
BASES DE DATOS

PROYECTO INTERDISCIPLINAR

Este es un proyecto interdisciplinar que involucra todos los diferentes módulos del curso (Programación, Entornos de Desarrollo, Bases de Datos y Lenguajes de Marcas). El conjunto del proyecto, cada archivo nuevo que se añada, se modifique, etc. Podrá verse en la siguiente carpeta de Drive: [Proyecto-Enrique-Iván](#)

ÍNDICE

INTRODUCCIÓN	1
MODELO ENTIDAD/RELACIÓN	2
Explicación E/R	2
Lucidchart	7
MODELO RELACIONAL	8
Tablas	8
Pérdidas semánticas	11

1. INTRODUCCIÓN

Nuestro proyecto interdisciplinar consiste en una tienda online de venta de libros y discos de música. En vez de hacer una librería o una disquería, decidimos juntar las dos en una sola y ofrecer como nuestro producto ambos: libros y discos.

Estos serán nuestros principales productos, que contarán con categorías propias que en este caso serán los géneros. Obviamente, los libros y los discos tendrán géneros diferentes.

Para los libros serán: fantasía, romance, ciencia ficción, terror, etc.

Mientras que para los discos: pop, rock, electrónica, hip hop, country, jazz, etc.

Estos serán los criterios de búsqueda, elegir entre libro o cd y luego navegar sobre los diferentes géneros de sus respectivos.

Los productos serán distribuidos hacia nosotros por las diferentes compañías que los producen: editoriales, discográficas, etc.

Cada uno de estos productos tendrá una serie de características, la primera es que cada uno de ellos podrá tener una serie de opiniones escritas por los usuarios: si les han gustado o no, si lo recomiendan...u opiniones sobre el servicio de la tienda: si los productos han llegado en buenas condiciones, comparaciones calidad-precio...En resumen, cada cliente podrá poner por escrito sus opiniones tanto del producto como de nuestro servicio.

Otra característica más será la capacidad de guardar estos productos en listas de deseos creadas por los propios usuarios.

Respecto a nuestros usuarios habrá dos tipos: administrador y cliente.

El usuario de tipo cliente será el que tenga la capacidad de realizar todas las acciones que hemos mencionado anteriormente. Navegar por los distintos productos, escribir una opinión y guardarlos en listas de deseos o seleccionarlos para así realizar un pedido.

El cliente realizará un pedido, al cual se le irán añadiendo los productos.

Para que este pedido sea posible, necesitamos una dirección (a la que llevar el producto). Como dato opcional, un cliente podrá vincular una tarjeta (con la que se realizará el pago del pedido).

Aparte, se generará una factura cada vez que se realice un pedido. Esta factura se enviará también al cliente (a la misma dirección del pedido o a otra distinta) y podrá pagarla una vez le llegue en el caso de no haberla pagado con tarjeta.

Una última característica de los clientes será la posibilidad de mandar tickets para solucionar problemas, quejas o reclamaciones.

Finalmente, el administrador se encargará de crear y modificar los diferentes productos, además de poder visualizar los pedidos, facturas, y tickets de los clientes.

2. MODELO ENTIDAD/RELACIÓN

2.1. Explicación E/R

Nuestro proyecto consiste en realizar una tienda online de venta de libros y discos de música; por lo tanto nuestra base de datos deberá tener las siguientes *entidades*: PROVEEDOR, PRODUCTOS (LIBRO; DISCO DE MÚSICA), OPINIÓN, CATEGORÍA, LISTA DE DESEOS, PEDIDO, USUARIO (ADMINISTRADOR; CLIENTE), DIRECCIÓN, FACTURA, TICKET y TARJETA.

A continuación se verán los *atributos* de cada *entidad*, y explicaremos las relaciones que hay entre cada una de ellas.

Los *atributos* de las *entidades* serán los siguientes:

- De los proveedores queremos poder identificarlos mediante un código, queremos saber de qué tipo serán (discográfica o editorial), el nombre de la empresa, su dirección y su teléfono.
- Respecto a los productos queremos saber lo siguiente: tendrán un código de producto (que los identificará), un nombre, una descripción (que tendrá sus características), su precio, la unidad de medida de dicho producto (UNIDADES), su IVA, la cantidad de la que disponemos (Stock), la cantidad mínima que deberá haber de cada producto y una foto.

