| UNIDADES A EVALUAR DEL PROGRAMA DE LA MATERIA:  • [Unidad 1: ¿Qué Es Una Base De Datos?](#bookmark=id.gjdgxs)  • [Unidad 2: Modelo Entidad Relación](#bookmark=id.30j0zll)  • [Unidad 3: Modelo Relacional, Mantener La Integridad Y La Consistencia](#bookmark=id.1fob9te)  • [Unidad 5: Mantener y documentar una base de datos](#bookmark=id.3znysh7) |
| --- |
| RESULTADOS DE APRENDIZAJE:  • [Diseña] + [ el Modelo Conceptual de Datos] + [ para representar las entidades y las relaciones] + [utilizando un Modelo Conceptual]  • [Elabora] + [un modelo Lógico de Datos]+ [para implementar una base de datos] + [utilizando el Modelo Relacional] |
| CRITERIOS DE CALIFICACIÓN:  Para aprobar el examen deberá sumar 60 puntos de un total de 100 (Teórico-Práctico), siendo, al menos el 60% de los aspectos conceptuales teóricos y al menos el 60% de los aspectos prácticos. |
| CRITERIOS DE RESOLUCIÓN:  Los alumnos/as recibirán la consigna del examen en la fecha de evaluación prevista por el cronograma de la asignatura.  El examen constará de 2 instancias:  • *Semana 1:* Entrega de las consignas y explicación de la metodología de evaluación por parte del docente a los alumnos/as.  • *Semana 2:* Entrega de la evaluación digital (archivo en MS Word y archivo generado con el software propuesto para el desarrollo de la parte práctica) por parte de los alumnos/as al docente. |

**PARTE TEÓRICA (100 PTS)**

Debe obtener al menos 60/100 para la aprobación de esta parte.

**UNIDAD 1: ¿QUÉ ES UNA BASE DE DATOS? (30 PTS)**

1. ***Concepto de base de datos. Ventajas del enfoque de base de datos frente a un sistema de Archivos.***

Una base de datos (o banco de datos) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Las ventajas de una base de datos frente a un sistema de archivos son:

* independencia de los datos;
* reduce la redundancia;
* permite un control centralizado;
* conservación de la integridad de los datos;
* permite el control de acceso;
* permite compartir datos;
* permite manejar requerimientos contradictorios;
* permite mantener los datos frente a actualizaciones de software y;
* permite aplicar restricciones de seguridad.

En resumen, las ventajas más destacadas frente a un sistema de archivos (que solo es una colección de registros, como por ejemplo una guía de teléfono) es que los datos están integrados, lo que implica una redundancia parcial o nula de los datos, y que los datos son compartidos, porque se puede acceder a ellos en forma concurrente por diferentes usuarios.

2.***Desarrolle el Sistema de Administración de Base de Datos (DBMS). Ejemplificar mediante gráfico.***

El Sistema de Administración de una Base de Datos (también llamado Database Management System o DBMS por sus siglas en inglés) es una capa de Software entre la base de datos física, las aplicaciones y los usuarios. Este sistema de administración se encarga de las siguientes tareas:

1. Desarrollo de base de datos.
2. Consulta de bases de datos.
3. Mantenimiento de bases de datos.
4. Desarrollo de aplicaciones.

Su objetivo principal es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera práctica y eficiente. Estos sistemas, se diseñan para gestionar grandes cantidades de información, lo que implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para su manipulación. Sumado a ello, deben garantizar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o de los intentos de acceso no autorizados.

Gráficamente lo podemos describir de la siguiente forma:

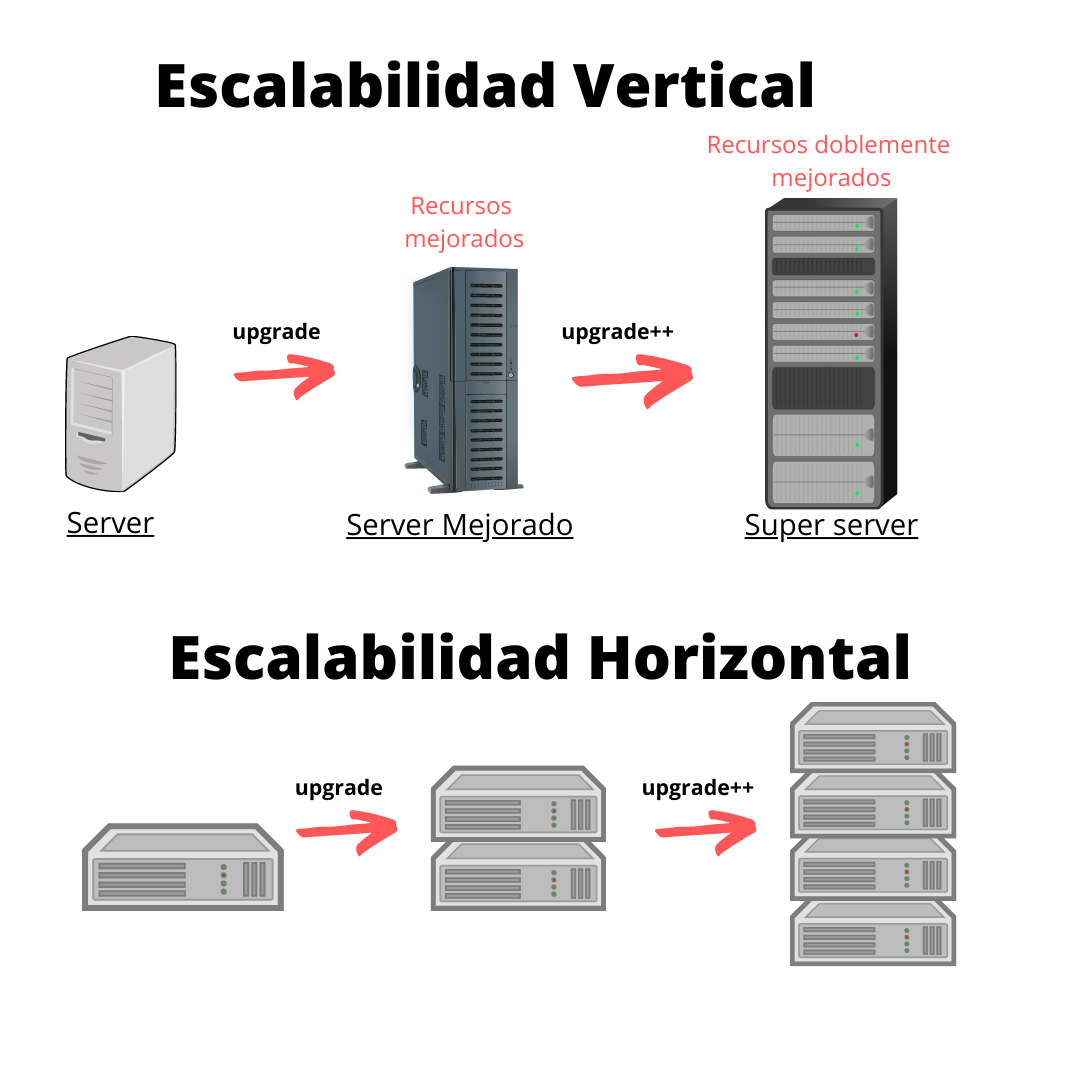


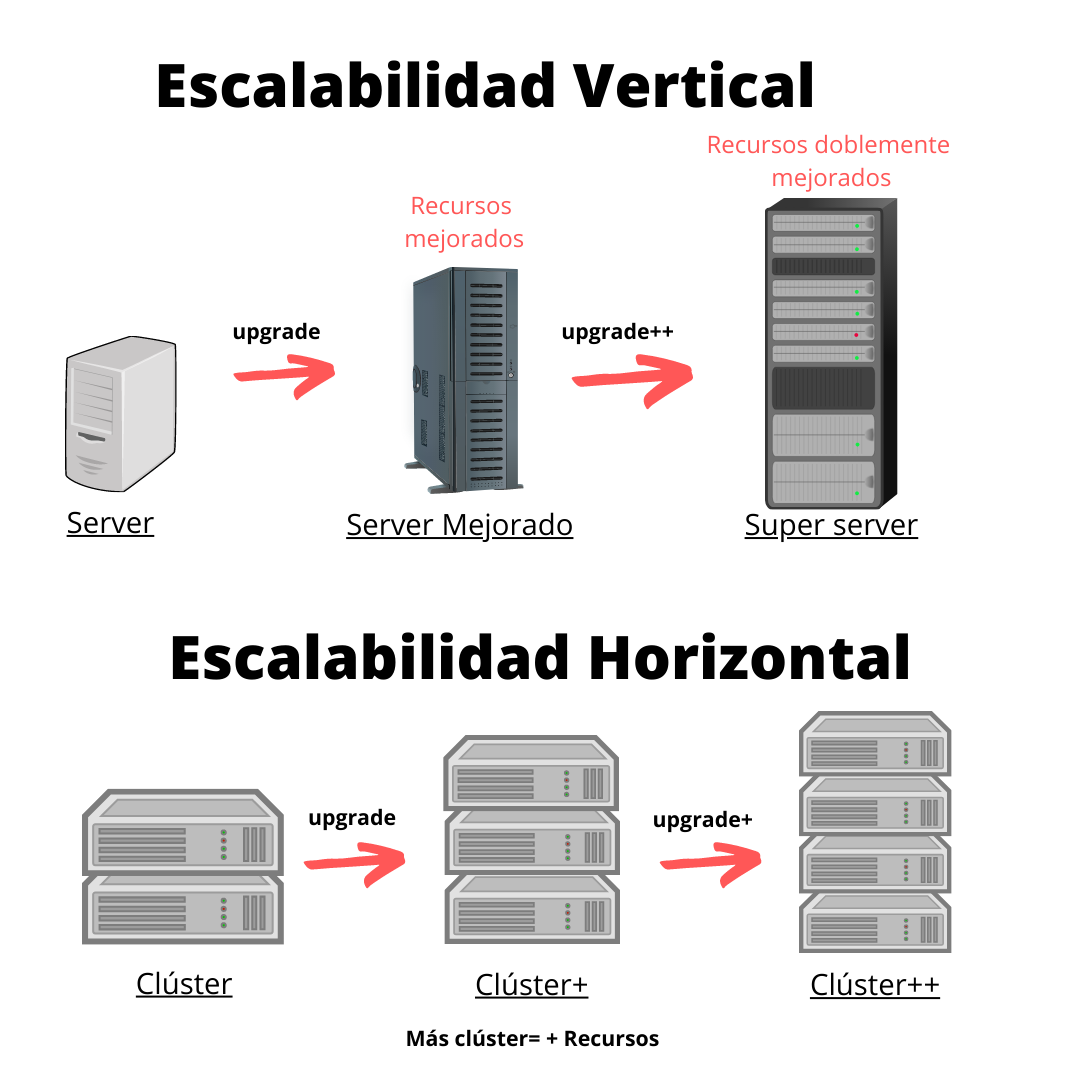
**3**. ***Diferencia entre Escalabilidad Vertical y Escalabilidad Horizontal. Desarrolle y Grafique.***

Tanto la escalabilidad vertical (scaling up) como la horizontal (scaling out) indican la habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida.

La diferencia principal entre ambas radica en que la escalabilidad vertical trata de mejorar cada nodo con más CPU, más memoria, más máquina y es mucho más sencilla (aunque también limitada), mientras que la otra, la escalabilidad horizontal, trata de incrementar el número de nodos, lo que implica más máquinas, distribuir y descentralizar, lo cual resulta más complicado y en su mayoría, requiere de una mayor inversión.

***Ejemplo de escalamiento vertical:***



***Ejemplo de escalamiento horizontal***

***4. Describa el Enfoque Jerárquico versus Enfoque de Red.***

El enfoque jerárquico se caracteriza por su capacidad de disminuir la complejidad y lograr buenas soluciones para economizar tiempo y requerimientos computacionales; se lo suele utilizar para la planificación y control de la producción. Las principales características de las bases de datos jerárquicas son:

* se organizan en forma de árbol invertido;
* tiene un único nodo raíz, nodos padre e hijos y;
* El árbol se organiza en un conjunto de niveles (donde el nivel 0 se corresponde al nodo raíz, el cual es el nivel más alto de la jerarquía).

El enfoque de red se caracteriza por estar conformado por una colección o set de registros (un conjunto de nodos), los cuales están conectados entre sí por medio de enlaces en una red. Una base de datos de red es una base de datos jerárquica tradicional, excepto que permite que cada objeto tenga varios padres en lugar de uno sólo (como ocurre en el enfoque jerárquico). Esto le permite modelar relaciones más complejas.

