

PROYECTO LIBRERÍA ON-LINE

2º curso DAM - Año académico 2021/2022

ACCESO E INTERFACES

CEINMARK

Autores:

Julen Carmona, Christian Jodra, Imanol Turiel

Índice

1. Proyecto: Librería On-Line.....	2
1.1. Especificación de requisitos.....	2
2. Arquitectura de la aplicación.....	2
2.1. Descripción.....	2
2.2. Esquema.....	3
3. Funciones de la Aplicación.....	3
3.1. Funciones de la WEB (Cliente).....	3
3.1.1. Página de inicio (Menú).....	3
3.1.2. Mostrar catálogo.....	3
3.1.3. Visualizar artículo.....	3
3.1.4. <i>Log-in/Log-out</i>	4
3.1.5. Registro.....	4
3.1.6. Mantenimiento del carrito.....	5
3.1.6.1. Añadir.....	5
3.1.6.2. Quitar.....	5
3.1.6.3. Modificar unidades.....	5
3.1.7. Totalizar la compra	5
3.1.8. Tramitar el pedido.....	6
3.1.9. Búsqueda de artículo.....	6
3.1.10. Mantenimiento de ficha cliente.....	6
3.1.11. Listar pedidos del cliente.....	6
3.1.12. Compartir artículo.....	6
3.1.13. Top 10 artículos en el <i>index</i>	7
3.2. Funciones de Escritorio (Admin).....	7
3.2.1. Menú de la aplicación (MainActivity).....	7
3.2.2. Mantenimiento del Catálogo.....	7
3.2.2.1. Libros.....	7
3.2.2.2. Categorías.....	8
3.2.2.3. Autores.....	9
3.2.2.4. Editoriales.....	9
3.2.3. Listado de ventas (Informes).....	10
4. Pantallas y descripción operativa de cada aplicación.....	11
5. Modelo de Datos.....	18
5.1. Diagrama de Entidad/Relación.....	18

1. Proyecto: Librería On-Line

1.1. Especificación de requisitos:

LIBRERÍA ONLINE

Se desea modelar un sistema online de venta de libros.

Para ello hay que mantener, en primer lugar, los autores (cada autor tendrá un código y un nombre), las editoriales (también código y nombre) y las categorías o familias de libros (también código y nombre).

Cada libro puede ser de uno o varios autores, de una editorial y de una categoría. De cada libro guardamos el ISBN (que lo identifica), el título, su precio, el número de libros que hay en stock, sus autores, la editorial y la categoría.

Todos estos datos son mantenidos por una aplicación de gestión interna de la empresa.

Los clientes de la tienda se conectarán a la misma a través de una aplicación web.

El cliente se puede dar de alta, introduciendo sus datos de: DNI, nombre, primer apellido, segundo apellido, dirección completa, email, fecha de nacimiento, y el usuario y contraseña con los que se identificará en la tienda.

Existe un campo identificador que identifica de manera unívoca a cada cliente.

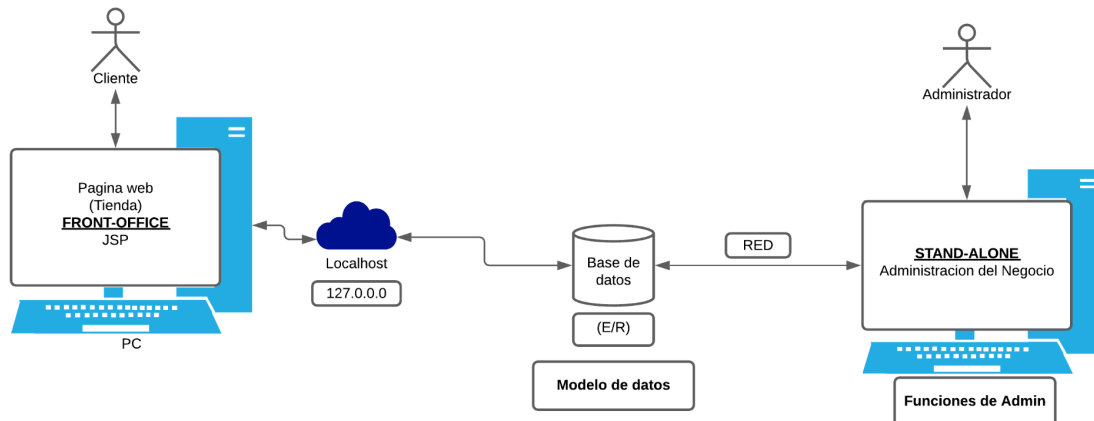
El cliente puede realizar compras de libros online, consistiendo cada compra (que se realiza en una fecha determinada) en varios libros seleccionados de entre los libros que están en stock.

2. Arquitectura de la aplicación:

2.1. Descripción:

La aplicación se compone de 3 partes. Por una parte, se encuentra la propia página web. En ella, se localiza la tienda (*Front-office*), la base de datos, y la aplicación de escritorio para la administración (*Back-office*). Será una web dinámica HTML/XML con ayuda de JavaServer Pages (JSP). Esta página web será el apartado al que accederán los clientes para la compra de libros. Por otra parte, la web se conectará a la Base de Datos (BBDD) mediante protocolo TCP/IP (en este caso, en *Localhost*). La BBDD se gestionará utilizando MySQL como Sistema Gestor de Base de Datos. Finalmente, en la parte de "Administración del Negocio", utilizaremos un programa *Stand-Alone* desarrollado en Java Swing/AWT, que gestionará los datos de la BBDD. Esta última parte sólo será accesible para los/as administradores/as de la web.

