Version 1.0

December 4, 2017

Integrantes: alejandra diaz  
ricardo carcassón  
docente: fernando sepulveda  
seccion: 006v



Sistema de Compra de Peajes

highway

INDICE

[Sistema de Compra de Peajes 2](#_Toc500167903)

[Introducción 2](#_Toc500167904)

[Descripción del Proyecto 3](#_Toc500167905)

[El Sistema 4](#_Toc500167906)

[Características 4](#_Toc500167907)

[Código Fuente Documentado 5](#_Toc500167908)

[Class Carretera 5](#_Toc500167909)

[Class Opcion 6](#_Toc500167910)

[Class Venta 7](#_Toc500167911)

[Class ctrlCarretera 8](#_Toc500167912)

[Class ctrlOpcion 9](#_Toc500167913)

[Class ctrlVenta 10](#_Toc500167914)

[Class srvProcesarPedido 13](#_Toc500167915)

[Class srvProcesarPedido 14](#_Toc500167916)

[Conclusión 15](#_Toc500167917)

[Bibliografía 16](#_Toc500167918)

# Sistema de Compra de Peajes

## Introducción

El presente informe tiene como finalidad el detallar de la manera más clara posible toda la información relacionada con el desarrollo de la aplicación solicitada por el cliente Highway, a modo de referencia para futuras consultas o mejoras que quieran implementarse al código del mismo. Como desarrolladores estamos muy contentos con el producto alcanzado para satisfacer las necesidades del cliente y esperamos que este sea el inicio para futuras mejoras para brindar un producto de calidad.

Durante el desarrollo de la aplicación fuimos aplicando cada uno de los conocimientos adquiridos a lo largo del tiempo, empleando métodos actuales de codificación en base a implementación MVC para un fácil entendimiento y manejo del código.

Para llevar a cabo este desarrollo utilizamos lenguaje de programación Java con base de datos MySQL bajo un entorno de desarrollo con Netbeans en un servidor GlassFish.

Alejandra Diaz

Ricardo Carcassón

## Descripción del Proyecto

La empresa concesionada Highway desea el desarrollo de un sistema para agilizar la compra de peajes por parte de empresas particulares y fiscales.

El proceso de compra consta de un formulario donde el comprador ingresara la información de la empresa, método de pago y de retiro de sus peajes comprados. El sistema presentará un listado con las cuatro carreteras que existen actualmente, en donde el cliente seleccionara la carretera deseada, ingresará la cantidad de peajes que desea comprar y los agregara al listado del formulario. El sistema le permitirá seguir ingresando otras carreteras más a la compra o modificar las ya existentes. Una vez procesada la compra el sistema desplegará la información del voucher de compra junto con el total a cancelar.

El sistema también permitirá buscar por rut todas las compras realizadas por un cliente para de esta manera generar una nueva compra en base a la información histórica con un nuevo voucher.

Adicionalmente el sistema tendrá la opción de mostrar un mapa con las carreteras cargadas en sistema para que el usuario sepa donde se encuentran ubicadas y cual es el tramo correspondiente de cada una.

El cliente ha solicitado una serie de requerimientos y validaciones al sistema para generar un resultado robusto, confiable y amigable para el usuario.

# El Sistema

## Características

El sistema posee una serie de características únicas en el mercador las cuales se detallan a continuación:

1. Validaciones en cada uno de sus campos para evitar errores de tipeo o de falta de información.
2. Manejo de errores.
3. Registro a través de un log de extensión TXT sobre todas las acciones que realiza el cliente en el sitio. Cada proceso que realiza el cliente es registrado con fecha y hora.
4. Link de Google Maps para visualizar ubicación de las carreteras.
5. Posibilidad de visualizar compras anteriores y en base a dichos resultados realizar una compra bajo un nuevo voucher sin tener que realizar todo el proceso de llenado de formulario.
6. Múltiples opciones de pago y de despacho de la compra.
7. Capacidad de poder modificar la cantidad de peajes a solicitar, eliminar o agregar más carreteras a la compra.
8. El sistema despliega los totales por cada transacción que realiza el cliente.

