

Documentación Proyecto

Profesional responsable: Martin Hildebrandt

Institución: Universidad Nacional Lanús

Área de conocimiento: Ingeniería Software 3

Denominación del Equipo: Grupo 6

Capacitación: Licenciatura en Sistemas

Proyecto a Analizar: Proyecto de Software SUBE

[Documentación Tablas y relaciones entre Tablas 6](#_bt9encp6miz9)

[Estación 6](#_7qaoqy6os2ap)

[Medio Transporte 6](#_67une6rc5hvh)

[Recarga 6](#_n4are4lfvcxl)

[Tarjeta 7](#_owicvhpv0cvc)

[Transacción 7](#_c7kad4m7uh4z)

[Usuario 8](#_quzgthqm6mfr)

[Viaje 8](#_ncjv1d6wlro)

[Documentación componentes software backend 10](#_ukvr9kw4aj9q)

[Controllers 10](#_597w5jtlr9hx)

[Global Advice Controller 10](#_d2jm5hv6hsqg)

[Home Controller 10](#_pzmbak1lh41n)

[Estación Controller 10](#_grmlmwt67zz0)

[Information Controller 10](#_qdti6dgm0xry)

[Login Controller 10](#_d6j1fnsquswl)

[Medio Transporte Controller 10](#_bu218evj5ioe)

[Recarga Controller 10](#_rm474av1eogg)

[Tarjeta Controller 11](#_jrp7kmk0oge)

[Transacciones Controller 11](#_53whry1od9vd)

[Usuario Controller 11](#_cmf8y1k3jnud)

[Viaje Controller 11](#_wa9cntr8gsge)

[DTOs 11](#_flnleg40txci)

[Estación DTO 11](#_mouyhadmmw9q)

[Medio Transporte DTO 11](#_sdo96eakmhfz)

[Recarga DTO 11](#_9lwgce9ydism)

[Recarga Form DTO 12](#_55vn181uz1hh)

[Tarjeta DTO 12](#_zc8yi52402qf)

[Tarjeta Form DTO 12](#_rc6ouny00b9y)

[Tarjeta Form Comprar DTO 12](#_r5omtkm6kupt)

[Transacción DTO 12](#_wgc6bd6dc6i9)

[Usuario DTO 12](#_4mgqe6hhn3vf)

[Viaje DTO 12](#_j1yunha7hhg7)

[Entity 13](#_eiavvex0qgzr)

[Estación 13](#_9qr6ldo5mf0s)

[Medio Transporte 13](#_fwhom9p10d60)

[Recarga 13](#_7631qmcql6hg)

[Tarjeta 13](#_xot23lyc79n5)

[Transacción 13](#_uo6pibtfgbap)

[Usuario 13](#_lg9pt4iqugp4)

[Viaje 13](#_tu3gdtbpqmru)

[Repository 14](#_hs4hxs69hvvi)

[Estación Repository 14](#_lhvhbq52s39p)

[Medio Transporte Repository 14](#_5xf9pr4pbxm)

[Recarga Repository 14](#_czogzw9xhz26)

[Tarjeta Repository 14](#_267dymi14wvk)

[Transaccion Repository 14](#_89zu8efh93sk)

[Usuario Repository 14](#_u83m5ym9wuy9)

[Viaje Repository 14](#_yxts6x7fxxvt)

[Service 15](#_6oyb7pbqa1e2)

[Estación Service 15](#_e92zu1ifgtan)

[Medio Transporte Service 15](#_2i1vgr20ub7k)

[Recarga Service 15](#_kbp2wls6adbm)

[Tarjeta Service 15](#_tagrn5s5scqd)

[Transacción Service 15](#_57sax3fcj6eg)

[Usuario Service 15](#_tmxkh1xa054j)

[Viaje Service 15](#_qp7hc1hwkm4s)

[Service Implementation 16](#_opca2kybk5n7)

[Estación Service Implementation 16](#_fjxbh4dle6db)

[Medio Transporte Service Implementation 16](#_j70540gnltz2)

[Recarga Service Implementation 16](#_9jrzyl29k30i)

[Tarjeta Service Implementation 16](#_zcbbgxfudble)

[Transaccion Service Implementation 16](#_6myymwa4bpmh)

[Usuario Service Implementation 16](#_e06mig614418)

[Viaje Service Implementation 16](#_h53ivjtgni09)

[Testing 17](#_1joxahli08q2)

[Estación Test 17](#_u70v4pn4tvzh)

[Medio Transporte Test 17](#_720112r94n4e)

[Recarga Test 17](#_hk4rn7iybskw)