2.2. Esquema:



3. Funciones de la Aplicación

3.1. Funciones de la WEB (Cliente):

3.1.1. Página de inicio (Menú):

- **Header:** Logo, catálogo, carrito, *log-in*.
- **Body:** top 4 artículos, opiniones
- Funciones del “Menú de navegación”: acceso al *log-in*, acceso al catálogo, acceso al carrito.

3.1.2. Mostrar catálogo

- **Header:** Logo, catálogo, carrito, *log-in*.
- **Body:** Estará compuesto por una tabla de 4 columnas, en la que se incluirán los diferentes libros que estén disponibles.
- **Template:** Cada artículo se mostrará según la siguiente plantilla: la foto del libro, el título del libro, el nombre del/la autor/a y su primer apellido, el precio y un botón que servirá para añadir el libro seleccionado al carrito y otro para ver los detalles.
- **Acciones:**
 - Al hacer click en detalles muestra una página sobre ese libro, al añadir a carrito se suma una unidad en carrito si hay stock

3.1.3. Visualizar artículo

- *Header*: Logo, catálogo, carrito, *log-in*.
- *Body*: Foto del libro que se ha seleccionado, título de dicho libro, nombre y primer apellido del/a autor/a, resumen del libro y detalles, la editorial. Incluirá el precio de cada ejemplar, siendo expresado en euros (€) y el botón de añadir al carrito.
- Funciones:
 - Añadir el libro al carrito.
- Restricciones:
 - En caso de no haber existencias del libro, al hacer click no añadirá nada

3.1.4. Log-in/Log-out:

- *Body*; esta sección incluirá:
 - Un formulario a rellenar con los datos necesarios para iniciar sesión (Usuario y contraseña)
 - Botón de iniciar sesión
 - Muestra mensajes informativos al cerrar sesión o al introducir la contraseña o usuario equivocados
- Restricciones:
 - Un/a usuario/a que ya esté conectado/a, deberá cerrar sesión para cambiar de cuenta o conectarse de nuevo.
 - Un/a usuario/a que no esté registrado/a no podrá iniciar sesión.
 - No es posible registrarse si ya estás *logueado*.

3.1.5. Registro:

- *Body*; esta sección incluirá:
 - Un formulario en el que el/la usuario/a creará su cuenta (DNI, *user*, nombre, primer apellido, segundo apellido, dirección, fecha de nacimiento, correo electrónico, *password*).

- Botón de crear cuenta.
- Restricciones:
 - No se puede registrar un correo electrónico ya registrado.
 - No se puede registrar un/a usuario/a ya registrado/a.
 - No se puede repetir un DNI ya registrado.

3.1.6. Mantenimiento del carrito:

3.1.6.1. Añadir:

- En la visualización del libro, se incorporará un botón de compra. Al clicar en él, el libro seleccionado se agregará al carrito para poder ser comprado más adelante.

3.1.6.2. Quitar:

- En el carrito, el/la usuario/a podrá almacenar los productos que desee para su posterior compra. En esta sección, se añadirá un botón o icono que, al ser pulsado, eliminará del carrito el producto que el/la usuario/a finalmente no desee comprar.

3.1.6.3. Modificar unidades:

- En la pantalla “Carrito”, el/la usuario/a podrá comprobar qué productos ha incluido en su posible futura compra. En esta sección, el/la usuario/a podrá añadir o eliminar unidades de alguno de los productos incluídos en su carrito. Para ello, se dibujará un contador que constará de dos botones, uno con el símbolo + para añadir más libros a la compra, y otro con el símbolo - en caso de querer quitar unidades.

3.1.7. Totalizar la compra:

- Su función es calcular el precio total de los libros que estén en el carrito para más adelante pasar ese dato al pago. La información que se muestra es el total de artículos, y el precio total de la compra se expresa en euros (€).

- Restricciones: No pueden ser números negativos. Para poder efectuar la compra tiene que tener un artículo mínimo.

3.1.10. Mantenimiento de ficha cliente:

- Esta página servirá para gestionar y modificar la información de la cuenta del usuario/a.
- Se podrá editar los siguientes datos personales de cada perfil → (Nombre, primer apellido, segundo apellido, DNI, contraseña, correo electrónico, fecha de nacimiento y dirección).

3.1.13. Top 4 artículos en el *index**:

- En la página de inicio (*index*), colocar una sección de los 4 libros más vendidos. Esta selección se actualizará dependiendo de cómo varíe la tendencia de compra y cuáles sean los libros que se conviertan en los últimos 4 más vendidos.

3.2. Funciones de Escritorio (Admin):

3.2.1. Menú de la aplicación (MainActivity):

- Contiene 4 botones para acceder a las siguientes funciones de mantenimiento: libros, autores, editoriales y categorías.

3.2.2. Mantenimiento del libros:

3.2.2.1. Libros:

- Pantalla con una tabla que nos lista los libros
- Tiene 2 Paneles con pestañas, una para ver los autores que tiene el libro y quitarlos y otra para añadir nuevos autores. Dentro de la de quitar hay un botón para eliminarlos que solo se activa si tenemos un autor seleccionado. En la pestaña de añadir autor hay 2 botones, uno para actualizar la lista de autores y otro para añadir un autor al libro seleccionado.
- Tiene los botones de añadir libro, que añade un libro con los datos escritos. Eliminar libro, que recoge el isbn del libro seleccionado y modificar libro.
- Tiene también un botón para actualizar los datos de las editoriales y de las categorías.

- Permite mostrar y añadir una imagen al libro que cuando le damos a modificar en guardar o añadir crea una copia de la imagen seleccionada.

