

Proyecto Almacén

Con PHP



Borja Mateo León

2º Desarrollo de aplicaciones web

Índice.

1. Introducción.....	2
2. Arquitectura.....	3
3. Diagrama de clases.....	4
4. Esquema de la base de datos.....	5
5. Casos de uso.....	5
6. Excepciones.....	14
7. Variables de sesión.....	14

1. Introducción.

En esta documentación expondré y explicare la aplicación web propuesta por nuestro profesor, en la que hemos estado trabajando todo este tiempo.

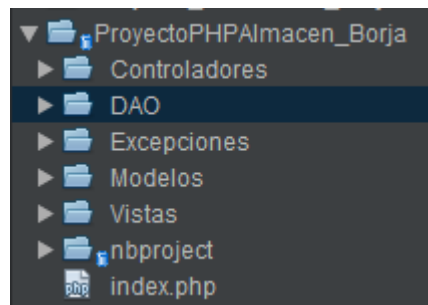
El proyecto simula la gestión de un almacén de cajas, donde podremos añadir estanterías a el almacén en una determinada ubicación para más tarde añadirle diferentes cajas en sus lejas. También podremos vender esas cajas y devolverlas. Podremos hacer un inventario exacto de nuestro almacén, también listar las cajas y estanterías.

Para la realización del proyecto he utilizado las siguientes tecnologías:

- Netbeans IDE 8.2
- PHP 7.0
- HTML5
- CSS3
- Generación dinámica de componentes web
- Bootstrap
- GIT y GitLab

2. Arquitectura.

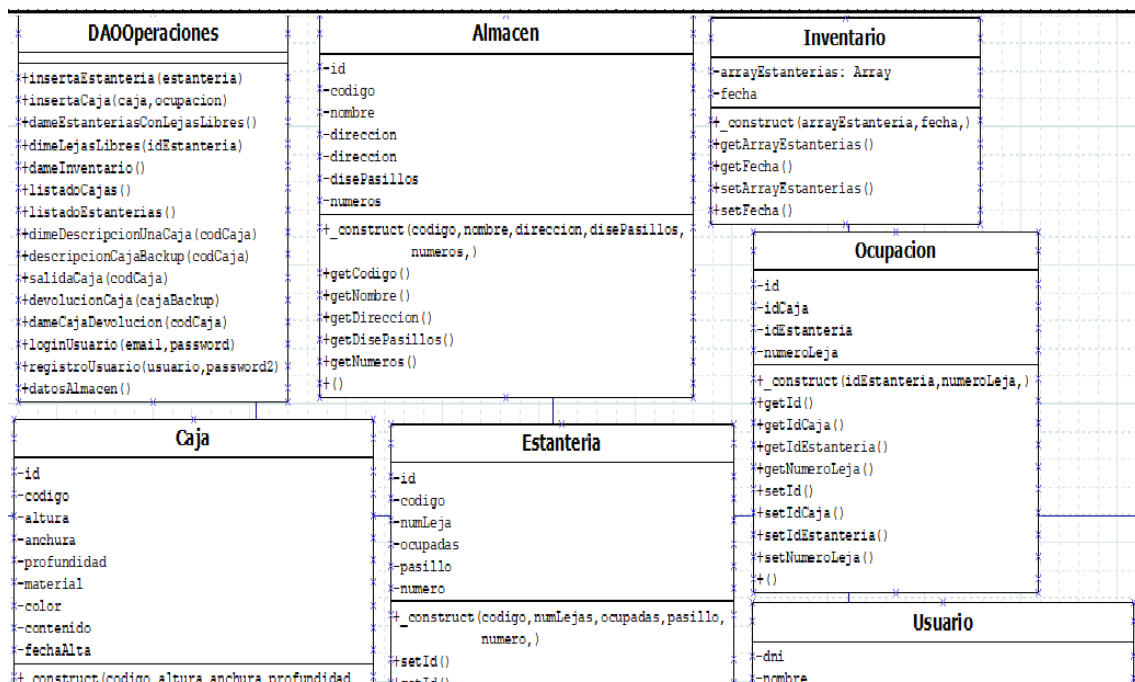
Para el desarrollo del proyecto hemos utilizado el patrón modelo vista controlador (MVC) con la capa DAO. Con el MVC conseguimos separar los datos y la lógica de negocio de una aplicación de su representación y el modulo encargado de gestionar los eventos y las comunicaciones



