

Documentación Proyecto JAVA Línea Autobuses

MIGUEL ÁNGEL GARCÍA MARTÍNEZ

Índice

| | | |
|--------|---|----|
| 1. | Información acerca del proyecto | 2 |
| 1.1. | Software utilizado para su desarrollo | 2 |
| 1.2. | Descripción del proyecto..... | 2 |
| 1.3. | Estructura de carpetas | 2 |
| 2. | Diagrama de tablas..... | 4 |
| 3. | Diagrama de clases..... | 5 |
| 4. | Casos de uso | 6 |
| 4.1. | Proceso completo para sacar un billete..... | 6 |
| 4.1.1. | Buscar Viaje | 7 |
| 4.1.2. | Seleccionamos el viaje..... | 8 |
| 4.1.3. | Recoger la información de los Viajeros | 9 |
| 4.1.4. | Seleccionar los asientos de los Viajeros | 11 |
| 4.1.5. | Registro del Cliente | 12 |
| 4.1.6. | Registro o selección de una tarjeta para realizar el pago | 13 |
| 4.1.7. | Proceso de inserción de todos los datos | 14 |
| 4.2. | Proceso de confirmación de llegada de un Viaje | 15 |
| 4.2.1. | Selección del Viaje que finaliza | 16 |
| 4.2.2. | Proceso de backup de los datos | 17 |
| 4.2.3. | Disparador | 18 |
| 5. | Variables de sesión..... | 19 |
| 6. | Datos usados para ejemplificar la aplicación | 19 |

1. Información acerca del proyecto

1.1. Software utilizado para su desarrollo

Para el desarrollo de este proyecto he 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

1.2. Descripción del proyecto

Aplicación Web desarrollada para gestionar una empresa de autobuses a nivel provincial, ejemplificada en este caso en la provincia de Albacete.

Con esta aplicación tenemos la posibilidad de ejemplificar la reserva de un viaje completo y la confirmación de llegada del mismo, realizando así sus correspondientes copias de seguridad, de los datos correspondientes.

1.3. Estructura de carpetas

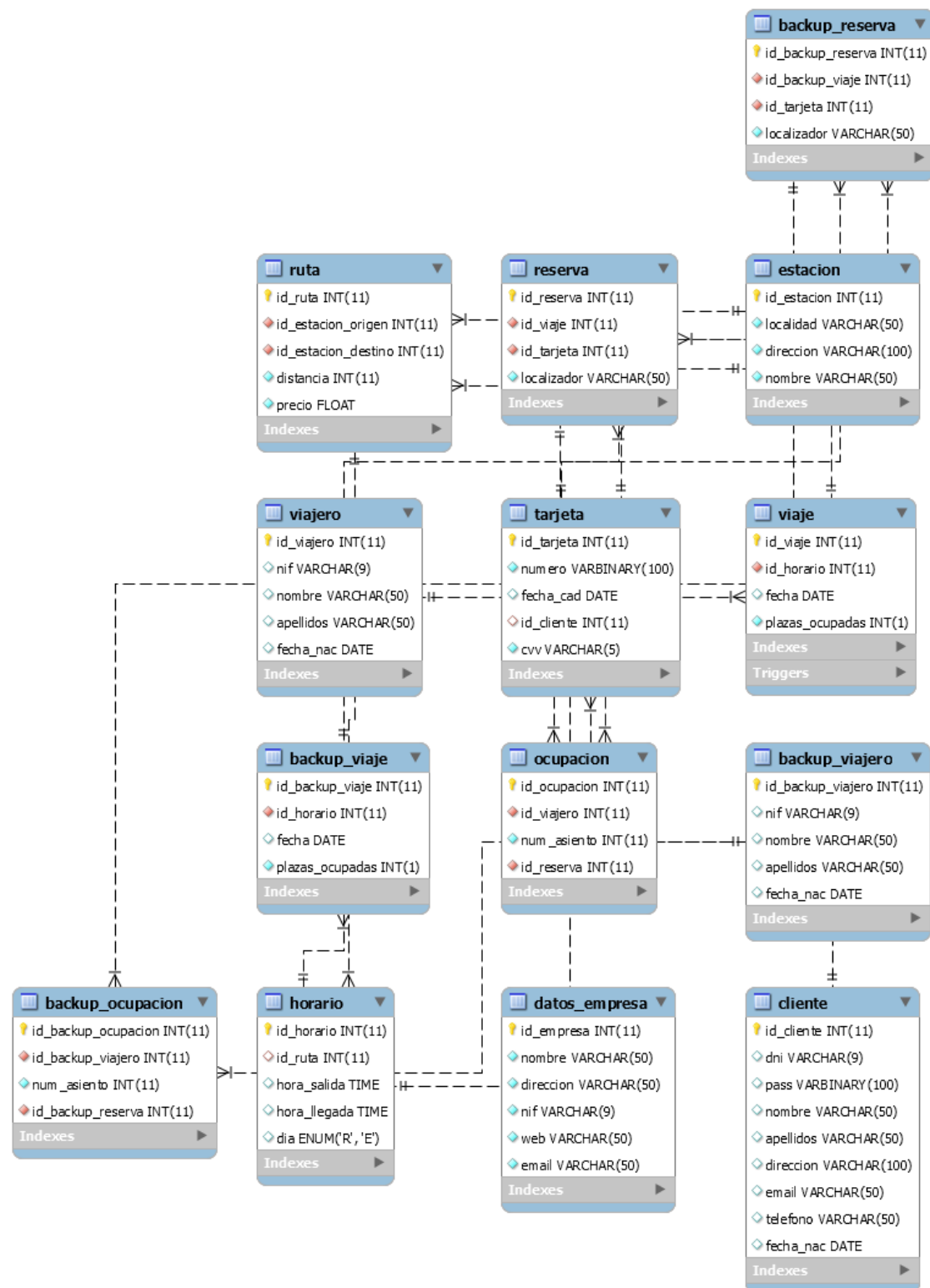
La aplicación esta desarrolla en JAVA Web respetando el modelo – vista – controlador (MVC) con esta estructura de carpeta y ficheros:

Carpetas principales:

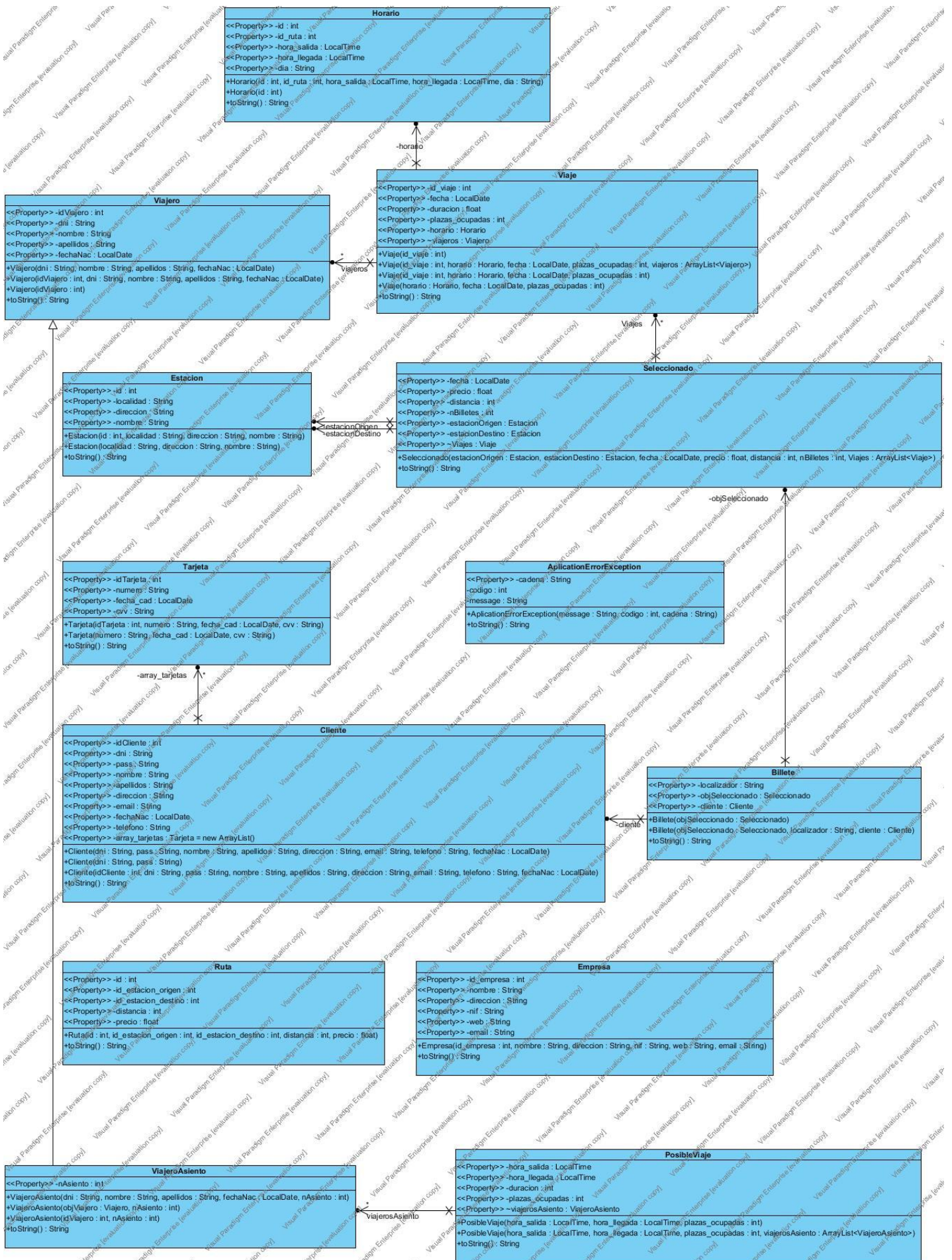
- Web: donde se encuentran todas las vistas (JSP) incluido el index.jsp
- Java->dao: donde se encuentran la conexión a la base de datos (Implementación el uso del patrón Singleton)
- Java->modelo: donde se encuentran las clases
- Java->servlet: donde se encuentran los controladores que interactúan con las vistas y la capa dao
- Js: donde se encuentra el archivo JavaScript



2. Diagrama de tablas



3. Diagrama de clases



4. Casos de uso

Vamos a hacer un resumen de las funciones principales de la aplicación web. Como se indica anteriormente respecto al modelo vista controlador, con lo que el control de la aplicación cambia constantemente entre estos.

Las operaciones principales que realiza la aplicación web están contenidas dentro de la clase Operaciones, de la cual mostraremos algún diagrama de flujo.

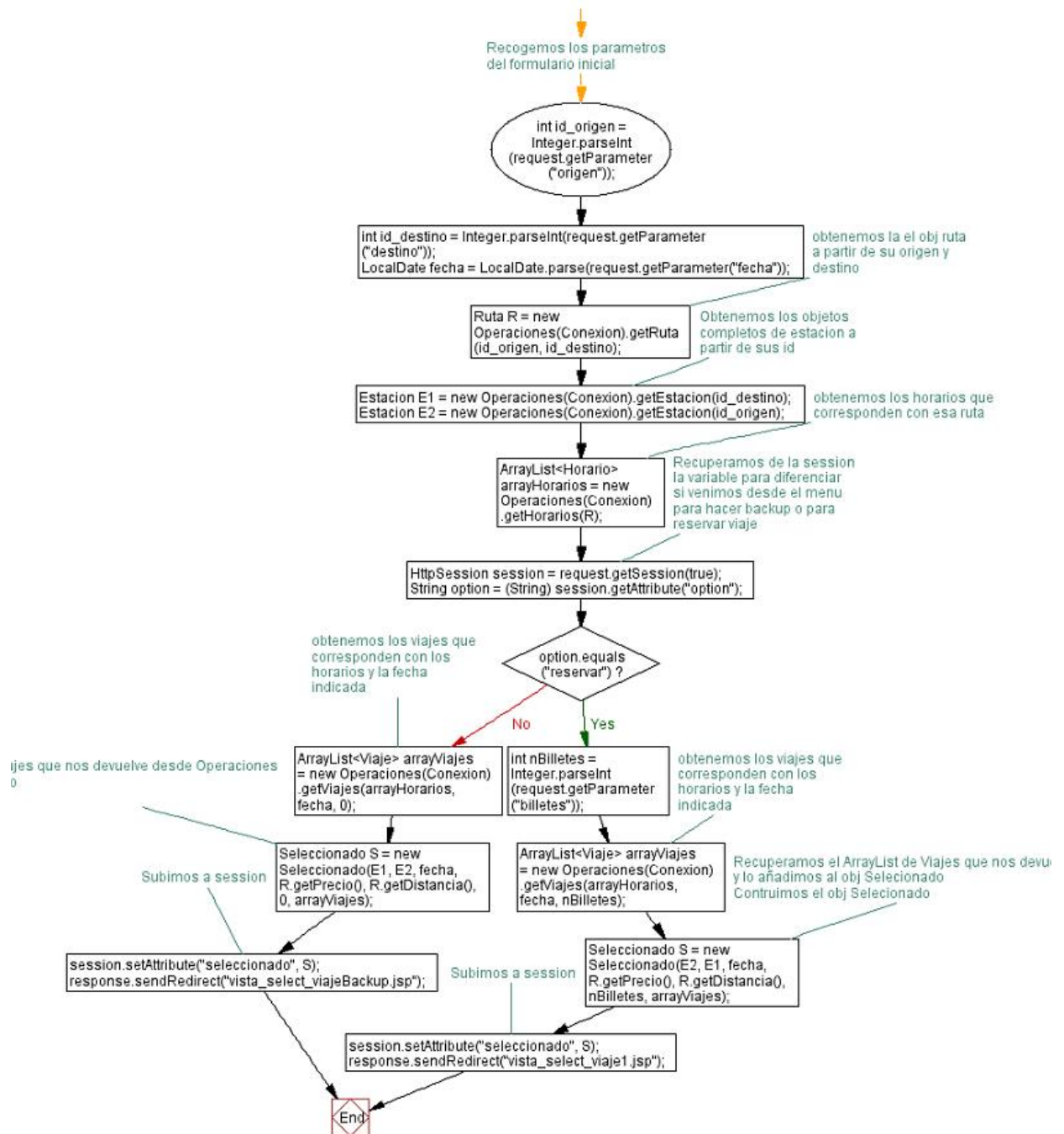
Nota: todos los controladores o servlet, están protegidos con un try-catch, con lo que lo obviamos de los dibujos.