- Restricciones:
 - No se podrán editar datos que estén referenciados a otras tablas por FK.
 - Si no existe el libro, se mostrará el mensaje “El libro no existe”.

3.2.2.2. Categorías:

- Pantalla con dos botones, uno para buscar libros ya existentes utilizando el texto del *Textfield* y el otro para añadir una nueva categoría.
- En este apartado, habrá una tabla con el listado de las categorías que se encuentren en la BBDD.
- Dentro de esa pantalla, se podrán modificar datos de una categoría que ya tengamos registrado: nombre de la categoría
- El botón añadir categoría añadirá la editorial en la BBDD
- Botón de editar para modificar los datos de la categoría
- Botón de “Eliminar” que eliminará de la BBDD el registro de la categoría seleccionada.
- Restricciones:
 - No se podrán editar o eliminar datos referentes a categorías que estén referenciados a algún libro por FK.
 - Si no existe la categoría de la búsqueda, se mostrará el mensaje “La categoría buscada no existe”.

3.2.2.3. Autores:

- Pantalla con tres botones y 4 campos de texto.
- En este apartado, habrá una tabla con el listado de todas los autores que se encuentren en la BBDD.
- Dentro de esa pantalla, se podrán modificar datos de un autor que ya tengamos registrado: nombre del autor, primer apellido y segundo apellido
- El botón añadir autor añadirá el autor en la BBDD
- Botón de editar para modificar los datos del autor
- Botón de “Eliminar” que eliminará de la BBDD el registro del autor seleccionada.
- Restricciones:
 - No se podrán editar o eliminar datos referentes a autores que estén referenciados a algún libro por FK.
 - Si no existe el autor de la búsqueda, se mostrará el mensaje “El autor buscado no existe”.

3.2.2.4. Editoriales:

- Pantalla con tres botones y 2 campos de texto.
- En este apartado, habrá una tabla con el listado de todas las editoriales que se encuentren en la BBDD.
- Dentro de esa pantalla, se podrán modificar datos de una editorial que ya tengamos registrado: nombre de la editorial
- El botón añadir editorial añadirá la editorial en la BBDD
- Botón de editar para modificar los datos de la editorial
- Botón de “Eliminar” que eliminará de la BBDD el registro de la editorial seleccionada.
- Restricciones:
 - No se podrán editar o eliminar datos referentes a editoriales que estén referenciados a algún libro por FK.
 - Si no existe la editorial de la búsqueda, se mostrará el mensaje “La editorial buscada no existe”.

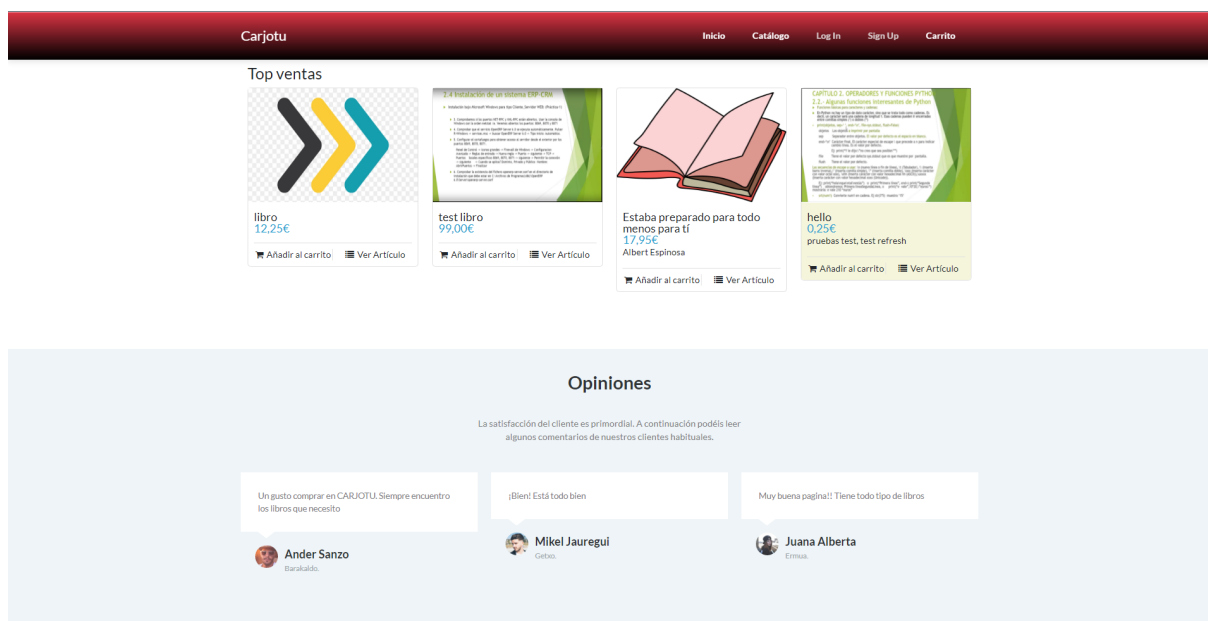
3.2.3. Listado de ventas (Informes):

- Genera un archivo de la venta seleccionada
- Se exportarán dichos datos a un archivo. txt, XML o PDF

4. Pantallas y descripción operativa de cada aplicación

4.1. Pantallas WEB

4.1.1. Pantalla de Inicio



4.1.2. Catálogo

Catálogo de libros

CAPÍTULO 2. OPERADORES Y FUNCIONES PYTHON
2.1 Algunas funciones importantes de Python

• La función print() muestra el contenido de una variable en pantalla.

