Entrega 2

Bases de Datos

Grupo 47

M. Clara Pinto

Jacinta Dumay

Catalina Río

Diagrama E/R

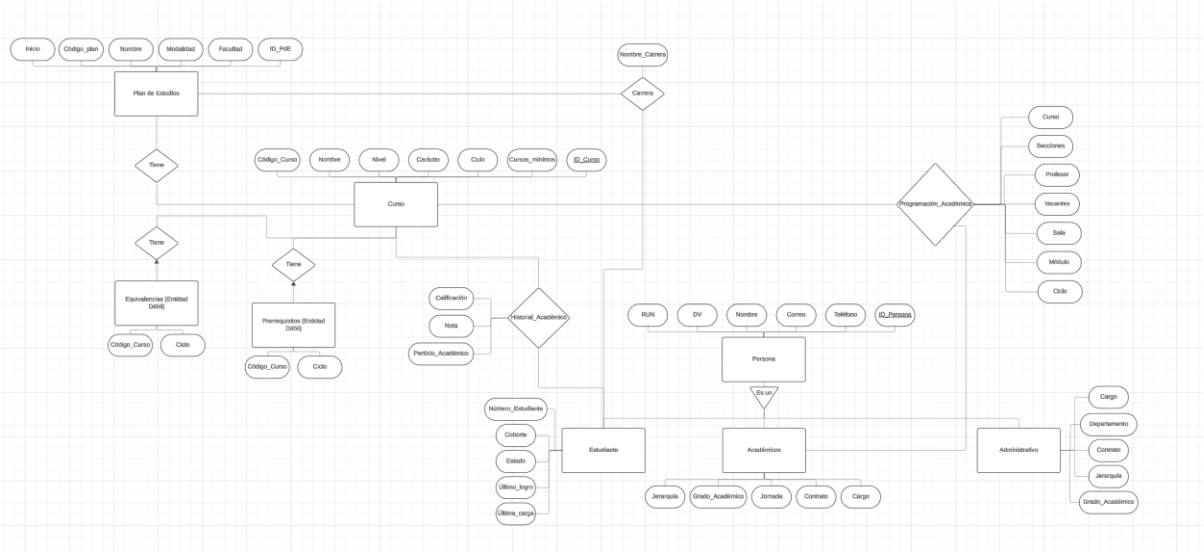

Imagen Nº1: Modelo E/R

Este es el link para ver más grande el diagrama en caso de que no se pueda ver bien desde aquí:

https://lucid.app/lucidchart/8528c965-5824-4976-892c-ad242e711a98/edit?viewport_loc=-740%2C-971%2C4587%2C2451%2C0_0&invitationId=inv_7adb9b3d-67fc-4451-a903-a8826bc31f88

Entidades Débiles del Modelo

Entendiendo que las entidades débiles son aquellas que dependen de otras entidades para existir, podemos identificar dos en este modelo.

Prerrequisitos es una entidad débil porque depende de la existencia de un curso. Los prerrequisitos son los cursos que un estudiante debe completar antes de tomar otro curso, por lo que no pueden existir sin estar vinculados a un curso específico.

Equivalencias es una entidad débil ya que depende completamente del curso al cual pertenece. No puede existir sin un curso que defina cuáles son las equivalencias con otros cursos, por lo que depende del curso para su existencia. Las equivalencias se utilizan para reconocer cursos alternativos que cumplen los mismos objetivos.

Llaves Primarias y Parciales en el modelo

Persona

- Llave primaria: RUN. Esta es una llave primaria dado que es un identificador único para cada persona, no pueden existir dos personas con el mismo RUN en el sistema.

- Llave primaria: ID_Persona. Es otra llave primaria que se utiliza para gestionar de los registros de personas, independiente de su RUN.

Estudiante

- Llave primaria: Número de Estudiante. Es única para cada estudiante y permite la identificación dentro del sistema.
- Llave parcial: RUN. Teniendo en cuenta que la entidad estudiante depende de la entidad persona, el RUN de la persona también es parte de la identificación del estudiante, pero para que sea estudiante, la persona debe tener un Número de estudiante, por lo que el RUN pasaría a ser una llave parcial.
- Llave foránea: ID_Persona. Esta es una llave foránea que conecta al estudiante con la entidad Persona.

