MANUAL TÉCNICO

Funcionamiento técnico de SI Urbanos

Elaborado por: Germán Andrés Piñeros M

Diego Fernando Pedraza G

Fabián Romero C

Iván Esteban Ballesteros

Oscar Suarez V

Andrés Tuta L

**ARQUITECRURA DE SOFTWARE – Grupo SIUrbanos**

**Marzo de 2015**

CONTENIDO

[INTRODUCCIÓN 2](#_Toc412903668)

[HERRAMIENTAS USADAS 3](#_Toc412903669)

[DIAGRAMA DE CLASES 4](#_Toc412903670)

[DIAGRAMA DE ARQUITECTURA (DE CAPAS) 5](#_Toc412903671)

# INTRODUCCIÓN

**SI Urbanos** es un sistema de información para la gestión de la central de buses urbanos en la capital, el propósito del sistema es ayudar a los usuarios de buses urbanos a conseguir sus tarjetas de una manera más sencilla, permitiendo a su vez realizar distintas acciones sobre la tarjeta y cuenta del usuario, tales como recargar, ver historial de viajes y ver historial de recargas.

Este manual técnico contempla todo lo relacionado a la elaboración del sistema de información, mostrando y explicando los respectivos diagramas que permiten evidenciar el buen uso y práctica de la arquitectura de software.

# HERRAMIENTAS USADAS

Para la elaboración del sistema se contó con las siguientes herramientas de trabajo en cada uno de los procesos:

La realización de la aplicación como tal, se empleó en el entorno de desarrollo Netbeans 8.0.2, que provee una interfaz de usuario amigable y de fácil manejo, que además de esto se acopla muy bien al resto de herramientas utilizadas, el lenguaje utilizado fue Groovy con su framework para desarrollo de páginas web Grails, que son sencillos de usar y muy intuitivos. Grails emplea una base de datos por defecto que es la utilizada es el desarrollo de la aplicación.

Para el desarrollo de las vistas web, se emplearon plantillas de Bootstrap y las que proveía por defecto Grails, aunque estas últimas solo son visibles para el rol de administrador.

 





# DIAGRAMA DE CLASES

Dentro de SI Urbanos hay 4 clases entidad que ayudará a manejar la persistencia de datos del sistema, estas clases son: Card, User, Recharge y Route. Se tiene un controlador para cada uno de ellas, es decir: userController, routecontroller, rechargeController y cardController. La clase userController, controla todas las posibilidades de un usuario dentro del sistema, ya sea un simple cliente o un administrador. La clase cardController permite la creación de una tarjeta para un usuario en específico, así como poder recargarla. La clase routeController maneja lo necesario para la creación de un viaje y además permite ver el historial de viajes. Por último, la clase rechargeController que maneja el historial de recargas.

Las vistas que se tienen y que pueden ser vistas en el manual de usuario son las de “index2” que es la vista general del sistema. Se tiene una vista de “menú” que contiene la vista principal para un usuario que está en sesión. Y también se tiene vistas específicas como “recarga”, “historialRecargas” e “historialViajes” que permiten la funcionalidad dentro del sistema. Los layouts con los que se cuentan son “login\_bar” barra que muestra si se está o no en sesión y “menu\_bar” que despliega un pequeño para navegar dentro del sistema.

El siguiente diagrama muestra mejor las clases dentro del sistema y su respectiva interacción:



Imagen 1. Diagrama de clases de SI Urbanos.

La siguiente tabla muestra una mejor descripción de las entidades y clases del sistema:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entidad | Atributos | Controlador | Vistas Asociadas | Relación con otras entidades |
| User | String idUser  String userName  String password  String rol | userController: Contiene las operaciones CRUD para un usuario en el sistema. | Index2: Primera vista del sistema.  Delete, show, edit, index: vistas solo permitidas para el administrador, se crean por debajo gracias a Grails. | Uno a uno con Card. |
| Card | String idCard  double amount  double valorPasaje | cardController: Contiene las operaciones CRUD para una card en el sistema. Permite recargar una tarjeta con una monto cualquiera. | Comprar: permite que el usuario adquiera una tarjeta.  Recarga: permite al usuario recargar la tarjeta.  Menú: Menú principal.  Delete, show, edit, index: vistas solo permitidas para el administrador, se crean por debajo gracias a Grails. | Uno a uno con usuario.  Uno a muchos con Recharge.  Uno a muchos con Route. |
| Route | String routeName  Date date | routeController: Contiene las operaciones CRUD para un viaje dentro del sistema. Permite la visualización del historial de viajes del usuario en el sistema. | historialRutas: Permite ver el historial de rutas del usuario en el sistema.  Menú: Menú principal.  Delete, show, edit, index: vistas solo permitidas para el administrador, se crean por debajo gracias a Grails. | De muchos a uno con card. |
| Recharge | Date dateRecharge double amountRecharge | rechargeController: Contiene las operación CRUD para un recarga en el sistema. Permite la visibilidad del historial de recargas. | historialRecarga: Permite ver el historial de recargas del usuario en el sistema.  Menú: Menú principal.  Delete, show, edit, index: vistas solo permitidas para el administrador, se crean por debajo gracias a Grails. | De muchos a uno con card. |

Tabla 1. Explicación de las clases de SI Urbanos.

# DIAGRAMA DE ARQUITECTURA (DE CAPAS)

El diagrama de capas trata de ejemplificar el modelo de arquitectura implementado para el desarrollo de SI Urbano. Dado que Groovy and Grails facilita la creación de un modelo vista controlador, este será el modelo usado en la creación de nuestra aplicación. En la Tabla 1 se pueden ver todas las entidades, controladores y vistas que se poseen en el sistema SI Urbanos. El siguiente diagrama trata de dar un mejor acercamiento de cómo se maneja el estilo de capas dentro de nuestro sistema.



Imagen 2. Diagrama de capas de SI Urbanos.

# DIAGRAMA DE MODULOS

Este diagrama muestra los módulos del sistema que son completamente funcionales y las vistas que tiene para cumplir con la funcionalidad pedida.



Imagen 3. Diagrama de modelos de SI Urbanos.