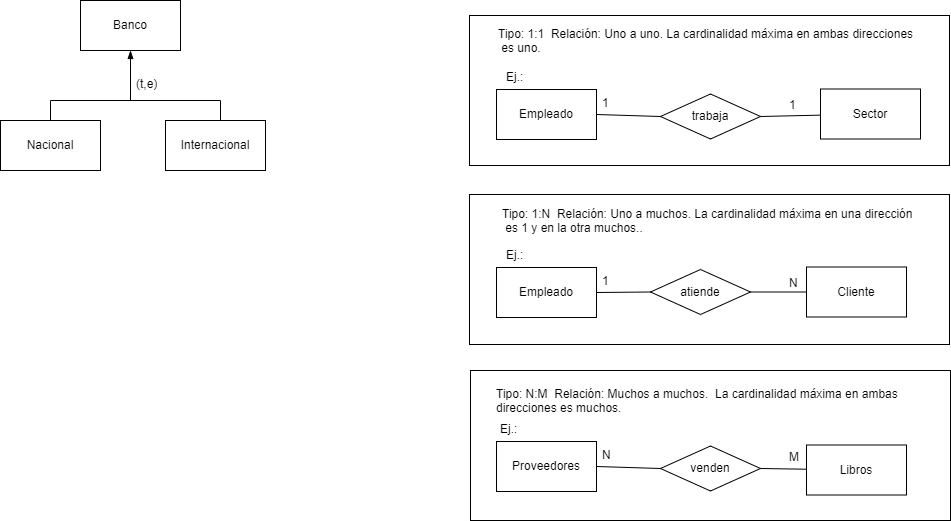
Las bases de datos de red se pueden representar como un gráfico en lugar de una estructura de árbol y se la suele utilizar cuando se requiere una relación más flexible entre los distintos registros de una base de datos ya que facilitan el acceso a la información porque permite realizar la consulta de los datos de manera rápida y sencilla.

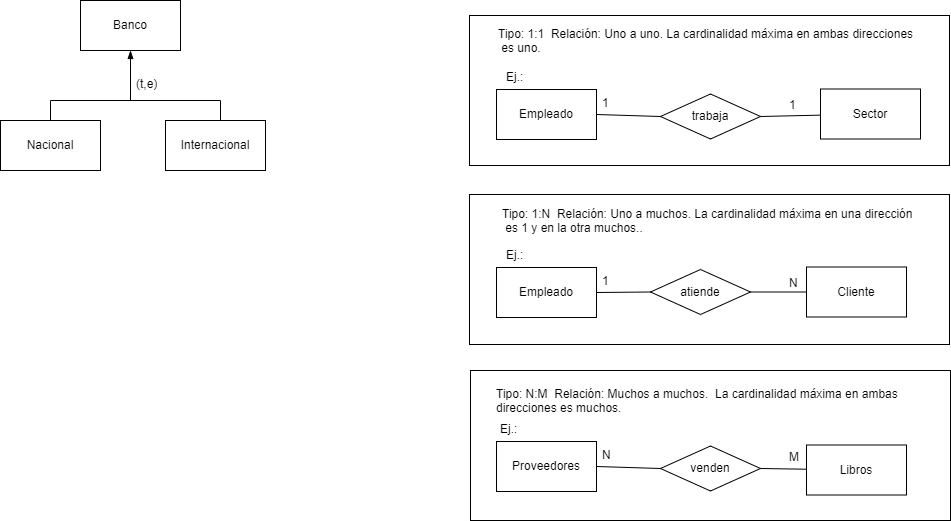
En síntesis, un sistema jerárquico no es más que una red cuya configuración obedece a un conjunto de reglas específicas (sólo existe un nodo raíz y un nodo padre que puede tener varios hijos, pero sólo un padre), por lo que este modelo resulta menos flexible que el modelo de red.

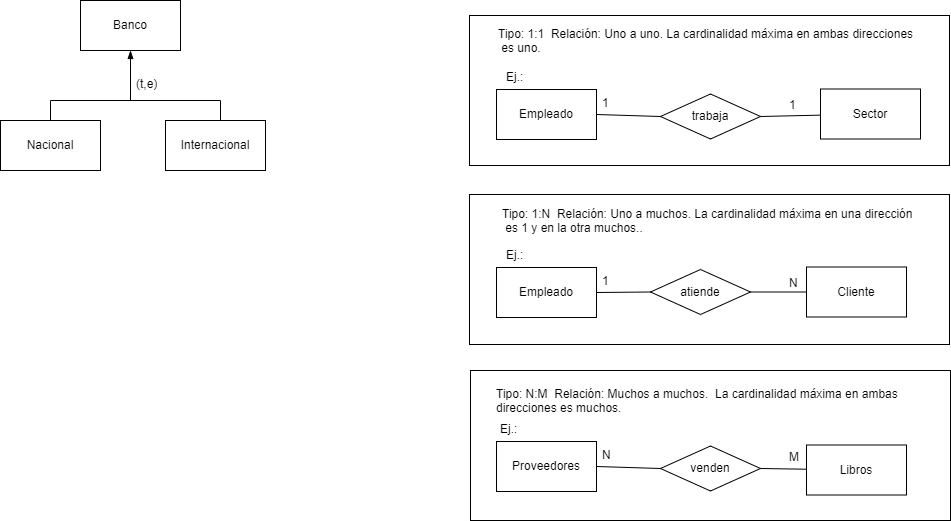
***5. Describa, ejemplifique y grafique:***

***a. ¿Qué relación hay entre cardinalidad y relación?***

La relación que existe entre ambas es que la cardinalidad en una relación es el número de veces que una entidad aparece asociada a otra entidad. Según el número de entidades relacionadas (o razón de cardinalidad), se pueden definir tres tipos de relaciones, las cuales se describen y ejemplifican a continuación:







***b. ¿Qué es el dominio de un atributo?***

El dominio de un atributo describe los valores disponibles de un tipo de campo (establece ciertos límites). Se utilizan para limitar los valores permitidos en cualquier atributo concreto de una tabla o clase de entidad.

Se diferencia con el tipo de dato porque el dominio establece los valores disponibles de un tipo de campo, mientras que el tipo de dato colabora con delimitar los valores permitidos disponibles para ese tipo de dato.

Ejemplo:

Para la siguiente entidad, denominada Empleado, cuyos atributos son nombre, teléfono y edad, en la descripción del dominio voy a limitarlos de la siguiente forma:

| **Atributo** | **Descripción dominio** |
| --- | --- |
| **Nombre** | cadena de hasta 30 caracteres alfabéticos |
| **Teléfono** | cadena de hasta 9 caracteres numéricos |
| **Edad** | números reales entre 18 y 80. |

***6. Concepto de la Independencia física y lógica de datos.***

En cuanto al concepto de la independencia física y lógica de los datos, cuando nos referimos a la independencia física, hacemos referencia a la capacidad de modificar la definición física y/o el método de almacenamiento, sin afectar el esquema, las aplicaciones de usuario ni vistas; en cambio, cuando nos referimos a la independencia lógica de los datos, hacemos referencia a la capacidad de modificar el esquema sin afectar las aplicaciones de usuario ni las vistas.

