

php7

php7

php7

php7

DOCUMENTACIÓN SOBRE ALMACÉN DE CAJAS

PRESENTACIÓN

La documentación de este proyecto se basa en un almacén de cajas en el cual se pueden registrar estanterías, las cuales poseen en su interior cajas. Se pueden realizar la salida de una o varias cajas y la devolución de la o las mismas.

El lenguaje de programación utilizado ha sido PHP.

Para llevar a cabo este proyecto, he utilizado el patrón de diseño arquitectónico de software denominado Modelo-Vista-Controlador (MVC). En este tipo de arquitectura lo que quiero conseguir es el control de las entradas y salidas del sistema, en el cual uno o varios modelos se encargan de buscar los datos y la información necesaria y mostrarlos en una interfaz que se mostrará para el usuario final.

Al dividirse en tres componentes, cada uno realiza una función.

El modelo es el encargado de manipular, gestionar y actualizar los datos. Por otro lado, la vista es el componente que se encarga de mostrar al usuario final, el resultado de una solicitud. Por último, el controlador es el componente encargado de gestionar las instrucciones que recibe, atenderlas y procesarlas. Por medio de él, se comunican el modelo y la vista, los cuales solicitan los datos necesarios, manipulándolos para obtener los resultados y así entregárselos a la vista para que se puedan mostrar.

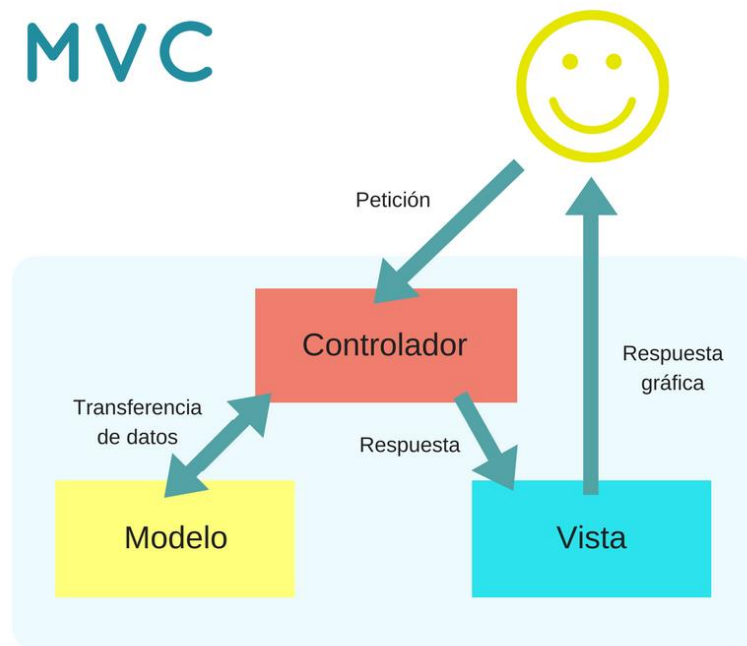
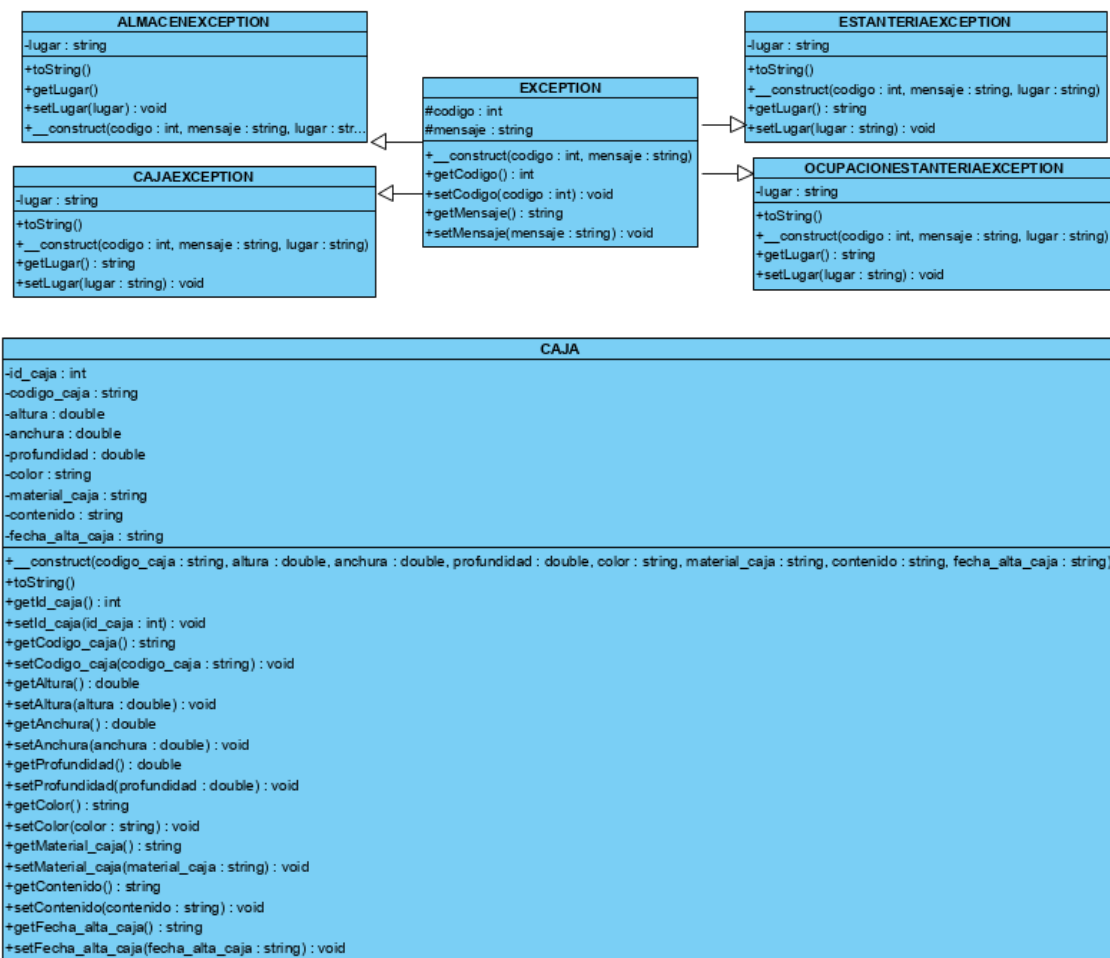