Todos estos datos serán obligatorios, a parte de estos, habrá una serie de datos opcionales que serán los siguientes:

- Qué usuario lo ha creado.
- Su fecha de creación.
- Que usuario lo ha modificado.
- La fecha de la última vez que ha sido modificado.

Estos productos serán de dos tipos: libros y discos de música. Esto será una *generalización*.

De los libros queremos saber: Género, autor, número de páginas, ISBN, editorial, formato.

De los discos de música: Género, artista, canciones, sello, ASIN y tipo.

- Cada producto pertenece a una categoría, que serán los diferentes géneros de nuestros productos. De estas categorías queremos saber::
 - Código (con la que se identifica).
 - Nombre (Libro o CD).

- Un producto podrá tener opiniones. Los atributos de una opinión son:
 - Código (identificador).
 - Mensaje.
 - La fecha en la que se ha publicado.
- Para las listas de deseos, se identificarán con un nombre.
- Respecto a los pedidos queremos saber: su código de pedido (su identificador), la fecha en la que se realiza, el precio total del pedido, la dirección del pedido.
- Los pedidos generarán una factura. De la factura necesitamos: su código de factura (identificador), la fecha en la que se ha realizado, una dirección de facturación. (podrá ser distinta de la del pedido).
- Para solucionar que hay dos posibilidades para una dirección (que sea de pedido o de factura), la convertiremos en una entidad. De la dirección queremos saber:
 - Número (será el identificador).
 - Tipo (pedido o factura).
 - CP.
 - Localidad.
 - Provincia.
 - Dirección.

Esta dirección estará asociada a un usuario.

- Usuarios: se identificarán con un código, vincularán un correo electrónico (ha de ser único), necesitan una contraseña, queremos saber su nombre completo y la última vez que se conectaron (última conexión).

A parte, habrá una serie de datos opcionales:

- Teléfono.
- Fecha de nacimiento.
- Foto.

Un usuario podrá ser de dos tipos: cliente o administrador. Esto será una *generalización*.

- Los usuarios podrán realizar tickets, de estos queremos saber: el mensaje, el asunto y el estado. Estos se identificarán con un código.
- Un usuario también podrá vincular una tarjeta, de la que se querrá saber su número.

Las relaciones de las diferentes *entidades* serán las siguientes:

- Un *producto* no existirá si no lo ha suministrado un *proveedor*. Un *proveedor* suministrará si o si un *producto* y solamente ese *producto*. Un *producto* puede ser suministrado por varios *proveedores*, y puede darse el caso de que un *proveedor* no suministre dicho *producto* (un *proveedor* de *discos de música* no suministrará *libros*), *cardinalidad 1:N*.
- Un *producto* será creado y modificado por *administradores*. Un *administrador* no necesariamente modificará/creará un *producto* pero puede llegar a modificar/crear varios. Un *producto* tiene que ser creado si o si por un *administrador* (*restricción de existencia*), pero no tiene porque ser modificado, al guardar solamente el último *administrador* que lo modifica, como máximo lo modificara o creará un *administrador*. *Cardinalidad 1:N*.
- Un *producto* podrá o no tener una *opinión*, y podrá tener varias *opiniones*. A su vez, una *opinión* habrá de estar si o si asociada a un *producto*, y cada *opinión* será única para cada *producto*. Por lo tanto, tendrá una *cardinalidad 1:N*. Una *opinión* no existirá si no pertenece a un *producto*, por lo tanto estamos hablando de una *restricción de existencia*.
- Estas *opiniones* son escritas por los *usuarios*, su relación seguirá la misma lógica que con los productos. No puede haber *opinión* si no la ha escrito un *usuario*, *restricción de existencia*.
- Un *producto* pertenece a una *categoría* como mínimo, no puede haber *productos* sin *categoría* (Ya que hemos mencionado que las *categorías* serán los propios *productos* (Libros y cd's). Por lo tanto, un *producto* tendrá a su vez una *categoría* como máximo (*cardinalidad 1:1*). Esto quiere decir que no puede haber una *categoría* sin *producto* o un *producto* sin *categoría* (*restricción de existencia*).
- A una *lista* se le podrán añadir como mínimo un *producto* y como máximo muchos. Un *producto* no tiene porque estar añadido a una *lista*, pero, podrá ser añadido a muchas. Con esto tenemos una *cardinalidad N:N*.
- Dicha *lista* será creada por un *cliente* y será única para cada uno, y un *cliente* podrá crear muchas, pero no necesariamente habrá de crear una. *Cardinalidad 1:N*. En resumen, una *lista* no existirá si no la crea un *cliente* y será una *restricción de ID*, ya que al estar identificadas con un nombre, necesitamos saber que *cliente* a creado dicha *lista*:

LISTA DE DESEOS	CLIENTE
Lista de reyes	01
Lista de reyes	02
Lista de cumpleaños	01
Lista de cumpleaños	02

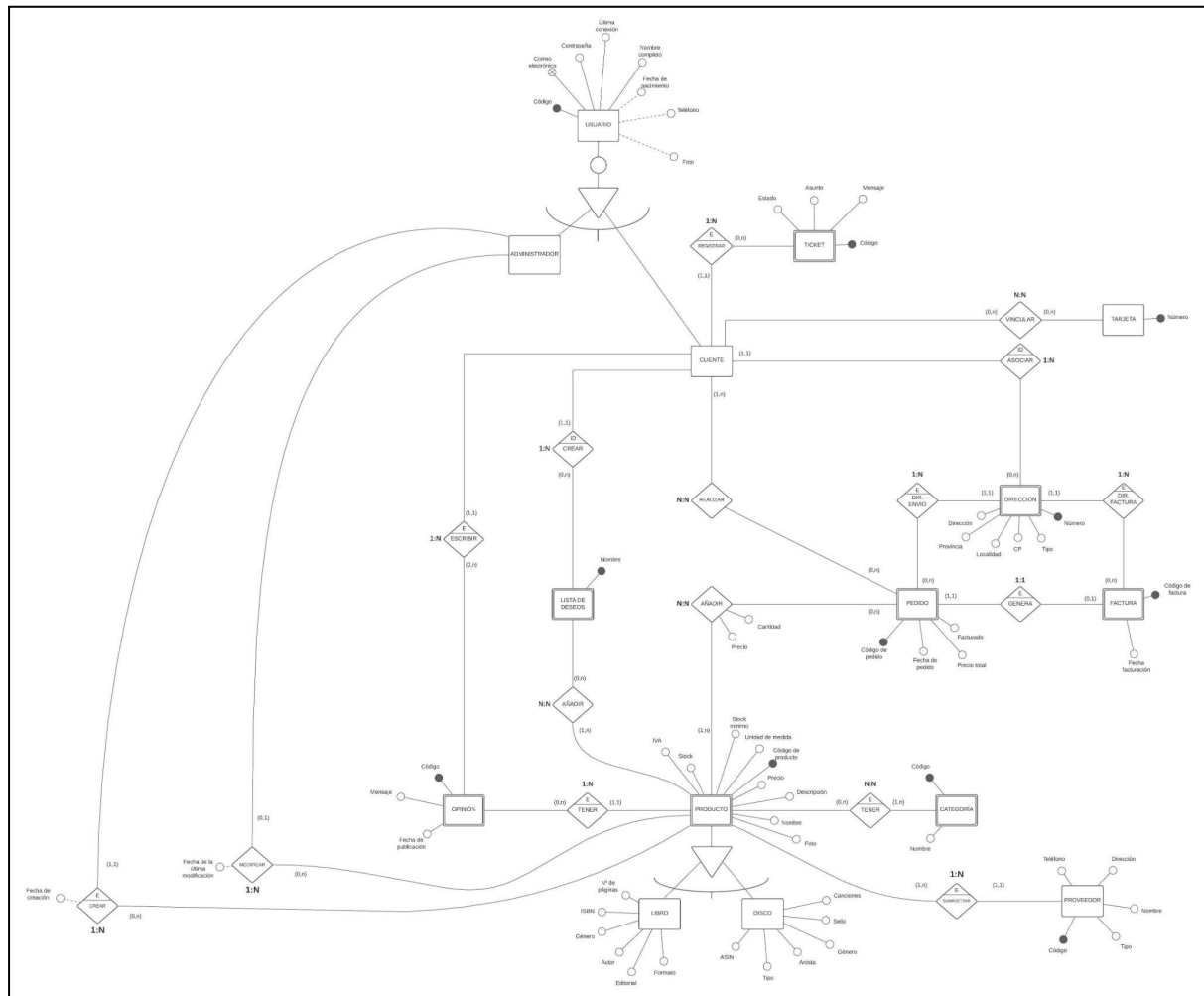
- Un *producto* puede ser añadido a varios *pedidos*, pero no tiene por qué ser añadido a uno. A un *pedido* se le ha de añadir un *producto*, y pueden ser añadidos varios. *Cardinalidad N:N*.
- Un *cliente* puede realizar varios *pedidos*, y no necesariamente realizará uno (puede haber *clientes* que aun no hayan realizado *pedidos* pero estén registrados). Un *pedido* es realizado como mínimo por un *cliente*, y puede ser realizado por varios. *Cardinalidad N:N*.
- Un *pedido* existirá solamente si se ha realizado una *cesta* (*restricción de existencia*), esté será único de cada *cesta*. *Cardinalidad 1:1*. Una *cesta* no necesariamente tiene que realizar un *pedido* (tú puedes crear una *cesta*, pero aun así no realizar el *pedido*). Además, un *pedido* realizado generará una *factura*, que no existirá si no se ha realizado dicho *pedido* (*restricción de existencia*). Cada *factura* será única de cada *pedido* y viceversa. *Cardinalidad 1:1*.
- Un *cliente* puede registrar *tickets*, puede haber *clientes* que no hayan registrado uno y al mismo tiempo puede haberlos que hayan registrado varios. Un *ticket* será registrado si o si por un *cliente*, y dicho *ticket* será únicamente de ese *cliente*. *Cardinalidad 1:N, restricción de existencia*.
- Un *cliente* podrá tener vinculadas varias *tarjetas* (no necesariamente tiene que tener una) y viceversa. Esto será una *cardinalidad N:N*. Dicho *cliente* también podrá tener o no asociadas varias *direcciones* y una *dirección* debe pertenecer a un *cliente* y será únicamente de este. *Cardinalidad 1:N*. Como un *cliente* puede tener varias *direcciones* necesitamos saber a qué *cliente* están asociadas para poder identificarlas ya que la *dirección* 02 no necesariamente tiene que ir relacionada con el *cliente* 02, por lo tanto será una *restricción de ID*.

