

Documentación Proyecto Almacén Cajas

MIRIAM MORENO SÁNCHEZ

2ºDAW

Índice

Relación de tablas	2
Diagrama de clases	3
Variables de sesión	4
Diagrama de flujo	5
Casos de uso	6
Log-in	7
Añadir Estantería	8
Añadir Caja.....	10
Inventario.....	11
Listar Estanterías	12
Listar Cajas.....	13
Sacar Cajas	14
Devolver Cajas	16

Relación de tablas

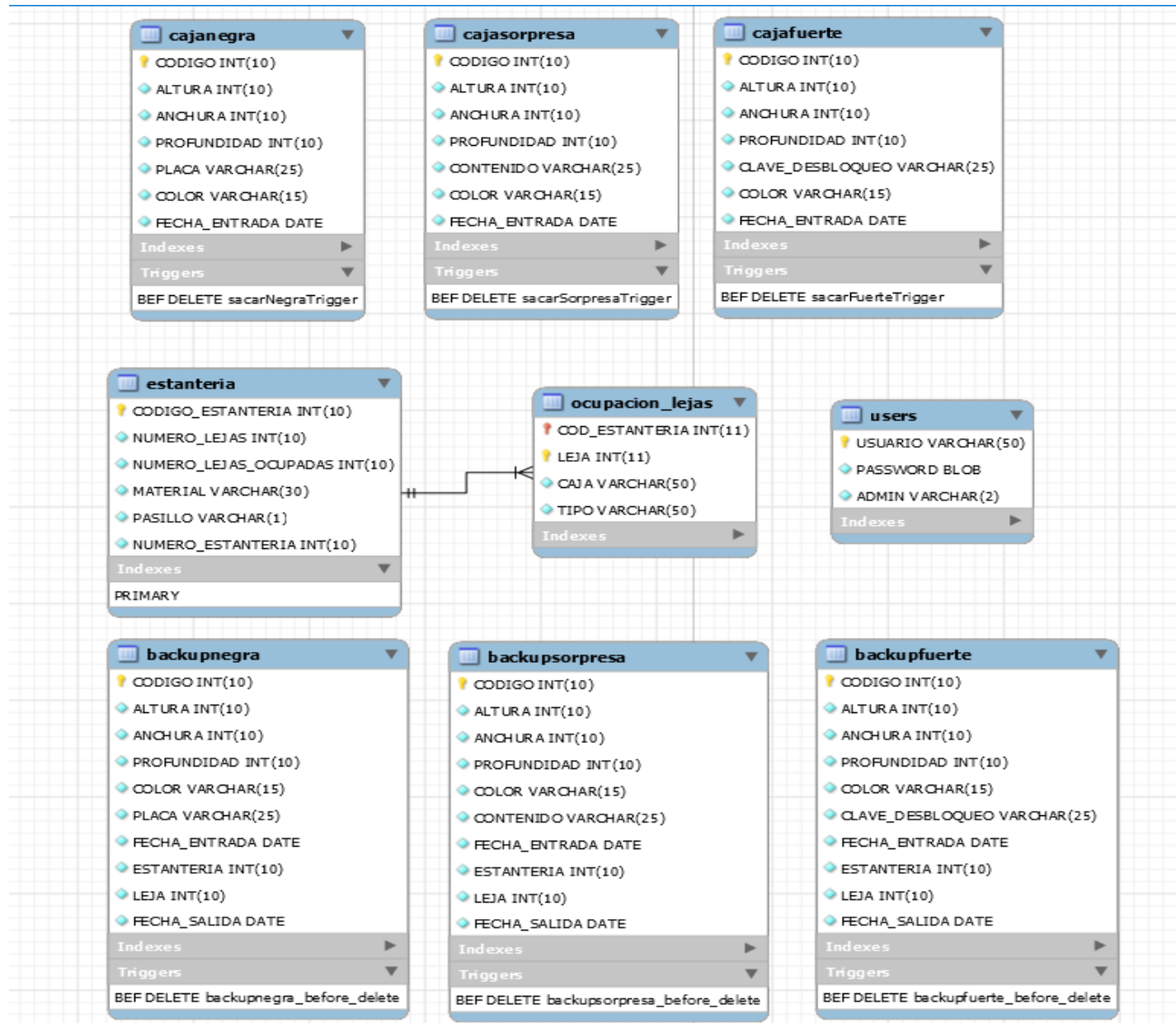
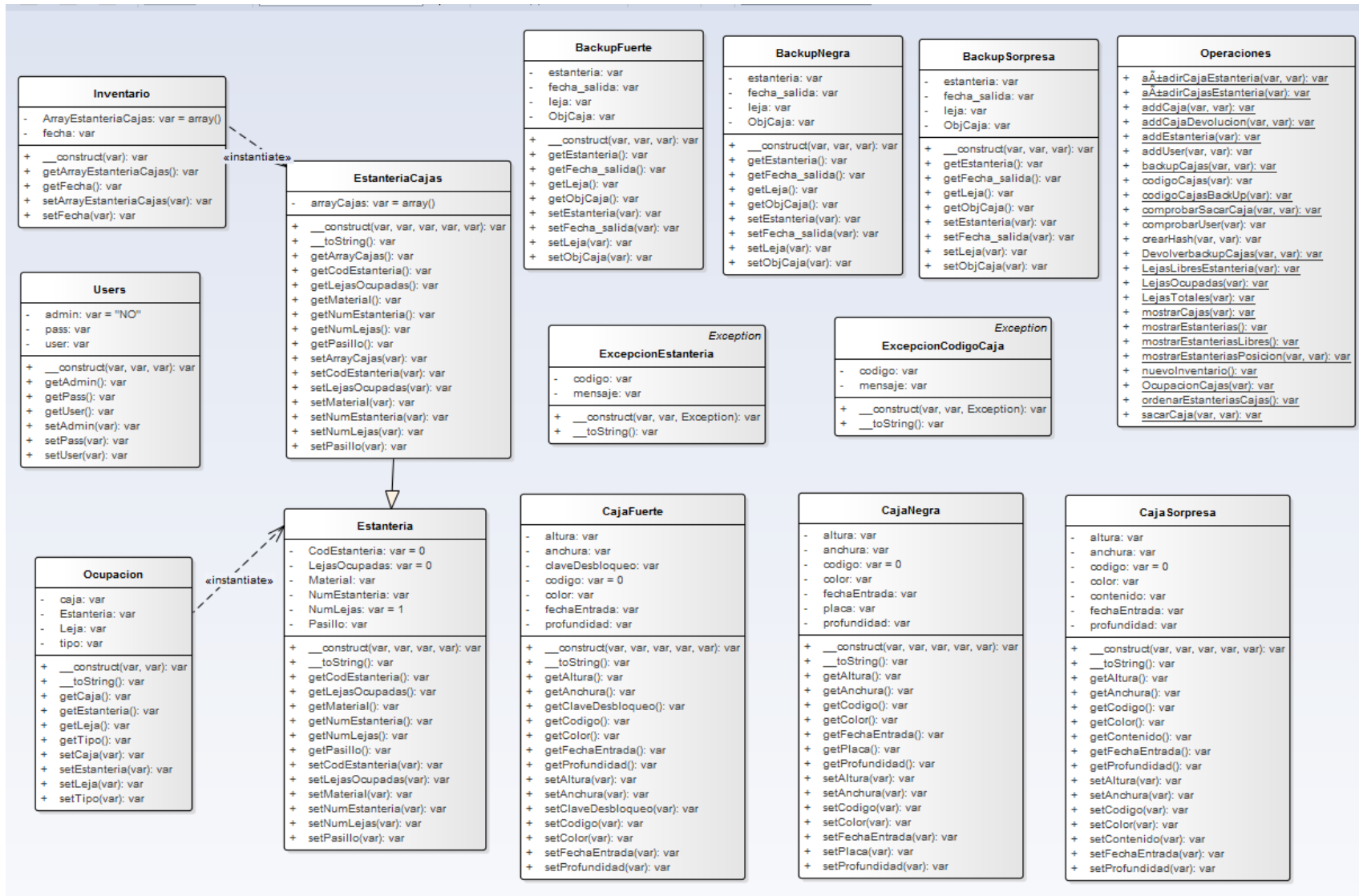