Académico

- Llave primaria: RUN. Heredada de Persona, es la llave primaria que identifica de manera única a cada académico en el sistema.
- Llave foránea: ID_Persona. Es una llave foránea que conecta la subclase Profesor con la entidad Persona.

Administrativo

- Llave primaria: RUN. Heredada de Persona, es la llave primaria que identifica de manera única a cada administrativo en el sistema.
- Llave foránea: ID_Persona. Es una llave foránea que conecta la subclase Administrativo con la entidad Persona.

Curso

- Llave primaria: Código del Curso. Es el código único del curso que actúa como llave primaria pública.
- Llave primaria adicional: ID_Curso. Es un identificador interno del sistema utilizado para gestionar los cursos.

Plan de Estudios

- Llave primaria: Código del Plan de Estudios. Es el código único que identifica un plan de estudios dentro del sistema.
- Llave primaria adicional: ID_PdE. Es el identificador interno que gestiona cada plan de estudios en la base de datos.

Equivalencias (Entidad Débil)

- Llave compuesta primaria: Código del Curso + Ciclo. Esta combinación actúa como llave primaria para garantizar la unicidad de cada equivalencia.
- Llave foránea: ID_Curso. Conecta cada equivalencia con su curso correspondiente.

Prerrequisitos (Entidad Débil)

- Llave compuesta primaria: Código del Curso + Ciclo. Esta combinación es la llave primaria que identifica de manera única los prerrequisitos.
- Llave foránea: ID_Curso. Conecta cada prerrequisito con el curso al que pertenece.

Llaves Compuestas

Equivalencias:
Llave compuesta: Código del Curso + Ciclo.
La entidad Equivalencias requiere una llave compuesta, ya que no se puede identificar una equivalencia solo con el Código del Curso o solo con el Ciclo, sino que necesita ambos atributos para generar una relación de equivalencia única.

Con la combinación del Código del Curso y el Ciclo se asegura que cada relación de equivalencia sea única dentro de ese ciclo académico.

Prerrequisitos:
Llave compuesta: Código del Curso + Ciclo.
La entidad Prerrequisitos requiere una llave compuesta, ya que no basta con identificar un prerrequisito solo mediante el Código del Curso o el Ciclo.
La combinación de Código del Curso y Ciclo garantiza que cada prerrequisito esté relacionado de manera única con un curso en un ciclo específico asegurando que los prerrequisitos sean adecuados para cada curso y ciclo.

Cardinalidad de las relaciones

EsDe: Estudiante – Plan de Estudios:

- Cardinalidad: Uno a muchos (Plan de Estudios --> Estudiantes). Cada Plan de Estudios puede ser de muchos estudiantes, sin embargo, un estudiante tiene un solo Plan de Estudios.

PerteneceA: Curso – Plan de Estudios:

- Cardinalidad: Muchos a muchos (Curso <--> Plan de Estudios). Un curso puede pertenecer a varios planes de estudio pero un plan de estudios puede incluir varios cursos.

Tiene: Curso – Equivalencias:

- Cardinalidad: Uno a muchos (Curso --> Equivalencias). Un curso puede tener varias equivalencias con otros cursos pero cada equivalencia pertenece a un solo curso.

TienePrerrequisitos: Curso – Prerrequisitos:

- Cardinalidad: Uno a muchos (Curso --> Prerrequisitos). Un curso puede tener muchos prerrequisitos pero cada prerrequisito está asociado a un solo curso.

Historial: Estudiante Historial Académico:

- Cardinalidad: Uno a muchos (Estudiante --> Historial Académico). Un estudiante puede tener varios registros en su historial académico pero cada registro de historial académico pertenece a un solo estudiante.

Asignación: Programación Académica - Curso:

- Cardinalidad: Uno a muchos (Programación Académica --> Curso). La programación académica puede estar asignando un curso a por ejemplo muchas secciones, pero cada curso tiene solo una asignación de programación académica para cada ciclo.

Hace: Académico - Programación Académica:

- Cardinalidad: Uno a muchos (Académico --> Programación Académica): Un académico puede hacer clases en varios cursos o secciones, pero cada curso o sección es asignada a un solo académico (o profesor) en cada ciclo académico.

TrabajaEn: Administrativo - Departamento:

- Cardinalidad: Uno a muchos (Administrativo --> Departamento). Un departamento puede tener varios administrativos, pero un administrativo solo pertenece a un departamento a la vez.

