A thick dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

29-11-2018

Documentación

ArtisanBeer - Proyecto final DAW

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Miguel Ángel García y Julián Rodenas
2º DAW

Indice

1.	Información acerca del proyecto	3
1.1.	Software utilizado	3
1.2.	Descripción del proyecto.....	3
1.3.	Modelo de organización	4
2.	Diagrama de tablas	5
3.	API de la aplicación	6
4.	Casos de uso.....	6
4.1.	Abrir aplicación.....	6
4.2.	Registrar Usuario.....	6
4.3.	Registrar vendedor	7
4.4.	Mostrar catálogo.....	8
4.5.	Mostrar más información sobre un artículo	8
4.6.	Añadir artículo a carrito	9
4.7.	Realizar pedido.....	10
4.8.	Casos de uso del panel de control	11
4.8.1.	Alta artículo.....	11
4.8.2.	Listado artículos	12
4.8.3.	Modificación y eliminación de un artículo.....	12
4.8.4.	Alta categoría	14
4.8.5.	Alta subcategoría.....	15
4.8.6.	Listado categorías y subcategorías	16
4.8.7.	Modificación y eliminación de categorías.....	16
4.8.8.	Modificación y eliminación de subcategorías	18
4.8.9.	Listado de pedidos activos y completados.	20
5.	Variables de sesión.....	21

DOCUMENTACION PROYECTO FINAL

Miguel Ángel García Martínez

Julián Díaz Rodenas

Fecha 2018

Ciclo de Grado Superior y Grupo Informática

IES LEONARDO DA VINCI

DESARROLLO DE APLICACIONES WEB

1. Información acerca del proyecto

1.1. Software utilizado

Para el desarrollo de este proyecto hemos utilizado:

- Lenguaje principal Java enfocado a desarrollo web
- Otros lenguajes utilizados: Bootstrap (HTML5 y CSS3), JavaScript, AJAX
- Entorno de desarrollo NetBeans en su versión 8.2
- Base de datos: MYSQL
- Gestor de base de datos: HeidiSQL
- Servidor de Aplicaciones: Glassfish Server 4.1.1
- Hibernate 4.1
- GitHub
- GitHub Desktop

1.2. Descripción del proyecto

Aplicación web enfocada a la compraventa de cerveza artesanal. La idea surge de las dificultades que existen a la hora de distribuir este tipo de producto, ya que suele estar elaborado por pequeños productores con dificultades de publicitarse.

La aplicación permite ver un amplio catálogo de productos subidos por los propios usuarios vendedores. La aplicación permite registrarse tanto como usuario consumidor o como usuario vendedor, de forma que un usuario vendedor pueda acceder a sus pedidos y realizarlos, y un usuario vendedor puede acceder a un panel de control para poder poner a la venta productos y categorizarlos a su gusto.

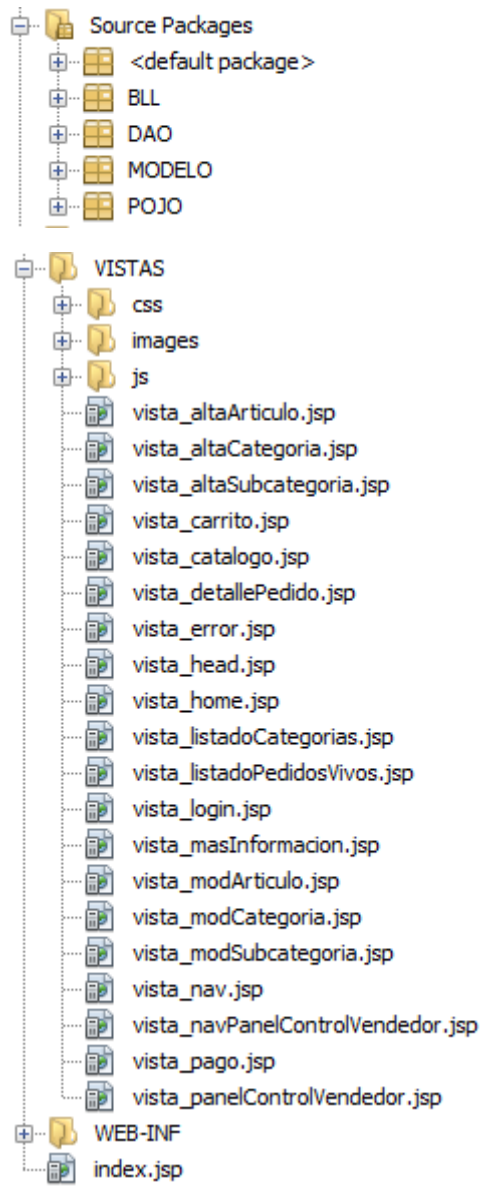
La aplicación está implementada con Java como lenguaje de programación principal utilizando el framework Hibernate para gestionar el acceso a la base de datos.

Las vistas han sido realizadas con HTML5 y hemos utilizado las librerías de Bootstrap 4 junto con CSS3 para realizar el diseño de la web.

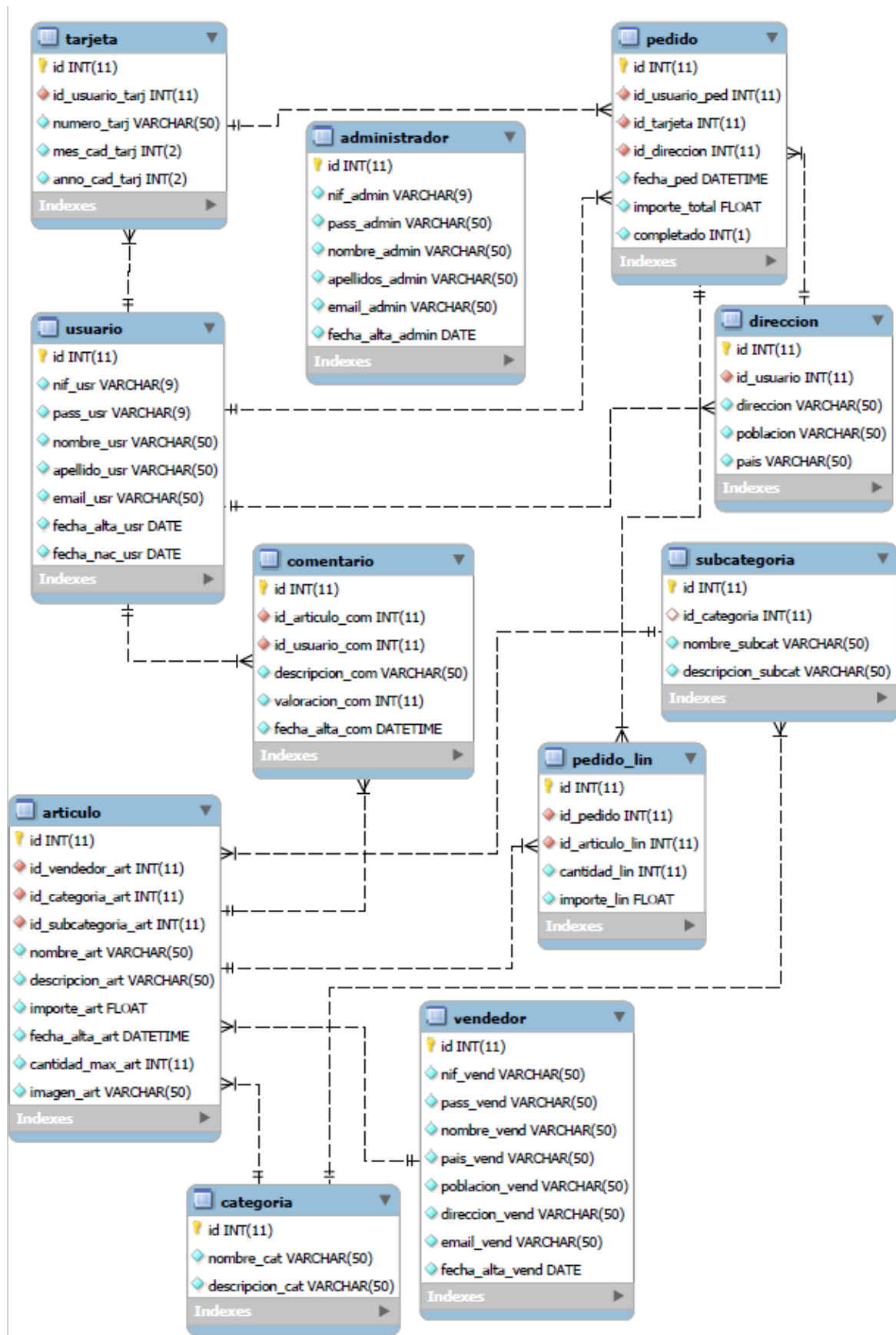
El proyecto al realizarlo de forma conjunta ha tenido que ser actualizado utilizando un repositorio común, en nuestro caso utilizamos Git-Hub y usamos su aplicación de escritorio Git-Hub Desktop para sincronizar nuestro trabajo.