4.1. Proceso completo para sacar un billete

Vamos a mostrar un diagrama de la función principal de la aplicación, por partes ya que es un proceso extenso.

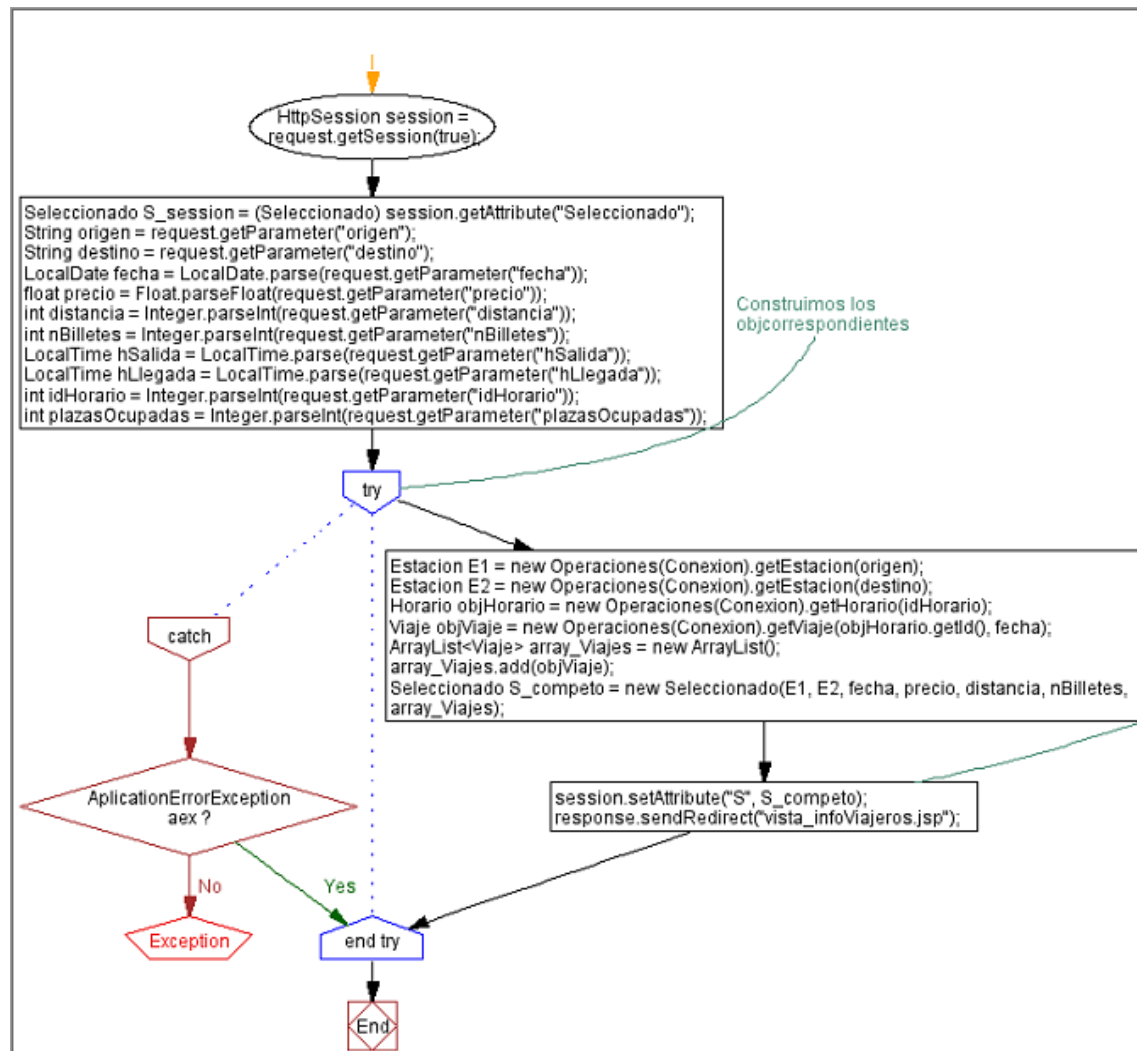
4.1.1. Buscar Viaje

Una vez completado el formulario de la vista “vista_home.jsp”, pasamos por el controlador y por la clase Operaciones para extraer los viajes disponibles en función de los parámetros que nos introduce el usuario. Este mismo controlador se utiliza tanto para buscar viaje para obtener un billete como para confirmar su llegada, que la veremos más adelante.