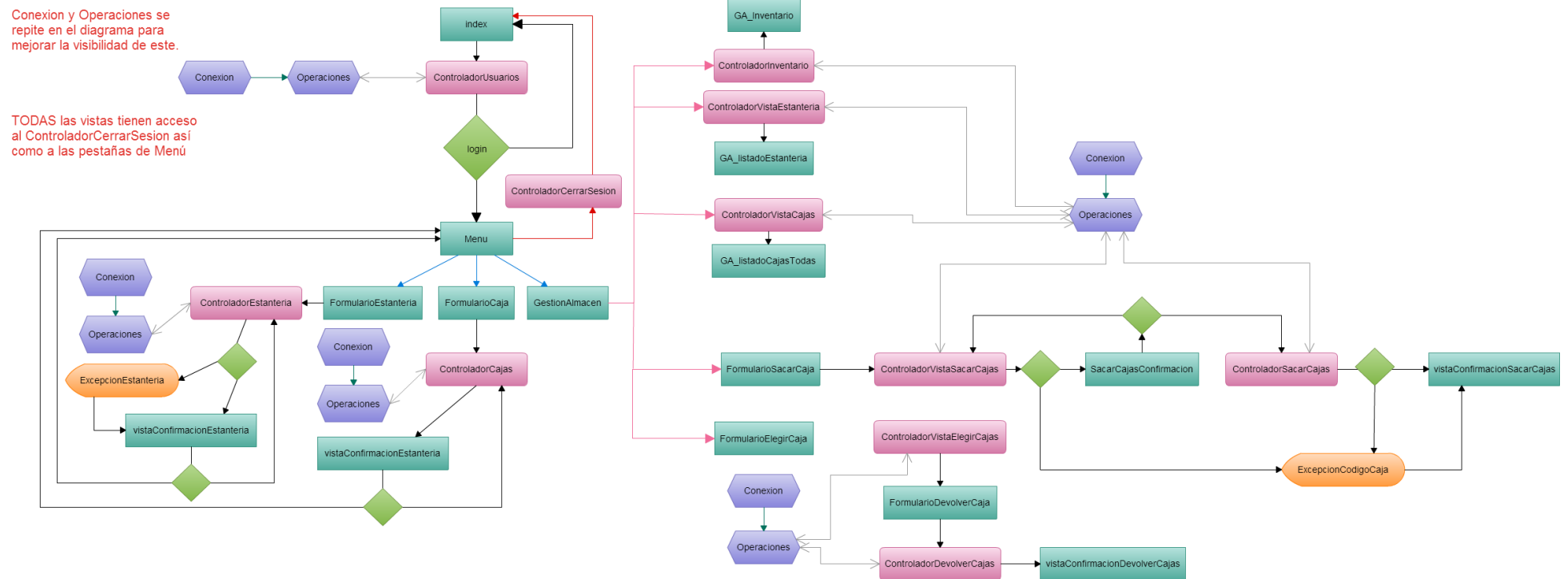
Diagrama de clases



Variables de sesión

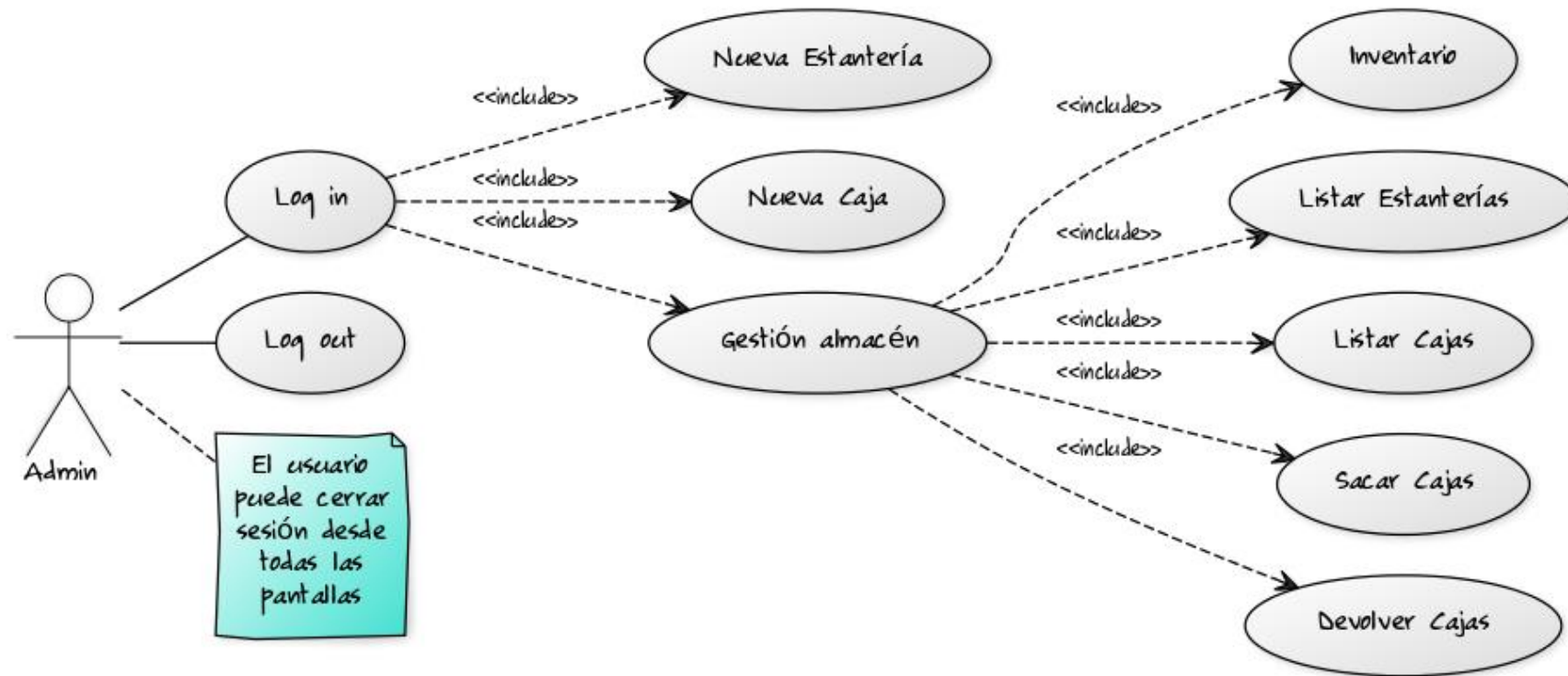
Nombre	Contenido	Creada en...	Usada en...
confirmacionCajas	TRUE/FLASE	ControladorCajas	vistaConfirmacionCajas
confirmacionDevolverCajas	TRUE/FALSE	ControladorDevolverCajas	VistaConfirmacionDevolverCajas
confirmacionEstanterias	TRUE/FALSE	ControladorEstanteria	vistaConfirmacionEstanteria
confirmacionSacarCajas	TRUE/FALSE	ControladorSacarCajas, ControladorVistaSacarCajas	vistaConfirmacionSacarCajas
inventarioCajas	Objeto Inventario	ControladorInventario, ControladorVistaCajas	GA_Inventario, GA_InventarioCajasTodas
mostrarDatosCaja	Objeto Ocupación	ControladorvistaSacarCajas	SacarCajasConfirmacion,, ControladorSacarCajas
mostrarDatosCajaDevol	Objeto Bakcup	ControladorvistaElegirCajas	ControladorDevolverCajas, FormularioDevolverCaja,
mostrarDatosEstanteria	Array Estanterías	ControladorVistaFormCajas, ControladorVistaelegirCajas	FormularioCaja, FormularioDevolverCaja,
mostrarEstanteria	Array Estanterías	ControladorVistaEstanteria	GA_listadoEstanteria
Usuario	Objeto User	ControladorUsuarios	TODAS LAS VISTAS ControladorCerrarSesion,
Validación	TRUE/FALSE	ControladorUsuarios	index