1.3. Modelo de organización

Se ha utilizado un modelo de organización modelo-vista-controlador (MVC):



2. Diagrama de tablas



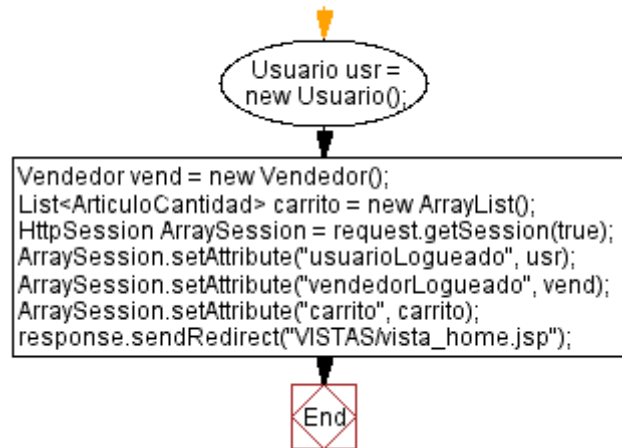
3. API de la aplicación

Se adjunta junto a la documentación del proyecto.

4. Casos de uso

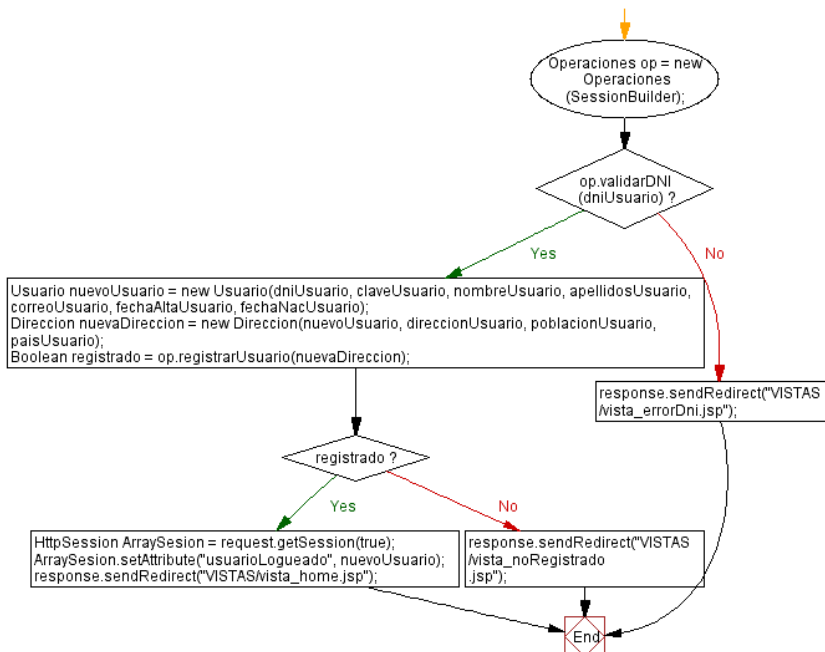
4.1. Abrir aplicación.

La aplicación al abrir crea en la sesión un carrito vacío (List de ArtículoCantidad), un usuario (que servirá como usuario logueado) y un vendedor (que servirá como vendedor logueado).



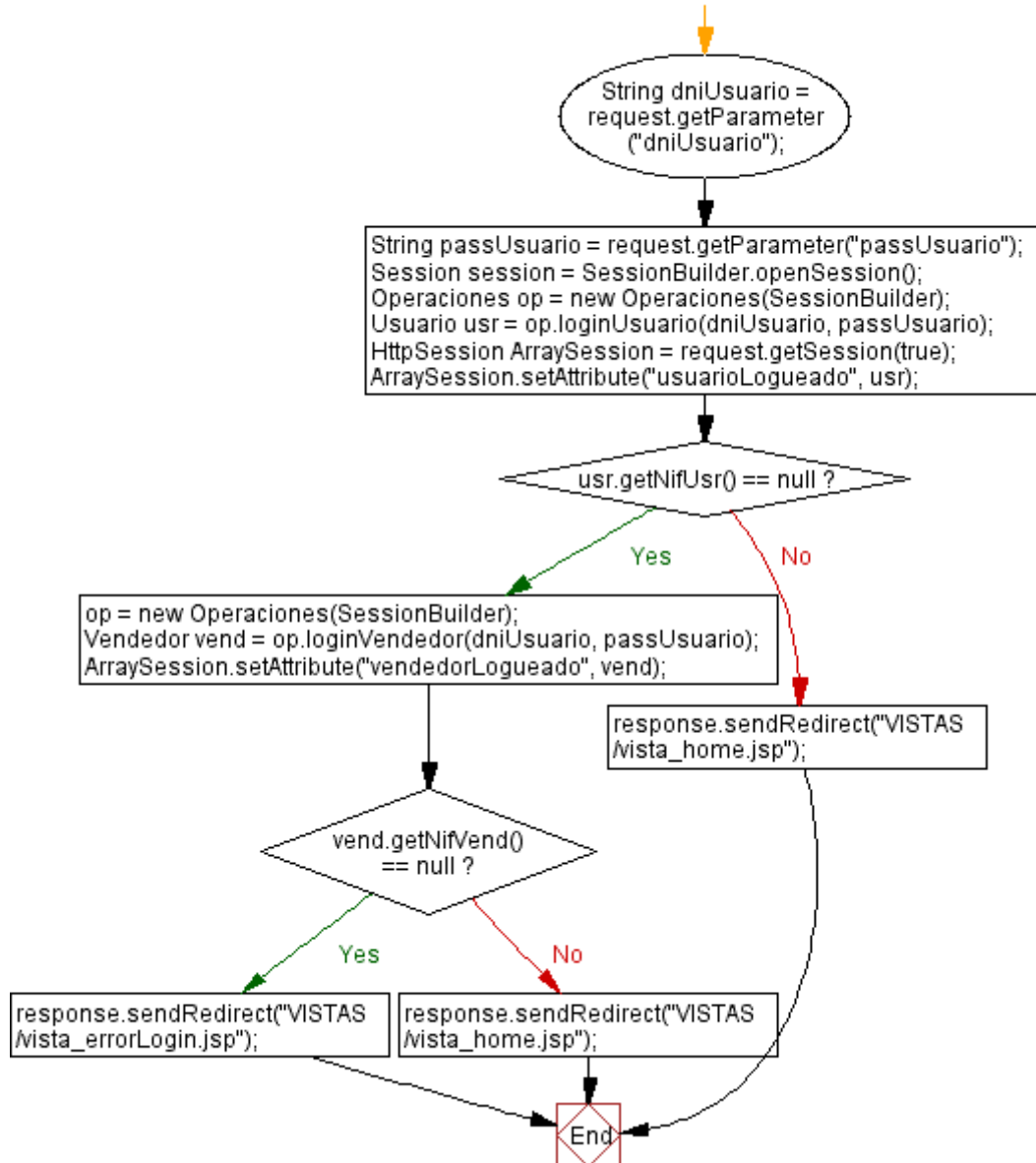
4.2. Registrar Usuario

Para registrar un usuario validamos que el DNI sea correcto y después comprueba que no exista ya un usuario con ese DNI. El registro de vendedor es similar, por lo que lo incluyo en el mismo apartado.



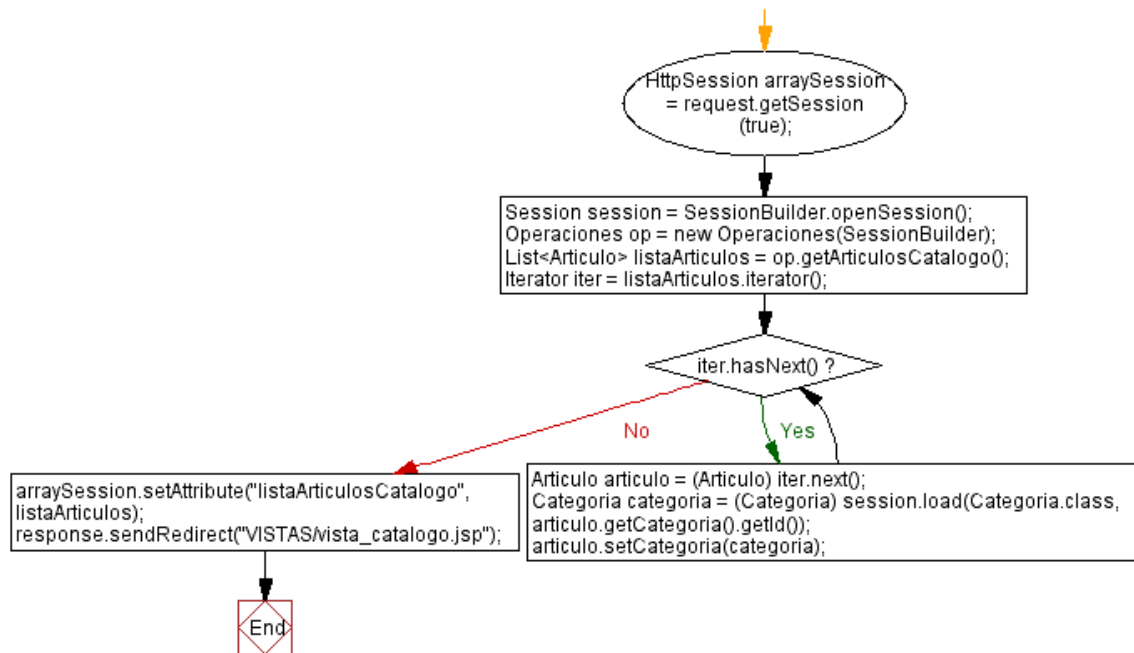
4.3. Registrar vendedor

Se comprueba si existe un usuario, y si es así inicia sesión guardándolo en la sesión, si no es así comprueba si existe ese vendedor, si existe lo guarda en la sesión, si no muestra un aviso de error.



