

Módulo 2: Práctica

Trabajo Práctico: Identificación de Componentes del Modelo Conceptual

- **Alumno:** Patricio Sussini Guanziroli
- **Materia:** Bases de Datos I
- **Tutor:** Constanza Uño

Escenario Propuesto:

Consideren el diseño de un sistema de información simple para gestionar una biblioteca universitaria pequeña. Esta biblioteca presta libros a estudiantes y profesores. Cada libro es identificado por un código único y tiene un título, autor(es), editorial y un número de ejemplares disponibles. Los estudiantes tienen un número de legajo único, nombre completo, dirección y correo electrónico.

Los profesores también tienen un identificador único, nombre completo y departamento. Un estudiante o profesor puede tomar prestados varios libros a la vez, y un mismo libro (si tiene varios ejemplares) puede prestarse a diferentes personas. Es importante registrar quién prestó qué libro y la fecha en que se realizó el préstamo. Se necesita saber qué libros están prestados en un momento dado y a quién.

1. Identificación de Entidades:

- **Libro:** es identificado por un código único, y sus atributos son el título, editorial y cantidad disponible. Es una entidad porque es posible describir sus atributos al igual que las demás.
- **Autor:** Identificado por id_autor. Sus atributos son nombre y apellido. Se relaciona con la cantidad de libros que ha escrito el mismo autor.
- **Estudiante:** es identificado por legajo único. Sus atributos son nombre completo, dirección e e-mail.
- **Profesor:** se identifica por el id_profesor. Sus atributos son el nombre completo y el departamento al cual pertenece.
- **Préstamo:** Es una entidad y relación clave, porque no alcanza solo con libro/persona. Se identifica por el id_prestamo. Sus atributos son fecha de préstamo y fecha de devolución. Se relaciona con uno o más libros y con Estudiante o Profesor.

2. Descripción de Atributos:

Entidad: Libro

- **codigo_libro** ⇒ Simple, Monovaluado. Es un identificador único, no se descompone. Se identifica con un código único y no con múltiples.
- **título** ⇒ Simple, Monovaluado. Es un único valor textual.
- **editorial** ⇒ Simple, Monovaluado. Se guarda como un string único.
- **cantidad_disponible** ⇒ Simple, Monovaluado. Representa un número entero.

Entidad: Autor

- **id_autor** ⇒ Simple, Monovaluado. Identificador único del autor.
- **nombre** ⇒ Simple, Monovaluado.
- **apellido** ⇒ Simple, Monovaluado.
- **libros_escritos** ⇒ Multivaluado. Un autor puede estar vinculado a varios libros. Un mismo autor puede escribir varios libros.

Entidad: Estudiante

- **legajo** ⇒ Simple, Monovaluado. Es un identificador único del estudiante.
- **nombre_completo** ⇒ Compuesto, Monovaluado. Puede descomponerse en nombre y apellido.
- **direccion** ⇒ Compuesto, Monovaluado. Puede dividirse a su vez en detalles como calle, número, ciudad, etc.
- **email** ⇒ Simple, Monovaluado. Cada estudiante tiene un solo correo registrado.

Entidad: Profesor

- **id_profesor** ⇒ Simple, Monovaluado. Identificador único de cada profesor.
- **nombre_completo** ⇒ Compuesto, Monovaluado. Se puede descomponer en nombre y apellido.
- **departamento** ⇒ Simple, Monovaluado. Cada profesor pertenece a un departamento.

Entidad: Préstamo:

- **id_prestamo** ⇒ Simple, Monovaluado. Identificador único del préstamo particular que ocurre.
- **fecha_prestamo** ⇒ Simple, Monovaluado. Fecha exacta de inicio.
- **fecha_devolucion** ⇒ Simple, Monovaluado. Puede estar vacía si el libro aún no fue devuelto.
- **libros_asociados** ⇒ Multivaluado. Un préstamo puede incluir uno o varios ejemplares de libros.
- **usuario** ⇒ Simple, Monovaluado. Cada préstamo pertenece a un único usuario.

3. Clasificación e Identificación de Relaciones:

❖ Relación: Libro – Autor

- **Descripción:** Un libro puede estar escrito por uno o varios autores, y un autor puede escribir varios libros.
- **Cardinalidad: N:M**
- **Justificación:** Ejemplo: un autor famoso puede tener varios libros; y un libro académico puede estar escrito en coautoría por varios autores.