DIAGRAMA DE CLASES

Un diagrama de clases es una herramienta para comunicar el diseño de un programa que se creó para orientar objetos y que permite modelar relaciones entre diferentes entidades.

Para realizar los diagramas de clases (las clases que se encuentran dentro del modelo), voy a utilizar la aplicación de Visual Paradigm 16.1.

- El signo + indica que el atributo o el método será visible tanto dentro como fuera de clase, es decir, es accesible desde todos lados.
- El signo – indica que el atributo o método sólo será accesible desde dentro de la clase.
- El signo # indica que el atributo o método no será accesible desde fuera de la clase, pero si podrá ser utilizado o manipulado por métodos de la clase y de sus subclases.
- El signo → indica que una clase derivada hereda los métodos y atributos especificados por una clase base, por lo que una clase derivada además de tener sus propios métodos y atributos, podrá acceder a los métodos y atributos visibles de su clase base.



| ESTANTERIA |
|---|
| -id_estanteria : int -codigo_estanteria : string -material_estanteria : string -lejas : int -pasillo : int -numero : int -lejas_ocupadas : int -fecha_alta_estanteria : string |
| +toString() +__construct(codigo_estanteria : string, material_estanteria : string, lejas : int, fecha_alta_caja : string, pasillo : int, numero : int) +getId_estanteria() : int +setId_estanteria(id_estanteria : int) : void +getCodigo_estanteria() : string +setCodigo_estanteria(codigo_estanteria : string) : void +getMaterial_estanteria() : string +setMaterial_estanteria(material_estanteria : string) : void +getLejas() : int +setLejas(lejas : int) : void +getPasillo() : int +setPasillo(pasillo : int) : void +getNumero() : int +setNumero(numero : int) : void +getLejas_ocupadas() : int +setLejas_ocupadas(lejas_ocupadas : int) : void +getFecha_alta_estanteria() : string +setFecha_alta_estanteria(fecha_alta_estanteria : string) : void |

| OCUPACIONESTANTERIA |
|--|
| -id : int -id_estanteria : int -lejaOcupada : int -id_caja : int |
| +toString() +__construct(id_estanteria : int, lejaOcupada : int) +getId() : int +setId(id : int) : void +getId_estanteria() : int +setId_estanteria(id_estanteria : int) : void +getLejaOcupada() : int +setLejaOcupada(lejaOcupada : int) : void +getId_caja() : int +setId_caja(id_caja : int) : void |