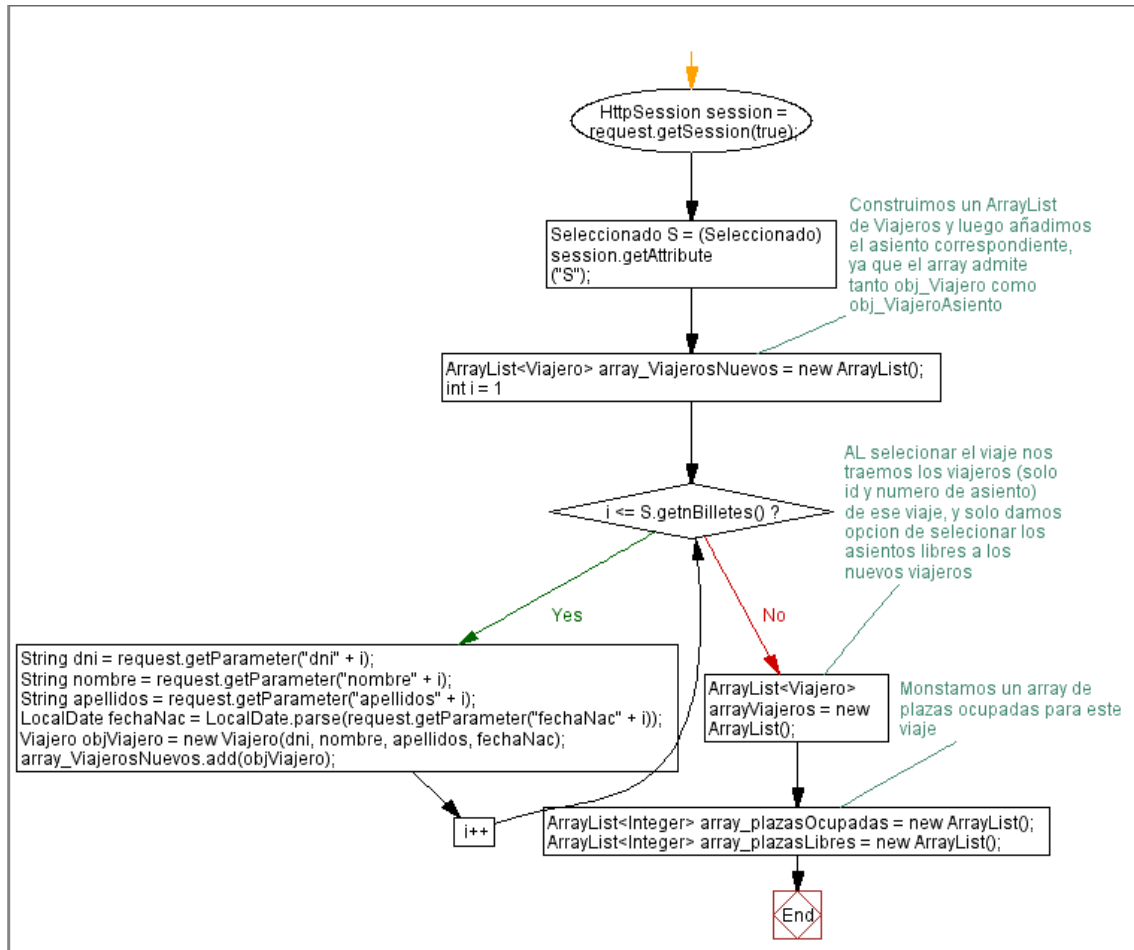
4.1.2. Seleccionamos el viaje

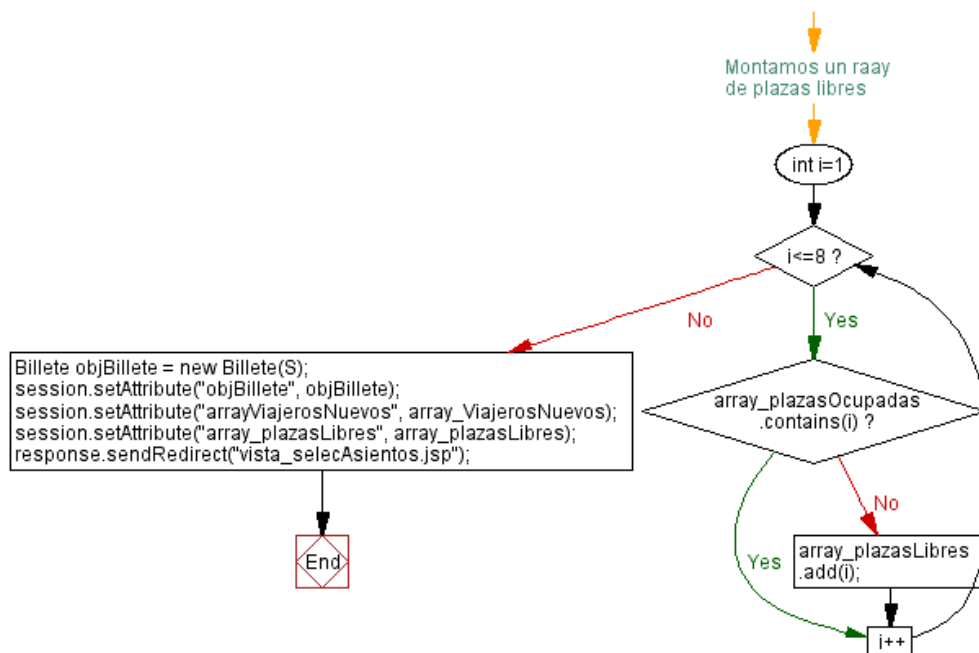
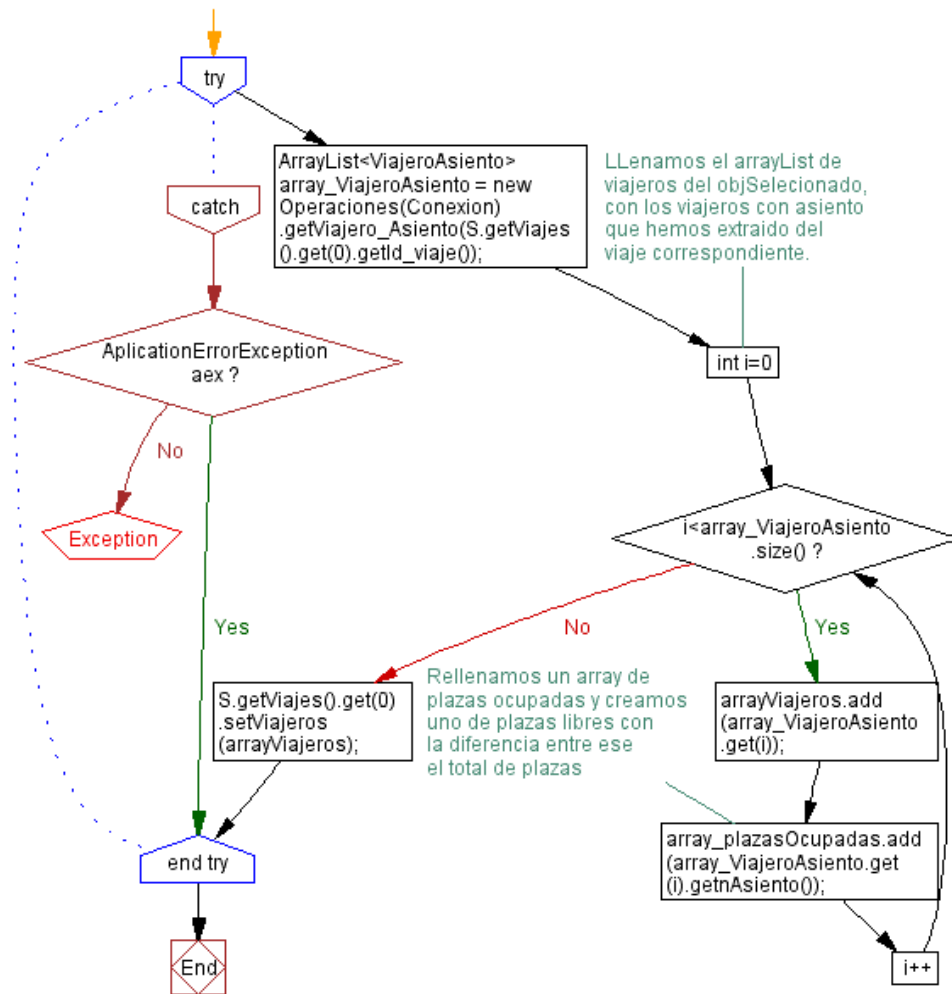
Recogemos todos los datos del viaje y construimos un objeto de la clase Seleccionado