Jerarquías de Clases

En el modelo E/R se ha logrado identificar una clara jerarquía que organiza a la entidad Persona junto con sus subclases: Estudiante, Académico y Administrativo. Esta jerarquía permite trabajar eficientemente los datos comunes a todas las personas de la universidad sin generar redundancias, mientras que se separan las características específicas de cada tipo de persona.

Jerarquía identificada:

Persona (Superclase):

- Atributos: RUN, DV, Nombre, Correo, Teléfono, ID_Persona.

Esta entidad agrupa a todos los miembros de la universidad, ya sean estudiantes, académicos o administrativos y contiene los atributos que son comunes entre todos estos.

Subclases de Persona:

1. **Estudiante**: Una persona que está inscrita o ha estado inscrita en cursos en la universidad.

- Atributos específicos: Número_Estudiante, Cohorte, Estado, último_logro, última_carga.

- Relaciones: Un estudiante puede inscribirse en secciones de cursos y tener un historial académico.

La subclase Estudiante es necesaria porque los estudiantes, además de los atributos comunes de Persona como RUN, DV, Nombre, Correo y Teléfono, tienen atributos que son exclusivos de su rol dentro de la universidad.

2. **Académico**: Una persona que realiza funciones de docencia o investigación en la universidad.

- Atributos específicos: Grado_Académico, Jerarquía, Jornada, Contrato, Cargo.
- Relaciones: Los académicos están relacionados con las secciones que dictan y revisan.

La subclase Académico es necesaria para diferenciar a los profesores e investigadores de otros roles dentro de la universidad. Los académicos tienen un rol específico como la docencia y la investigación lo que requiere atributos que no son necesarios para los estudiantes o administrativos.

3. **Administrativo**: Una persona que ocupa un cargo administrativo en la universidad.

- Atributos específicos: Cargo, Contrato, Departamento, Jerarquía, Grado_Académico.

La subclase Administrativo se utiliza para representar al personal que tiene funciones administrativas dentro de la universidad. Este tipo de Persona tiene responsabilidades distintas a las de los estudiantes o académicos, por lo que es necesario capturar sus características de manera separada.

Esquema Relacional

Persona (RUN: int, DV: int, Nombre: string, Correo: string, Telefono: int, ID_Persona: int)

Estudiante (Numero_Estudiante: int, Cohorte: string, Estado: booleano, Ultimo_logro: string, Ultima_carga: string, Persona.ID_Persona: int, Persona.RUN: int, Persona.DV: int, Persona. Nombre: string, Persona.Correo: string, Persona.Telefono: int)

Academico (Jerarquia: string, Grado_academico: string, Jornada: string, Contrato: string, Cargo: string, Persona.ID_Persona: int, Persona.RUN: int, Persona.DV: int, Persona. Nombre: string, Persona.Correo: string, Persona.Telefono: int)

Administrativo (Cargo: string, Departamento: string, Contrato: string, Jerarquia: string, Grado_Academico: string, Persona.ID_Persona: int, Persona.RUN: int, Persona.DV: int, Persona. Nombre: string, Persona.Correo: string, Persona.Telefono: int)

Curso (Codigo_curso: string, Nombre: string, Nivel: string, Caracter: string, Ciclo: string, Cursos_minimos: string, ID_Curso: int)

Prerrequisitos (Codigo_Curso: string, Ciclo: string, Curso.ID_Curso: int)

Equivalencias (Codigo_Curso: string, Ciclo: string, Curso.ID_Curso: int)

Plan_de_Estudios (Inicio: date, Codigo_plan: string, Nombre: string, Modalidad: string, Facultad: string, ID_PdE: int)

Carrera (Nombre_Carrera: string, Estudiante.ID_Persona: int, Plan_de_estudios.ID_PdE: int)

Historial Académico (Calificacion: string, Nota: booleano, Periodo_Academico: string, Persona.ID_Persona: int, Curso.ID_Curso: int)

Programacion_Academica (Curso: string, Secciones: int, Profesor: string, Vacantes: int, Sala: string, Modulo: int, Ciclo: string, Persona.ID_Persona: int, Curso.ID_Curso: int)

Carga de datos

Para cargar y validar los datos, primero se debe correr el archivo *pasar_csv.py*, el cual se encarga de generar un archivo csv, de cada hoja del archivo entregado por enunciado *Datos proyecto E2.xlsx*. Luego de esto, se debe correr el archivo corredor_general.php, el cuál correrá los archivos 7 archivos cargadores que se encargan de validar y separar los datos correctos para las tablas proporcionadas en el Excel. Al terminar de correr, se generarán dos nuevos archivos csv por cada cargador, los cuales corresponden a un archivo con los datos rescatados y editados (correcto_....csv) y otro archivo con los datos con errores que no se pudieron reparar (errores_....csv).