## Código Fuente Documentado

cl.duoc.modelo

### Class Carretera

java.lang.Object

cl.duoc.modelo.Carretera

public class **Carretera**

extends java.lang.Object

Esta clase gestionará la información de la carretera. Usará tres propiedades: id de tipo int para el identificador unico de la carretera, nombre de tipo string para el nombre de la carretera y valor de tipo int para el precio del peaje asociado de dicha carretera.

cl.duoc.modelo

### Class Opcion

java.lang.Object

cl.duoc.modelo.Opcion

public class **Opcion**

extends java.lang.Object

Esta clase se encargará de la gestión de la información de las opciones seleccionadas por el usuario con cuatro propiedades. ID de tipo int para el identificador unico de la opción dentro de la base de datos, idventa de tipo int que indica el número de venta al que está asociada la opción, pago de tipo int para indicar el tipo de pago que selecciono el usuario y retiro de tipo int para indicar la opción de retiro seleccionada por el usuario en el proceso de compra.

cl.duoc.modelo

### Class Venta

java.lang.Object

cl.duoc.modelo.Venta

public class **Venta**

extends java.lang.Object

Clase que gestionará la información de la venta realizada en el sistema. Posee 5 propiedades: id de tipo int que será el identificador único dentro la base de datos. Rut de tipo string para el rut de la empresa, carretera de tipo string que guardara el nombre de la carretera, cantidad de tipo int que indica la cantidad de peajes ingresados a comprar para dicha carretera por parte del cliente y total de tipo int para registrar el total de la compra en base al valor \* con la cantidad de peajes solicitados por el cliente.

cl.duoc.controlador

### Class ctrlCarretera

java.lang.Object

cl.duoc.controlador.ctrlCarretera

public class **ctrlCarretera**

extends java.lang.Object

#### Method Detail

#### ObtenerCarreteras

public java.util.ArrayList<Carretera> ObtenerCarreteras()

Este método obtiene las carreteras que se encuentran cargadas en el sistema junto con sus valores.

**Returns:**

un ArrayList del tipo Carretera con la información.

#### ObtenerCarretera

public Carretera ObtenerCarretera(int id)

Este método retorna una carretera en específico dependiendo del id indicado y retornara un objeto Carretera con la información del nombre y del valor de la carretera cuyo id coincida con el indicado.

**Parameters:**

id - de tipo entero, representa el id único de la carretera a buscar

**Returns:**

un objeto de tipo Carretera con la información

#### ObtenerValor

public int ObtenerValor(int id)

Este método permite obtener el valor de una carretera en base al id ingresado.

**Parameters:**

id - de tipo entero, representa el id único de la carretera

**Returns:**

un valor de tipo int que representa el valor de la carretera a buscar

cl.duoc.controlador

### Class ctrlOpcion

java.lang.Object

cl.duoc.controlador.ctrlOpcion

public class **ctrlOpcion**

extends java.lang.Object

#### Method Detail

#### GuardarOpciones

public boolean GuardarOpciones(Opcion o)

Este método permite guardar las opciones de forma de pago y de retiro de los peajes comprados por el cliente. El método guardar el id de la venta junto con las opciones previamente indicadas.

**Parameters:**

o - De tipo clase Opcion con los valores seleccionados por el cliente

**Returns:**

boolean para indicar si se guardó de manera correcta la información

cl.duoc.controlador

### Class ctrlVenta

java.lang.Object

cl.duoc.controlador.ctrlVenta

public class **ctrlVenta**

extends java.lang.Object

#### Method Detail

#### ObtenerNuevoID

public int ObtenerNuevoID()

Método que permite obtener un nuevo id de venta. El id es generado en base a la cantidad de registros que existen en la tabla de ventas, considerando solo aquellos IDs que no se repiten.

**Returns:**

un int con el total de registros que coincidan con la consulta.

#### GuardarVenta

public boolean GuardarVenta(int idVenta,

java.util.ArrayList<Venta> v)

Método que registra la venta cuando el cliente procesa el pedido. El método recibe dos parámetros: idVenta, que posee el número de id de venta asignado y v que posee la información de las carreteras seleccionadas en el proceso de pedido junto con las cantidads y totales correspondientes. Adicionalmente el método guardará en el log del sistema la acción realizada o el error que ocurrio si este fuese el caso.

**Parameters:**

idVenta - De tipo int con el número de la venta

v - De tipo ArrayList Venta que posee la información de las carreteras seleccionadas en el proceso de compra con sus respectivos valores

**Returns:**

Un boolean para indicar que el proceso se registro de manera correcta.

#### ListarUsuarios

public java.util.ArrayList<Venta> ListarUsuarios()

Este método obtiene la lista de usuarios únicos que han realizado compras en el sistema. Permite conocer el rut de los usuarios que han realizado compras anteriores.

**Returns:**

un ArrayList del tipo Venta con la información del rut de los usuarios.

#### BuscarDetallePorUsuario

public java.util.ArrayList<Venta> BuscarDetallePorUsuario(java.lang.String rut)

Este método obtiene la lista de ventas por usuario, filtrado por rut. Permite obtener información histórica de las ventas por usuario.

**Parameters:**

rut - De tipo string, recibe el rut del usuario a consultar.

**Returns:**

un ArrayList del tipo Venta con la información del idventa, rut, carretera, cantidad, total del rut consultado.