❖ Relación: Estudiante – Préstamo

- **Descripción:** Un estudiante puede realizar varios préstamos, pero cada préstamo pertenece a un solo estudiante.
- **Cardinalidad: 1:N**
- **Justificación:** Un mismo estudiante puede tener varios préstamos registrados en distintos momentos, pero un préstamo no puede pertenecer a más de un estudiante.

❖ Relación: Profesor – Préstamo

- **Descripción:** Un profesor puede realizar varios préstamos, pero cada préstamo pertenece a un único profesor.
- **Cardinalidad: 1:N**
- **Justificación:** Igual que con los estudiantes: un profesor puede sacar varios libros, pero un préstamo no puede registrarse para más de un profesor.

❖ Relación: Préstamo – Libro

- **Descripción:** Un préstamo puede incluir uno o varios libros y un libro puede ser prestado en distintos préstamos a diferentes usuarios.
- **Cardinalidad: N:M**
- **Justificación:** Si un préstamo se define como la acción de retirar uno o varios ejemplares en una fecha, se requiere una relación N:M entre Préstamo y Libro. Se detalla en una tabla intermedia.

4. Propuesta de Claves Primarias y Candidatas:

❖ Entidad: Libro

- **Clave primaria (PK):** codigo_libro
 - Justificación: cada libro en el sistema tiene un código único asignado, lo que garantiza su identificación sin ambigüedades. No cambia y no se repite.
- **Clave candidata:** combinación de (titulo, editorial, autor).
 - No es la mejor opción porque distintos libros pueden compartir título o autor, y las editoriales no garantizan unicidad confiable.

❖ Entidad: Autor

- **Clave primaria (PK):** id_autor
 - Justificación: un identificador numérico o alfanumérico único evita ambigüedades entre autores con nombres iguales o similares.
- **Clave candidata:** combinación de (nombre, apellido).
 - Problema: no garantiza unicidad, ya que pueden existir autores con nombres parecidos.

❖ Entidad: Estudiante

- **Clave primaria (PK):** legajo
 - Justificación: el número de legajo universitario es único para cada estudiante, no se repite y es más estable que atributos como nombre o email.
- **Clave candidata:** email.
 - Podría ser única en la práctica, pero no es recomendable como PK porque puede cambiar si el estudiante modifica su dirección de correo.

❖ **Entidad: Profesor**

- **Clave primaria (PK):** id_profesor
 - Justificación: es un identificador único asignado por la universidad. Evita ambigüedades y es más estable que el nombre o el departamento.
- **Clave candidata:** combinación de (nombre, apellido, departamento).

❖ **Entidad: Préstamo**

- **Clave primaria (PK):** id_prestamo
 - Justificación: un identificador único para cada transacción de préstamo garantiza que pueda diferenciarse de cualquier otro préstamo, incluso si es del mismo usuario y con los mismos libros.
- **Clave candidata:** combinación de id_usuario, fecha_prestamo, codigo_libro.

5. Identificación de Claves Foráneas y su Rol:

❖ Relación Libro – Autor

- **Clave foránea:** codigo_libro en la tabla intermedia **Libro_Autor** → referencia a codigo_libro en **Libro**.
- **Clave foránea:** id_autor en **Libro_Autor** → referencia a id_autor en **Autor**.
- **Rol:** garantiza que no se pueda registrar la relación de un libro con un autor inexistente.

❖ Relación Estudiante – Préstamo

- **Clave foránea:** legajo en **Préstamo** → referencia a legajo en **Estudiante**.
- **Rol:** asegura que un préstamo siempre esté asociado a un estudiante válido.

❖ Relación Profesor – Préstamo

- **Clave foránea:** id_profesor en **Préstamo** → referencia a id_profesor en **Profesor**.
- **Rol:** asegura que un préstamo siempre esté asociado a un profesor válido.

❖ Relación Préstamo – Libro

- **Clave foránea:** id_prestamo en la tabla intermedia **Detalle_Prestamo** → referencia a id_prestamo en **Préstamo**.
- **Clave foránea:** codigo_libro en **Detalle_Prestamo** → referencia a codigo_libro en **Libro**.
- **Rol:** asegura que solo se registren préstamos sobre libros existentes y vinculados a un préstamo real.

El beneficio de usar claves foráneas es que mantiene la **integridad de las referencias**:

- No se podrá registrar un préstamo de un libro que no exista en la base de datos.
- No se podrá asignar un préstamo a un estudiante o profesor inexistente.
- Garantiza coherencia en las relaciones, evitando “huérfanos” en las tablas.