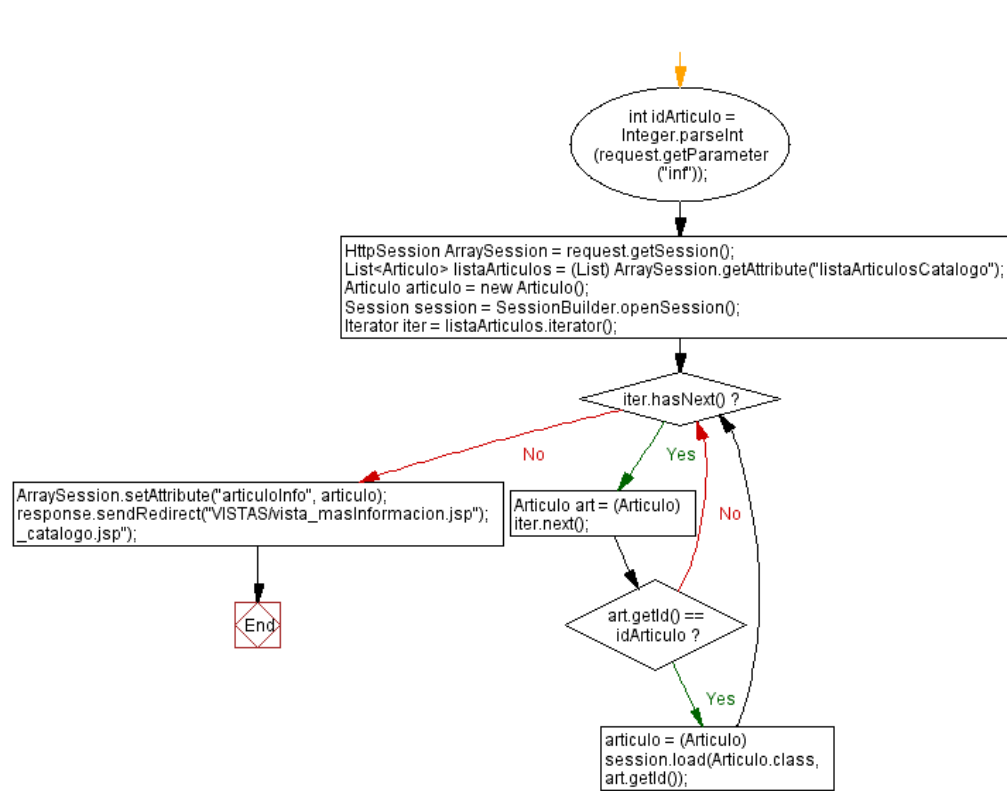
4.4. Mostrar catálogo

Extrae los artículos disponibles en la base de datos y los guarda en una variable de sesión como List.



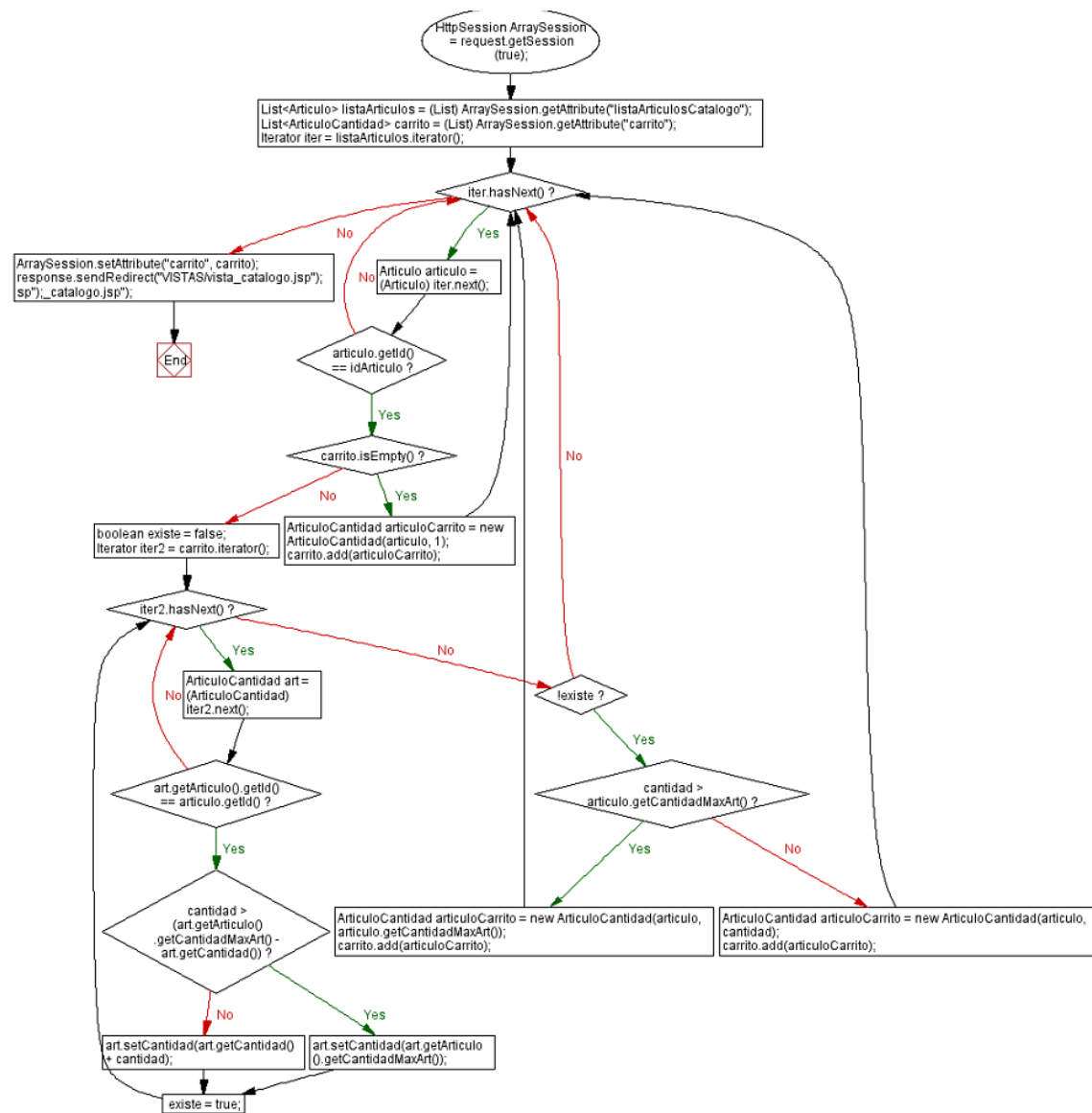
4.5. Mostrar más información sobre un artículo

Busca en la lista de artículos de la sesión el artículo seleccionado y hace un load para cargarlo en la sesión de Hibernate y hacer visibles los datos de los objetos que contiene.