Diagrama de flujo



***** Este es el flujo de ejecución de la aplicación. Todos los controladores tienen acceso a operaciones. En este caso Operaciones y Conexión están repetidas varias veces para mejorar la visualización del diagrama. Ocurre lo mismo con el acceso al controladorCerrarSesion, todas las vistas tienen acceso a él.

Casos de usos

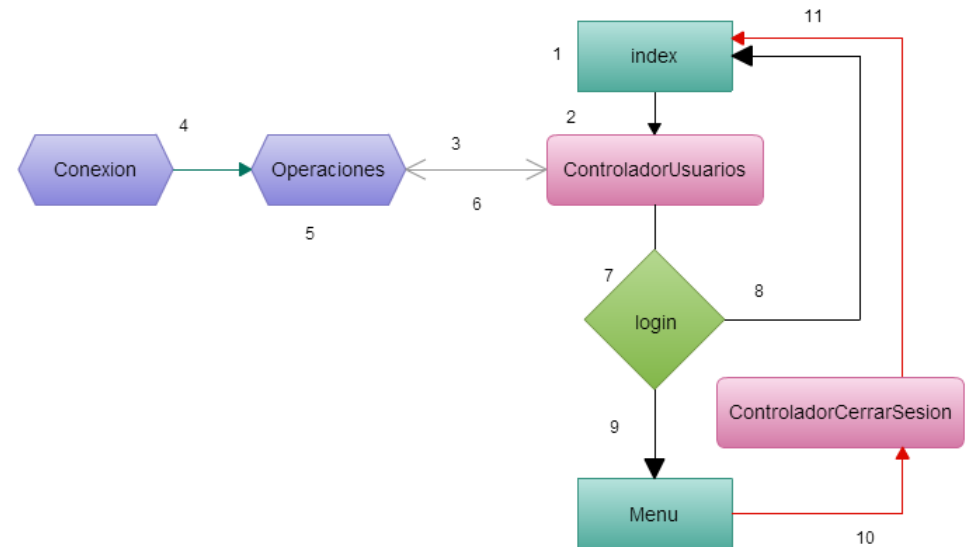


Log In

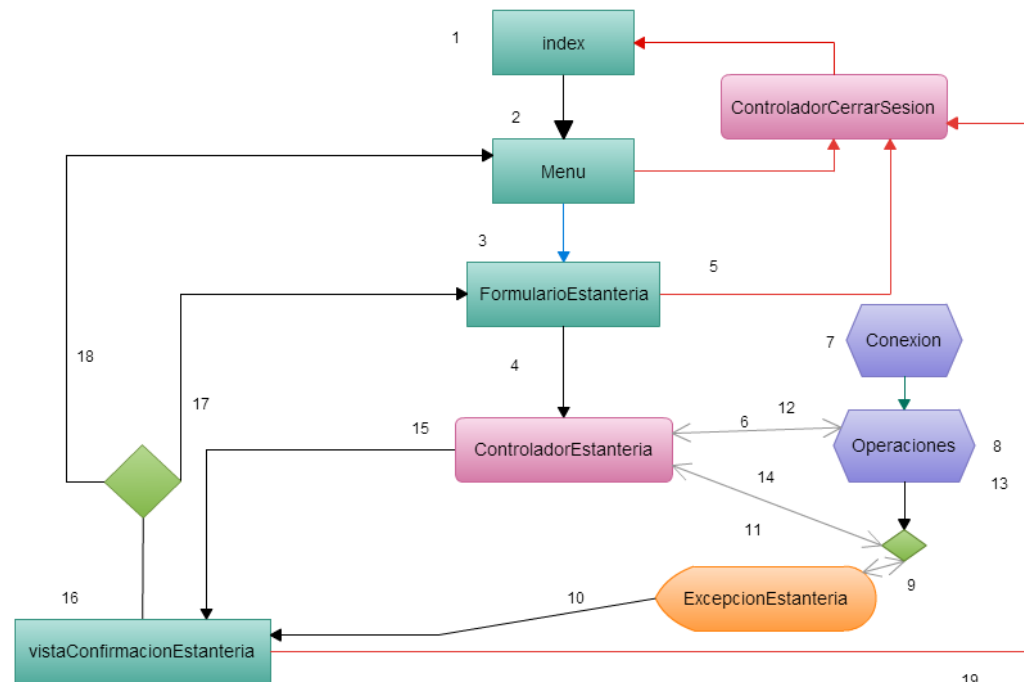
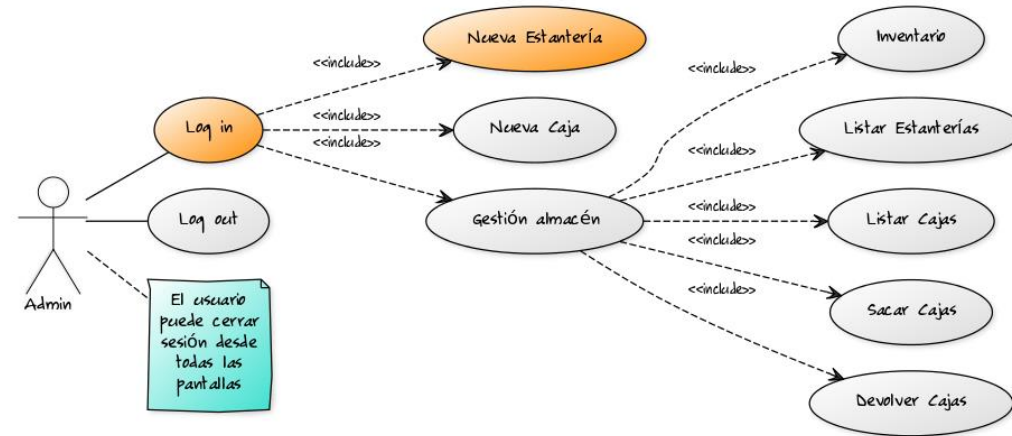
USUARIO: Admin