**UNIDAD 2: MODELO ENTIDAD RELACIÓN (30 PTS)**

1. ***Explique y ejemplifique Cobertura en Generalización. Caso cobertura total y exclusiva.***

Las jerarquías de generalización presentan la propiedad de cobertura, la misma puede ser parcial o total y exclusiva o superpuesta. Cuando en un diagrama no se encuentra especificado el tipo, su valor por omisión es total y exclusiva.

Como se solicita en el enunciado, se procederá a ejemplificar y explicar la cobertura total y exclusiva de una entidad BANCO.

La cobertura total permite especificar una restricción entre el tipo de entidad genérica y sus tipos de entidad subconjunto, donde todos los elementos del tipo de entidad genérico deben pertenecer a alguno de sus tipos de entidad subconjunto (si existiera la posibilidad que pertenecieran, sería una cobertura parcial). En el caso ejemplificado, todos los bancos son nacionales o internacionales, pero no ambas cosas simultáneamente; es decir que todos los bancos están clasificados como nacionales o internacionales.

La cobertura exclusiva, también permite especificar una restricción entre los tipos de entidad subconjunto, donde los elementos que pertenecen a un tipo de entidad subconjunto, no pueden pertenecer a otro tipo de entidad subconjunto (en caso de que pudiera pertenecer a más de una entidad del subconjunto, sería superpuesta). En el caso ejemplificado, un banco puede clasificarse como Nacional o Internacional, no ambos.

Diagrama

Descripción generada automáticamente

**2. *Realizar un cuadro con las características de los siguientes modelos (Ejemplifique en cada caso):***

***a. Modelo Entidad Relación***

***b. Modelo Conceptual***

***c. Modelo Relacional***

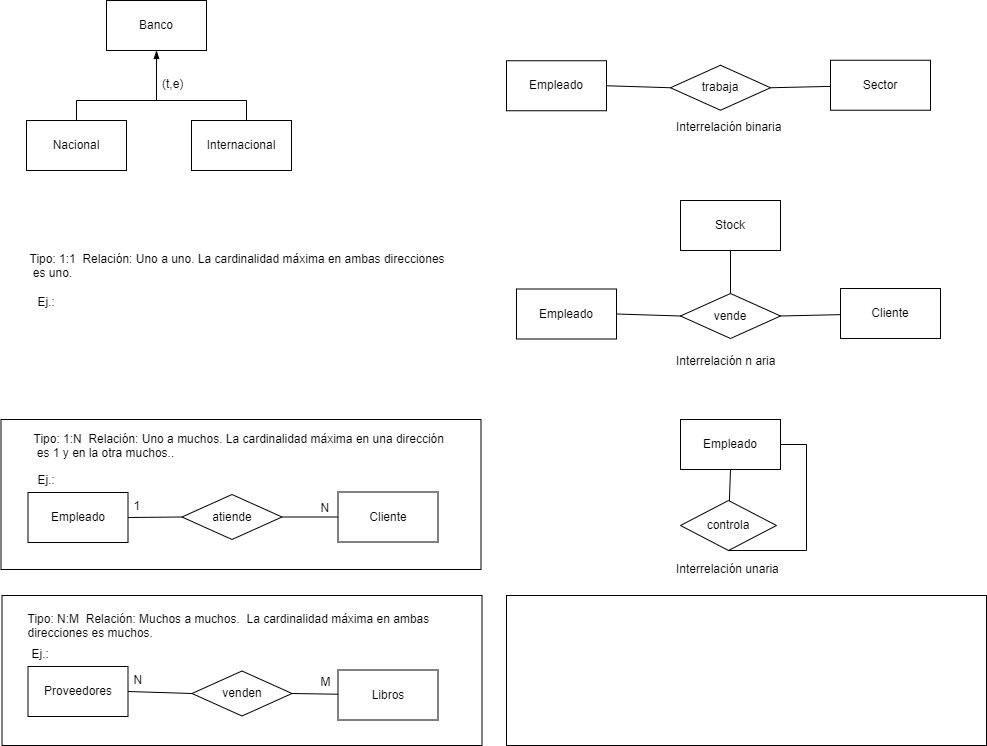
***d. Modelo Lógico***

| Modelo | **Entidad- relación** | **Relacional** | **Lógico** | **Conceptual** |
| --- | --- | --- | --- | --- |
| Enfoque | Entidades y sus relaciones; se describe el  problema textualmente | Modelar problemas reales y administrar datos  dinámicamente | Operaciones | Estructuras de datos y restricciones de  integridad |
| Utilización | enfoque natural | Dinamismo | Implementación en un gestor de base de  datos. | Etapa de análisis |
| Característica | Busca posibles relaciones y atributos entre las entidades | Pasa por un proceso llamado normalización | El ejemplo más típico es el Modelo Relacional | Ejemplo, el MODELO ENTIDAD/RELACIÓN. |
| ¿Qué es? | Un modelo que se basa en el modelo conceptual | El ejemplo más típico es el Modelo Relacional | Representación abstracta de una posible implementación (sin estar  vinculado a una implementación específica). | Representación de alto nivel de los requisitos comerciales y los  conjuntos y relaciones de datos conectados. |