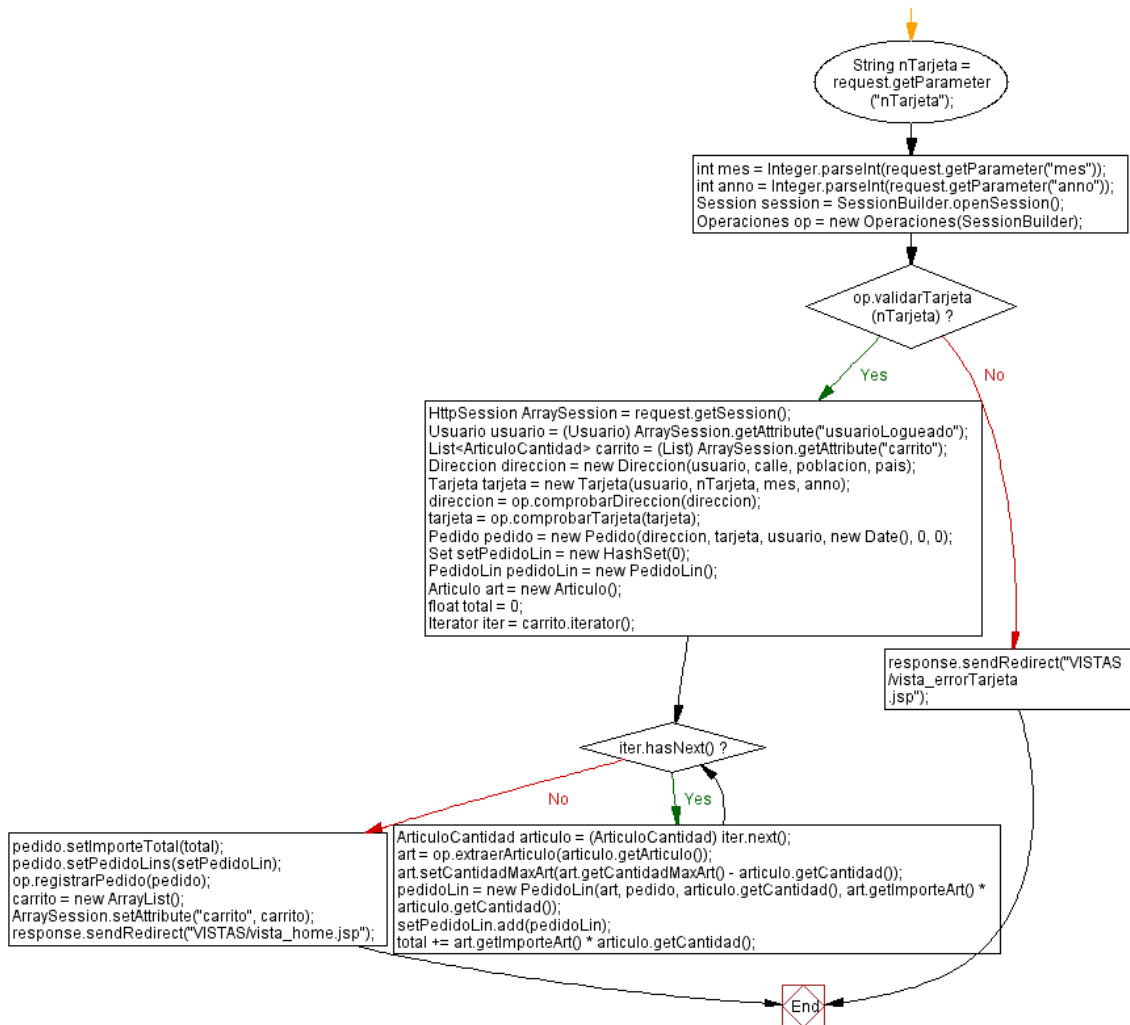
4.6. Añadir artículo a carrito

Comprueba si el artículo a añadir existe en el carrito, si existe modifica la cantidad que hay en él, también comprueba que la cantidad a añadir no sea mayor al stock disponible, si es así establece la cantidad en la máxima disponible en stock.



4.7. Realizar pedido

Valida que la tarjeta sea válida. Luego comprueba si la dirección y la tarjeta existen en la base de datos, y si es así la extrae y la añade al objeto pedido. Luego extrae de la base de datos los artículos que lleva el pedido (Ya que los objetos que tenemos al haberlos cargado anteriormente son transitorios y para insertar en la base de datos deben ser persistentes), crea un Set de líneas de pedido para insertarlo en el pedido. Hecho todas estas comprobaciones, inserta pedido en la base de datos.

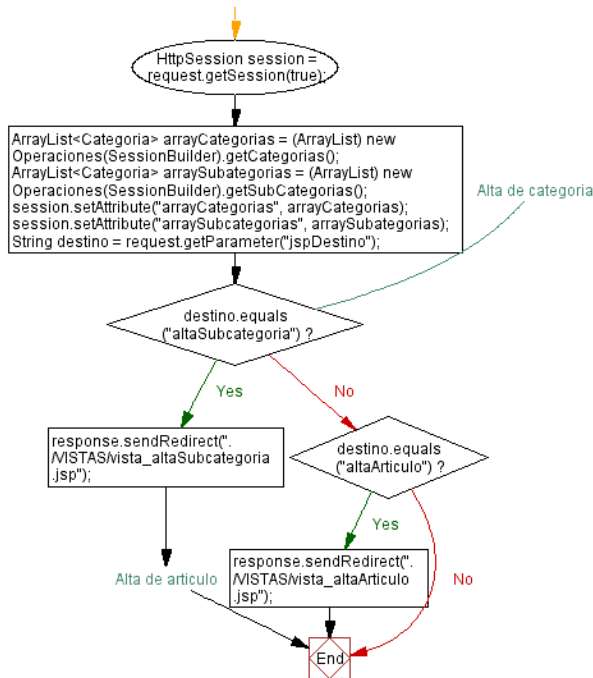


4.8. Casos de uso del panel de control

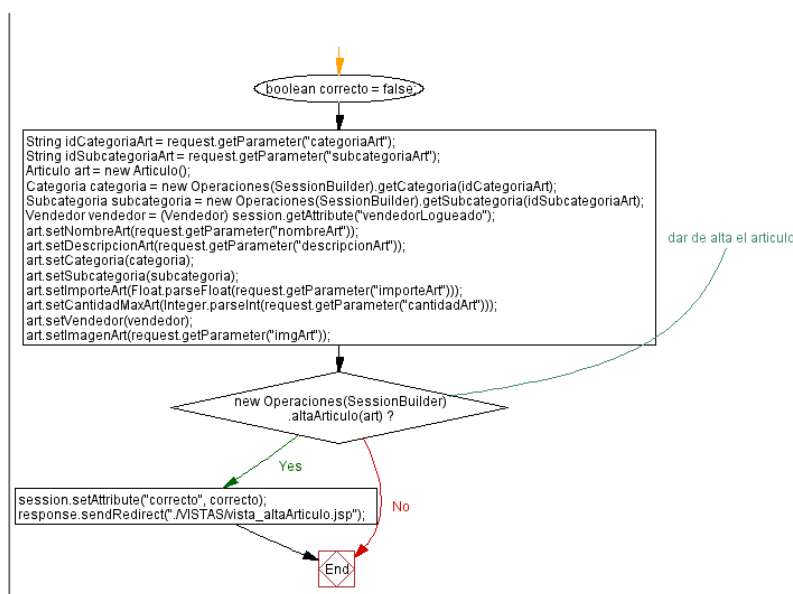
Ahora se mostraran los casos de uso de cada uno de los puntos de menú del panel de control. Se mostraran todos sin importar el usuario que tengamos logueado.

4.8.1. Alta artículo

Desde este punto de menú vamos al servlet_cargaCategorias, ya que las necesitamos para elegir cuales queremos para nuestro artículo.

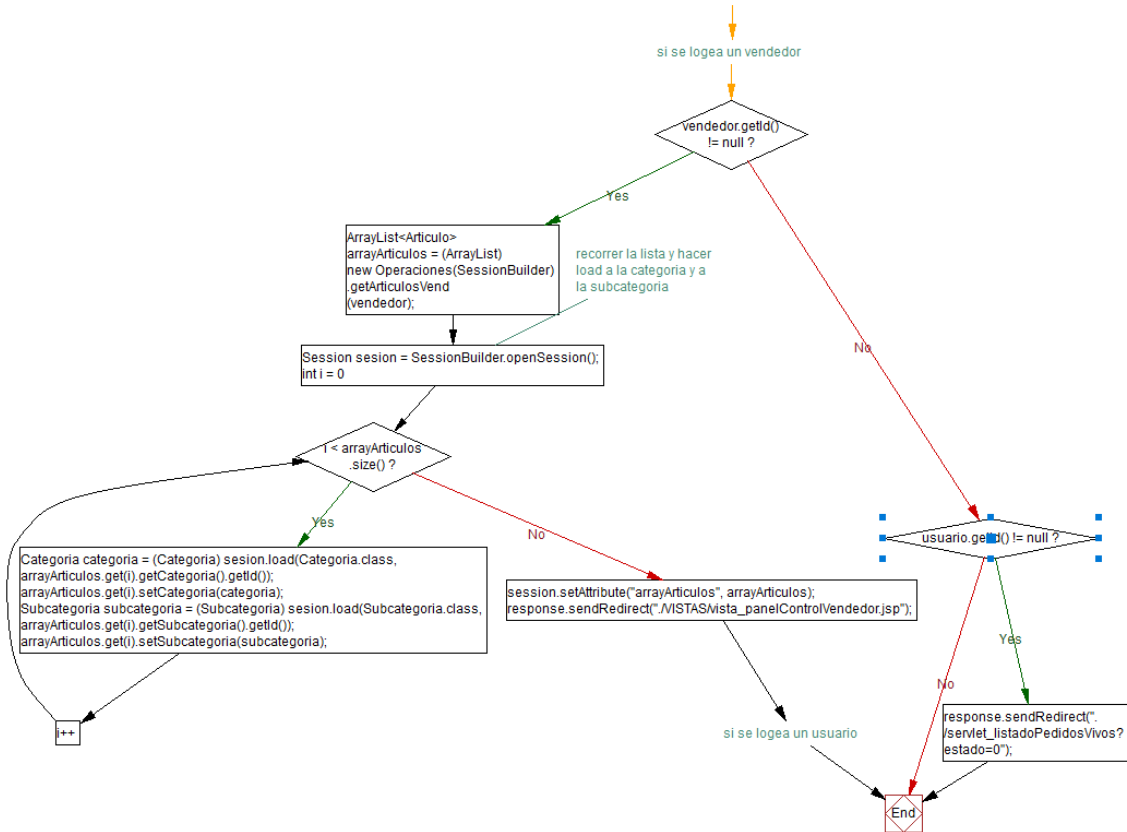


Este controlador una vez, subidas a sesión las categorías y subcategorías, nos redirecciona a la vista_altaArticulo, donde se rellenan los datos de nuestro artículo. Una vez confirmado el envío de los datos del formulario no lleva al sevlet_altaUsuario, allí recogemos los datos del artículo y lo damos de alta.



