



PROYECTO INTERDISCIPLINAR

IVÁN AYUSO OLIVERA

ENRIQUE AZORÍN CASTELLANO

NOMBRE TIENDA

LOGO

Aún no tenemos ni NOMBRE ni LOGO

Tanto la nomenclatura de los apartados, así como su diseño no son definitivos, pueden ir variando a lo largo del desarrollo del proyecto **sin alterar** los apartados ya **corregidos**.

ÍNDICE

1) INTRODUCCIÓN	1
1) BASES DE DATOS	2
a) <i>El modelo Entidad/Relación</i>	2
b) <i>Nuestro modelo Entidad/Relación</i>	3
c) <i>Modelo relacional</i>	8
2) PROGRAMACIÓN	8
a) <i>Explicación de la funcionalidad de la aplicación con capturas de pantalla</i>	8
3) LENGUAJE DE MARCAS	8
a) <i>Mockup</i>	8
b) <i>Diagrama de flujo de pantallas</i>	8
4) ENTORNOS DE DESARROLLO	8
a) <i>Enlace de GitHub</i>	8
b) <i>Diagramas de clases</i>	8
c) <i>Diagramas de casos de uso</i>	8
5) CONCLUSIONES Y DIFICULTADES ENCONTRADAS	8
6) POSIBLES MEJORAS O AMPLIACIONES	8
7) BIBLIOGRAFÍA	8

MEMORIA

1) INTRODUCCIÓN

Nuestro proyecto interdisciplinar consiste en una tienda online de venta de libros y cd's 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 cd's.

Estos serán nuestros principales productos, que a su vez serán su propia categoría, es decir, cuando tu vayas a seleccionar una categoría elegirías entre libros o cd's. Después, dentro de dichas categorías, estarán sus subcategorías que en este caso serán los géneros. Obviamente, los libros y los cd's tendrán géneros diferentes.

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

Mientras que para los cd's: 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.

Siguiendo con más características de estos productos, 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 o bien seleccionarlos en una cesta de la compra. Tanto la una como la otra serán 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 llevarlos a la cesta de la compra para así realizar un pedido.

Para que este pedido sea posible, necesitamos una serie de datos: la cesta con los productos y 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).

Finalmente, 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.

1) BASES DE DATOS

a) El modelo Entidad/Relación

Para diseñar una *base de datos* debemos realizar un esquema que tenga en cuenta todos los requisitos y funciones que pueda tener nuestra futura aplicación informática. Uno de los esquemas utilizados para ello es el *modelo Entidad/Relación*.

Este modelo plasma todos esos requisitos y funciones mediante una serie de diagramas que son un conjunto de *entidades* y las *relaciones* que hay entre ellas.

Las entidades, son cualquier tipo de objeto, elemento o concepto de los que necesitemos recoger información. No necesariamente tienen que ser algo que exista físicamente. Ejemplos: cliente, departamento, trabajador, cuenta, sucursal, libro, asignatura, ciclo, etc.

Existen dos tipos de *entidades*:

- Fuertes; son independientes, existen sin necesidad de otra y en el caso de que se modifiquen o eliminen otras *entidades* estas no se verán afectadas.
- Débiles; son las que dependes de otras. Si por ejemplo Libro depende de Autor, si se borran todos los datos de ese Autor, también desaparecerán los del Libro.

Las *entidades* tienen *atributos* asociados a ellas; estos, son nombres que identifican diferentes cualidades, propiedades o características. Son además identificadores.

Hay atributos de diferentes tipos:

- Principales o secundarios.
- Simples o compuestos.
 - Compuestos: aquellos que se pueden dividir en varios valores. Ejemplo: dirección (calle, ciudad, provincia, código postal); fecha de nacimiento (día, mes y año).
 - Simples: no se dividen en otros valores. Ejemplo: color, código, contraseña...
- Almacenados o derivados.
 - Derivados:
 - Almacenados:
- Monovaluados o multivaluados.
- Obligatorios u Opcionales. **SIN ACABAR**

b) Nuestro modelo Entidad/Relación

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*: PRODUCTOS (LIBRO; DISCO DE MÚSICA), OPINIÓN, CATEGORÍA, LISTA DE DESEOS, CESTA, PEDIDO, USUARIO (ADMINISTRADOR; CLIENTE), DIRECCIÓN, FACTURA 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:

- 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, una foto.

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

- Su fecha de creación.
- 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, discográfica, ASIN.

- Cada producto pertenece a una categoría, de la que se querrá saber:
 - Código (con la que se identifica).
 - Nombre (Libro o CD).

Las categorías tendrán subcategorías (con nombre propio).