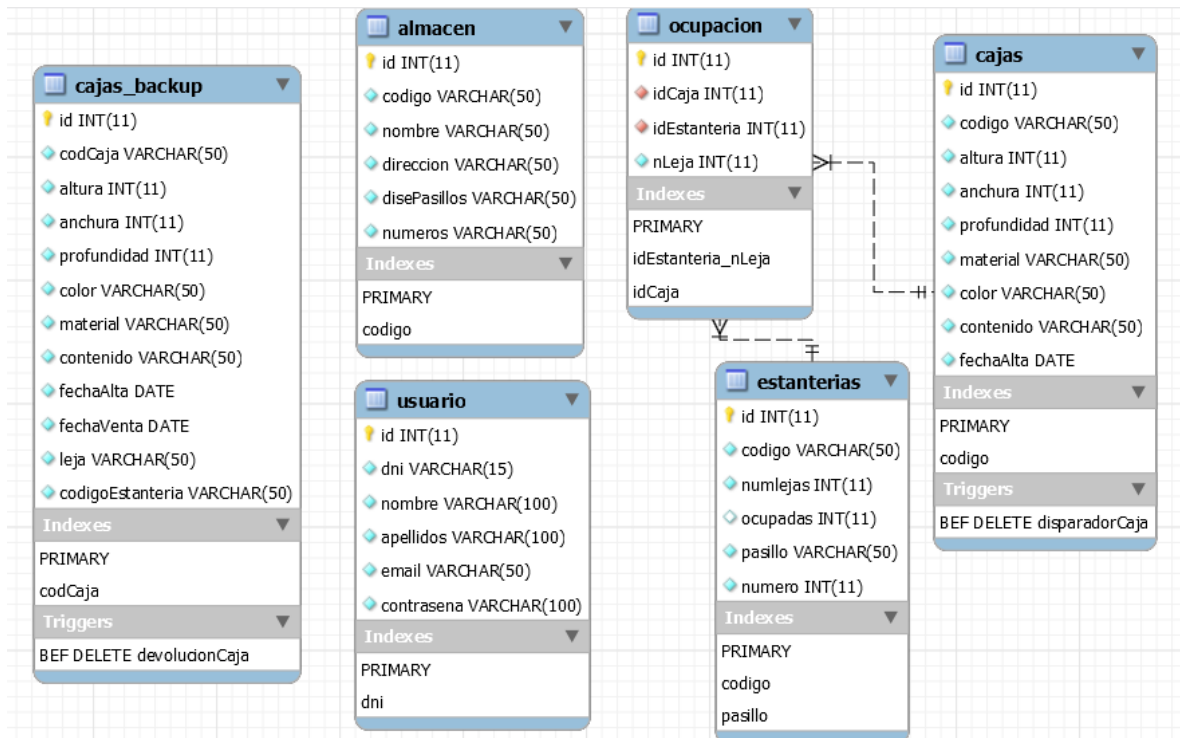
- **Controladores:** estos se encargan de recibir y filtrar datos que le llegan de las vistas, llamar a los modelos y ejecutar los métodos de la capa DAO
- **DAO:** aquí tenemos la conexión con la base de datos y otra clase DAOOperaciones.php que realiza las peticiones a la base de datos y le devuelve la información a los controladores entonces estos pasan los datos a la vista.
- **Excepciones:** en esta carpeta tenemos una clase que hereda de Exception que es quien captura las excepciones que se produzcan en nuestro código.

- **Modelos:** aquí tenemos los modelos, estas clases servirán para crear objetos de ese tipo de entidad.
- **Vistas:** aquí irán las vistas, es decir, donde se imprimirán los datos y lo que verá el usuario.