4.8.2. Listado artículos

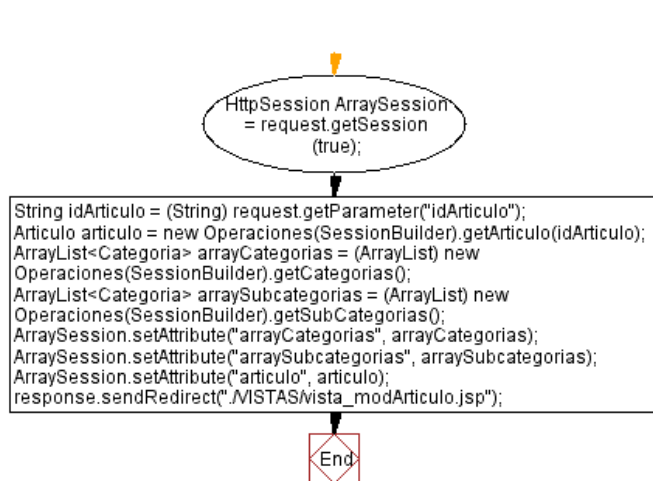
Desde este punto de menú vamos al que hemos decidido que sea el home. Listado de pedidos activos. Por ello vamos al servlet_panelControlVendedor



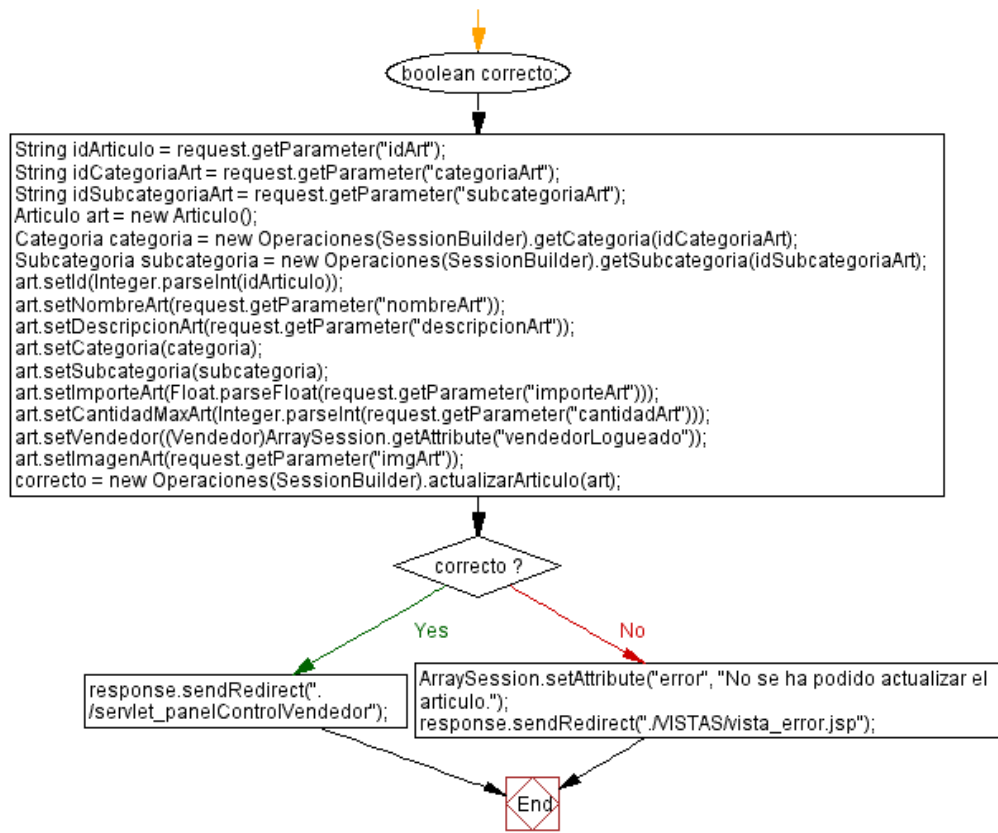
Dentro de este si estamos registrados como vendedor, sube a sesión un array con los artículos correspondientes y no lleva a la vista_panelControlVendedor para mostrarlos.

4.8.3. Modificación y eliminación de un artículo

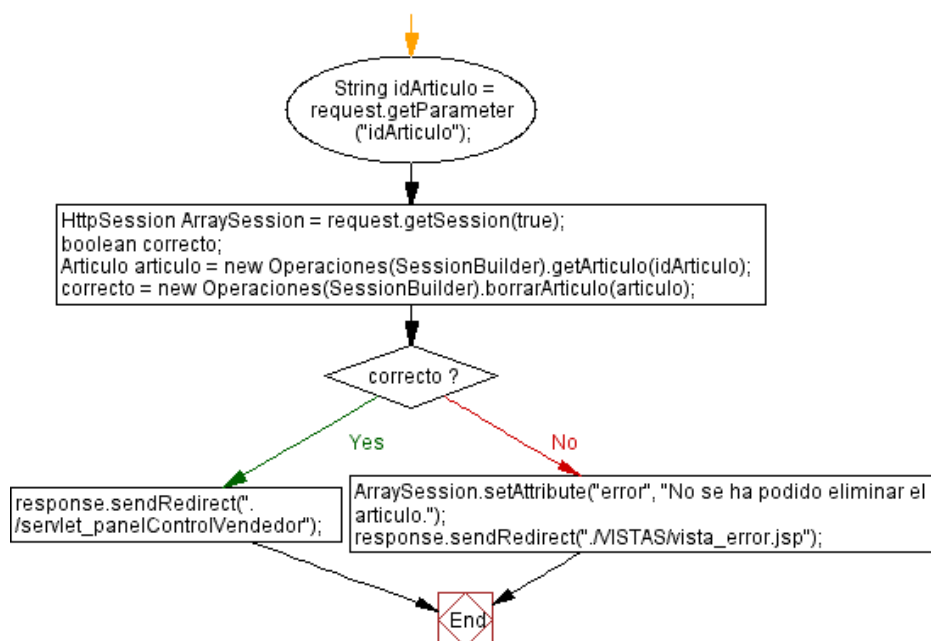
Una vez en la lista de artículos, en el icono que tenemos a la derecha nos lleva al servlet_preModArticulo, el cual carga todos los datos de ese artículo y nos lleva la vista_modArticulo.



Desde esta vista una vez modificado los datos necesarios, desde el botón actualizar vamos al servlet_modArticulo. El cual extrae los datos del artículo y llama al método para actualizarlo

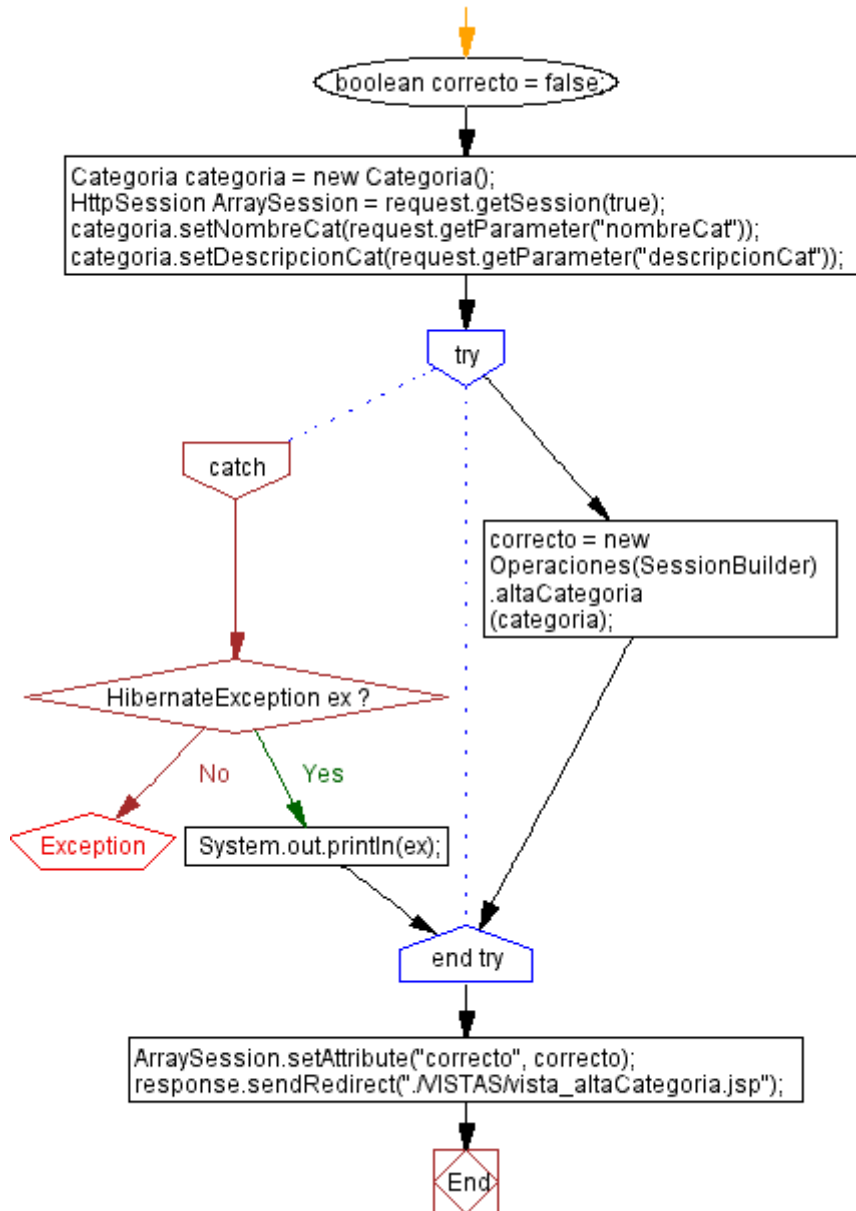


Y sin embargo si hacemos clic en eliminar, nos lleva al servlet_borrarArticulo, para eliminarlo.