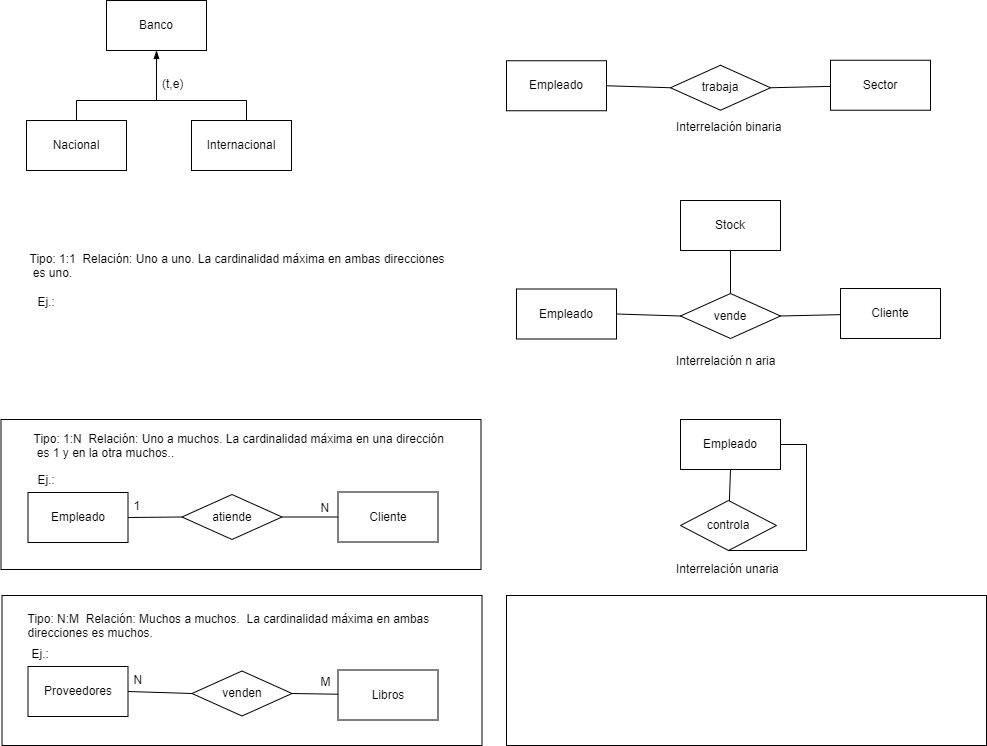
**3.** **Explique y ejemplifique: Interrelación Unaria, Binaria y n-aria.**

Se procederá a explicar y ejemplificar las relaciones que puede tener una entidad.

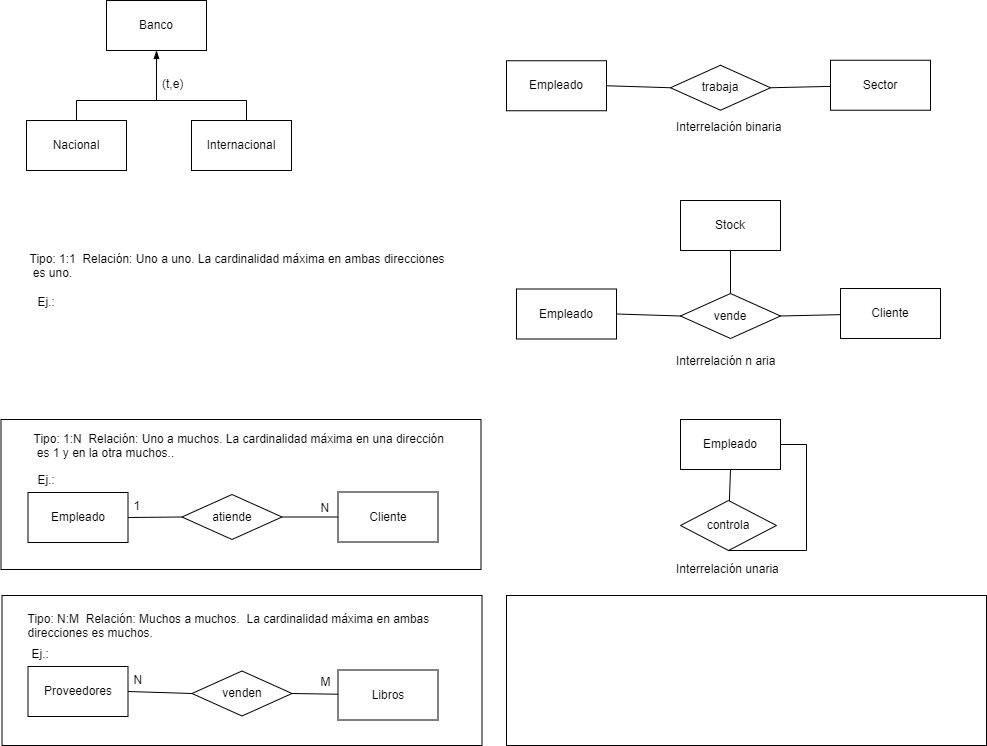
*Interrelación unaria:* es una relación que se da en una entidad consigo misma. En el ejemplo, se detalla el caso en el que un empleado, el supervisor, se encarga de controlar al resto.



*Interrelación binaria:* es una relación que se da entre dos entidades. En este ejemplo, la relación se da entre la entidad Empleado que trabaja en Sector.



*Interrelación n aria:* es una relación que se da entre más de dos entidades (n+1). En este ejemplo la relación se da entre la entidad empleado, stock y cliente; estas entidades se relacionan cuando hay una venta.



**4. ¿Cuál es la diferencia entre el Diseño Conceptual y el Diseño Lógico?**

La diferencia fundamental es que elmodelo conceptual se orienta a la descripción de estructuras de datos y restricciones de integridad y se usa durante la etapa de Análisis de un problema dado (representando los elementos que intervienen en ese problema y sus relaciones); en cambio, el modelo lógico está orientado a las operaciones más que a la descripción de una realidad y está implementado en algún Gestor de Base de Datos. El ejemplo más típico de modelo lógico es el Modelo Relacional.

Sintetizando, el modelo de datos lógico es una representación abstracta de una posible implementación (sin estar vinculado a una implementación específica); en cambio, el modelo de datos conceptual es una representación de alto nivel de los requisitos comerciales y los conjuntos y relaciones de datos conectados.

**UNIDAD 3: MODELO RELACIONAL, MANTENER LA INTEGRIDAD Y LA CONSISTENCIA (20 PTS)**

***1. Explique y ejemplifique Clave Primaria, Superclaves, Claves Candidatas y Claves Foráneas.***

Considerando que una clave es un conjunto de atributos que identifican de forma unívoca una entidad, se procede a explicar y ejemplificar los siguientes tipos de claves solicitados:

*Clave Primaria:*también se la suele llamar principal; es un atributo o un conjunto de atributos que identificarán de forma única cada tupla en una relación. Contiene la información que necesita el gestor de base de datos para modificar el registro. Se caracteriza por aplicar unicidad a la tabla, Implementar integridad referencial y mejorar sustancialmente el acceso a los datos.

***Ejemplos:*** DNI, CUIT e ID (identificador único).

*Superclaves:*es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de entidades.

***Ejemplos:*** DNI+Cód.Asignatura o Id\_cliente+Cód.Artículo .

*Claves candidatas:*son aquellas que podrían utilizarse como claves primarias (no pueden existir dos o más registros con el mismo valor en el campo o campos que componen la clave candidata). Estas claves se diferencian de las claves alternativas porque son aquellas claves candidatas que no han sido seleccionadas como clave primaria, pero que también pueden identificar de forma única a una fila dentro de una tabla.