3. Diagrama de clases.



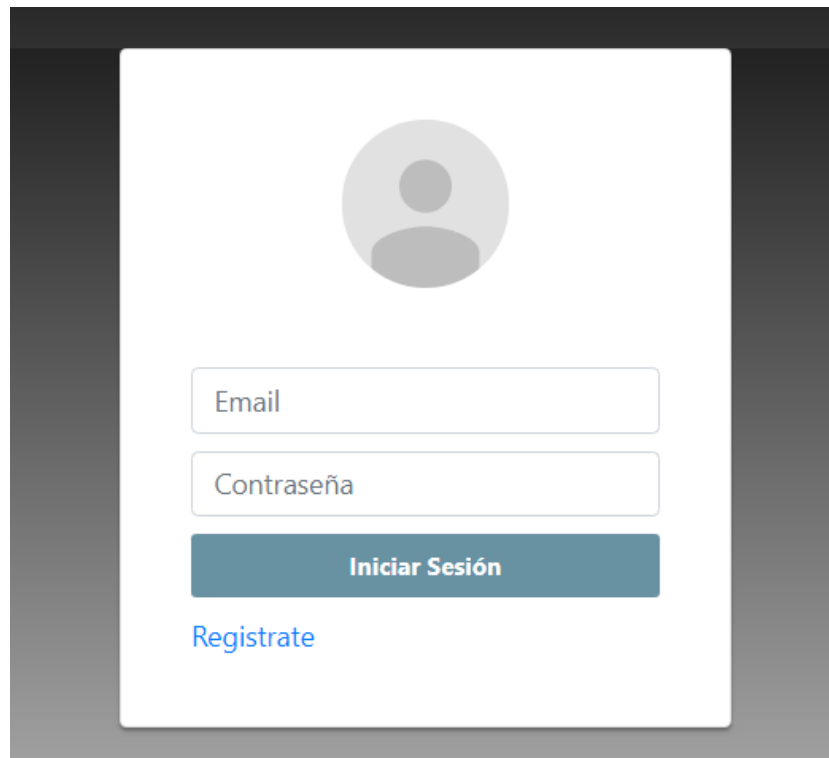
4. Esquema de la base de datos.



5. Casos de uso.

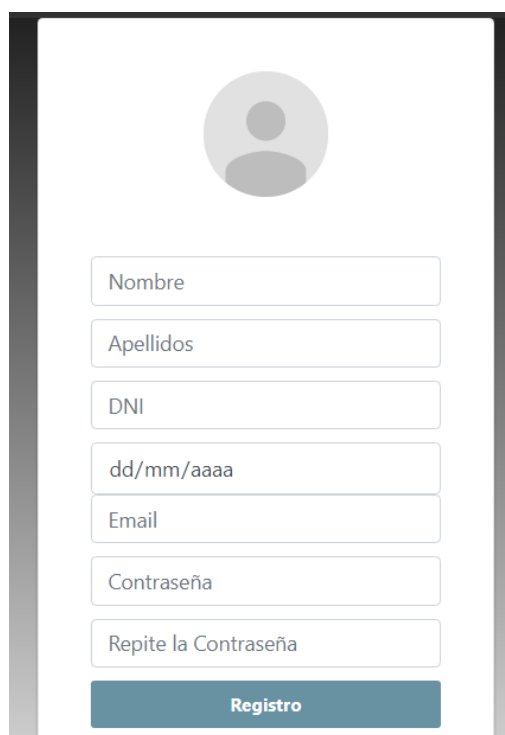
A. Login y registro en el sistema.

Lo primero que encontraremos al entrar en nuestra aplicación será la pantalla del Login donde tendremos dos opciones, si ya tenemos un usuario y contraseña podremos entrar a la gestión del almacén y sino debemos registrarnos en la base de datos.



A login form interface with a dark gray background. At the top center is a circular placeholder for a profile picture. Below it are two white input fields with gray borders, labeled 'Email' and 'Contraseña'. Under these fields is a blue button with white text that says 'Iniciar Sesión'. At the bottom left of the form is a blue link that says 'Registrate'.

Por el contrario si es la primera vez que utilizamos la aplicación debemos pulsar sobre “Registrate”, entonces aparecerá una nueva pantalla en la que podremos darnos de alta en la base de datos del sistema.



A registration form interface with a light gray background. At the top center is a circular placeholder for a profile picture. Below it are seven white input fields with gray borders, labeled 'Nombre', 'Apellidos', 'DNI', 'dd/mm/aaaa', 'Email', 'Contraseña', and 'Repite la Contraseña'. At the bottom center is a blue button with white text that says 'Registro'.

Flujo de datos.

1. Index.php
2. Login -> ControladorLogin.php ->



3. Registro -> VistaRegistro.php -> ControladorRegistro.php



B. Cajas.

Una vez estamos dentro de nuestro sistema ya podremos ver nuestro menú con las diferentes opciones y la información del almacén que estamos gestionando.