DIRECCIÓN	CLIENTE
01	01
01	02
02	01
02	02
03	02

- Una *dirección* irá sí o sí asociada a una *factura* o a un *pedido* (podrán ser *direcciones* diferentes) y será única de cada una. Como hemos dicho que pueden ser diferentes, una *dirección* no tiene porque estar como dirección de pedido, si no que será la dirección de facturación y otra diferente será la de pedido. Una *dirección* podrá estar en muchos *pedidos* y *facturas*, cardinalidad 1:N. Además, tanto *factura* como *pedido* no existirán si no hay *dirección*, restricción de existencia.

2.2. Lucidchart

Enlace: [Lucidchart](#)



3. MODELO RELACIONAL

3.1. Tablas

Todos los atributos son considerados “no nulos” excepto los opcionales. Estos atributos opcionales estarán marcados con un “ * ”.

USUARIOS (**código**, **correo_electrónico**, password, última_conexión, nombre, fecha_nacimiento*, teléfono*, foto*)

PK: código

UK: correo_electrónico

ADMINISTRADORES (**código**)

PK: código

FK: código → USUARIO.código

CLIENTES (**código**)

PK: código

FK: código → USUARIO.código

TICKETS (**código**, mensaje, asunto, estado, cliente)

PK: código

FK: cliente → CLIENTE.código

NN (cliente)

TARJETAS (**número**)

PK: número

TARJETAS_USUARIOS (**cliente**, **tarjeta**)

PK: cliente, tarjeta

FK: cliente → CLIENTES.código

FK: tarjeta → TARJETAS.número

DIRECCIONES (**cliente**, **código**, tipo, cp, localidad, provincia, dirección_completa)

PK: cliente, código

FK: cliente → CLIENTES.código

PEDIDOS (**código**, fecha, precio_total, facturado, cliente, dirección_envio)

PK: código

FK: cliente, dirección_envio → DIRECCIONES. cliente, código

NN (cliente, dirección_envio)

FACTURAS (**código**, fecha, cliente, dirección_factura, pedido)

PK: código

UK: pedido

FK: cliente, dirección_factura → DIRECCIONES. cliente, código

NN (cliente, dirección_factura)

FK: pedido → PEDIDOS.código

NN (pedido)

PEDIDOS_CLIENTES (**cliente**, **pedido**)

PK: cliente, pedido

FK: cliente → CLIENTES.código

FK: pedido → PEDIDOS.código

LISTAS_DESEOS (**cliente**, **nombre**)

PK: cliente, nombre

FK: cliente → CLIENTES.código

OPINIONES (**código**, mensaje, fecha_publicación, cliente, producto)

PK: código

FK: cliente → CLIENTES.código

FK: producto → PRODUCTOS.código

NN (cliente)

NN (producto)

PROVEEDORES (**código**, teléfono, dirección, nombre, tipo)

PK: código

CATEGORÍAS (código, nombre)

PK: código

PRODUCTOS (**código**, foto, nombre, descripción, precio, unidad_medida, stock_mínimo, stock, iva, proveedor, modificador*, creador, fecha_creación, fecha_ultima_modificacion*)

PK: código

FK: proveedor → PROVEEDORES.código

FK: modificador → ADMINISTRADORES.código

FK: creador → ADMINISTRADORES.código

NN (proveedor)

NN (creador)

CATEGORIAS_PRODUCTOS (categoría, producto)

PK: categoría, producto

FK: categoría → CATEGORIAS.código

FK: producto → PRODUCTOS.código

LIBROS (código, formato, editorial, autor, género, isbn, número_páginas)

PK: código

FK: código → PRODUCTOS.código

DISCOS (código, canciones, sello, género, artista, tipo, asin)

PK: código

FK: código → PRODUCTOS.código

PEDIDOS_PRODUCTOS (producto, pedido, cantidad, precio)

PK: producto, pedido

FK: producto → PRODUCTOS.código

FK: pedido → PEDIDOS.código

LISTAS_DESEOS_PRODUCTOS (producto, nombre_lista, cliente)

PK: producto, nombre_lista, cliente

FK: producto → PRODUCTOS.código

FK: nombre_lista, cliente → LISTAS_DESEOS.cliente, nombre

3.2. Pérdidas semánticas

Generalización → solamente hay pérdida si una generalización es total o exclusiva.

- La generalización de USUARIO es tanto total como exclusiva. Se pierden tanto la totalidad como la exclusividad puesto que no podemos plasmarlas en el modelo relacional.
- La generalización de PRODUCTO es exclusiva. Se pierde la exclusividad ya que no se puede plasmar en el modelo relacional.

(1,n) → Se pierde el uno, no se puede representar. Ya que en el relacional sólo podemos captar la existencia mínima de ese producto en relaciones con restricción de existencia.

- Se pierde el 1 del (1,n) de PRODUCTO-PROVEEDOR.
- Se pierde el 1 del (1,n) de PRODUCTO-LISTA_DESEOS.
- Se pierde el 1 del (1,n) de PRODUCTO-PEDIDO.
- Se pierde el 1 del (1,n) de CLIENTE-PEDIDO.

1:N o 1:1 → Si hay atributo en la relación, no se puede saber su origen. En las relaciones 1:N ADMINISTRADOR-MODIFICAR-PRODUCTO y ADMINISTRADOR-CREAR-PRODUCTO, se pierde el origen de los atributos “Fecha de creación” y “Fecha de última modificación” porque no hacemos una tabla de la relación.