4.1.3. Recoger la información de los Viajeros

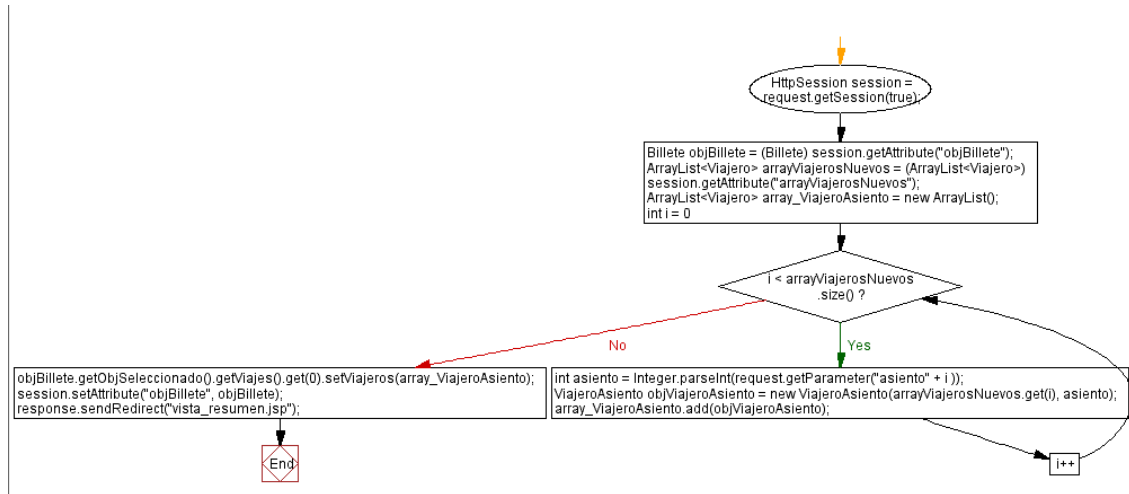
Recogemos la información que el usuario nos facilita en los formularios. Y montamos una serie de ArrayList para poder gestionar los asientos





4.1.4. Seleccionar los asientos de los Viajeros

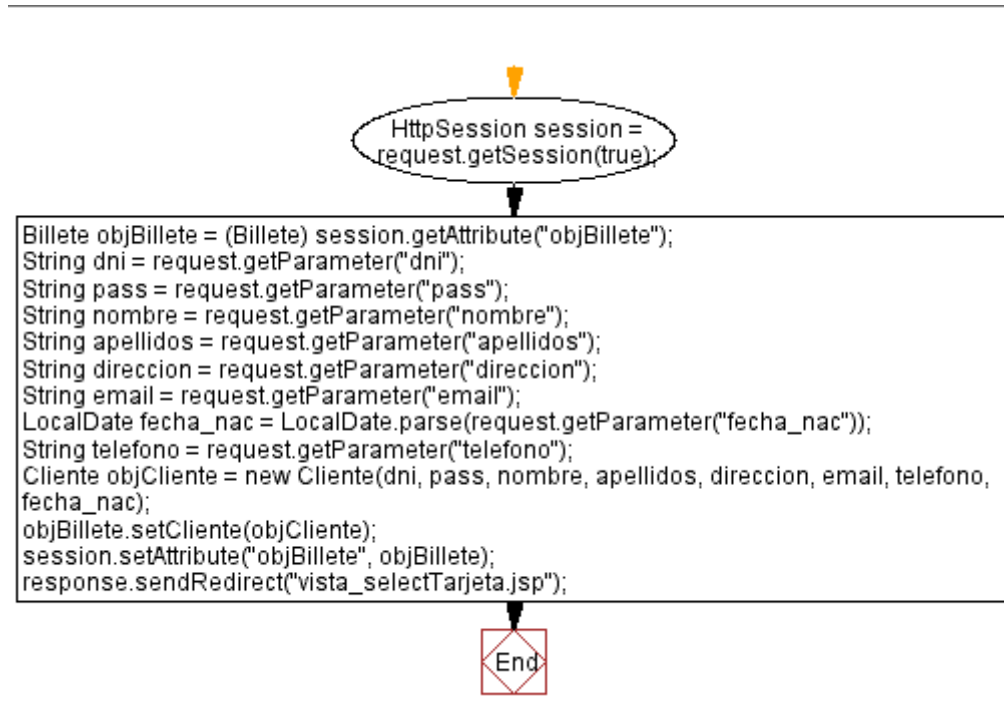
Una vez recogidos los datos del formulario añadimos el asiento correspondiente a cada Viajero y lo redirigimos a una vista resumen de su billete.



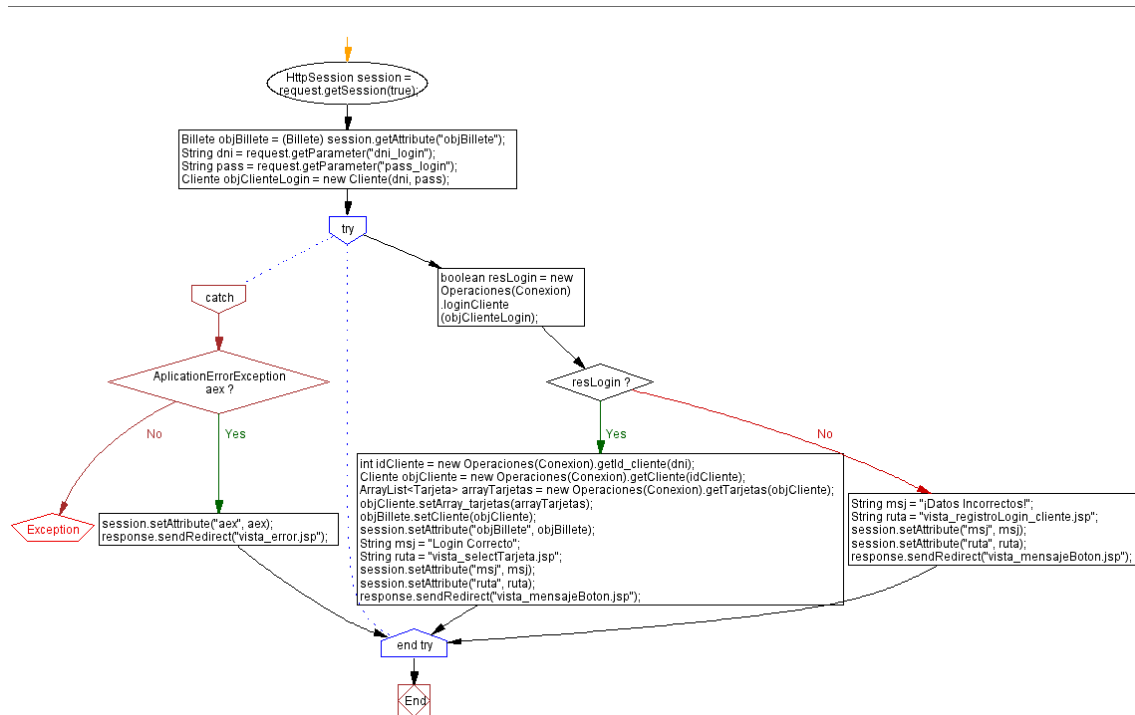
4.1.5. Registro del Cliente

Después de la vista resumen, procedemos al paso de un formulario del Cliente, donde podemos hacer login si ya disponemos de una cuenta o registrarnos con una cuenta nueva.