Password: 123456

1. Entramos al **log-in** de la aplicación, introducimos el **usuario** y la **contraseña** y pulsamos **enviar**.
2. Los datos los recoge el **controladorUsuarios**,
3. Llama a **Operaciones** pasándole a **comprobarUser** el **objeto user**.
4. En **operaciones** se llama a **Conexión** para conectar con la base de datos.
5. Si la conexión se ha establecido, **comprobarUser** hace una consulta a la base de datos buscando el usuario, si lo encuentra comprueba que la password que tiene es la misma que la introducida en el log in.
6. **Operaciones** devuelve **false**, si el usuario no ha sido encontrado, **true**, si lo ha encontrado y además coincide la contraseña o **false** si el usuario existe pero no es correcta la contraseña.
7. En el controlador se recoge el resultado de la operación.
8. Si ha sido **false**, crea una **sesión (validación = false)** y vuelve a la vista del **log-in** mostrando un mensaje de error.
9. Si ha sido **true**, crea dos **sesiones (usuario = objUser, validación = true)** y accede a la vista de **Menú**.
10. Desde el **menú** podemos **cerrar la sesión**. Al pulsar el botón nos lleva al **ControladorCerrarSesion**.
11. El controlador **destruye la sesión usuario** y nos lleva a la vista del **log-in (index)**.



Añadir Estantería.



1. Logeamos y nos lleva al **menú**.
2. En el menú podemos elegir varias opciones, pero en este caso seguiremos el flujo de como añadir una estantería.
3. Una vez elegida la opción de estantería, rellenamos el **formulario** introduciendo los datos de la nueva estantería.
 - a. **Material**: material del que está construida la estantería.
 - b. **Nº lejas**: cantidad de lejas que tendrá la estantería.
 - c. **Pasillo**: letra del pasillo en el que estará ubicada (A, B, C,...).
 - d. **Estantería**: número que le corresponde en el pasillo. (1, 2, 3,...).
4. Al pulsar el botón **Añadir** nos llevara al **controladorEstanteria** que recogerá los datos del formulario y compondrá el **objeto Estantería**.
5. Desde la vista del formulario para añadir estantería **podemos cerrar sesión**.
6. Desde el **controlador**, llamamos a **operaciones** pasándole a **mostrarEstanteriasPosicion** el **pasillo** y el **número de estantería** para así comprobar si la posición está libre.
7. En **operaciones** se llama a **conexión** para establecer la conexión con la base de datos.
8. Si la conexión se ha establecido correctamente **mostrarEstanteriasPosicion** buscara si hay una estantería en la posición indicada.
9. Si la hay lanza la **excepción ExcepcionEstanteria**, porque ya está ocupada esa posición.
10. Nos lleva a la vista para mostrarnos el **mensaje de error**.
11. Si no la hay devuelve **false** y el controlador sigue ejecutando las demás acciones.
12. **controladorEstanteria** llama a **operaciones** pasándole a **addEstanteria** el **objeto estantería**. Crea una **sesión** llamada **confirmacionEstanterias**.
13. **Operaciones** ejecuta la función insertando en la base de datos una nueva estantería.
14. Devuelve al controlador **True/False** dependiendo de si se ha realizado o no.
15. Del controlador pasamos a la **vistaConfirmacionEstanteria**.
16. **vistaConfirmacionEstanteria** nos muestra el **mensaje correspondiente**. Desde aquí podemos tomar tres caminos:
17. Podemos ir de vuelta al **formulario** para añadir otra estantería
18. Podemos volver al **menú** para realizar otra acción
19. Podemos **cerrar la sesión** y volver al log-in.

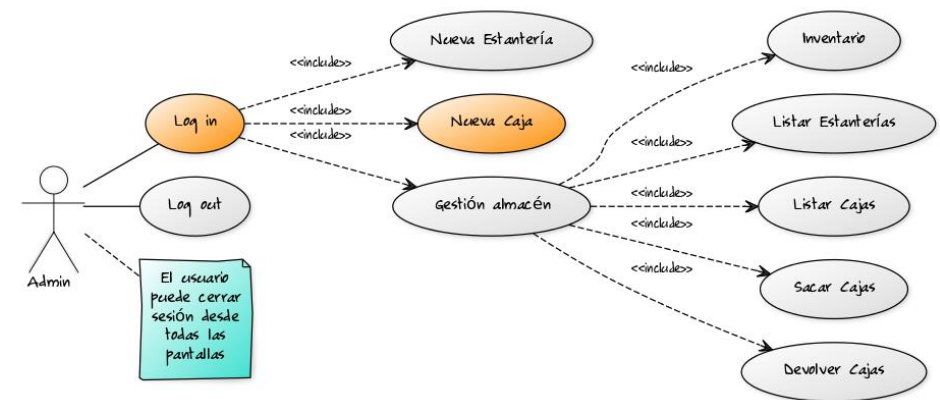
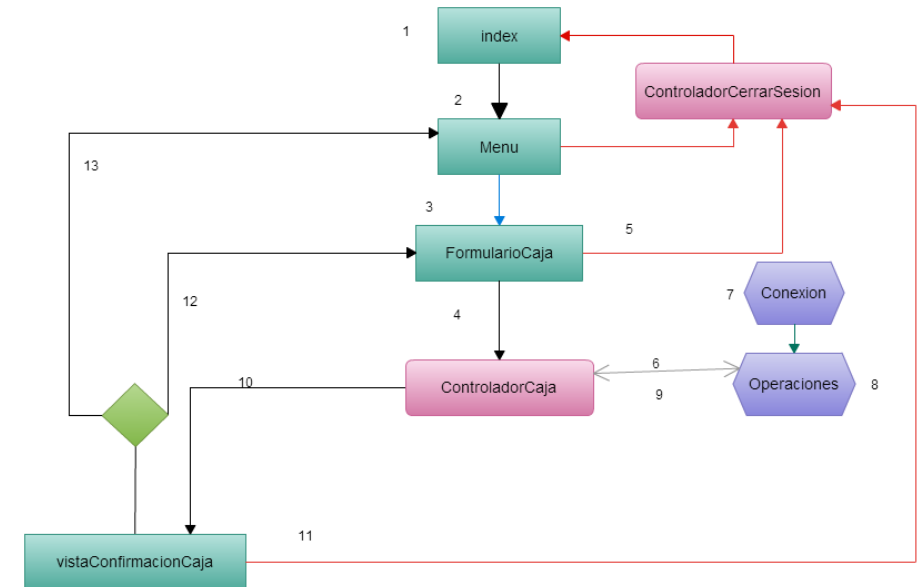
Añadir Caja

10

2. Accedemos al **menú**, desde el que podremos tomar varios caminos. Uno de ellos es **cerrar sesión**.
3. Accedemos el **formulario** para añadir una nueva caja, donde introduciremos los datos en los campos correspondientes del formulario, que son:
 - a. **Tipo**: se nos abre un desplegable con los 3 tipos de caja que podemos añadir.
 - b. **Estantería**: es otro desplegable con la lista de estanterías que tienen alguna leja libre. Nos muestra el código y la posición de cada una.
 - c. **Lejas libres**: de nuevo otro desplegable con la lista de lejas libres que tiene la estantería elegida.