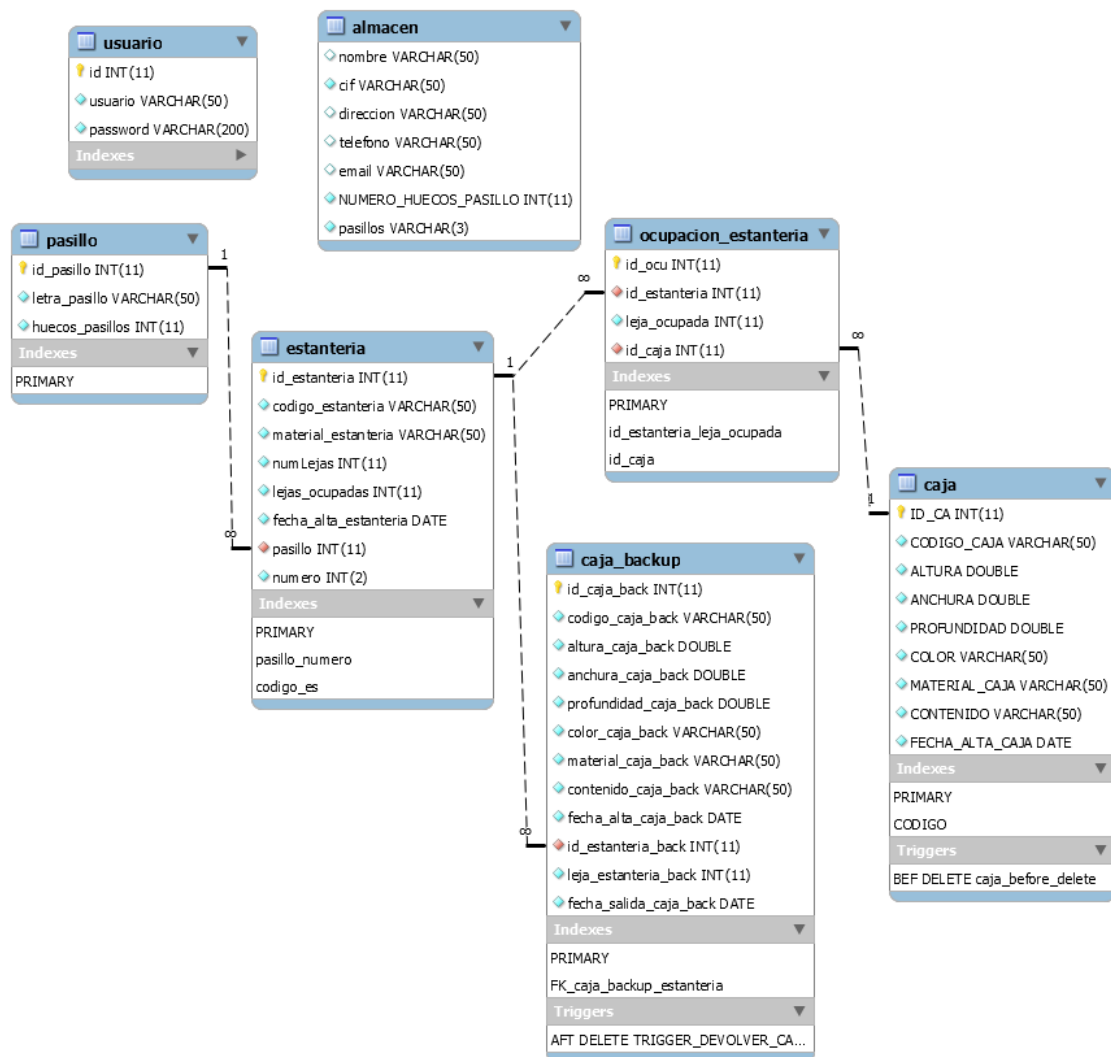
| PASILLO |
|--|
| -id_pasillo : int -letra_pasillo : string -huecos_ocupados : int |
| +__construct(id_pasillo : int, letra_pasillo : string) +toString() +getId_pasillo() : int +setId_pasillo(id_pasillo : int) : void +getLetra_pasillo() : string +setLetra_pasillo(letra_pasillo : string) : void +getHuecos_ocupados() : int +setHuecos_ocupados(huecos_ocupados : int) : void |

Visual Paradigm deja una marca de agua en la imagen que exporta del diagrama de clases, de todas maneras, la imagen se encuentra guardada en la carpeta.

ESQUEMA DE TABLAS Y SUS RELACIONES

El esquema de tablas describe la estructura de una base de datos, en un lenguaje formal soportado por un sistema de gestión de datos. El esquema va a definir las tablas, los campos de cada tabla y las relaciones entre campos y tablas.

Para realizar el esquema de tablas y sus relaciones uso el programa Workbench.