Ejemplificamos el registro



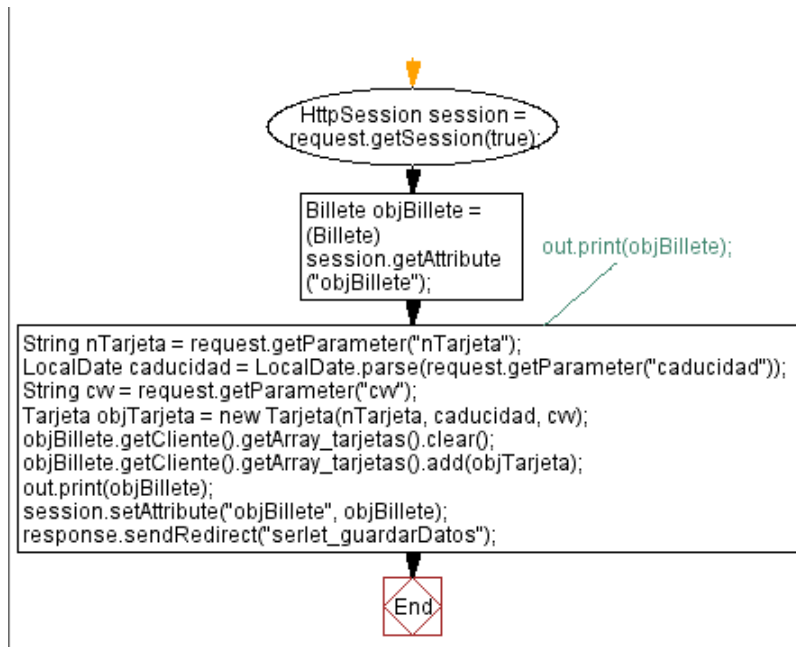
Ejemplificamos login del cliente



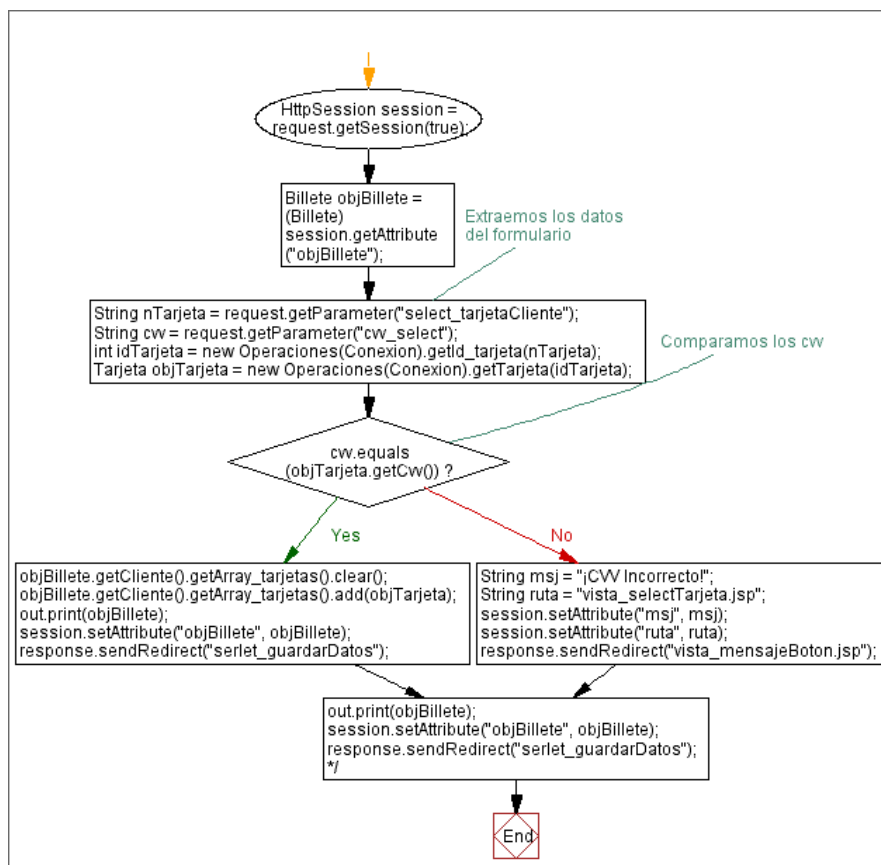
4.1.6. Registro o selección de una tarjeta para realizar el pago

Una vez el cliente se registra o inicia sesión con su cuenta, pasamos a un formulario de tarjetas, ya se para seleccionar una que ya tiene como registrar una nueva tarjeta

Registro de Tarjeta:



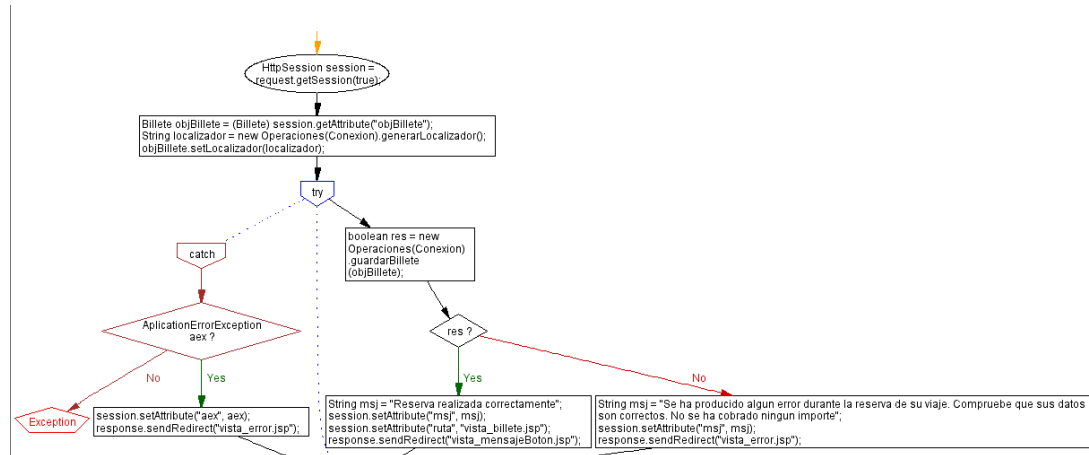
Selección de una tarjeta existente



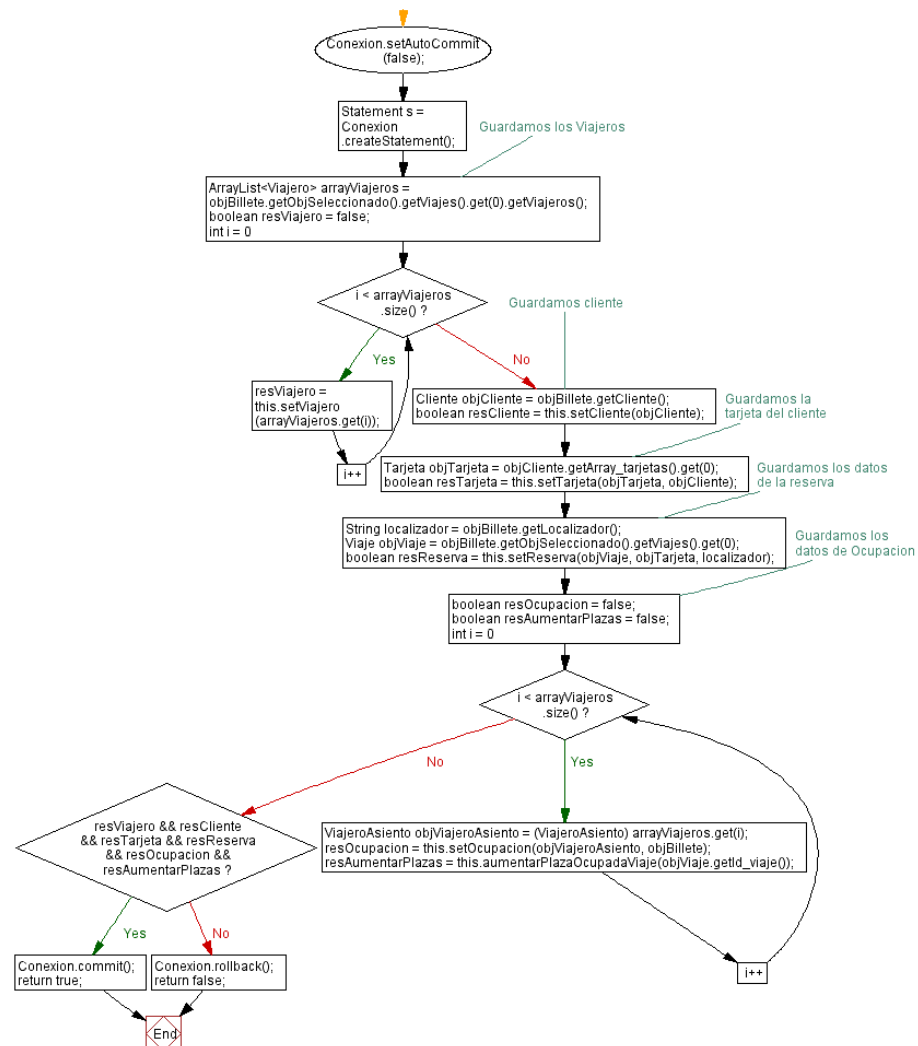
4.1.7. Proceso de inserción de todos los datos

Una vez comprobados los diferentes dato necesarios para realizar el billete, Datos de los Viajeros, su número de asiento correspondiente, los datos del cliente, forma de pago, etc.

Interactuamos con la clase Operaciones para almacenar todos los datos.



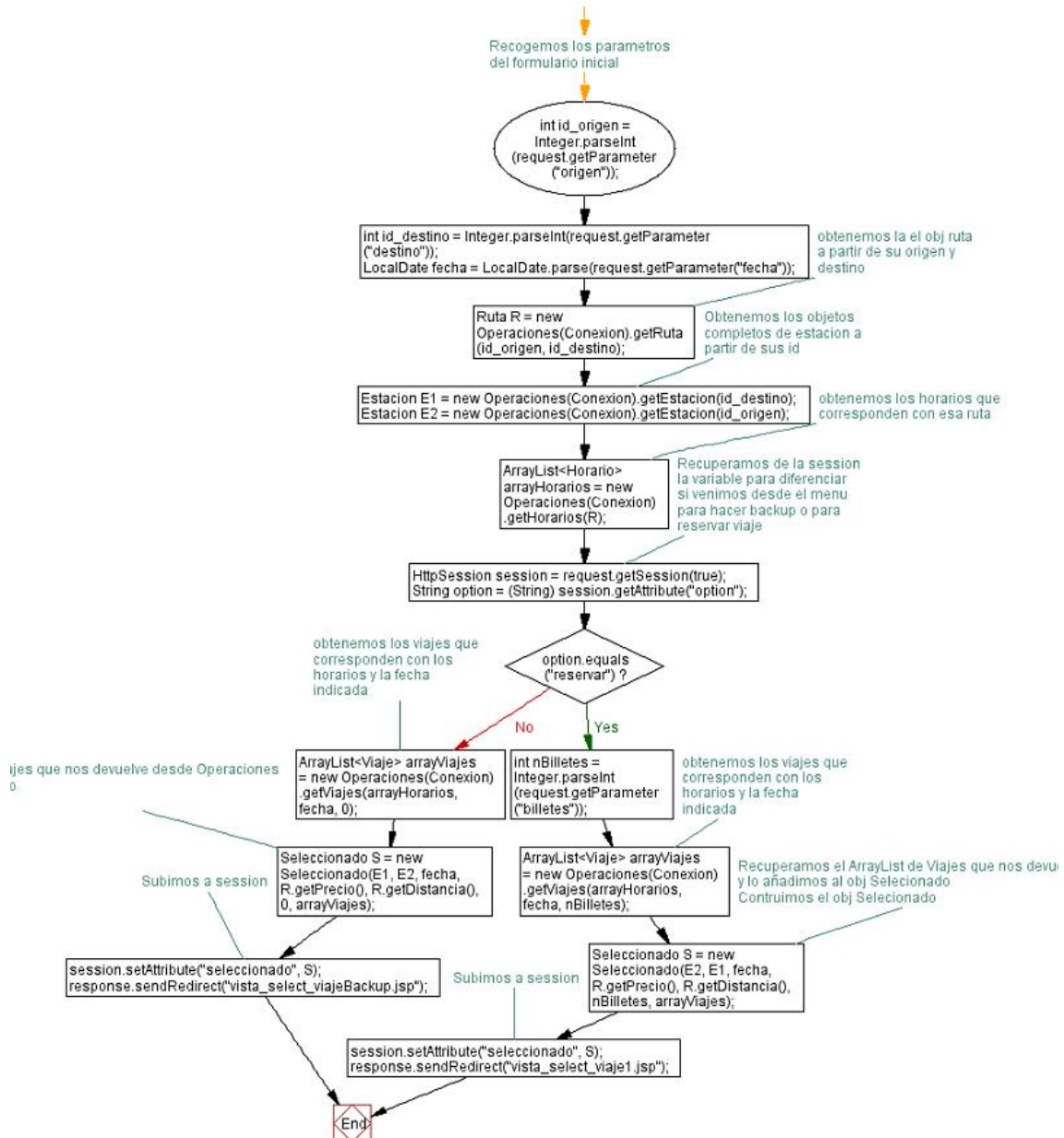
Método de la clase Operaciones que se encarga de guardar todos los datos en la base de datos



4.2. Proceso de confirmación de llegada de un Viaje

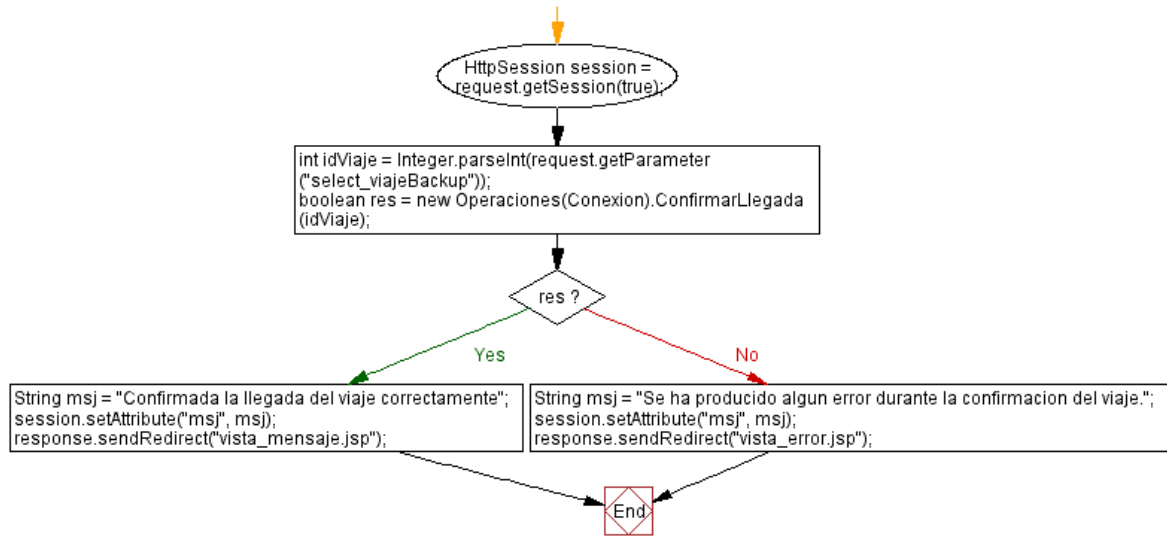
La aplicación también nos permite la posibilidad de confirmar la llegada de un Viaje y realizar una copia de seguridad de los datos que corresponden.