• La función input() permite al usuario introducir datos por teclado.

• La función len() devuelve la longitud de un objeto iterable.

• La función type() devuelve el tipo de dato de una variable.

• La función isinstance() verifica si un objeto es instancia de una clase.

• La función dir() devuelve una lista con los atributos de un objeto.

• La función help() muestra la documentación de un módulo o función.

• La función eval() evalúa expresiones Python.

• La función exec() ejecuta código Python.

• La función globals() devuelve un diccionario con las variables globales.

• La función locals() devuelve un diccionario con las variables locales.

• La función id() devuelve el identificador único de un objeto.

• La función hash() devuelve el valor hash de un objeto.

• La función memoryview() crea una vista sobre un objeto de memoria.

• La función bytearray() crea un array de bytes mutable.

• La función range() genera una secuencia de números enteros.

• La función enumerate() devuelve una tupla con el índice y el elemento de una iteración.

• La función zip() combina dos o más iterables en una única iteración.

• La función sorted() devuelve una nueva lista ordenada a partir de otra lista.

• La función reversed() devuelve un iterador que itera sobre una colección en orden inverso.

• La función filter() filtra los elementos de una iteración basándose en un criterio.

• La función map() aplica una función a cada elemento de una iteración.

• La función reduce() reduce una iteración a un solo valor acumulando resultados.

• La función lambda se utiliza para crear funciones anónimas.

• La función def define una función.

• La función return devuelve un valor desde una función.

• La función pass no hace nada, pero sirve como marcador de posición.

• La función yield se utiliza en funciones generadoras para producir valores uno a uno.

• La función *args permite pasar un número arbitrario de argumentos.

• La función **kwargs permite pasar un número arbitrario de argumentos clave-valor.

• La función nonlocal se utiliza para modificar variables de ámbito local dentro de una función.

• La función global se utiliza para acceder y modificar variables globales desde una función.

• La función __name__ devuelve el nombre del módulo actual.

• La función __doc__ devuelve la documentación de un módulo o función.

• La función __file__ devuelve el archivo donde se encuentra el código.

• La función __path__ devuelve la ruta de acceso al directorio del módulo.

• La función __package__ devuelve el paquete al que pertenece el módulo.

• La función __spec__ devuelve la especificación del módulo.

• La función __annotations__ devuelve un diccionario con las anotaciones de tipo de las variables.

• La función __dict__ devuelve un diccionario con los atributos de un objeto.

• La función __getattr__() se llama cuando no se encuentra un atributo en el diccionario de atributos.

• La función __setattr__() se llama cuando se intenta asignar un nuevo valor a un atributo.

• La función __delattr__() se llama cuando se intenta eliminar un atributo.

• La función __str__() devuelve una representación legible en texto de un objeto.

• La función __repr__() devuelve una representación oficial y no ambigua de un objeto.

• La función __lt__() se llama cuando se compara un objeto con otro usando el operador <.

• La función __le__() se llama cuando se compara un objeto con otro usando el operador <=.

• La función __gt__() se llama cuando se compara un objeto con otro usando el operador >.

• La función __ge__() se llama cuando se compara un objeto con otro usando el operador >=.

• La función __eq__() se llama cuando se compara un objeto con otro usando el operador ==.

• La función __ne__() se llama cuando se compara un objeto con otro usando el operador !=.

• La función __add__() se llama cuando se usa el operador + para sumar dos objetos.

• La función __sub__() se llama cuando se usa el operador - para restar un objeto de otro.

• La función __mul__() se llama cuando se usa el operador * para multiplicar dos objetos.

• La función __div__() se llama cuando se usa el operador / para dividir un objeto entre otro.

• La función __mod__() se llama cuando se usa el operador % para calcular el resto de una división.

• La función __pow__() se llama cuando se usa el operador ** para elevar un objeto a una potencia.

• La función __len__() se llama cuando se usa el operador len() para obtener la longitud de un objeto.

• La función __getitem__() se llama cuando se accede a un elemento de un objeto mediante corchetes.

• La función __setitem__() se llama cuando se intenta asignar un nuevo valor a un elemento de un objeto mediante corchetes.

• La función __delitem__() se llama cuando se intenta eliminar un elemento de un objeto mediante corchetes.

• La función __iter__() devuelve un iterador para un objeto iterable.

• La función __next__() devuelve el siguiente elemento de un iterador.

• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __getitem__(), __setitem__() y __delitem__() se llaman cuando se accede a un elemento de un objeto mediante corchetes.

• La función __iter__() y __next__() se llaman cuando se itera sobre un objeto iterable.

• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __len__() se llama cuando se usa el operador len() para obtener la longitud de un objeto.

• La función __str__() y __repr__() se llaman cuando se convierte un objeto a cadena de texto.

• La función __lt__(), __le__(), __gt__() y __ge__() se llaman cuando se comparan dos objetos.

• La función __add__(), __sub__(), __mul__() y __div__() se llaman cuando se realizan operaciones aritméticas.

• La función __mod__() y __pow__() se llaman cuando se realizan operaciones matemáticas avanzadas.

• La función __getitem__(), __setitem__() y __delitem__() se llaman cuando se accede a los elementos de una colección.

• La función __iter__() y __next__() se llaman cuando se itera sobre los elementos de una colección.

• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __len__() se llama cuando se usa el operador len() para obtener la longitud de un objeto.

• La función __str__() y __repr__() se llaman cuando se convierte un objeto a cadena de texto.

• La función __lt__(), __le__(), __gt__() y __ge__() se llaman cuando se comparan dos objetos.