- Código (su identificador).
- Nombre (el género).
- Un producto podrá tener opiniones. Los atributos de una opinión son:
 - Número (identificador).
 - Mensaje.
- Las cestas solamente se identifican con un nombre.
- Para las listas de deseos lo mismo, 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*.

- 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* 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*. Como un *producto* puede tener varias *opiniones* necesitamos saber a qué *producto* se está refiriendo para poder identificarla ya que la *opinión 02* no necesariamente tiene que ir relacionada con el *producto 02*, por lo tanto será una *restricción de ID*.

OPINIÓN	PRODUCTO
01	01
01	02
02	01
02	02

- Estas *opiniones* son escritas por los *usuarios*, su relación seguirá la misma lógica que con los productos.
- 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*).
- Respecto a las *subcategorías*, una *categoría* tendrá sí o sí una *subcategoría*, ya que tanto los libros como los cd's tienen como mínimo un género asociado (el género serán las *subcategorías*). Habrá la misma cantidad de *subcategorías* que de géneros dispongamos de nuestros diferentes *productos*. Una *subcategoría* es exclusiva de una *categoría* (no pueden haber libros de género rock al igual que no hay cd's de género de fantasía), y una subcategoría no existirá sin categoría asociada, por lo tanto habrá una restricción de existencia y tendrá una *cardinalidad 1:N*.
- A una *cesta* se le podrán añadir como mínimo un *producto* y como máximo muchos. Un *producto* no tiene porque estar añadido a una *cesta*, pero, podrá ser añadido a muchas. Con esto tenemos una *cardinalidad N:N*.
- Dicha *cesta* 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 cesta 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 *cesta*:

CESTA	CLIENTE
Cesta de reyes	01
Cesta de reyes	02
Cesta de cumpleaños	01
Cesta de cumpleaños	02

- Se seguirá la misma lógica que las *cestas* para las *listas de deseos*.
- 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* 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 la 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*.

Enlace: [Lucidchart](#)



c) Modelo relacional

2) PROGRAMACIÓN

a) Explicación de la funcionalidad de la aplicación con capturas de pantalla

3) LENGUAJE DE MARCAS

a) Mockup

b) Diagrama de flujo de pantallas

4) ENTORNOS DE DESARROLLO

a) Enlace de GitHub

b) Diagramas de clases

c) Diagramas de casos de uso

5) CONCLUSIONES Y DIFICULTADES ENCONTRADAS

5.1) Errores y correcciones (Esto es temporal)

5.1.1) Base de Datos

5.1.1.1) Modelo Entidad/Relación

- Faltas de ortografía: o opiniones (u opiniones), cesto de la compra (será cesta, supongo), aparte junto.
 - Corrección: se han corregido las faltas de ortografía.
- En el usuario tenéis dos claves, debería tener solo una. O le ponéis código o correo (podéis poner correo como alternativa, pero usar otra simbología en el esquema).
 - Corrección: he cambiado el correo a clave alternativa.
- Entre usuario y producto debería haber dos relaciones, una de crear y otra de modificar (no se debe poner como atributos el usuario porque ya está implícito en la propia relación).
 - Corrección: he añadido la relación CREAR con su fecha de creación, y he eliminado los atributos relacionados con el usuario en la relación de MODIFICAR.
- Al poner proveedor, lista de deseos, opiniones y tickets estáis haciendo el esquema más complejo de lo que se os pide. No pasa nada, podéis hacerlo y añadir la funcionalidad si os da tiempo.



- Corrección: hemos decidido mantener todas las entidades salvo TICKET, ya que hemos considerado que no era necesaria para la página web.
- Vuestros ejemplos de tablas no son correctos del todo, aunque el razonamiento y el esquema sí. Por ejemplo, Opinión producto podríamos tener el producto 01 con la opinión 01 y el producto 02 con la opinión 01 que es distinta de la opinión del producto 01. En categorías y subcategorías está perfectamente explicado, aunque podría no ser de ID, sino simplemente 1:N de existencia (el código de la subcategoría no se repite), en la cesta está bien explicado pero luego la cesta la estáis repitiendo (es la misma siempre y no es correcto). La restricción de ID significa que el cliente 01 puede crear la cesta de Enrique y el cliente 02 puede crear la cesta de Enrique también, que será distinta porque es distinto cliente. En direcciones de cliente os ha pasado lo mismo.
- Corrección: se han modificado las tablas para expresar de una manera más correcta la explicación de las relaciones de ID. Se ha modificado la relación CATEGORÍA - SUBCATEGORÍA de ID a Existencia.

6) POSIBLES MEJORAS O AMPLIACIONES

7) BIBLIOGRAFÍA

[Enlace al enunciado del proyecto](#)