

Documentación Proyecto JAVA – Autobuses



**Ángel David
Castillo Arenas**

Contenido

1.	Información sobre el 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.	Casos de uso	5
3.1.1.	Buscar Viaje	5
3.1.2.	Registro del Cliente	6
3.1.3.	Registro o selección de una tarjeta para realizar el pago	6
3.1.4.	Proceso de inserción de todos los datos	6
3.2.	Proceso de confirmación de llegada de un Viaje	6
3.2.1.	Selección del Viaje que finaliza	6
3.2.2.	Disparador	6
4.	Datos usados para ejemplificar la aplicación	7

1. Información sobre el proyecto

1.1. Software utilizado para su desarrollo

- Lenguaje principal Java Web
- Bootstrap (HTML5 y CSS3), JavaScript
- Base de datos: MYSQL
- Servidor de Aplicaciones: Glassfish Server 4.1.1

1.2. Descripción del proyecto

Aplicación Web desarrollada para gestionar una empresa de autobuses.

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

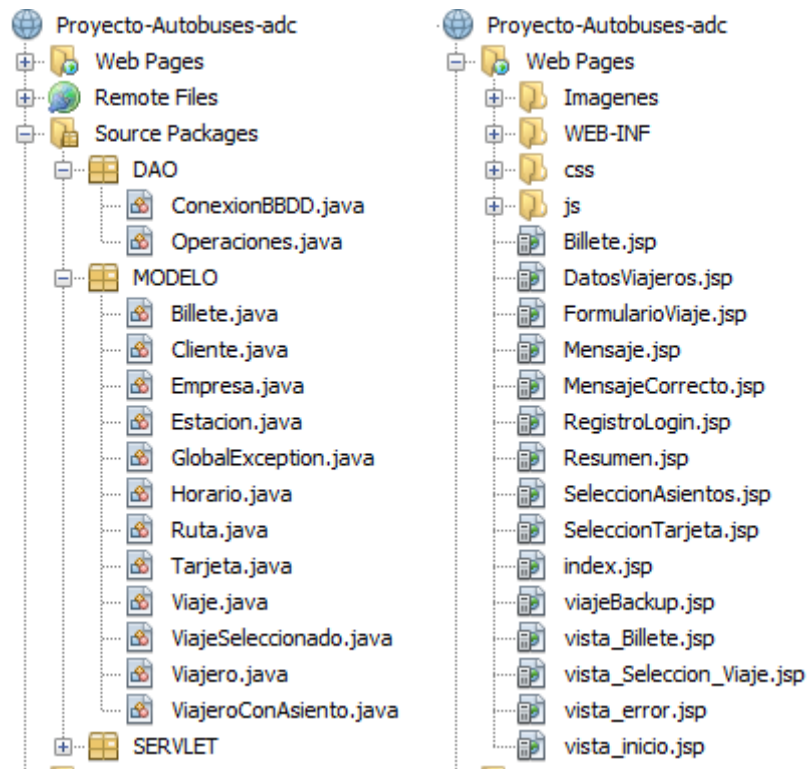
1.3. Estructura de carpetas

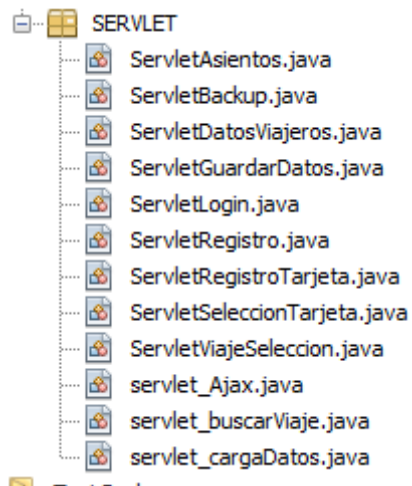
La aplicación esta desarrolla en JAVA Web respetando el modelo – vista – controlador (MVC) con esta estructura de carpeta y ficheros:

Carpetas principales:

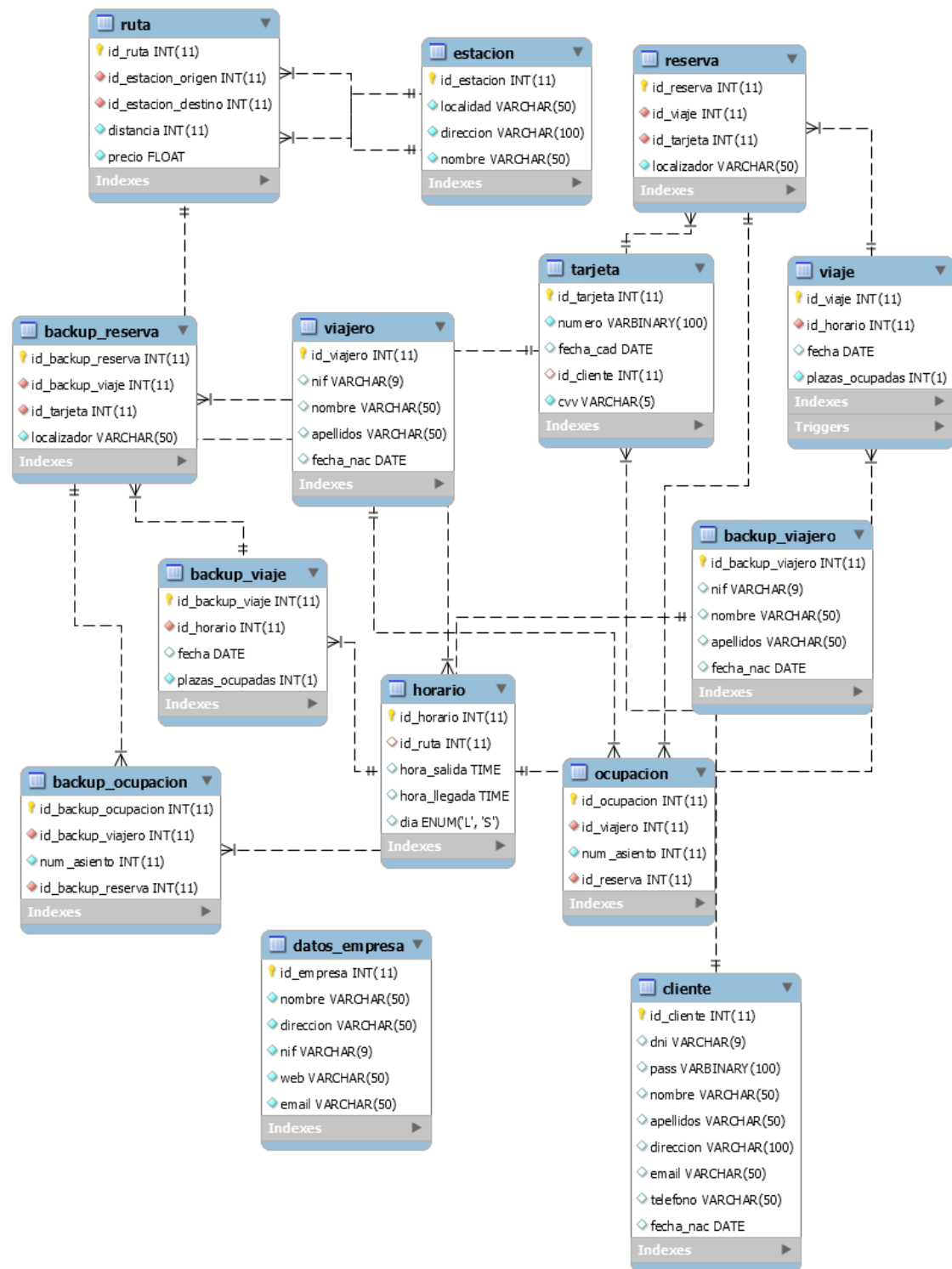
- Web: donde se encuentran todas las vistas (JSP) incluido el index.jsp
- Java->DAO: donde se encuentran la conexión a la base de datos (Implementación el uso del patrón Singleton).
- Java->MODELO: donde se encuentran las clases

- 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. Casos de uso

3.1.1. Buscar Viaje

Una vez completado el formulario, 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 y obtener un billete o para confirmar su llegada.

3.1.2. 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.

3.1.3. 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 sea para seleccionar una que ya tiene, como registrar una nueva tarjeta.

3.1.4. Proceso de inserción de todos los datos

Una vez comprobados los diferentes datos 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.

Por último confirmamos la llegada de un viaje.

3.2. 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.

3.2.1. Selección del Viaje que finaliza

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

3.2.2. 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
```

4. Datos usados para ejemplificar la aplicación

Para probar la aplicación esta ejemplificado para las siguientes fechas:

- Días: 15/06/2018 y 16/06/2018