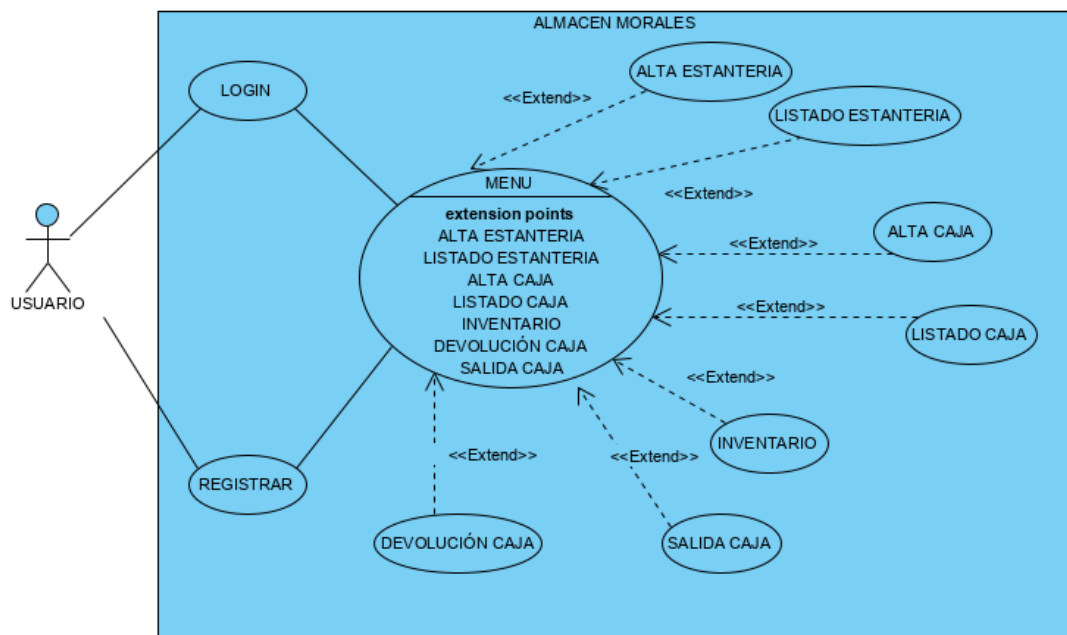


CASOS DE USO Y FLUJO DE DATOS

Un caso de uso es la descripción de una acción o actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso.

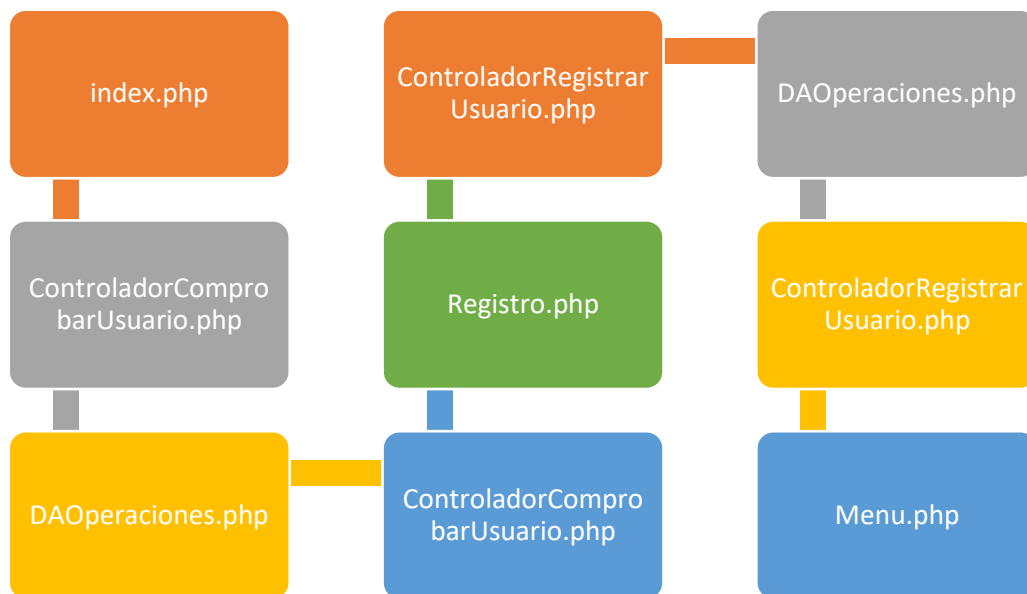
En este caso, muestro el caso de uso general de la aplicación.

Para más aclaraciones, ALMACEN MORALES actúa como sistema.



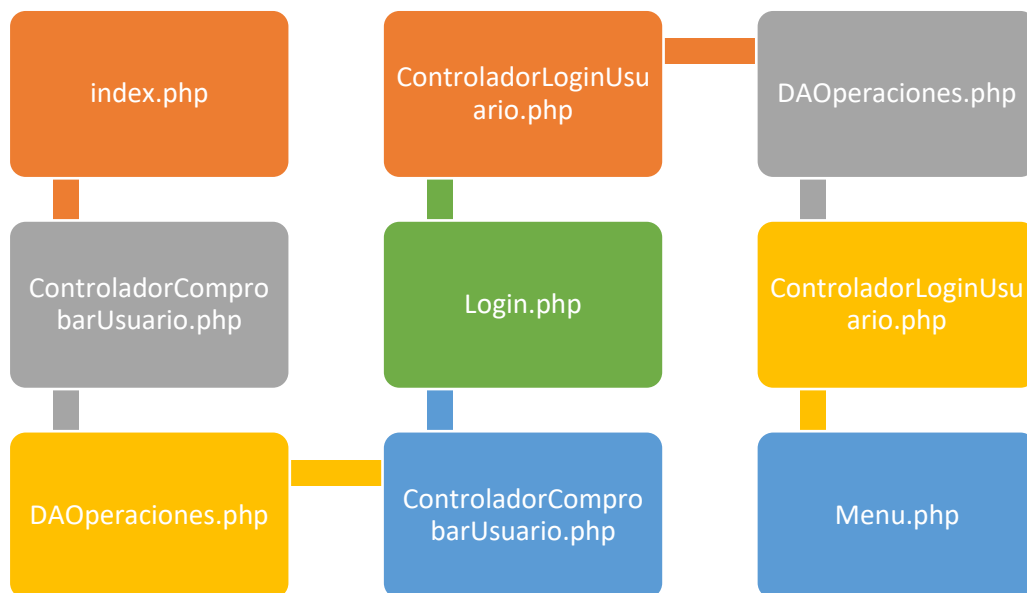
El flujo de datos de cada caso de uso se puede documentar mientras depuras el proyecto para así comprobar por qué archivos pasa.