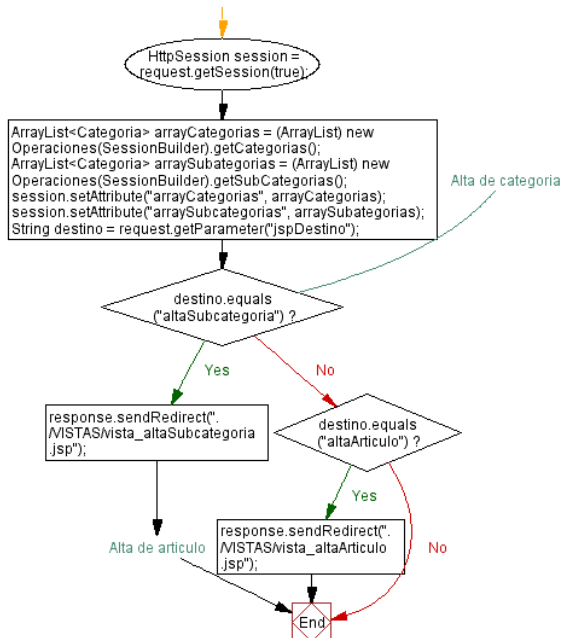
4.8.4. Alta categoría

Desde este punto de menú de nuestro panel de control vamos a la vista_altaCategoria, donde rellenamos los datos necesarios para nuestra categoría. Una vez rellenado nuestro formulario nos envía a servlet_altaCategoria, donde se recogen los datos necesarios y se da de alta la categoría desde Operaciones.

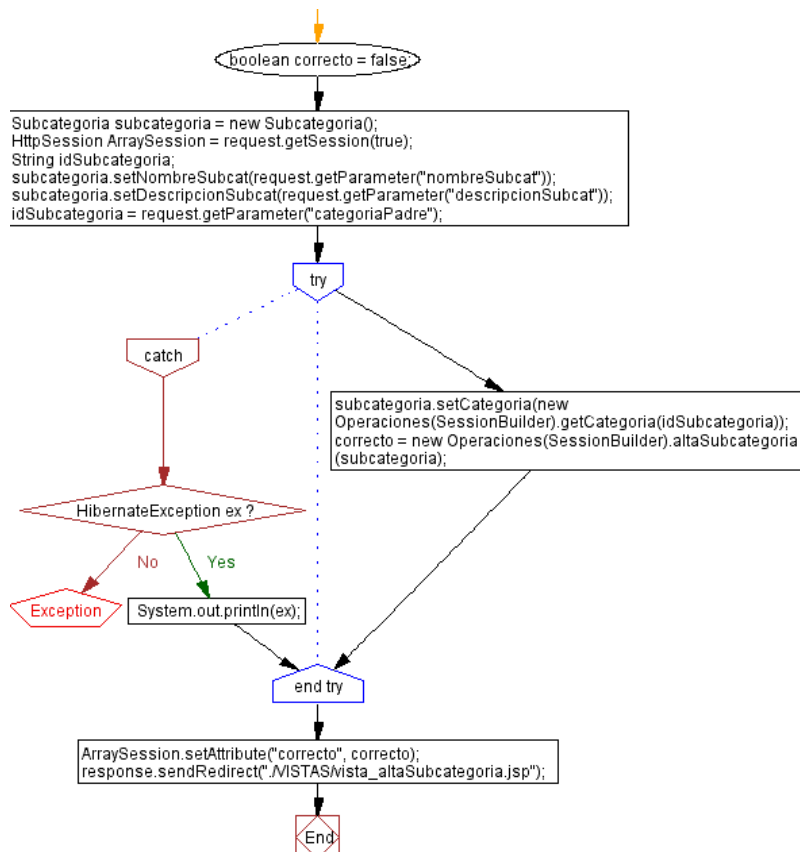


4.8.5. Alta subcategoría

El siguiente punto de menú es el de alta de subcategoría, desde el vamos al servlet_cagaCategorias, que ya hemos ido anteriormente desde el alta de artículo. Ya que para los 2 necesitamos las categorías.

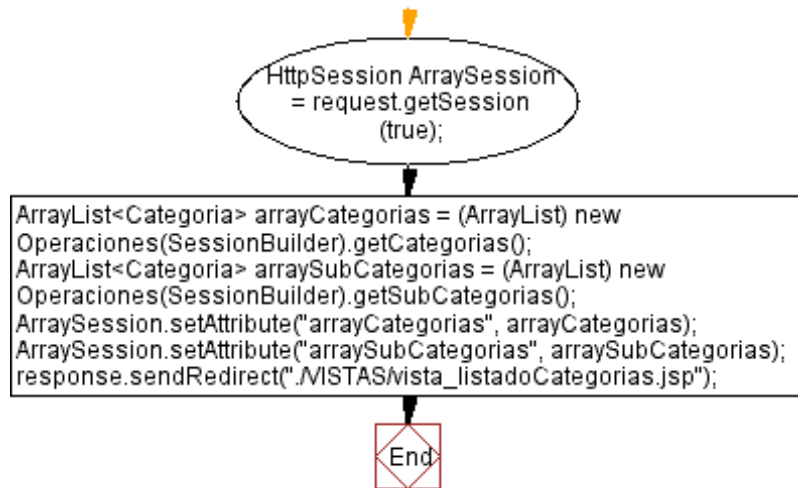


Desde este nos envía a la vista_altaSubcategoria donde rellenamos el formulario y enviamos los datos al servlet_altaSubcategoria donde se recogen los datos y se llama al método para dar de alta nuestra subcategoría.



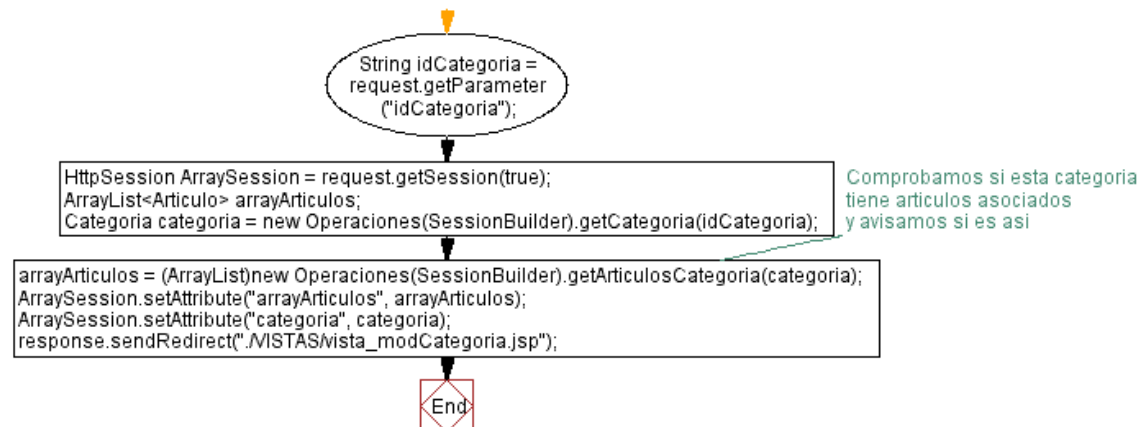
4.8.6. Listado categorías y subcategorías

El siguiente punto de menú nos sirve para listar tanto las categorías como las subcategorías. Para ellos nos lleva al servlet_listadoCategorias, donde se extraen y se suben a sesión para que desde la vista, podamos listarlas