Validación de datos

A continuación, se muestran las principales validaciones que se realizaron en cada uno de los archivos.

Hoja prerrequisitos:

Se normalizaron los nombres de las asignaturas en mayúsculas y, en los casos donde no había nivel, se colocó una "X". Además, se estandarizaron valores como "Ingreso" y "Egreso" para asegurar que estuvieran escritos de forma consistente en todas las filas.

Hoja Notas:

Se verificó que la calificación coincidiera con la nota; si no, se ajustó la calificación en función de la nota. Se eliminaron los espacios entre los valores y se estandarizaron las asignaturas en mayúsculas para evitar problemas de formato. Además, en casos donde el periodo de la asignatura es "2024-02", se aceptaron calificaciones y notas nulas, pero se reemplazaron por una "X".

Hoja planes:
Se validó que no hubiera campos nulos y se convirtieron los nombres de los planes a mayúsculas para asegurar consistencia y evitar errores de formato.

Hoja Asignaturas:
Las asignaturas se normalizaron en mayúsculas. También se verificó que el ID de la asignatura correspondiera con el plan (basado en los primeros tres dígitos del ID). Por último, las filas sin nivel se descartaron.

Hoja estudiantes:
Se validó que no existieran campos nulos en esta hoja. Además, dado que en excel los nombres estaban en columnas separadas, se decidió juntar el primer y segundo nombre de cada estudiante en una sola columna.

Hoja planeación:
Se estandarizaron las mayúsculas para las sedes y los nombres de los docentes. También se verificó que el número de inscritos en cada curso no superara el cupo disponible; en caso contrario, la fila fue descartada.

Hoja docentes planificados:
Se verificó que los números telefónicos tuvieran exactamente 9 dígitos y solo contuvieran números. Además, se descartaron filas con campos nulos. Se validó que el RUN solo contuviera números. En el caso de los correos electrónicos, se corrigieron aquellos con doble "@" para dejar solo uno, y se ajustó el formato de los correos institucionales para que terminen en "@lamejor.com".

Conexión Bananer
En nuestro servidor grupal se encuentra una carpeta llamada bananer, la cual contiene todo lo necesario para nuestra base de datos.

Configurar la Base de Datos
- Crear la base de datos en MySQL:
    1. Abre tu terminal o consola de comandos.
    2. Accede a MySQL con tu usuario y contraseña: mysql -u tu_usuario -p
    3. Crea la base de datos: CREATE DATABASE bananer;
    4. Usa la base de datos: USE bananer;

- Ejecutar el script de creación de tablas:
    - Importa el archivo tablas.sql para crear las tablas necesarias.

- Desde la terminal de MySQL, ejecuta: SOURCE /ruta/a/tu/archivo/tablas.sql;

Configurar el Archivo de Conexión

- Editar el archivo db_connection.php:
  - Asegúrate de que las credenciales para la base de datos sean correctas.
  - Cambia los siguientes parámetros en el archivo:
  -

```
$host = 'localhost'; // Cambia si tu base de datos está en otro servidor
$db = 'bananer';     // Nombre de tu base de datos
$user = 'tu_usuario'; // Usuario de la base de datos
$password = 'tu_contraseña'; // Contraseña de la base de datos
```

Ejecutar el Servidor Web

- Iniciar el servidor PHP:
  - Navega al directorio de tu aplicación Bananer usando la terminal: cd /ruta/a/tu/proyecto/bananer
  - Inicia el servidor PHP en el puerto 8000 (o el puerto que prefieras): php -S localhost:8000

6. Acceder a la Aplicación

- Abrir el navegador:
  - Accede a la aplicación ingresando la siguiente URL en tu navegador: http://localhost:8000