• La función __add__(), __sub__(), __mul__() y __div__() se llaman cuando se realizan operaciones aritméticas.

• La función __mod__() y __pow__() se llaman cuando se realizan operaciones matemáticas avanzadas.

• La función __getitem__(), __setitem__() y __delitem__() se llaman cuando se accede a los elementos de una colección.

• La función __iter__() y __next__() se llaman cuando se itera sobre los elementos de una colección.

• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __len__() se llama cuando se usa el operador len() para obtener la longitud de un objeto.

• La función __str__() y __repr__() se llaman cuando se convierte un objeto a cadena de texto.

• La función __lt__(), __le__(), __gt__() y __ge__() se llaman cuando se comparan dos objetos.

• La función __add__(), __sub__(), __mul__() y __div__() se llaman cuando se realizan operaciones aritméticas.

• La función __mod__() y __pow__() se llaman cuando se realizan operaciones matemáticas avanzadas.

• La función __getitem__(), __setitem__() y __delitem__() se llaman cuando se accede a los elementos de una colección.

• La función __iter__() y __next__() se llaman cuando se itera sobre los elementos de una colección.

• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __len__() se llama cuando se usa el operador len() para obtener la longitud de un objeto.

• La función __str__() y __repr__() se llaman cuando se convierte un objeto a cadena de texto.

• La función __lt__(), __le__(), __gt__() y __ge__() se llaman cuando se comparan dos objetos.

• La función __add__(), __sub__(), __mul__() y __div__() se llaman cuando se realizan operaciones aritméticas.

• La función __mod__() y __pow__() se llaman cuando se realizan operaciones matemáticas avanzadas.

• La función __getitem__(), __setitem__() y __delitem__() se llaman cuando se accede a los elementos de una colección.

• La función __iter__() y __next__() se llaman cuando se itera sobre los elementos de una colección.

• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __len__() se llama cuando se usa el operador len() para obtener la longitud de un objeto.

• La función __str__() y __repr__() se llaman cuando se convierte un objeto a cadena de texto.

• La función __lt__(), __le__(), __gt__() y __ge__() se llaman cuando se comparan dos objetos.

• La función __add__(), __sub__(), __mul__() y __div__() se llaman cuando se realizan operaciones aritméticas.

• La función __mod__() y __pow__() se llaman cuando se realizan operaciones matemáticas avanzadas.

• La función __getitem__(), __setitem__() y __delitem__() se llaman cuando se accede a los elementos de una colección.

• La función __iter__() y __next__() se llaman cuando se itera sobre los elementos de una colección.

• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __len__() se llama cuando se usa el operador len() para obtener la longitud de un objeto.

• La función __str__() y __repr__() se llaman cuando se convierte un objeto a cadena de texto.

• La función __lt__(), __le__(), __gt__() y __ge__() se llaman cuando se comparan dos objetos.

• La función __add__(), __sub__(), __mul__() y __div__() se llaman cuando se realizan operaciones aritméticas.

• La función __mod__() y __pow__() se llaman cuando se realizan operaciones matemáticas avanzadas.

• La función __getitem__(), __setitem__() y __delitem__() se llaman cuando se accede a los elementos de una colección.

• La función __iter__() y __next__() se llaman cuando se itera sobre los elementos de una colección.

• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __len__() se llama cuando se usa el operador len() para obtener la longitud de un objeto.

• La función __str__() y __repr__() se llaman cuando se convierte un objeto a cadena de texto.

• La función __lt__(), __le__(), __gt__() y __ge__() se llaman cuando se comparan dos objetos.

• La función __add__(), __sub__(), __mul__() y __div__() se llaman cuando se realizan operaciones aritméticas.

• La función __mod__() y __pow__() se llaman cuando se realizan operaciones matemáticas avanzadas.

• La función __getitem__(), __setitem__() y __delitem__() se llaman cuando se accede a los elementos de una colección.

• La función __iter__() y __next__() se llaman cuando se itera sobre los elementos de una colección.

• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __len__() se llama cuando se usa el operador len() para obtener la longitud de un objeto.

• La función __str__() y __repr__() se llaman cuando se convierte un objeto a cadena de texto.

• La función __lt__(), __le__(), __gt__() y __ge__() se llaman cuando se comparan dos objetos.

• La función __add__(), __sub__(), __mul__() y __div__() se llaman cuando se realizan operaciones aritméticas.

• La función __mod__() y __pow__() se llaman cuando se realizan operaciones matemáticas avanzadas.

• La función __getitem__(), __setitem__() y __delitem__() se llaman cuando se accede a los elementos de una colección.

• La función __iter__() y __next__() se llaman cuando se itera sobre los elementos de una colección.

• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __len__() se llama cuando se usa el operador len() para obtener la longitud de un objeto.

• La función __str__() y __repr__() se llaman cuando se convierte un objeto a cadena de texto.

• La función __lt__(), __le__(), __gt__() y __ge__() se llaman cuando se comparan dos objetos.

• La función __add__(), __sub__(), __mul__() y __div__() se llaman cuando se realizan operaciones aritméticas.

• La función __mod__() y __pow__() se llaman cuando se realizan operaciones matemáticas avanzadas.

• La función __getitem__(), __setitem__() y __delitem__() se llaman cuando se accede a los elementos de una colección.

• La función __iter__() y __next__() se llaman cuando se itera sobre los elementos de una colección.

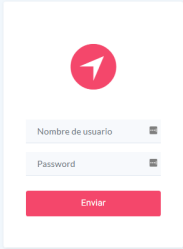
• La función __call__() se llama cuando se trata a un objeto como si fuera una función.