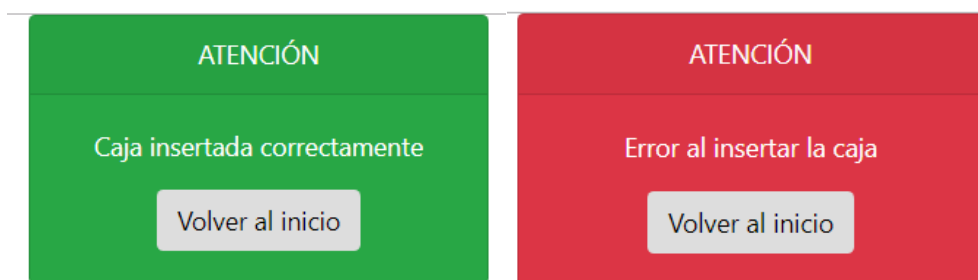


A01 ALMACEN BORJA
Codigo: A01
Nombre: ALMACEN BORJA
Direccion: C/CANTARRANAS
Pasillos: A-Z
Numeros: 1-100

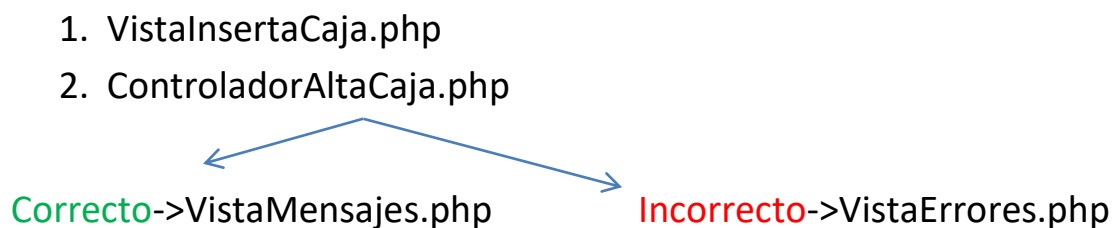
En el apartado Cajas podremos listar las cajas, añadir nuevas cajas, venderlas y devolverlas.

Alta de Caja.

En este apartado permitimos a un usuario registrado dar de alta nuevas cajas en el almacén. Para poder ingresar nuevas tablas debemos de tener al menos una estantería con una leja libre, sino debemos añadirla. Una vez en el registro de una nueva caja debemos ingresar el código, altura, anchura, profundidad, material, color, contenido, estantería y leja. Si ese código no existe en nuestra BD nos aparecerá un mensaje confirmando la operación que acabamos de hacer, si por el contrario surge cualquier error nos aparecerá un mensaje informándonos que la caja no se ha podido añadir correctamente.



Flujo de datos.



Listado de Cajas.

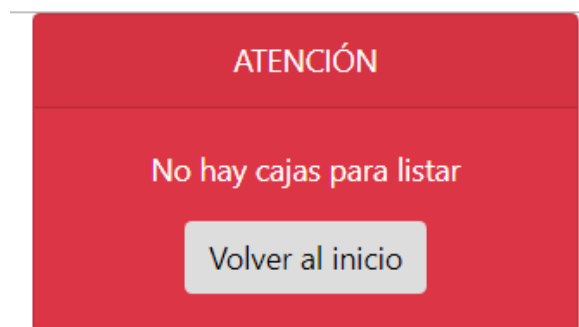
Nos muestra un listado detallado de todas las cajas dadas de alta en nuestra BD.

Listado de Cajas

Volver al inicio

Código	Altura	Anchura	Profundidad	Material	Color	Contenido	Fecha Alta
C01	10	10	10	PLASTICO		TELEFONOS MOVILES	26/11/2018
C02	10	10	10	MADERA		BOTELLAS	26/11/2018
C03	15	15	15	ALUMINIO		TORNILLOS	26/11/2018

Si no tenemos cajas en la BD nos mostrará un mensaje de error.



Flujo de datos.

1. ControladorListadoCajas.php

Correcto->VistaListadoCajas.php

Incorrecto->VistaErrores.php

Venta de Cajas.

Podemos vender las cajas existentes en la BD. Simplemente tenemos que introducir el código de la caja que queremos mostrar, mediante AJAX nos mostrara un mensaje de error si esa caja no existe o nos mostrara la descripción rápida de la caja. Una vez pulsamos sobre “Vender Caja” borramos esa caja de nuestra BD y se ejecuta un trigger que nos insertara esa caja en cajas_backup.