| |
|--|
| Nombre: Registrar |
| ID: CU-1 |
| Descripción: El usuario carga la aplicación, como es la primera vez y no existen usuarios en la base de datos, se registra introduciendo el nombre de usuario y la contraseña. |
| Actores: Usuario. |
| Precondiciones: No debe haber ningún usuario almacenado en la base de datos. |
| Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario carga la aplicación. 2. El usuario escribe el nombre de usuario y la contraseña. 3. El usuario pulsa el botón de registrar. 4. El sistema registra el usuario. |
| Postcondiciones: El sistema registra al usuario introducido en la base de datos. |
| Alternativa: <ol style="list-style-type: none"> 1. Si no introduce alguno de los dos datos requeridos, no podrá registrarse. |
| Fin de caso de uso. |



En el caso de que hubiera cualquier error, se lanzaría el catch y nos llevaría hasta VistaErrores.php.

| |
|---|
| Nombre: Login |
| ID: CU-2 |
| Descripción: El usuario carga la aplicación y como ya hay datos almacenados en la base de datos, introduce los datos correspondientes. |
| Actores: Usuario. |
| Precondiciones: Debe haber un usuario ya existente en la base de datos. |
| Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario carga la aplicación. 2. El usuario escribe el nombre de usuario y la contraseña. 3. El usuario pulsa el botón de iniciar sesión. 4. El sistema comprueba que los datos son correctos 5. El usuario accede al menú. |
| Postcondiciones: <ol style="list-style-type: none"> 1. El sistema comprueba que existe un usuario en la base de datos. 2. El sistema comprueba que el nombre de usuario y la contraseña son correctas. 3. El usuario accede al menú de la aplicación. |
| Alternativa: <ol style="list-style-type: none"> 1. Si no introduce alguno de los datos correctamente no podrá iniciar sesión. 2. Si deja algún campo vacío tampoco podrá iniciar sesión. |
| Fin de caso de uso. |



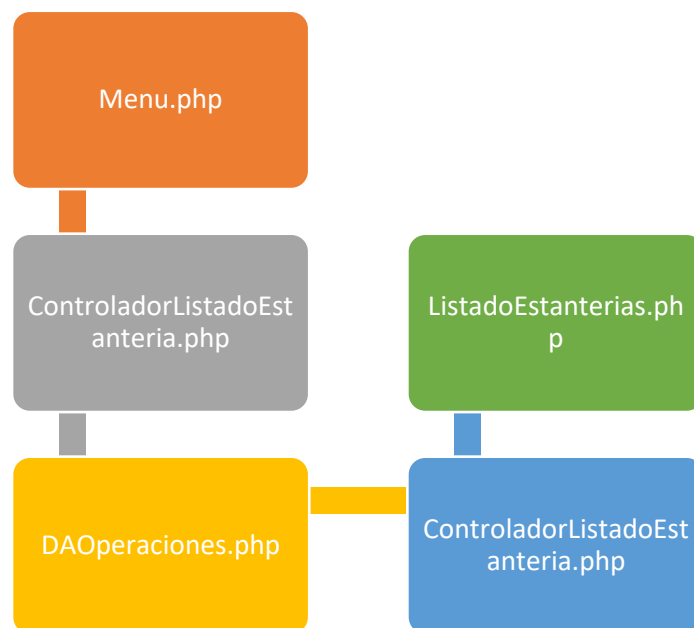
En el caso de que hubiera cualquier error, se lanzaría el catch y nos llevaría hasta VistaErrores.php.

| |
|--|
| Nombre: Alta Estantería |
| ID: CU-3 |
| Descripción: El usuario decide dar de alta una estantería. |
| Actores: Usuario |
| Precondiciones: Debe haber huecos libres en alguno de los tres pasillos. |
| Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario está en el menú y clicca sobre Estantería-> Alta. 2. El sistema comprueba que hay huecos libres en alguno de los tres pasillos. 3. El usuario introduce los datos correspondientes y cuando llegue a la elección del pasillo, selecciona el que desea. 4. El sistema muestra los huecos disponibles que tiene ese pasillo. 5. El usuario selecciona el número que quiere. 6. El usuario clicca sobre Dar de Alta. |
| Postcondiciones: <ol style="list-style-type: none"> 1. El sistema comprueba que estén todos los datos introducidos. 2. El sistema comprueba que no hay ninguna excepción ni error para dar de alta una estantería. 3. El sistema lo registra en la base de datos. |
| Alternativa: <ol style="list-style-type: none"> 1. Si no introduce alguno de los datos correctamente no podrá dar de alta una estantería. 2. Si el código de la estantería ya existe en la base de datos. 3. Si el usuario deja un campo vacío no podrá dar la estantería de alta tampoco. |
| Fin de caso de uso. |