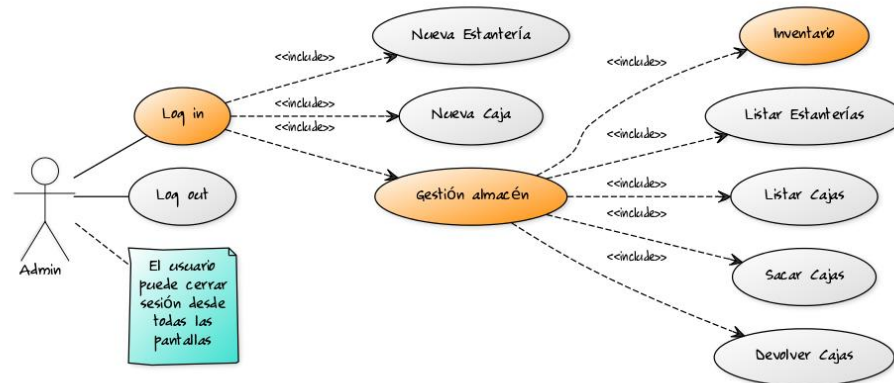
AjaxLejasLibres.php -> Recoge del formulario el código de la estantería. Llama a **operaciones – LejasLibresEstanteria** pasándole el código de la estantería, para sacar el array de lejas libres de esa estantería. Lo guarda en una variable que será descompuesta y mostrada en la vista. El JavaScript se encarga de actualizar la página cada vez que elegimos una estantería nueva.

- d. **Alto, ancho y profundidad**: medidas que tiene la caja
- e. **Color**: elegimos el color de la caja.
4. Al pulsar añadir pasamos al **controladorCaja** que recoge los datos del formulario para crear un **objeto Caja** (dependiendo del tipo crea una CajaSorpresa, CajaNegra o CajaFuerte) y un **objeto ocupación**.
5. Desde el **formularioCaja** también podemos **cerrar sesión**.
6. Llama a **operaciones – addCaja** pasándole los dos objetos.
7. Operaciones conecta con la base de datos.
8. Si la conexión es correcta ejecuta la función que descompondrá los dos objetos para usarlos en la consulta a la base de datos. Dependiendo del tipo realiza una consulta distinta y **actualiza las tablas** relacionadas con la caja (ocupación_lejas y estantería). Si todas las consultas se han realizado bien devuelve **true**, si alguna ha fallado realiza un **rollback** y devuelve **false**.
9. Se recoge el resultado de la operación en una **sesión llamada confirmacionCajas**.
10. Nos lleva a **VistaconfirmacionCajas** y nos muestra el mensaje correspondiente al resultado de la operación.
11. Desde esta vista podemos **cerrar sesión**.
12. Podemos volver al formulario para **añadir otra caja**



13. Podemos ir al **Menú** para realizar otra operación.

Inventario



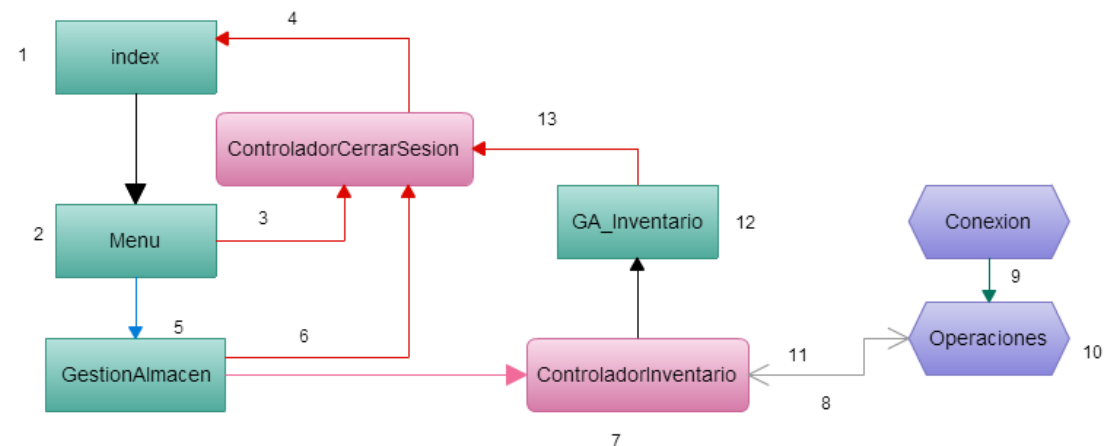
1. Una vez logeados podemos:
2. Acceder al **menú**.
3. Desde el menú podremos **cerrar sesión**.
4. El controlador de cerrar sesión nos llevara a la página de **logeo**.
5. Desde el menú podemos ir a **GestionAlmacen**
6. Desde **GestionAlmacen** también podremos **cerrar sesión**.
7. Para mostrar el **inventario** pasamos por el **controladorInventario**.
8. Que llamara al método de **operaciones nuevoinventario**.
9. Operaciones se conecta con la base de datos.

10. Si la conexión se ha realizado el método **nuevolnventario** llamara al método **ordenarEstanteriasCajas**, que devolverá un **array con las estanterías ordenadas** por pasillo y número. A continuación llamara a **añadirCajasEstanteria** pasándole el **objeto estantería**. Este método a su vez llama a **OcupacionCajas** que nos devuelve un **array con las estanterías** que aparecen en la tabla Ocupacion_Lejas. A continuación llama al método **AñadirCajaEstanteria** que ir añadiendo las cajas a cada estantería ordenadas por leja, crea el **objeto estanteríaCajas** y lo devuelve.

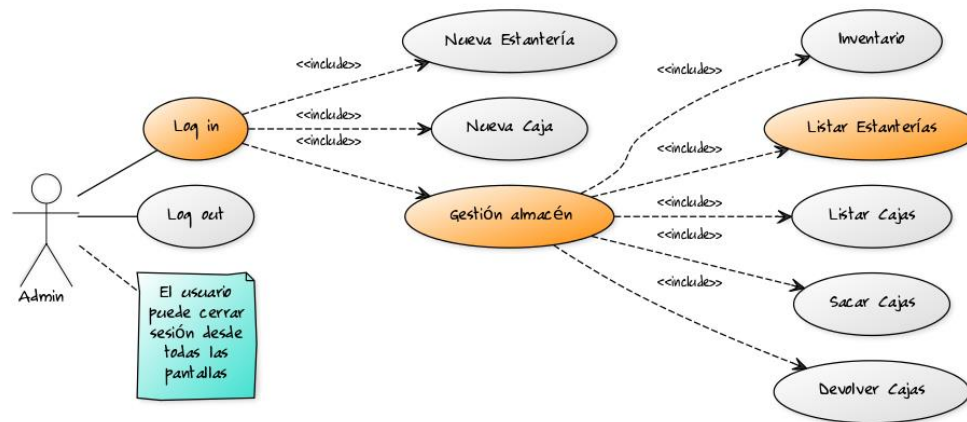
11. El controlador recoge el array de inventario devuelto en la operación y lo guarda en una variable de **sesión llamada inventarioCajas**. Nos lleva a la **vista del inventario**.

12. En la vista **GA_Inventario** nos muestra una lista con todas las estanterías ocupadas y sus cajas, ordenadas por Pasillo y número.