***Ejemplos:*** CUIL, DNI y Cód\_Cliente.

*Claves foráneas:*una clave foránea es una clave externa; es un campo (o conjunto de campos) cuyos posibles valores están limitados a los que se hayan definido en otra tabla como clave principal.

***Ejemplos:*** teniendo a “Pacientes” (con su clave primaria Id\_Paciente) y “Turnos” (con su clave primaria Id\_Turnos):

-> Id\_Paciente (llave primaria de Pacientes), será clave foránea de Turnos.

**2. ¿Qué son las Operaciones fundamentales del álgebra relacional? Desarrollo ejemplo de Producto cartesiano versus Proyección.**

El álgebra relacional es un conjunto de operaciones que describen paso a paso cómo computar una respuesta sobre las relaciones.

Estas operaciones (de tipo procedimental porque indican qué obtener y cómo obtenerlo a diferencia del cálculo relacional que es de tipo declarativo), describen la manipulación de datos. Se utilizan como una representación intermedia de una consulta a una base de datos y, debido a sus propiedades algebraicas, sirven para obtener una versión más optimizada y eficiente de dicha consulta.

Las operaciones fundamentales del álgebra relacional son:

**Tradicionales:**

• Unión: UNIÓN

• Intersección: INTER

• Diferencia: MENOS

**• Producto cartesiano: VECES**

**Especiales:**

• Selección: DONDE

**• Proyección**

• Reunión: REUNIÓN

• División: DIVISIÓN

Si una base de datos no soporta algunas de estas operaciones, se considera un sistema de base de datos semi relacional.

Por otra parte, se procede a realizar el desarrollo de las operaciones solicitadas:

*Proyección (Project):*

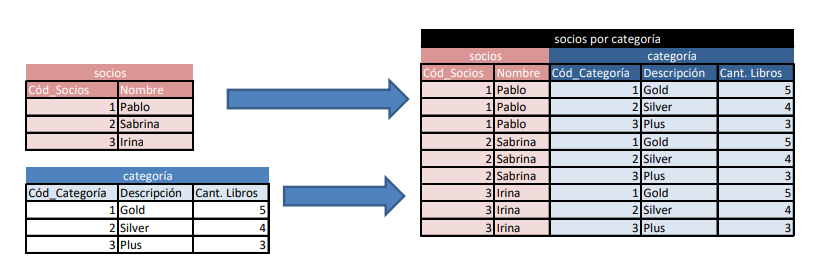
Se utiliza para producir una nueva relación desde una tabla. Esta nueva relación contiene sólo algunos de los atributos de la tabla, es decir, realiza la selección de algunas de las columnas de una tabla en función de lo solicitado. En el ejemplo podemos ver como de tabla Curso, se realiza una seleccion del ID y Nombre.

| TABLA CURSO | | | |
| --- | --- | --- | --- |
| ID | NOMBRE | EDAD | MATERIA |
| 123 | Julieta | 33 | Geografía |
| 234 | Tomas | 28 | Historia |
| 143 | Maria | 22 | Matemática |

π (id,nombre) Curso —>

| ID | NOMBRE |
| --- | --- |
| 123 | Julieta |
| 234 | Tomas |
| 143 | Maria |

*Producto cartesiano:*

Permite combinar información de dos relaciones. Al aplicarse un producto a las tablas, las columnas se suman y se multiplican las filas. En el ejemplo detallado a continuación podemos ver cómo se realiza el producto cartesiano de socios y categoría. 

En síntesis, la proyección permite realizar una nueva relación de una relación (en el ejemplo gráfico se realiza con los atributos ID y Nombre, de la tabla Curso); en cambio, el producto cartesiano nos permite generar una nueva relación con todos los atributos de relaciones existentes, ya que se suman y multiplican las filas (en el ejemplo gráfico, se crea la nueva relación de socios por categoría usando como base las tablas socios y categoría).

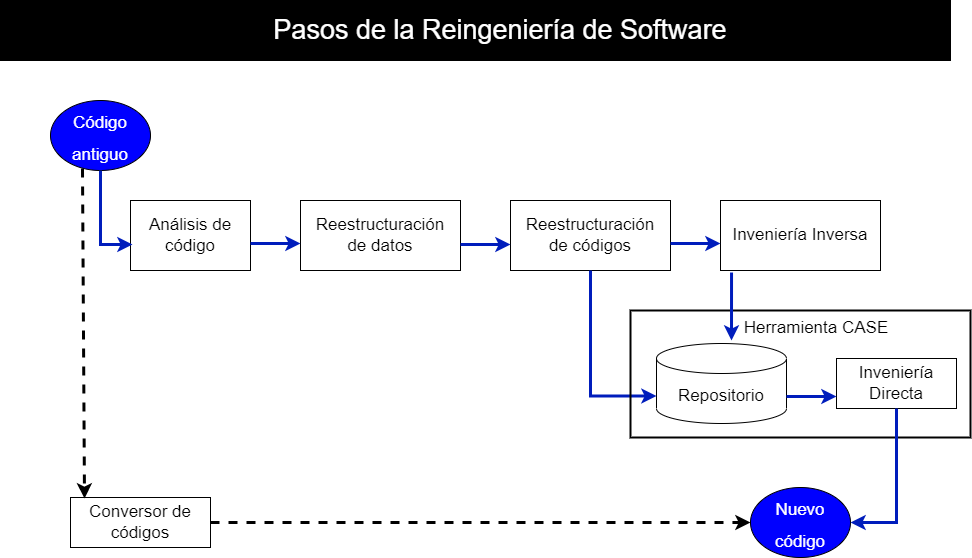
**UNIDAD 5: MANTENER Y DOCUMENTAR UNA BASE DE DATOS (20 PTS)**

**1. *¿Cuáles son las diferentes actividades que involucra la reingeniería del software?******Explique y grafique pasos.***

Las diferentes actividades que involucra la reingeniería del software con la finalidad de crear versiones de programas ya existentes que sean de mejor calidad y con mayor facilidad de mantenimiento son:

* *Análisis de inventarios:* los candidatos a la reingeniería aparecen cuando se ordena esta información en función de su importancia para el negocio, es entonces cuando es posible asignar recursos a las aplicaciones candidatas para el trabajo de reingeniería.
* *Ingeniería directa:* La ingeniería directa no solo recupera la información de diseño a partir del software existente, también utiliza esta información para alterar o reconstruir el sistema existente con la finalidad de mejorar su calidad global. En la mayoría de los casos el software sometido a reingeniería vuelve a implementar la función del sistema existente y también añade nuevas funciones o mejoras.
* *Reestructuración de documentos:* la documentación débil es la marca de muchos sistemas heredados. La documentación debe actualizarse pero, en general, se tienen recursos limitados; es por ello que se puede utilizar un enfoque de “documentar cuando se toque” o, cuando el sistema es crucial para el negocio y se deba volver a documentarse por completo, conviene adoptar un enfoque inteligente recortando la documentación a un mínimo esencial (cabe destacar que una organización de software debe elegir la opción más apropiada para su caso).
* *Ingeniería inversa:* es el proceso de análisis de un programa con el fin de crear una representación de programa con un nivel de abstracción más elevado que el código fuente (es un proceso de recuperación de diseño). Con las herramientas de la ingeniería inversa se extraerá del programa existente información del diseño arquitectónico y de proceso, e información de los datos.
* *Reestructuración de programas (código):* El tipo más común de reingeniería, donde se requiere analizar el código fuente empleando una herramienta de reestructuración, se indican las violaciones de las estructuras de programación estructurada, y entonces se reestructura el código (esto se puede hacer automáticamente). El código reestructurado resultante se revisa y se comprueba para asegurar que no se hayan introducido anomalías. Se actualiza la documentación interna del código.
* *Reestructuración (de datos):* es una actividad de reingeniería a gran escala que comienza con una actividad de ingeniería inversa. La arquitectura de datos actual se analiza con minuciosidad y se definen los modelos de datos necesarios, se identifican los objetivos de datos y los atributos, y después se revisa la calidad de las estructuras de datos existentes. Dado que la arquitectura de datos tiene una gran influencia sobre la arquitectura del programa, y también sobre los algoritmos que lo pueblan, los cambios en los datos darán lugar invariablemente a cambios de arquitectura o de código.

Es importante destacar que todas las organizaciones de software deberían tener un inventario de todas sus aplicaciones; de esta forma, es mucho más fácil encontrar los candidatos a la reingeniería, ya que aparecen cuando se ordena esta información en función de su importancia para el negocio y luego se asignan los recursos a las aplicaciones candidatas para el trabajo de reingeniería.

Gráficamente, los pasos de la reingeniería son: 

**2. *Desarrolle Herramientas CASE.***

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son herramientas, cada vez más sofisticadas, que nos ayudan a aplicar la reingeniería orientada a objetos.

En un mundo ideal, se podría crear un “motor” de reingeniería automatizado, pero los fabricantes de CASE han presentado herramientas que proporcionan un subconjunto limitado de capacidades para analizar el programa existente, reestructurar y regenerar los programas (lo que haría este motor automatizado), aunque se enfrentan con dominios de aplicaciones específicas. Es importante destacar que estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software, por ejemplo en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

En cuanto a su clasificación, podemos decir que no es fácil y no existe una forma única de hacerlo, pero se lo puede realizar teniendo en cuenta los siguientes parámetros:

* Plataformas que soportan;
* fases del ciclo de vida del desarrollo de sistemas que cubren;
* arquitectura de las aplicaciones que producen y;
* por su finalidad.

3. ***¿Cuáles son las herramientas para aplicar la ingeniería inversa?***

Las herramientas para aplicar ingeniería inversa son:

a- *Depuradores:* La depuración de programas es el proceso de identificar y corregir errores de programación.

b- *Herramientas de Inyección de Fallos:* la inyección de fallos es una técnica para mejorar la cobertura de una prueba de software mediante la introducción de fallos para probar las rutas de código, en particular de tratamiento de errores rutas de código, que de otra manera rara vez se seguirán. A menudo se utiliza con las pruebas de estrés y es ampliamente considerado como una parte importante del desarrollo robusto software; por ejemplo las pruebas de robustez, también conocido como Sintaxis Testing, Fuzzing o pruebas de la pelusa, es un tipo de inyección de fallos utilizada para la prueba de vulnerabilidades en interfaces de comunicación tales como los protocolos, parámetros de línea de comando, o APIs.

c) *Los Desensambladores:* un desensamblador es un programa de computador que traduce el lenguaje de máquina a lenguaje ensamblador (la operación inversa de la que hace el ensamblador).

El código fuente en lenguaje ensamblador generalmente permite el uso de constantes y comentarios del programador. Estos son generalmente eliminados, por el ensamblador, del código ensamblador a código máquina. De esta manera, un desensamblador operando sobre el código de máquina, produciría el desensamblado careciendo de estas constantes y comentarios; la salida desensamblada se vuelve más difícil de ser interpretada por un humano que el código fuente con comentarios originales.

d) *Los compiladores Inversos o Decompiladores:* un decompilador es un programa de ordenador que realiza la operación inversa a un compilador, es decir, traduce código de bajo nivel de abstracción a un lenguaje de mayor nivel de abstracción.

e) *Herramientas CASE*: estas herramientas fueron desarrolladas en el punto anterior.

**PRÁCTICA (100 PTS)**

Debe obtener al menos 60/100 para la aprobación de esta parte.

Para todos los puntos, se deberán realizar los diagramas correspondientes con un software de modelado de datos.

