ag, ID_G} F 15= {ID_Ag, ID_G → ID_Ag, ID_G}R16: U 16= {ID_Art, ID_Alb} F 16= {ID_Art, ID_Alb → ID_Art, ID_Alb}R17: U 17= {ID_G, ID_Alb} F 17= {ID_G, ID_Alb → ID_G, ID_Alb}R18: U 18= {ID_Alb, ID_Ca} F 18= {ID_Alb, ID_Ca → ID_Alb, ID_Ca}R19: U 19= {ID_Art, ID_Act_I} F 19= {ID_Art, ID_Act_I → ID_Art, ID_Act_I}R20: U 20 = {ID_G, ID_Act_G} F 20 = {ID_G, ID_Act_G → ID_G, ID_Act_G}
--	---

Esta partición cumple con la 3FN y garantiza la preservación de dependencias y la unión sin pérdida.

Conclusiones y Próximos Pasos

Este informe presenta el diseño conceptual y relacional de una base de datos para la gestión de agencias y grupos de K-pop, cumpliendo con los requerimientos académicos de las asignaturas **Bases de Datos II e Ingeniería de Software**.

Logros Alcanzados

- Identificación completa de requerimientos funcionales e informacionales
- Diseño de modelo conceptual robusto
- Definición de esquema relacional con restricciones de integridad
- Análisis de dependencias funcionales
- Proceso de normalización hasta 3FN

Beneficios del Sistema

- Automatización de procesos de gestión
- Centralización de información
- Mejora en la toma de decisiones
- Control de conflictos de horarios
- Generación de reportes detallados

Próximos Pasos

- Implementación física de la base de datos
- Desarrollo de la interfaz de usuario
- Pruebas de funcionalidad
- Optimización de consultas
- Documentación técnica completa

El diseño propuesto proporciona una base sólida para el desarrollo del sistema de gestión, asegurando la integridad de los datos y la eficiencia en las operaciones de la industria del entretenimiento coreano.