13. Desde la vista del inventario podemos cerrar sesión.

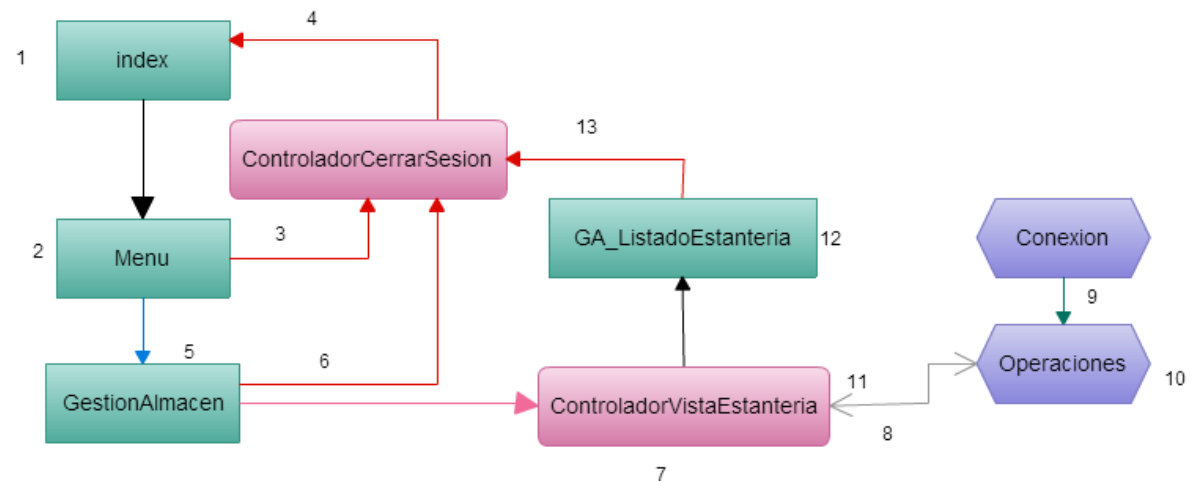


Listar Estanterías

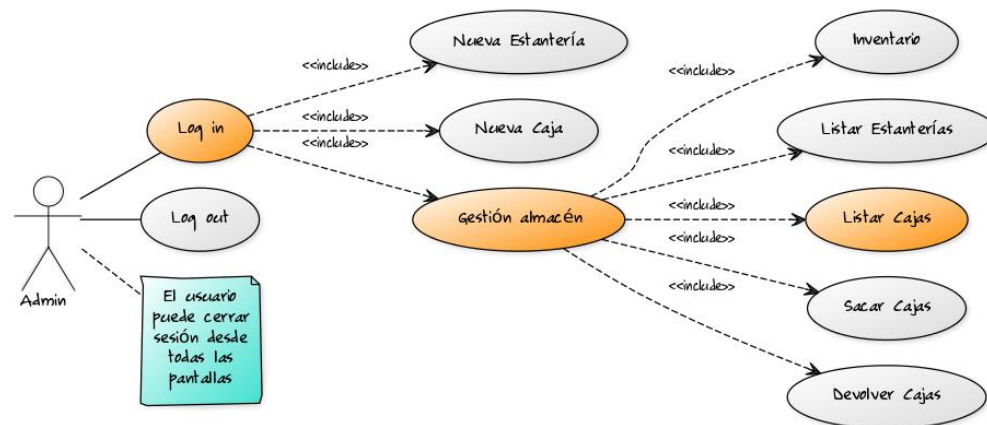


1. Una vez logeados podemos:
2. Acceder al menú.
3. Desde el menú podremos cerrar sesión.
4. El controlador de **cerrarSesión** nos llevara a la página de logeo.
5. Desde el menú podemos ir a **GestionAlmacen**
6. Desde **GestionAlmacen** también podremos cerrar sesión.
7. Para mostrar el inventario pasamos por el **controladorVistaEstanteria**.
8. Que llamara al método de **operaciones mostrarEstanterias**.
9. Operaciones se conecta con la base de datos.

10. Si la conexión se ha realizado el método **mostrarEstanterias** realiza una consulta a la base de datos sacando todas las estanterías que hay y guardándolas en un array.
11. El controlador recoge el array de estanterías devuelto en la operación y lo guarda en una variable de **sesión** llamada **mostrarEstanteria**. Nos lleva a la vista del **listadoEstanterias**.
12. En la vista **GA_ListadoEstanterias** nos muestra una lista con todas las estanterías y sus datos, ordenadas por código.
13. Desde la vista del listado podemos **cerrar sesión**.