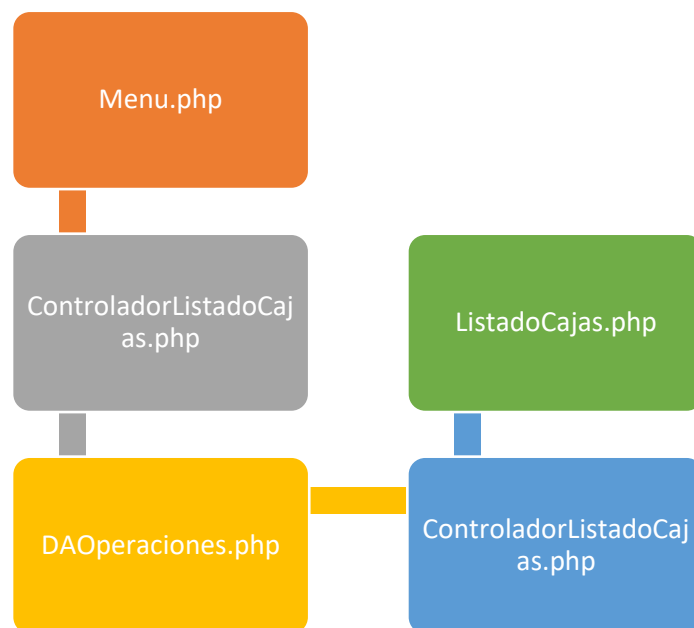
En el caso de que hubiera cualquier error, se lanzaría el catch y nos llevaría hasta VistaErrores.php.

| |
|--|
| Nombre: Listado Estantería |
| ID: CU-4 |
| Descripción: Muestra un listado completo de todas las estanterías. |
| Actores: Usuario. |
| Precondiciones: Debe haber estanterías dadas de alta. |
| Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario está en el menú y clicla sobre Estantería-> Listado. 2. El sistema comprueba que hay estanterías en la base de datos. 3. El sistema muestra el listado de estanterías. |
| Postcondiciones: <ol style="list-style-type: none"> 1. El sistema comprueba que existen estanterías dadas de alta. |
| Alternativa: <ol style="list-style-type: none"> 1. No existan estanterías en la base de datos. |
| Fin de caso de uso. |



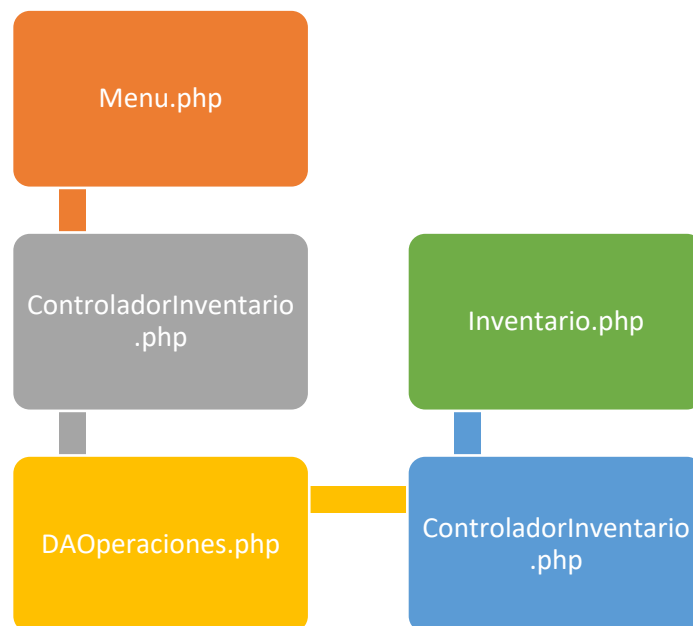
En el caso de que hubiera cualquier error, se lanzaría el catch y nos llevaría hasta VistaErrores.php.

| |
|---|
| Nombre: Listado Caja |
| ID: CU-6 |
| Descripción: Muestra un listado completo de todas las cajas. |
| Actores: Usuario. |
| Precondiciones: Debe haber cajas dadas de alta. |
| Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario está en el menú y clicla sobre Caja-> Listado. 2. El sistema comprueba que hay caja en la base de datos. 3. El sistema muestra el listado de cajas. |
| Postcondiciones: <ol style="list-style-type: none"> 1. El sistema comprueba que existen cajas dadas de alta. |
| Alternativa: <ol style="list-style-type: none"> 1. No existan cajas en la base de datos. |
| Fin de caso de uso. |