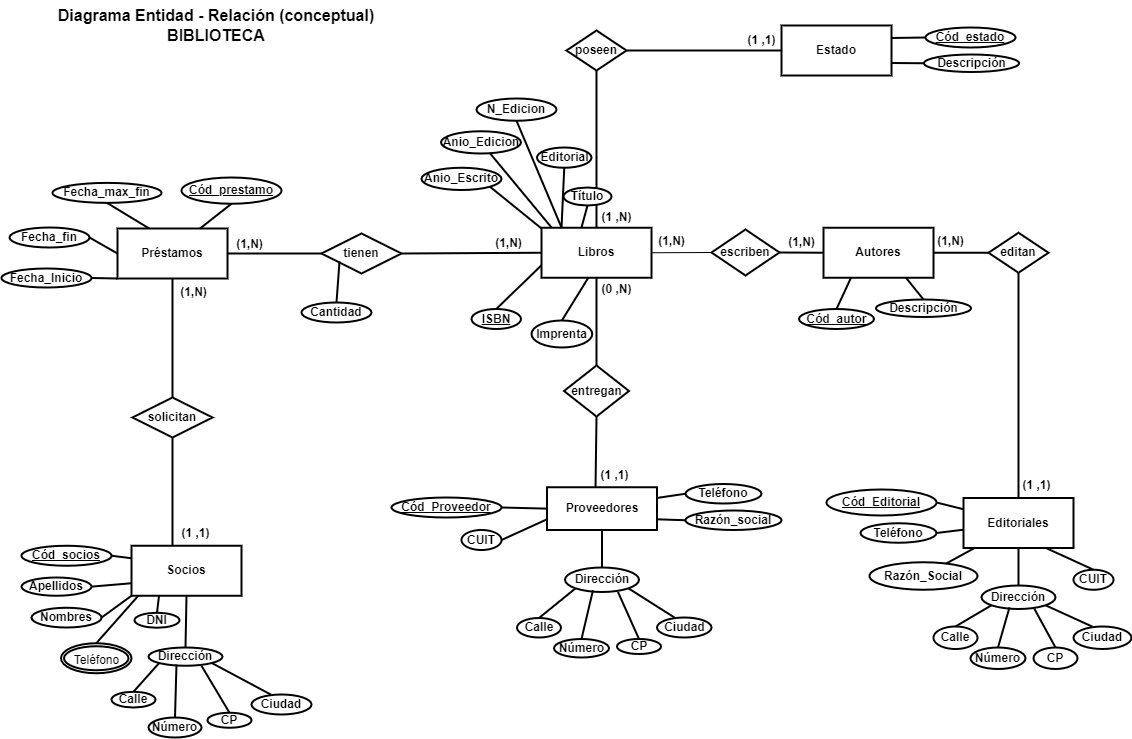
***1. Desarrolle un modelo de entidad relación (conceptual) para el siguiente enunciado*** **(30 PTS)**:

*Crear un diseño entidad relación que permita gestionar los datos de una biblioteca de modo que las personas socias de la biblioteca disponen de un código de socio y datos en general, como ser: dni, dirección, teléfono, nombre y apellidos.*

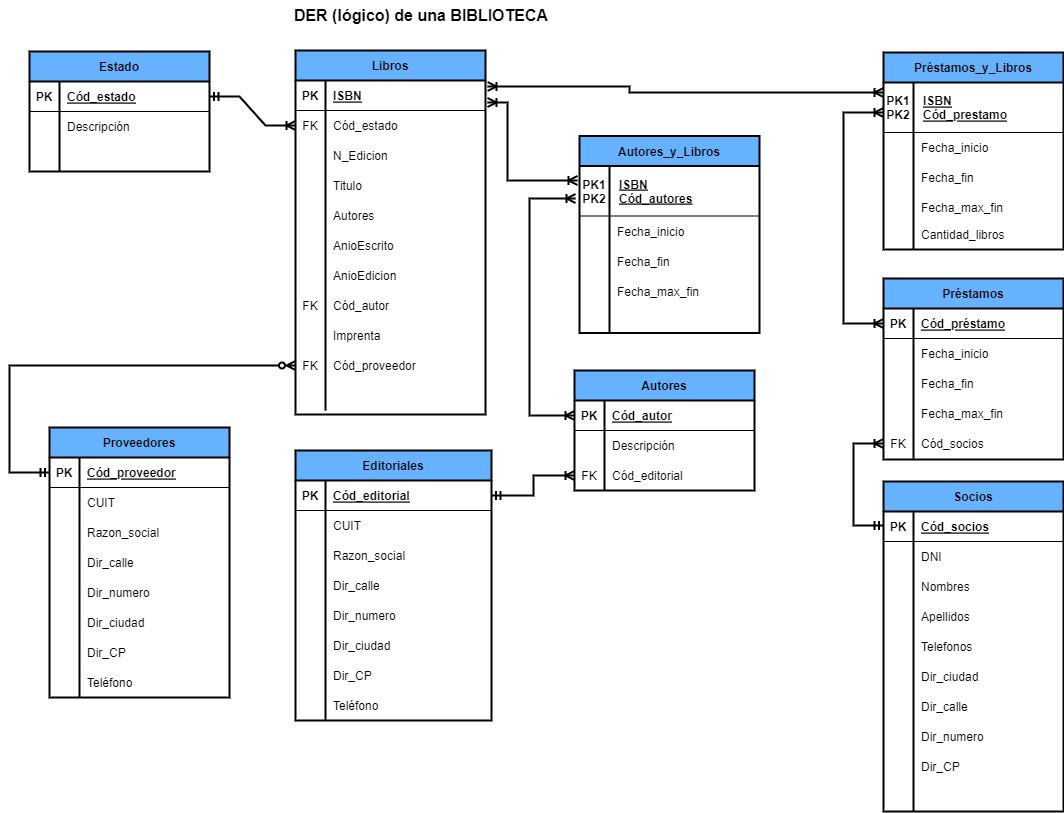
*La biblioteca almacena libros que presta a los socios. De los mismos se almacena su título, su editorial, el año en el que se escribió el libro, el nombre completo del autor (o autores), el año en que se editó, en qué editorial fue y el ISBN.*

*Necesitamos poder indicar si un volumen en la biblioteca está deteriorado o no. Además, queremos controlar cada préstamo que se realiza almacenando la fecha en la que se realiza, la fecha tope para devolver (que son 15 días más que la fecha en la que se realiza el préstamo) y la fecha real en la que se devuelve.*

*Se necesita la administración de Libros, Editoriales y Proveedores de libros.*

**

***2. Realizar el modelo lógico del punto anterior (30 PTS).***

******

***3. En base al modelo final lógico (DER lógico), incorporar 3 entidades justificando su cardinalidad y relación con el resto del diagrama (40 PTS).***

Las entidades que se agregarán al diagrama son:

**Empleados:** son los encargados de gestionar la entrega de los libros que se prestan a los socios. Siempre atenderá un empleado por préstamo de libro, pudiendo el mismo empleado atender a varios préstamos de libros o no atender a ninguno (si trabaja en el sector administrativo, organización de eventos o mantenimiento).

**Categoría\_Socios:** dependiendo la categoría de socio, le permitirá al socio pedir prestado una mayor cantidad de libros por mes. Las categorías disponibles serán plus, silver y gold. Cada socio tendrá su categoría (no pudiendo tener más de una).

**Eventos:** son todas las charlas, firmas de libros, lanzamientos de libros que podrán solicitar las editoriales realizar en la biblioteca. Cada editorial puede solicitar la cantidad de eventos que desee, pudiendo no solicitar realizar ningún evento. Asimismo, cada evento corresponderá sólo a una editorial. Se solicita agendar fecha de inicio, fecha fin y horario del evento, así como su nombre.