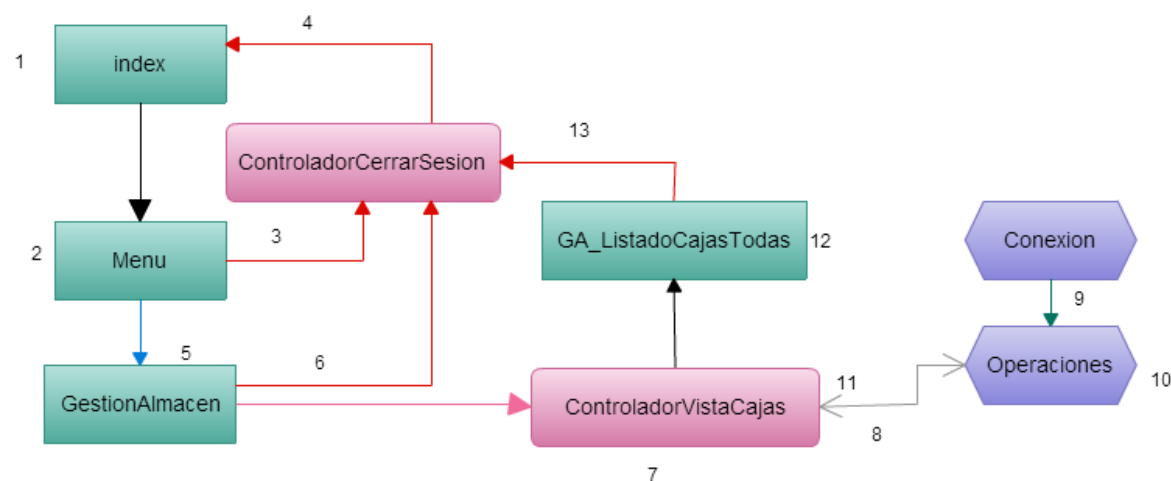


Listar Cajas

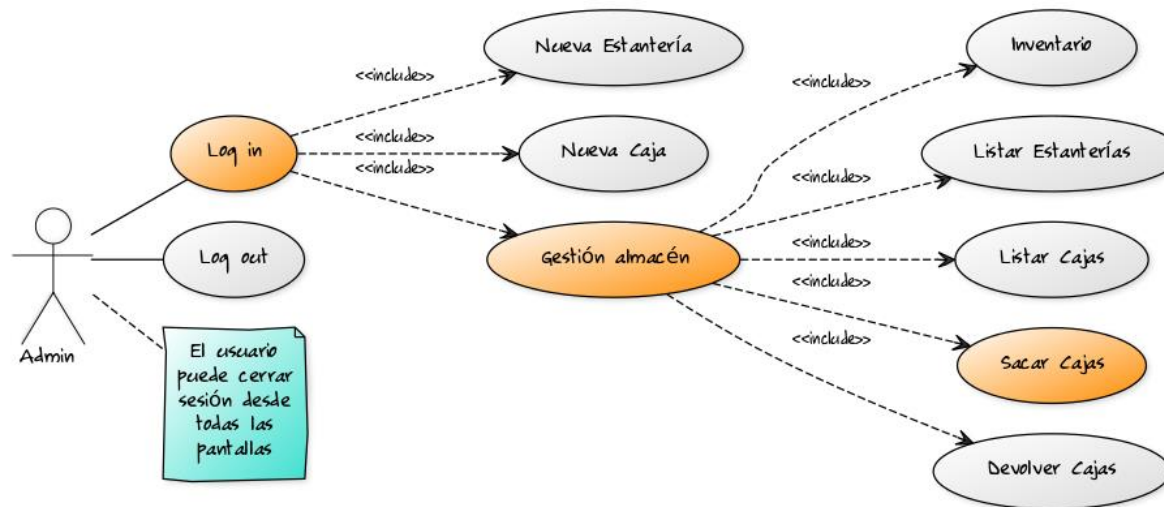


1. Una vez logeados podemos:
2. Acceder al menú.
3. Desde el menú podremos cerrar sesión.
4. El controlador de cerrar sesión nos llevara a la página de logeo.
5. Desde el menú podemos ir a GestionAlmacen
6. Desde GestionAlmacen también podremos cerrar sesión.
7. Para mostrar las cajas pasamos por el **controladorVistaCajas**.
8. Que llamara al método de **operaciones nuevoinventario**, reutilizándolo.
9. Operaciones se conecta con la base de datos.

10. Si la conexión se ha realizado se ejecuta el método **nuevolnventario** explicado anteriormente.
11. El controlador recoge el array de inventario devuelto en la operación y lo guarda en una variable de **sesión** llamada **mostrarEstanteria**. Nos lleva a la vista del **listadoCajasTodas**.
12. En la vista **GA_ListadoCajasTodas** nos muestra una lista con las cajas existentes, **pudiendo elegir si ver todas las cajas o por tipo**.
13. Desde la vista del listado podemos **cerrar sesión**.