#### BuscarPedidoPorUsuario

public java.util.ArrayList<Venta> BuscarPedidoPorUsuario(java.lang.String rut)

Este método obtiene la lista de ventas por usuario, filtrado por rut. Informa el valor total a pagar de cada venta anterior.

**Parameters:**

rut - De tipo string, recibe el rut del usuario a consultar.

**Returns:**

un ArrayList del tipo Venta con la información del idventa, rut y el valor total a pagar por cada venta.

#### BuscarVentas

public java.util.ArrayList<Venta> BuscarVentas(int idventa)

Este método obtiene la lista de ventas, filtradas por id. Permite conocer los detalles de la venta anterior del usuario.

**Parameters:**

idventa - De tipo int, recibe el id de venta a consultar.

**Returns:**

un ArrayList del tipo Venta con la información del idventa, rut, carretera, cantidad y total filtrado por idventa.

#### BuscarOpcion

public Opcion BuscarOpcion(int idventa)

Este método informa la opcion de pago y retiro seleccionada por el usuario, filtrado id de venta. Este método permite conocer la opción seleccionada por el usuario en compras anteriores.

**Parameters:**

idventa - De tipo int, recibe el id de venta a consultar.

**Returns:**

un objeto de tipo opcion, con la informacion del idventa, pago y retiro filtrado por idventa.

cl.duoc.servlet

### Class srvProcesarPedido

java.lang.Object

javax.servlet.GenericServlet

javax.servlet.http.HttpServlet

cl.duoc.servlet.srvProcesarPedido

**All Implemented Interfaces:**

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

#### Method Detail

#### processRequest

protected void processRequest(javax.servlet.http.HttpServletRequest request,

javax.servlet.http.HttpServletResponse response)

throws javax.servlet.ServletException,

java.io.IOException

Este servlet se encarga de procesar cada una de las solicitudes que se hagan desde la página de compras del sitio, ya sea agregar una carretera al pedido, modificar una cantidad de peajes de una carretera ingresada, eliminar una carretera en específico, procesar el pedido completo o reiniciar el formulario para una nueva compra en caso de errores a la hora de ingresar los datos del cliente en el formulario.

**Parameters:**

request - servlet request

response - servlet response

**Throws:**

javax.servlet.ServletException - if a servlet-specific error occurs

java.io.IOException - if an I/O error occurs

cl.duoc.servlet

### Class srvProcesarPedido

java.lang.Object

javax.servlet.GenericServlet

javax.servlet.http.HttpServlet

cl.duoc.servlet.srvProcesarPedido

**All Implemented Interfaces:**

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

#### Method Detail

#### processRequest

protected void processRequest(javax.servlet.http.HttpServletRequest request,

javax.servlet.http.HttpServletResponse response)

throws javax.servlet.ServletException,

java.io.IOException

Este servlet se encarga de procesar cada una de las solicitudes que se hagan desde la página de compras del sitio, ya sea agregar una carretera al pedido, modificar una cantidad de peajes de una carretera ingresada, eliminar una carretera en específico, procesar el pedido completo o reiniciar el formulario para una nueva compra en caso de errores a la hora de ingresar los datos del cliente en el formulario.

**Parameters:**

request - servlet request

response - servlet response

**Throws:**

javax.servlet.ServletException - if a servlet-specific error occurs

java.io.IOException - if an I/O error occurs

# Conclusión

Llevar a cabo el desarrollo de esta aplicación nos ha enriquecido de una manera excepcional. Hemos logrado cumplir a cabalidad con los requerimientos del cliente, obteniendo como resultado un sistema que se adapta a las necesidades diarias de su uso, con una interfaz amigable y acorde a los más altos standares de diseño.

Esperamos que el sistema sea del total agrado, estamos felices de que nos hayan elegido para dar vida a tan importante proyecto. Hemos adquirido nuevos conocimientos y hemos pulido nuestras técnicas de desarrollo con nuevas implementaciones como la opción de visualizar las carreteras mediante un mapa.

Gracias a la potencia de Java, Netbeans y de MySQL hemos logrado un resultado funcional y robusto en muy poco tiempo y eso nos da la confianza para seguir experimentado y desarrollando nuevas herramientas bajo estos potentes entornos.

Alejandra Diaz

Ricardo Carcassón

# Bibliografía

Bootstrap - <https://getbootstrap.com/docs/3.3/>

Layoutit! - <http://www.layoutit.com/build>

Como documentar Código - <https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=646:documentar-proyectos-java-con-javadoc-comentarios-simbolos-tags-deprecated-param-etc-cu00680b&catid=68&Itemid=188>

Implementación de Mapas - <https://www.google.com/maps/d/u/0/>

NetBeans - <https://netbeans.org/>

Java SDK - <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

MySQL - <https://dev.mysql.com/>