• La función __contains__() se llama cuando se usa el operador in para verificar si un objeto está en una colección.

• La función __len__() se llama cuando se usa el operador len() para

4.1.3 Detalles del Artículo

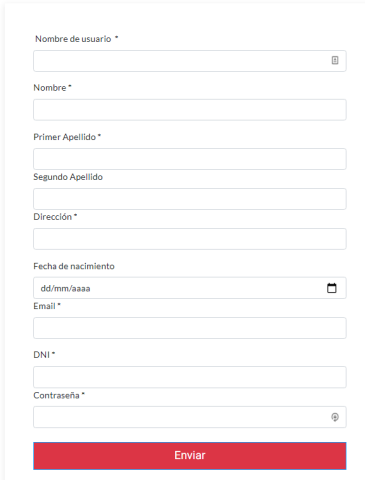
4.1.4. Log-in



A login form centered on a light blue background. It features a red circular logo with a white arrow pointing right. Below the logo are two input fields: 'Nombre de usuario' and 'Password', each with a small eye icon to its right. At the bottom is a red button labeled 'Enviar'.

4.1.5 Registro

REGISTRATE









A registration form with multiple input fields. The fields are: 'Nombre de usuario *' (with an eye icon), 'Nombre *', 'Primer Apellido *', 'Segundo Apellido', 'Dirección *', 'Fecha de nacimiento' (with a date format 'dd/mm/yyyy' and a calendar icon), 'Email *', 'DNI *', and 'Contraseña *' (with an eye icon). A red button labeled 'Enviar' is at the bottom.

4.1.6. Carrito

Carjotu

InicioCatálogotestLog outCarrito

PRODUCT	PRECIO	CANTIDAD	SUBTOTAL	BORRAR
 Estaba preparado para todo menos para ti Alfaguara	17,95€	+ 1 -	17,95€	
 ejemplo.tes Alfaguara	22,00€	+ 2 -	44,00€	
 HARRY POTTER Y LA PIEDRA FILOSOFAL Salamandra	9,45€	+ 1 -	9,45€	