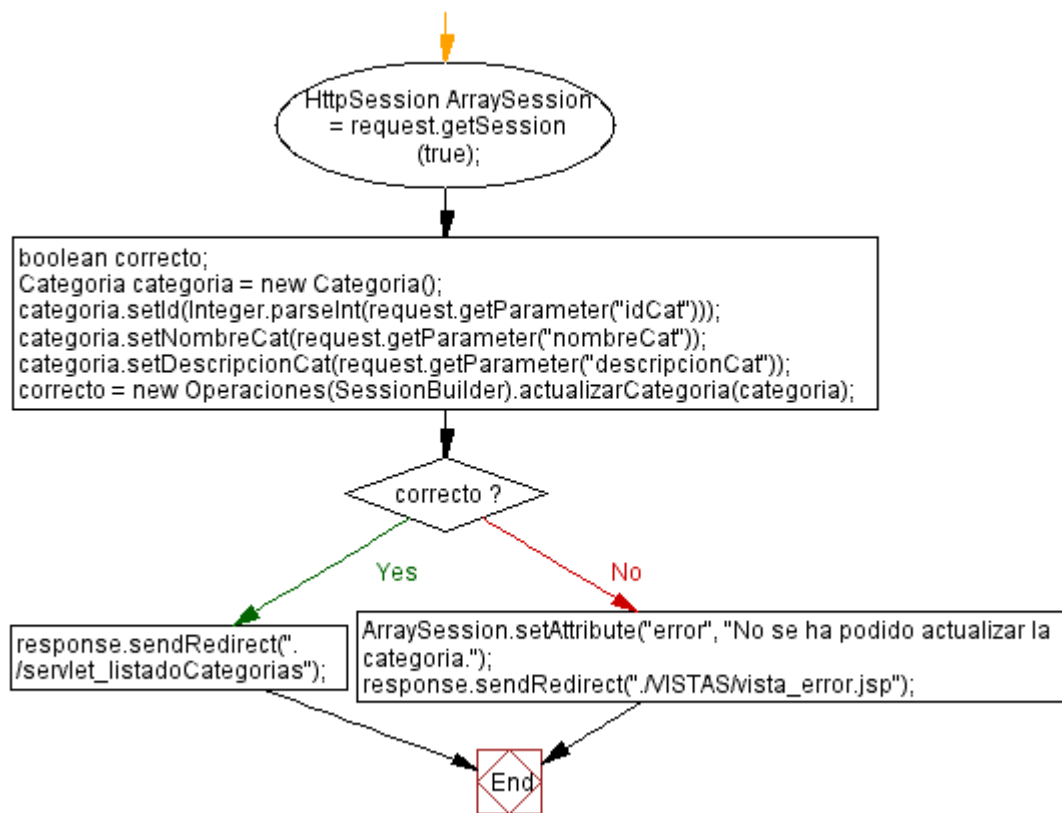


4.8.7. Modificación y eliminación de categorías

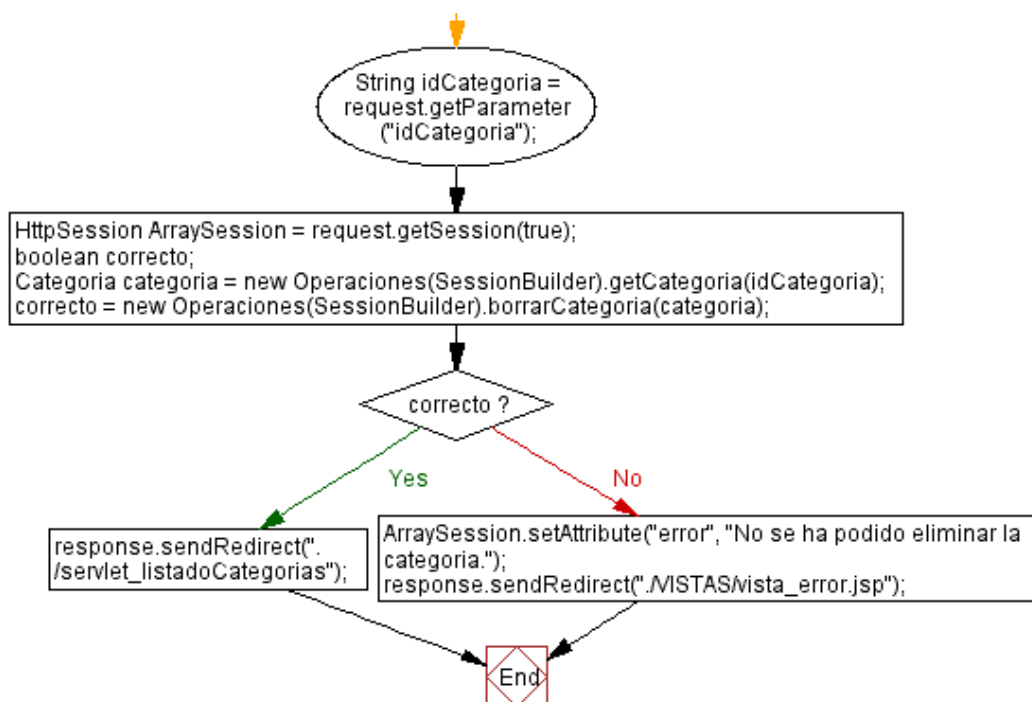
Desde el mismo listado anterior, como en el caso de artículos tenemos un pequeño icono que nos permite editar y eliminar la categoría. Este icono nos lleva al servlet_preModCategoria, este controlador extrae los datos de esta categoría y nos lleva a la vista para que podamos modificar los datos de la misma.



Una vez modificado el formulario correspondiente, no dirigimos al controlador servlet_modCategoria, en el cual recogernos todos los datos y llamamos al método que nos actualiza la categoría.

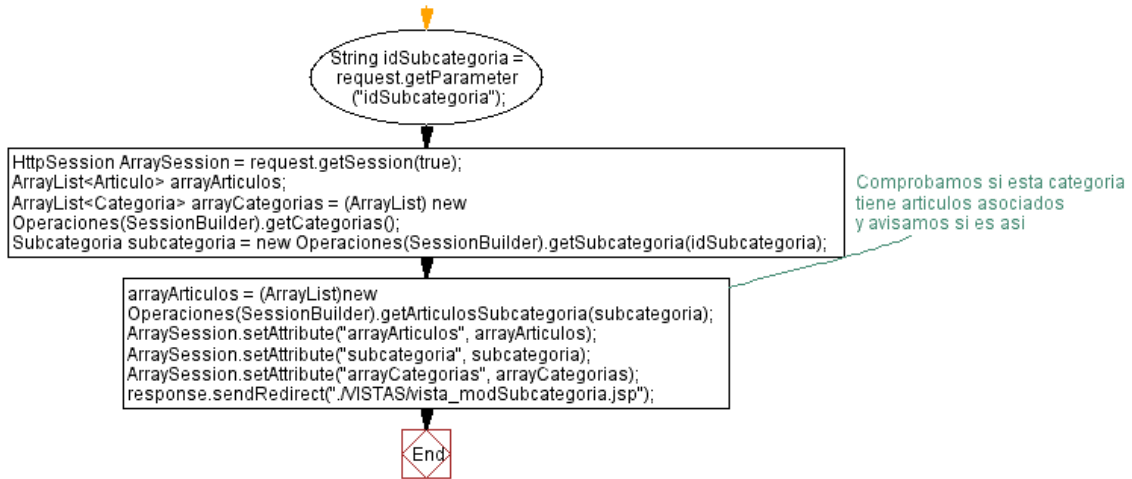


En el caso de que seleccionemos el botón de eliminar categoría, nos envía al `servlet_borrarCategoria`, donde se obtiene la categoría y la elimina.