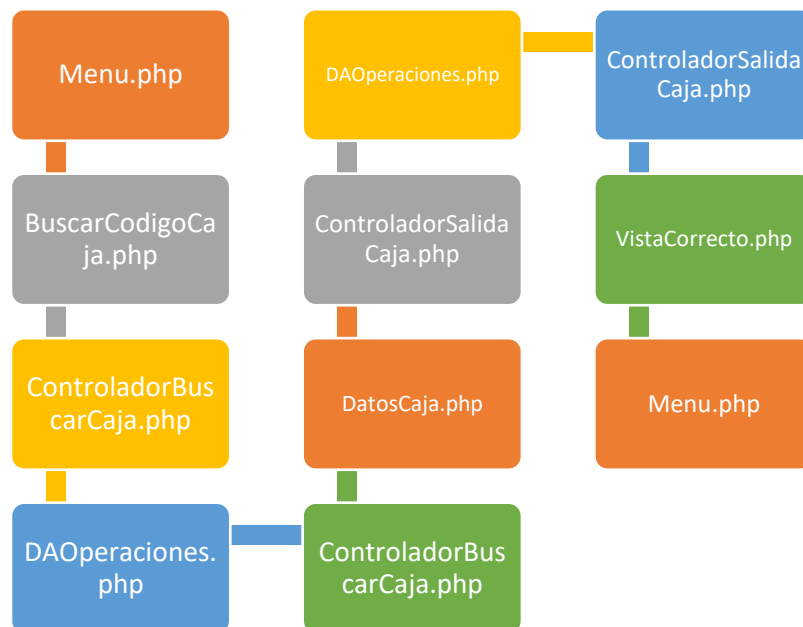
En el caso de que hubiera cualquier error, se lanzaría el catch y nos llevaría hasta VistaErrores.php.

| |
|--|
| Nombre: Inventario |
| ID: CU-7 |
| Descripción: Muestra un listado completo de todas las estanterías con sus cajas correspondientes. |
| Actores: Usuario. |
| Precondiciones: Debe haber como mínimo registradas estanterías en la base de datos. |
| Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario está en el menú y clicla sobre Inventario. 2. El sistema comprueba que hay estanterías y/o cajas en la base de datos. 3. El sistema muestra el listado de las estanterías con las cajas que contiene. |
| Postcondiciones: <ol style="list-style-type: none"> 1. El sistema comprueba que existen como mínimo estanterías dadas de alta. |
| Alternativa: <ol style="list-style-type: none"> 1. No existan estanterías en la base de datos. |
| Fin de caso de uso. |



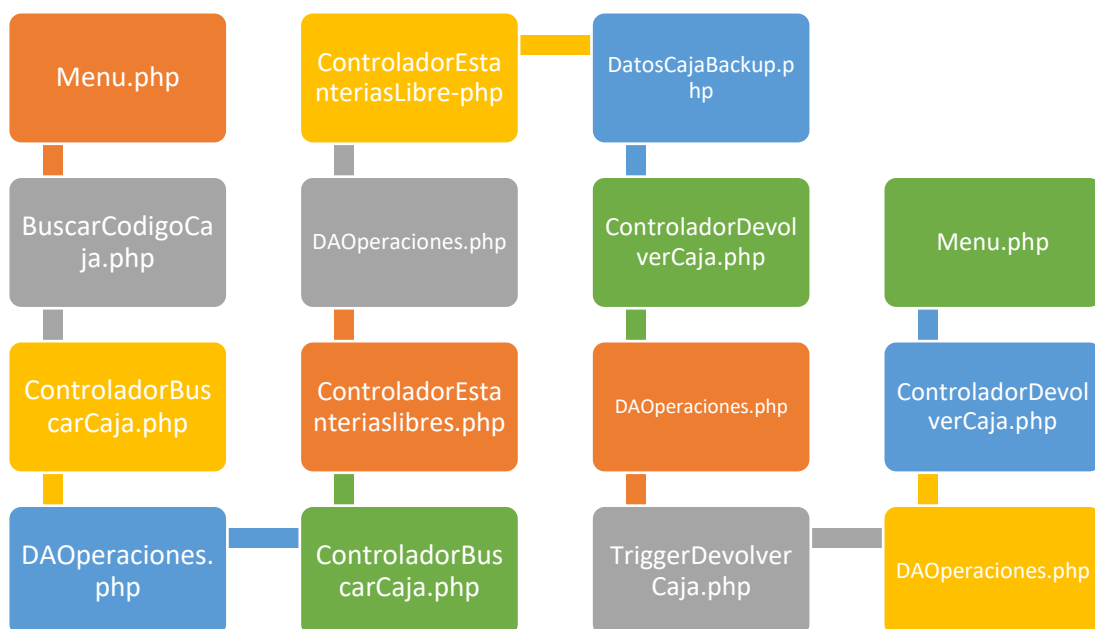
En el caso de que hubiera cualquier error, se lanzaría el catch y nos llevaría hasta VistaErrores.php.