RESUMEN DEL PEDIDO

Desglose del pedido antes y despues de impuestos

Subtotal	71,40€
Impuestos	14,99€
Total	86,39€

Proceder a la compra

4.1.7. Mantenimiento Usuario

Carjotu

InicioCatálogotestLog outCarrito

ACTUALIZAR DATOS

test

Nombre *

Nombre

Primer Apellido *

Apellido

Segundo Apellido

Apellido

Dirección *

Iturribide 99

Fecha de nacimiento

12/12/2011

Email *

test@test.com

00000000A

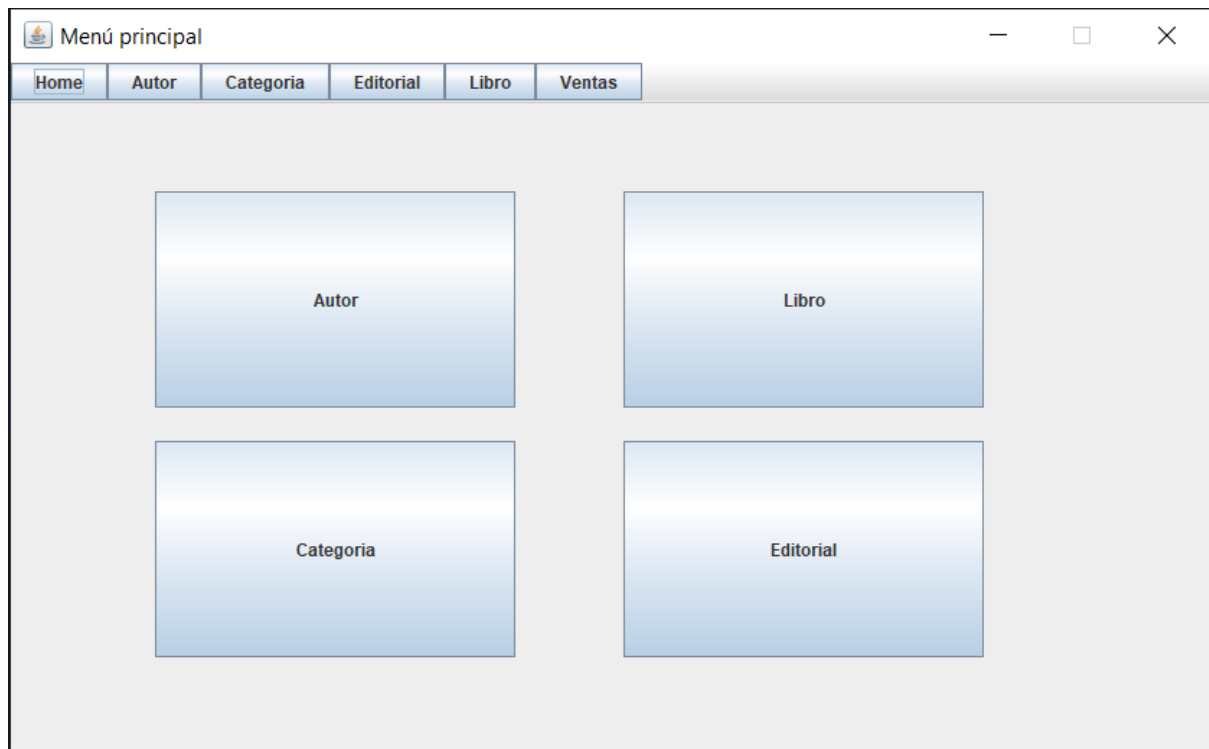
Contraseña *

Guardar cambios


Mapa de navegación de la web

4.2 Pantallas Aplicación escritorio

4.2.1. Menú principal



4.2.2. Mantenimiento Libros

 Menú principal

Home

Autor

Categoría

Editorial

Libro

Ventas

Listado de Libros

ISBN	Título	Precio	Stock	Imagen	Descripción	Editorial	Categoría
111111111	hello	0.25	2	111111111.png	descripcion.	Alfaguara	Test
123123123	ejemplo tes	22.0	2	1110001110.p...	algo por ponep...	Alfaguara	Test
123456789	test libro	99.0	8	123456789.png	ksfjdjfsod	Planeta	Misterio
420460499	El Italiano	21.9	27	default.png		Alfaguara	Novela Conte...
425360503	Estaba prepar...	17.95	5	default.png	algo	Alfaguara	Test
448028831	La cuenta atra...	17.9	3	default.png		Alfaguara	Test
490669983	Los vencejos	23.75	9	patatas.png		Tusquets Edito...	Novela Conte...
1110001110	libro	12.25	27	1110001110.p...	prueba editar p...	Alfaguara	Test
1110002220	test bugs	1.11	2	default.png	Test edicion d...	Tusquets Edito...	Misterio
1122334455	libro para hace...	12.5	200	1122334455.p...	texto de ejemplo	Alfaguara	Test
8498382662	HARRY POTTE...	9.45	16	8498382662.p...	Harry Potter y l...	Salamandra	Fantasia
9999999999	aaaaaaaaaaa...	1.01	0	default.png	descripcion.	prueba	Test

Autores Actuales

Añadir Autor

Código Autor	Nombre	Primer Apellido	Segundo Apellido

DROP Autor

Imagen:

ISBN:

Precio:

€

Añadir Libro

Título:

Editar Libro

Descripción:

Eliminar Libro

cambiar IMG

Stock:

+

0

-

Editorial


Alfaguara

Categoría

Fantasia

Refresh Ed. Cat.

4.2.3. Mantenimiento Autores

 Menú principal

Home

Autor

Categoría

Editorial

Libro

Ventas

Listado de Autores

Código	Nombre	Primer Apellido	Segundo Apellido
AAA000	pruebas	test	algo
AAA001	test	refresh	libros
AES001	Albert	Espinosa	Romero
APR001	Arturo	Pérez	Reverte
CRZ001	Carlos	Ruiz	Zafon
EAP001	Edgar	Allan	Poe
FAR001	Fernando	Aramburu	
FKA001	Franz	Kafka	
GGM001	Gabriel	Garcia	Marquez
GOR001	George	Orwell	
JAU001	Jane	Austin	
JKR001	Joanne	Rowling	
JRR001	John Ronald	Reuel	Tolkien
JSC001	Javier	Santaolalla	Camino
KLE001	Kai	Lenny	
LVR001	La Vecina	Rubia	

Editar Autor

Eliminar Autor

Añadir Autor


Codigo de Autor:

Nombre:

Primer Apellido:

Segundo Apellido:

4.2.4. Mantenimiento Categorías

 Menú principal

Home

Autor

Categoría

Editorial

Libro

Ventas

Listado de Categoría

Código	Nombre
FAN001	Fantasia
jkl192	Test
MIS001	Misterio
NCO001	Novela Contemporánea

Editar Categoría

Eliminar Categoría

Añadir Categoría

Codigo de Categoría:

Nombre:

4.2.5. Mantenimiento Editoriales

Menú principal

Home Autor Categoría **Editorial** Libro Ventas

Listado de Editoriales

Código	Nombre
ALF001	Alfaguara
asd213	prueba
GEO001	GeoPlaneta
GRI001	Grijalbo
LES001	La Esfera
PLA001	Planeta
SAL001	Salamandra
TED001	Tusquets Editores
TMA001	Timun mas

Editar Editorial

Eliminar Editorial

Añadir Editorial

Codigo de Editorial:

Nombre:

4.2.6. Generar Informe de ventas

Menú principal

Home Autor Categoría Editorial Libro **Ventas**

Listado de Ventas:

Nº de pedido	Fecha	Usuario que ha hecho el pedido:
1	2021-10-14	test
2	2021-12-23	chris