Devolución de Cajas.

También podremos devolver las cajas que han sido vendidas, una vez introducido el código de la caja, si encuentra la caja nos muestra toda la información de esa caja y nos deja darle una nueva ubicación en el almacén. Si no encuentra la caja nos muestra un mensaje de error. Una vez devolvemos la caja se borra de la tabla cajas_backup, se

inserta en la tabla Cajas, añade una ocupación más en la tabla estantería y una fila nueva en la tabla ocupación.

B. Estanterías.

Listado de Estanterías.

Nos muestra un listado detallado de todas las estanterías existentes en nuestra BD.

Listado de Estanterías

[Volver al inicio](#)

Código	Lejas	Lejas Ocupadas	Pasillo	Numero
ES01	10	2	A	1
ES02	15	0	A	2

Flujo de datos.

1. ControladorListadoEstanterias.php



Correcto->VistaListadoEstanterias.php Incorrecto->VistaErrores.php

Insertar estanterías.

En el apartado de Estanterías también podremos insertarlas a la base de datos. En este caso no podremos tener en la BD dos estanterías ni con el mismo código ni con el mismo id de caja ni id de estantería y leja.



Formulario de Alta de Estanteria

Codigo

Numero de Lejas

Pasillo

Numero

Flujo de datos.

1. ControladorInsertarEstanteria.php



Correcto->VistaMensajes.php Incorrecto->VistaErrores.php

C. Inventario.

Por último, en nuestra aplicación web podremos sacar un inventario detallado que nos mostrará la fecha en la que estamos solicitando el inventario, todas las estanterías de la BD y cada estantería las cajas que tiene guardadas. Si por el contrario no tenemos ninguna estantería añadida en nuestra BD nos mostrará un mensaje de error.

Inventario con Fecha: 26-11-2018, 20:58

[Volver al inicio](#)

Código	Lejas	Lejas Ocupadas	Pasillo	Numero
ES01	10	2	A	1

Código	Altura	Anchura	Profundidad	Material	Color	Contenido	Fecha Alta	Leja
C08	1	1	1	PLASTICO		TORNILLOS	26/11/2018	1
C01	10	10	10	#ff0000		TELEFONOS MOVILES	26/11/2018	2

Código	Lejas	Lejas Ocupadas	Pasillo	Numero
ES02	15	0	A	2

Flujo de datos.

1. ControladorInventario.php



Correcto->VistaInventario.php

Incorrecto->VistaErrores.php

6. Excepciones.

He tratado todos los errores que se produzcan en mi aplicación web con una excepción propia que no es más que una clase que hereda de Exception. Los errores pueden producirse en la capa DAO en la clase DAOOperaciones.php, aquí lanzamos la excepción que la recogemos en los controladores mediante un try- catch y decidimos si dirigimos a una vista de errores si se ha producido la excepción o dirigimos a una vista de éxito.

7. Variables de sesión.

Una sesión es una variable que se crea en el servidor y esta variable puede ejecutarse sin que el usuario de la web tenga conocimiento alguno de ella. Ahora vamos a ver las sesiones que utilizo en el proyecto y para que las utilizo:

- **Almacén:** guarda un objeto Almacén completo, lo utilizamos en la vista menu.php para sacar la información del almacén.
- **listadoEsteranterias:** guardamos un array de objetos Esterantería, la utilizamos en VistaListadoEsteranterias.php para sacar un listado de las esteranterías.
- **listadoCajas:** guardamos un array de objetos de tipo Caja, la utilizamos en VistaListadoCajas.php para sacar un listado de cajas.

- **Inventario:** guardamos un objeto Inventario y lo utilizamos en VistaInventario.php para sacar un listado del inventario del almacén.
- **CajaBackup:** guardamos un objeto de tipo CajaBackup para utilizarlo en VistaDevolucionCajaMuestra.php
- **Estanterías:** guardamos un array de objetos de tipo Estantería para utilizarlo al dar una nueva ubicación a la caja.
- **Lejas:** guardamos un array de lejas libres para utilizarlo al dar una nueva ubicación a la caja.
- **Caja:** guardamos la descripción de una caja para visualizarla cuando estamos vendiendo esa caja.