Para ello se utiliza una vista distinta, pero con el mismo formulario y el mismo controlador que la primera parte del proceso anterior. Junto:



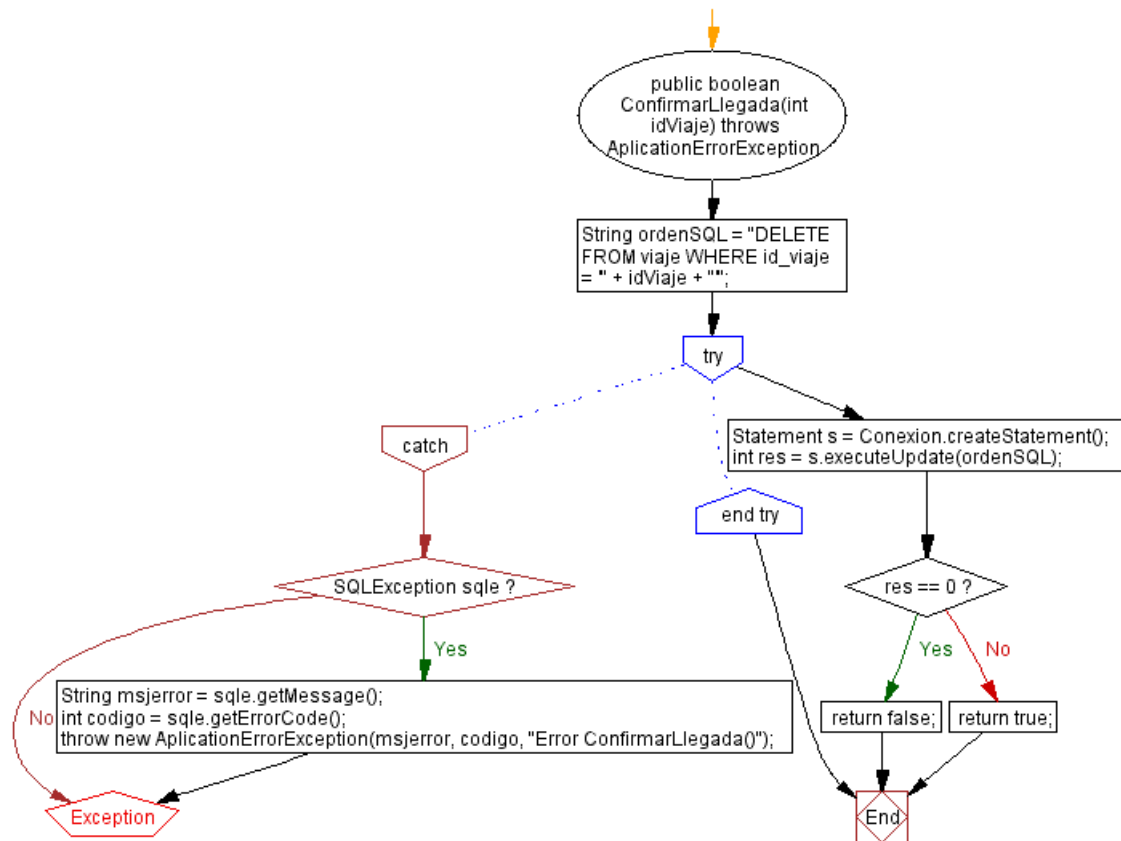
4.2.1. Selección del Viaje que finaliza

Una vez seleccionado el viaje que ha realizado su llegada por el administrador, recogemos esos datos y realizamos las diferentes operaciones, tanto de eliminación de datos de las tablas activas e inserción del mismo en las tablas de backup. La mayor parte de este proceso se realiza desde un disparador en la base de datos.



4.2.2. Proceso de backup de los datos

Método de la clase Operaciones que se encargará del Backup



4.2.3. Disparador

Disparador que realiza las inserciones y eliminaciones de los datos necesarios para realizar el backup

```
BEGIN

INSERT INTO backup_viajero SELECT * FROM viajero WHERE id_viajero IN
(SELECT id_viajero FROM ocupacion WHERE id_reserva IN (SELECT
id_reserva FROM reserva WHERE id_viaje = OLD.id_viaje));

INSERT INTO backup_viaje SELECT * FROM viaje WHERE id_viaje =
OLD.id_viaje;

INSERT INTO backup_reserva SELECT * FROM reserva WHERE id_viaje =
OLD.id_viaje;

INSERT INTO backup_ocupacion SELECT * FROM ocupacion WHERE id_reserva
IN (SELECT id_reserva FROM reserva WHERE id_viaje = OLD.id_viaje);

DELETE FROM ocupacion WHERE id_reserva IN (SELECT id_reserva FROM
reserva WHERE id_viaje = OLD.id_viaje);

DELETE FROM reserva WHERE id_viaje = OLD.id_viaje;

DELETE FROM viajero WHERE id_viajero NOT IN (SELECT id_viajero FROM
ocupacion);

END
```

5. Variables de sesión

Resumen de las variables de sesión que uso en la aplicación, principalmente son usadas para transferir los datos necesarios desde los servlet o controladores hasta las vistas o jsp.

| Nombre | Contenido |
|----------------------------|--|
| Msj | Mensaje para informar al usuario de diferentes eventos que se producen en la aplicación |
| Ruta | Dirección a la que conduce el botón de una vista de mensaje al usuario |
| aex | Contiene un objeto de la excepción propia de la aplicación |
| objBillete | Contiene un objeto de la clase Billete que se va rellenando de los datos que se necesitan para realizar un viaje |
| Seleccionado | Contiene un objeto de la clase Seleccionado, que tiene los datos que ha seleccionado el usuario para buscar su viaje |
| arrayViajerosNuevos | Contiene un ArrayList de objetos de la clase viajero |
| array_plazasLibres | Contiene un ArrayList con las plazas libres de un viaje |
| arrayEstaciones | Contiene un ArrayList con objetos de la clase Estación para mostrarlas a usuario |

6. Datos usados para ejemplificar la aplicación

He ejemplificado los siguientes datos para la aplicación:

- Días: 23/03/2018 (2 viajes) y 24/03/2018 (1 viaje)
- Ruta: Albacete-Cenizate
- Máximo de plazas por autobús: 8