| |
|---|
| Nombre: Salida Caja |
| ID: CU-8 |
| Descripción: El usuario decide dar de baja una caja. |
| Actores: Usuario. |
| Precondiciones: Debe estar dentro de una estantería. |
| Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario está en el menú y clicla sobre Operaciones Caja. 2. El usuario introduce el código de la caja y clicla sobre Salida de Caja. 3. El sistema muestra los datos de la caja en una vista. 4. El usuario comprueba los datos y clicla sobre dar de baja. 5. El sistema elimina la caja y la introduce en la tabla CajaBackup. |
| Postcondiciones: <ol style="list-style-type: none"> 1. El sistema comprueba que introduce el código de la caja. 2. El sistema comprueba que existe la caja. 3. Si todo es correcto, regresa al menú. |
| Alternativa: <ol style="list-style-type: none"> 1. No introduzca el código de la caja. 2. Introduzca el código de la caja incorrectamente. 3. La caja no se encuentre en la base de datos. |
| Fin de caso de uso. |



En el caso de que hubiera cualquier error, se lanzaría el catch y nos llevaría hasta VistaErrores.php.

| |
|--|
| Nombre: Devolución Caja |
| ID: CU-9 |
| Descripción: El usuario decide devolver una caja. |
| Actores: Usuario. |
| Precondiciones: Para devolver una caja a su estantería, tiene que estar la caja en backup y haberlejas libres en la estantería. |
| Curso normal del caso de uso: <ol style="list-style-type: none"> 1. El usuario está en el menú y clicla sobre Operaciones Caja. 2. El usuario introduce el código de la caja y clicla sobre Devolver. 3. El sistema muestra los datos de la caja en una vista. 4. El usuario comprueba los datos y clicla sobre Devolución. 5. El sistema introduce la caja en tabla Caja y la borra de la tabla backup. |
| Postcondiciones: <ol style="list-style-type: none"> 1. El sistema comprueba que introduce el código de la caja. 2. El sistema comprueba que existe la caja en backup. 3. Si todo es correcto, regresa al menú. |
| Alternativa: <ol style="list-style-type: none"> 1. No introduzca el código de la caja. 2. Introduzca el código de la caja incorrectamente. 3. La caja no se encuentre en la base de datos. |
| Fin de caso de uso. |



En el caso de que hubiera cualquier error, se lanzaría el catch y nos llevaría hasta VistaErrores.php.

VARIABLES DE SESIÓN

| VARIABLE | CONTENIDO | CREACIÓN | UTILIZACIÓN |
|-------------------|--|------------------------------|-------------------|
| \$ArrayPasillo | <div><div><div><div><div></div><div></div></div><div>\$ArrayPasillo</div><div>array[3]</div></div><div><div><div><div></div><div></div></div><div>[0]</div><div>Pasillo</div></div><div><div><div><div></div><div></div></div><div>id_pasillo</div><div>string</div></div><div><div><div><div></div><div></div></div><div>letra_pasillo</div><div>string</div></div></div><div><div><div><div></div><div></div></div><div>[1]</div><div>Pasillo</div></div><div><div><div><div></div><div></div></div><div>id_pasillo</div><div>string</div></div><div><div><div><div></div><div></div></div><div>letra_pasillo</div><div>string</div></div></div><div><div><div><div></div><div></div></div><div>[2]</div><div>Pasillo</div></div><div><div><div><div></div><div></div></div><div>id_pasillo</div><div>string</div></div><div><div><div><div></div><div></div></div><div>letra_pasillo</div><div>string</div></div></div></div></div></div></div></div></div></div></div> | ControladorCargarPasillo | AltaEstanteria |
| \$ArrayEstanteria | <div><div><div><div><div></div><div></div></div><div>\$ArrayEstanterias</div><div>array[1]</div></div><div><div><div><div></div><div></div></div><div>[0]</div><div>stdClass</div></div></div></div></div> | ControladorListadoEstanteria | ListadoEstanteria |
| \$ArrayEstanteria | <div><div><div><div><div></div><div></div></div><div>\$ArrayEstanteria</div><div>array[1]</div></div><div><div><div><div></div><div></div></div><div>[0]</div><div>Estanteria</div></div></div></div><div></div></div> | ControladorEstanteriasLibres | AltaCaja |
| \$ArrayInventario | <div><div><div><div><div></div><div></div></div><div>\$ArrayInventario</div><div>array[2]</div></div><div><div><div><div></div><div></div></div><div>[0]</div><div>Estanteria</div></div><div><div><div><div></div><div></div></div><div>[1]</div><div>Caja</div></div></div></div></div></div> | ControladorInventario | Inventario |
| \$ArrayCaja | <div><div><div><div><div></div><div></div></div><div>\$ArrayCaja</div><div>array[1]</div></div><div><div><div><div></div><div></div></div><div>[0]</div><div>stdClass</div></div></div></div></div> | ControladorBuscarCaja | DatosCaja |
| \$ArrayCajaBck | <div><div><div><div><div></div><div></div></div><div>\$ArrayCajaBck</div><div>array[1]</div></div><div><div><div><div></div><div></div></div><div>[0]</div><div>stdClass</div></div></div></div></div> | ControladorBuscarCaja | DatosCajaBackup |