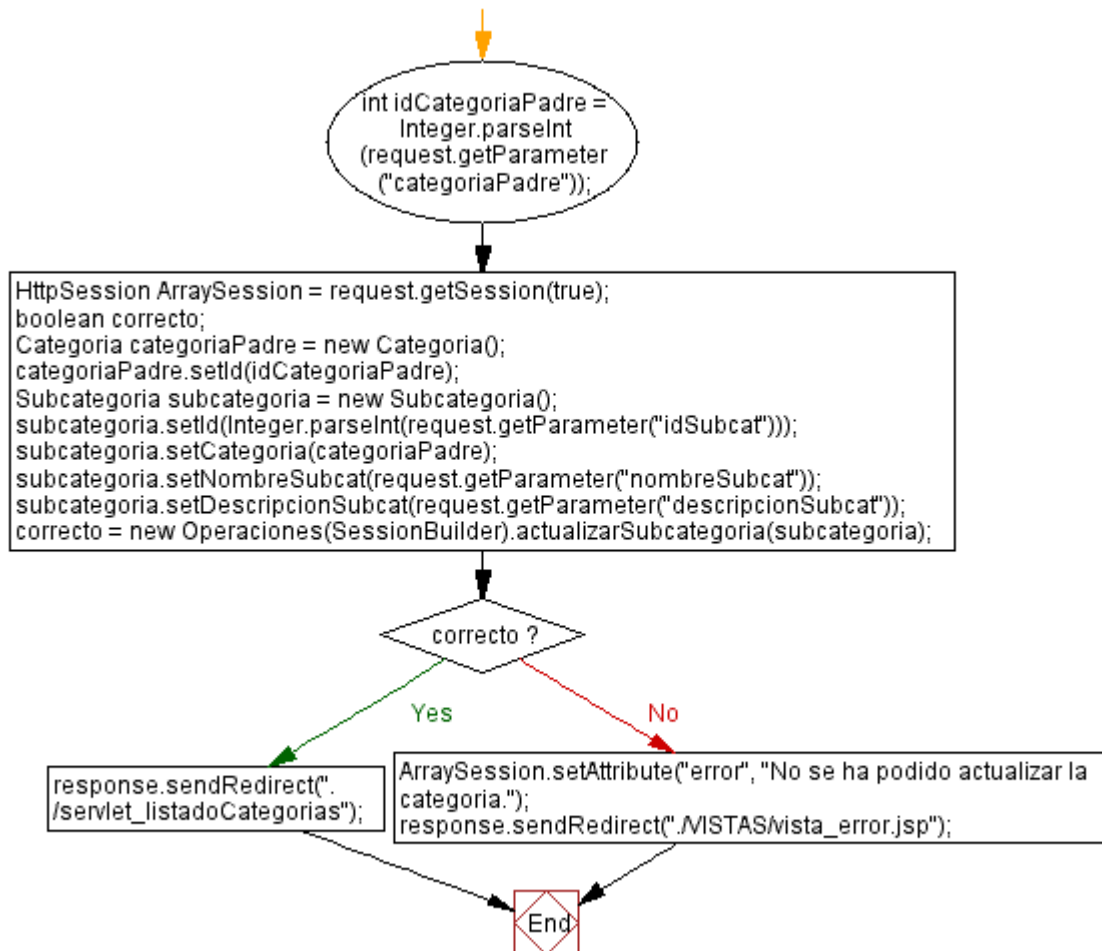


4.8.8. Modificación y eliminación de subcategorías

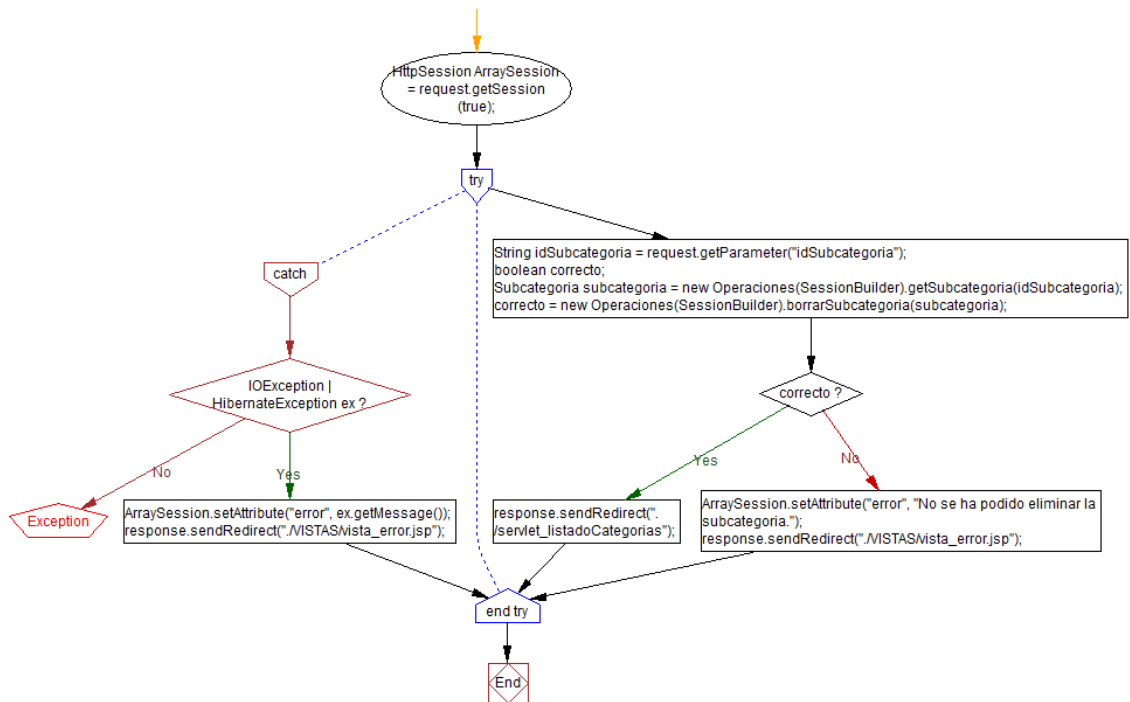
Para este caso, el cual es muy parecido al anterior, se llega de la misma forma a partir del pequeño icono en el listado. Que nos lleva al servlet_preModSubcategoria, que carga todos los datos necesarios para modificar nuestra subcategoría.



Este controlador nos envía a la vista donde nos aparecen todos los datos de nuestra subcategoría, una vez modificados podemos confirmar la actualización de los datos. Con lo que vamos al controlador servlet_modSubcategoria, que extrae los datos y actualiza.



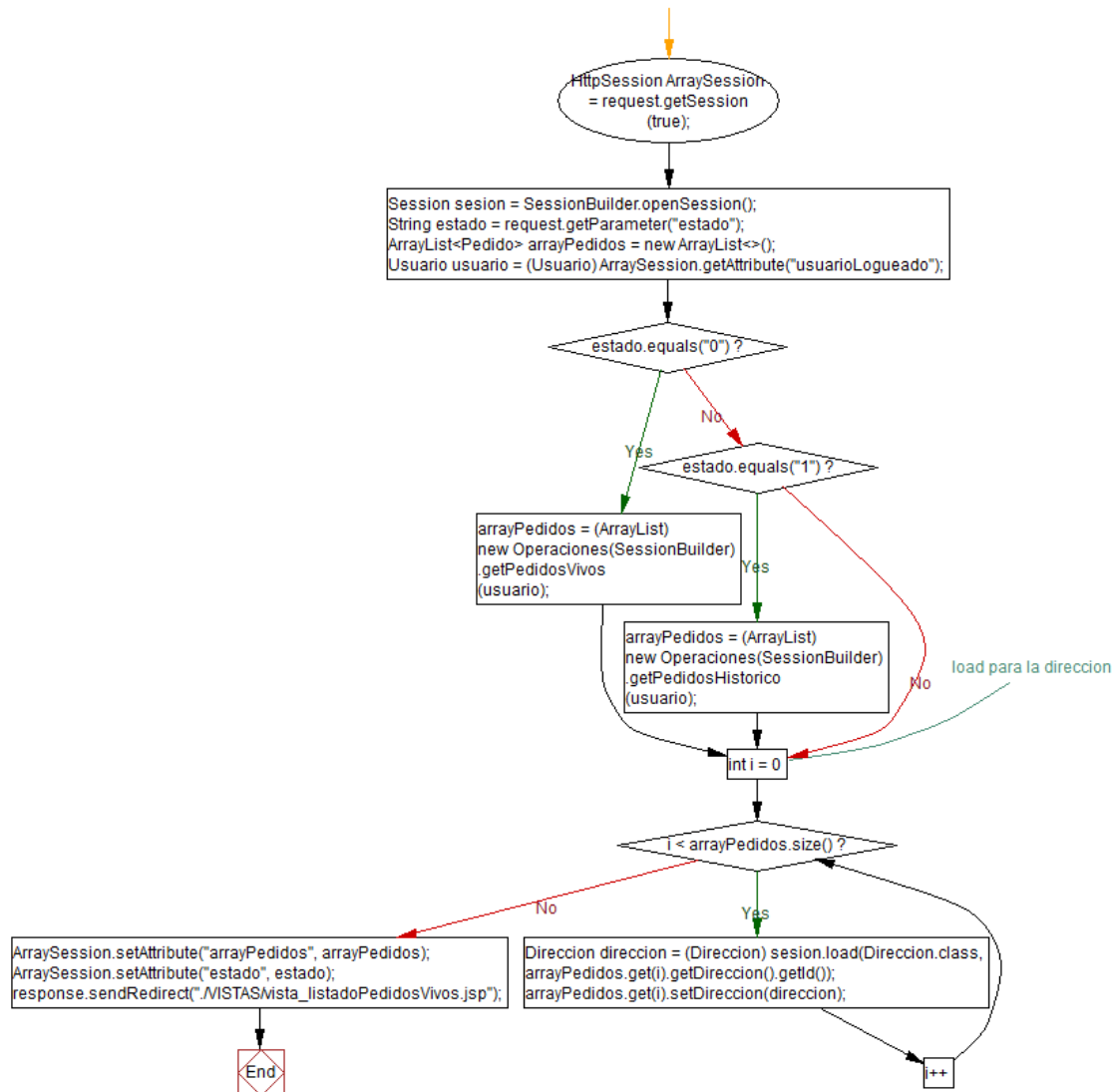
Y en el caso de que seccionemos eliminar la subcategoría nos dirigiremos a servlet_borrarSubcategoria, donde como en el caso anterior, llamamos al método correspondiente para eliminar nuestra subcategoría



4.8.9. Listado de pedidos activos y completados.

El siguiente punto de menú de nuestro panel de control es de listar los pedidos activos, este se podría seleccionar cuando hace login con un usuario.

Nos lleva al controlador servlet_listadoPedidosVivos, desde el cual obtenemos los pedidos activos en este caso con sus direcciones, subiéndolos a sesión para poder listarlos desde la vista_listadoPedidosVivos



Para el caso de listar pedidos completados se realizaría desde el mismo controlador, pero diferenciando por los pedidos completados.

En los dos casos se nos redirige a la misma vista, donde listamos los pedidos de un usuario logueado.

5. Variables de sesión

Nombre	Descripción
usuarioLogueado	(Usuario) Usuario que ha iniciado sesión.
vendedorLogueado	(Vendedor) Vendedor que ha iniciado sesión.
carrito	(List<ArticuloCantidad>) Lista que contiene los artículos añadidos al carrito y la cantidad.
listaArticulosCatalogo	(List<Ariculo>) Lista que contiene los artículos a la venta.
correcto	(boolean) Variable que contiene información de si se ha realizado una operación.
error	(boolean) Variable que contiene información de si ha fallado una operación.
arrayCategorias	(ArrayList<SubCategorias>) Lista de las subcategorías existentes.
arraySubcategorias	(ArrayList<SubCategorias>) Lista de las subcategorías existentes.
arrayPedidoLin	(ArrayList<PedidoLin>) Lista de las líneas de pedido de un usuario.
arrayPedidos	(ArrayList<Pedido>) Lista de los pedidos de un usuario.
estado	(String) Variable de si un pedido se ha completado.
articuloInfo	(Articulo) Articulo cargado con toda la información.
articulo	(Articulo) Articulo a modificar.
categoría	(Categoría) Categoría a modificar
subcategoria	(SubCategoría) Subcategoría a modificar.