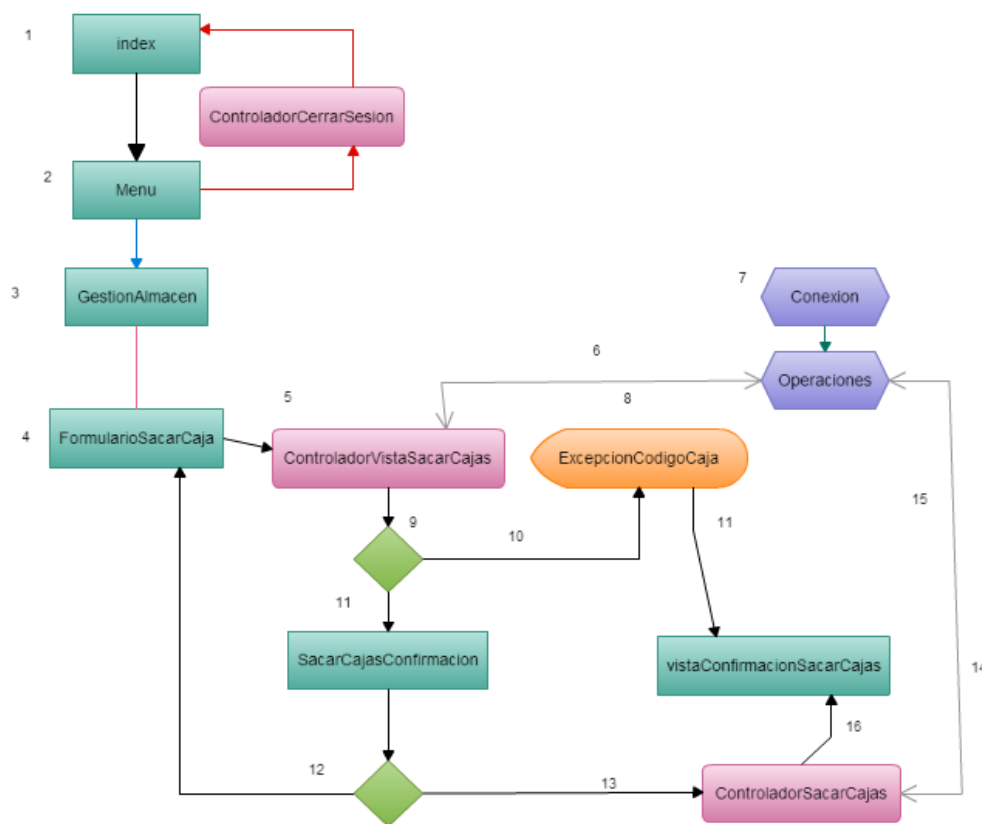


Sacar Cajas

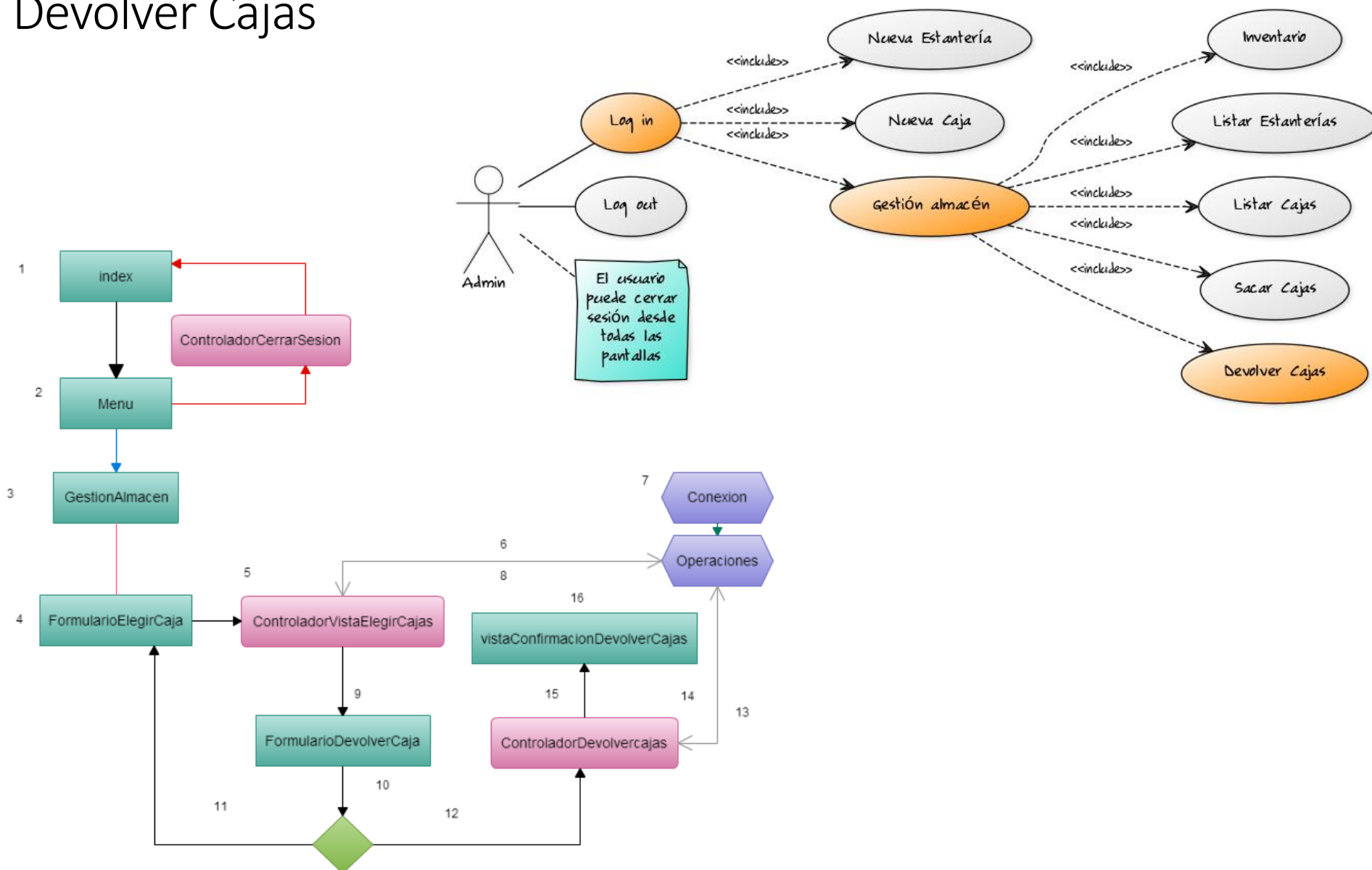


1. Una vez logeados podemos:
2. Acceder al menú. Desde el menú podremos cerrar sesión. El controlador de cerrar sesión nos llevara a la página de logeo.
3. Desde el menú podemos ir a GestionAlmacen. Desde GestionAlmacen también podremos cerrar sesión.
4. Pasamos al formulario de **sacarCaja**, donde elegiremos el **tipo** y el **código** de la caja que queremos sacar.
5. Al pulsar aceptar vamos a **ControladorVistaSacarCajas** que recogerá del formulario los datos
6. El controlador llamará al método de **operaciones backupCajas**, pasándole el **tipo** y el **código**.
7. Operaciones se conecta con la base de datos. Si la conexión se ha realizado se ejecuta el **método backupCajas** que realiza una consulta a la base de datos buscando una caja con el código y el tipo introducido en el formulario, en la tabla ocupación_lejas.
8. Si la consulta se ha realizado con éxito y encuentra datos **devuelve el objeto ocupación**. Si hay algún error o no existe una caja con ese código y tipo lanza una **excepción** que será recogida en el controlador
9. El controlador recoge el resultado de la operación. Si no hay errores, lo guarda en una **sesión llamada mostrarDatosCaja** y nos dirige a la **vistaSacarCajasConfirmacion**.
10. Si la **excepción ha sido lanzada**, el controlador lo recoge y guarda en una sesión el mensaje de dicha excepción y nos lleva a la **vistaConfirmacionSacarCajas** mostrando el mensaje que hemos pasado en la sesión.

11. En la vista **sacarCajasConfirmacion** nos muestra la información de la caja que vamos a sacar para decidir si sacarla o no.
12. Si pulsamos **cancelar** nos lleva de nuevo al **formularioSacarCaja**.
13. Si pulsamos **Si** nos lleva al **controladorSacarCaja**.
14. El controlador recoge la sesión que se creó en el **ControladorVistaSacarCajas** y descompone el objeto para llamar a operaciones-sacarCaja pasándole el código y el tipo.
 - a. Operaciones conecta con la base de datos.
 - b. **sacarCaja** realiza una consulta para borrar la fila de la tabla del tipo que se le ha pasado (CajaSorpresa, CajaNegra, CajaFuerte) con el código correspondiente. **Esta acción ejecuta los triggers de la base datos, para actualizar las tablas relacionadas.**
15. El **controlador** recoge el resultado de la operación y lo guarda en una variable de sesión llamada **confirmacionSacarCajas**, Si se realiza la operación recoge **True**, si no recoge **False**, y nos envía a **vistaConfirmacionSacarCajas**.
16. **vistaConfirmacionSacarCajas** nos muestra el mensaje correspondiente dependiendo de si se ha realizado la operación o no. Da dos opciones:
 - a. Volver al **menú**.
 - b. Volver al **formulario** sacarCaja.



Devolver Cajas



1. Una vez logeados:
2. Accedemos al **menú**, desde el que podremos tomar varios caminos. Uno de ellos es cerrar sesión.
3. Accedemos a la **GestionAlmacen**.
4. Accedemos el **formularioElegirCaja** para elegir la caja que vamos a devolver, donde seleccionaremos el **tipo** y el **código** de caja. Desde aquí podemos **cerrar sesión**.
5. Al pulsar añadir pasamos al **controladorVistaElegirCaja** que llama al método **operaciones-mostrarEstanteriasLibres**, usado en **añadirCaja**. Recoge los datos del formulario para pasárselo al método **DevolverbackupCajas**.
6. Llamamos a operaciones desde los dos métodos mencionados antes.
7. **Operaciones** conecta con la **base de datos**.
8. **Operaciones** ejecuta los métodos.
 - a. **mostrarEstanteriasLibres** que devuelve un **array de objetos Estantería**.
 - b. **DevolverbackupCajas** que recoge los datos de las tablas backup **devolviendo el objeto**.
9. El controlador recoge los datos de **mostrarEstanteriasLibres** y lo guarda en la **sesión mostrarDatosEstanteria** y el resultado de **DevolverbackupCajas** lo guarda en la **sesión mostrarDatosCajaDevol**. Nos lleva a **FormularioDevolverCaja**.
10. En este formulario nos aparecen todos los **datos** de la caja que vamos a devolver **rellenados** en el **formulario añadirCaja**, excepto los **select de Estantería y leja**.

AjaxCajaElegirDevolucion.php -> -> Recoge del formulario el **código de la estantería**. Llama a **operaciones – LejasLibresEstanteria** pasándole el código de la estantería, para sacar el **array de lejas libres de esa estantería**. Lo guarda en una **variable que será descompuesta y mostrada en la vista**. El JavaScript se encarga de actualizar la página cada vez que elegimos una estantería nueva.

11. Podemos **cancelar** la operación y esto nos llevaría al **formularioElegirCaja**.
12. Si pulsamos **Devolver** nos lleva al **controladorDevolverCajas** que recoge todos los datos del formulario crea un **objeto Caja** (sorpresa, negra o fuerte) y un **objeto ocupación**.
13. Le pasa a **addCajaDevolucion** el **objetoCaja** y el **objetoOcupacion**.
14. Si la conexión es correcta ejecuta la función que descompondrá los dos objetos para usarlos en la consulta a la base de datos. Dependiendo del tipo realiza una consulta distinta y **crea un trigger que actualiza la estantería, añade una fila a ocupación e inserta otra en la tabla de la caja correspondiente**. Este trtigger **se ejecutara cuando en esta misma función eliminemos la fila que contenga la caja con el código correspondiente de la tabla backup [tipo]** Si todas las consultas se han realizado bien devuelve **true**, si alguna ha fallado realiza un **rollback y devuelve false**.
15. Se recoge el resultado en la **sesión confirmacionDevolverCajas** y nos envía a la vista **vistaConfirmacionDevolverCajas**.
16. La vista muestra el **mensaje correspondiente** al resultado de la operación. Después podremos volver a **Menú** o al **formularioElegirCaja**.