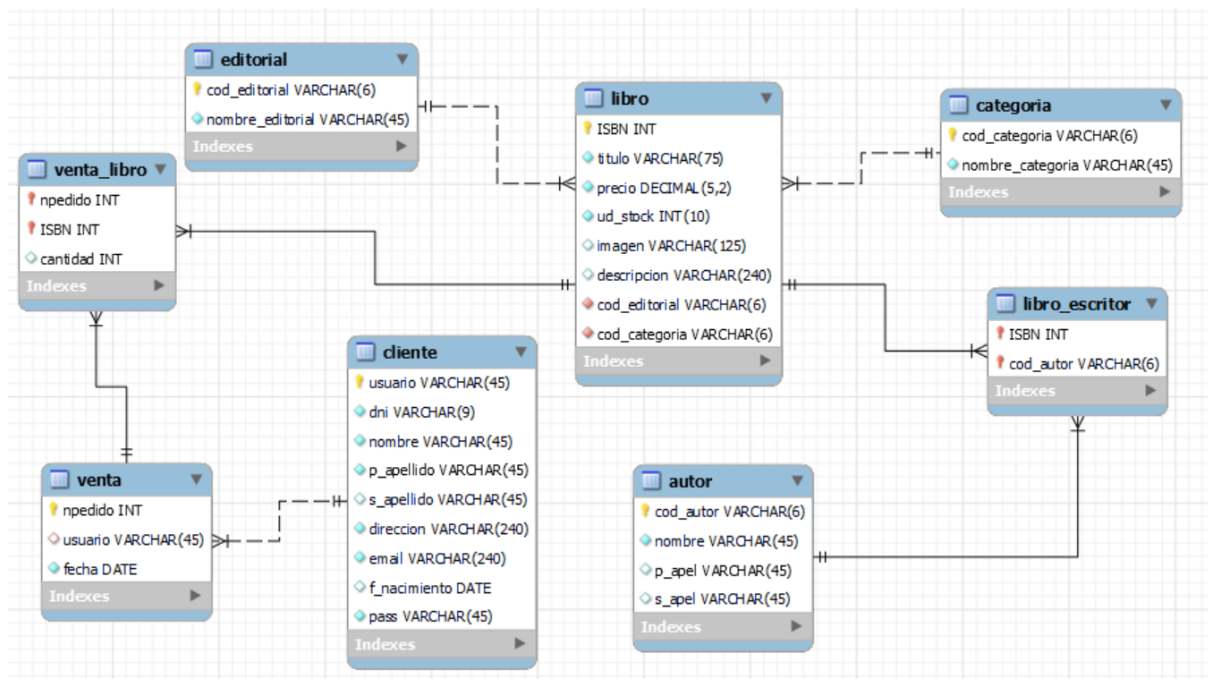
Generar TXT

Generar XML

Generar PDF

5. Modelo de Datos:

5.1. Diagrama de Entidad/Relación



6. Dependencias

Librerías y clases en el standalone:

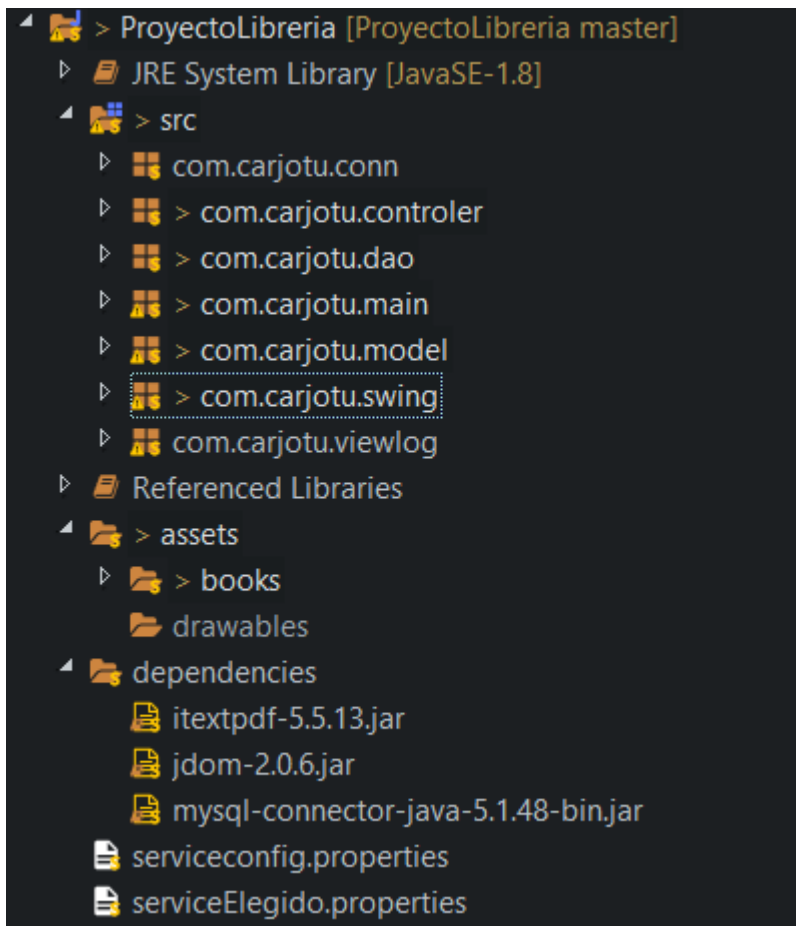
- Jdbc- mysql: conexión con la BBDD.
- ltext : crear pdfs.
- Jdom: Librería para crear documentos (Creación del documento XML).
- Clase JFileChooser : Elegir archivos, para seleccionar las imágenes.

Librerías y clases aplicación web:

- Jdbc- mysql: conexión con la BBDD.
- Framework Bootstrap : Framework para diseño de la web.























7. Estructura de los proyectos

Proyecto Standalone:



- conn → Archivos para la conexión a la base de datos
- controler → Controladores
- dao → DAOs para acceder a los datos de la base de datos
- main → Class main para iniciar el programa
- model → Modelo único
- swing → Vista principal y paneles, en swing
- viewlog → Vistas en consola
- assets
 - books → Imágenes de los libros
 - drawables → imágenes para iconos(sin usar)
- dependencies → librerías necesarias
- archivos de configuración

Proyecto web:

- ▼  ConnPass [ConnPass master]
 - >  JAX-WS Web Services
 - ▼  Java Resources
 - ▼  src
 - >  com.carjotu.beans
 - >  com.carjotu.conn
 - >  com.carjotu.controler
 - >  com.carjotu.dao
 - >  com.carjotu.model
 - >  com.carjotu.servletses
 - >  Libraries
 - >  build
 - ▼  WebContent
 - >  assets
 - >  META-INF
 - >  WEB-INF
 -  carrito.jsp
 -  catalogo.jsp
 -  index.jsp
 -  login.jsp
 -  signup.jsp
 -  user.jsp