[Tarjeta Test 17](#_k6qbz0ri8io4)

[Test Runner 17](#_ralnu01t39sk)

[Transacción Test 17](#_e9t1p0vzpiwb)

[Usuario Test 17](#_ppgmorw0fmqu)

[Viaje Test 17](#_6h2496tca4ox)

[Documentación componentes software web 18](#_yj2whhhzswz)

[Static 18](#_44sxt6wlvjs)

[Css 18](#_d94vu99wp20k)

[Conseguila-style.css 18](#_d5jcib33nzue)

[Estaciones-style.css 18](#_sseltochney0)

[General-style.css 18](#_h4n72thn8x4r)

[Index-style.css 18](#_ks1t6231q04c)

[Login-style.css 18](#_a9z169p2qwq)

[Medios-transporte-style.css 19](#_di7f4ljeve31)

[Recargas-style.css 19](#_nse0xmdw5xns)

[Register-style.css 19](#_2kw71i6529bu)

[Tarjetas-style.css 19](#_hvoudhnl7co5)

[Transacciones-style.css 19](#_ton538e2npx5)

[Usuarios-style.css 19](#_2lg061wesxa5)

[Viajes-style.css 19](#_11rtl3ic6tfw)

[Images 20](#_druzqnq0yae7)

[Js 20](#_amybpjfxjqlg)

[Templates 20](#_fn3crersnoql)

[Estaciones 20](#_4vyh1ulyypuh)

[● Formulario.html: formulario para crear una estación.. 20](#_lc555wx3djzt)

[● Lista.html: muestra un listado de todas las estaciones. 20](#_9rr2vfvk9zk8)

[● Detalles.html: vista de detalle de una sola estación. 20](#_i35xelvofpu7)

[● Borrar.html: confirma eliminación de la estación. 20](#_56asdywkc2e)

[Fragments 20](#_snw1asq4veem)

[● Head.html: define los importes de los .css, el título, etc. 20](#_k5udjlarxm2p)

[● Navbar.html: barra de navegación principal, con enlaces URL correspondientes. 20](#_k5udjlarxm2p)

[● Footer.html: pie de página compartido derechos de autor. 20](#_k5udjlarxm2p)

[● Contacto.html: sección de información de contacto, número de teléfono para soporte. 20](#_7ilc2brqe5ju)

[Información 20](#_s3koe1eog5ff)

[● Conseguila.html: página acerca del procedimiento para obtener una tarjeta SUBE. 21](#_6trl523fxk8y)

[Login 21](#_zfwfuw6k28ml)

[● Login.html: formulario de acceso a la página. 21](#_vbrc8cjww5ib)

[● Register.html: formulario para ingresar como un usuario nuevo. 21](#_75r9vmjrruq5)

[Medios-transporte 21](#_dvcy47peg51)

[● Formulario.html: formulario para crear un medio de transporte. 21](#_a8gmbycai3z2)

[● Lista.html: muestra un listado de todas los medios de transporte 21](#_44qqzhgbk9z)

[● Detalles.html: vista de detalle de un solo medio de transporte . 21](#_b6psgnmcwd00)

[● Borrar.html: confirma eliminación del medio de transporte. 21](#_azwkhw7qt33k)

[Recargas 21](#_dyuvyh3l7408)

[● Formulario.html: formulario para crear una recarga. 21](#_pwdsmu3uhpai)

[● Lista.html: muestra un listado de todas las recargas. 21](#_n5prht67s6j5)

[● Detalles.html: vista de detalle de una sola recarga. 21](#_rbarsoifktta)

[● Borrar.html: confirma eliminación de la recarga. 21](#_6bhdlnqom6tz)

[Tarjetas 21](#_uwgv9kn1wstf)

[● Formulario.html: formulario para crear una tarjeta. 22](#_m7ox2l9rqalt)

[● Lista.html: muestra un listado de todas las tarjetas. 22](#_c9numgaxh2n7)

[● Detalles.html: vista de detalle de una sola tarjeta. 22](#_59ij4588o7mr)

[● Borrar.html: confirma eliminación de la tarjeta. 22](#_p53l97hghks)

[● Formulario-comprar.html: procedimiento alternativo de compra de tarjeta por medio de un link y condiciones específicas en torno a datos del usuario solicitados. 22](#_ruvu4gk00s98)

[● Buscar.html / Buscar-tarjeta.html: formularios para búsqueda de tarjetas por código o documento del usuario. 22](#_ruvu4gk00s98)

[Transacciones 22](#_qof7yps4buyo)

[● Formulario.html: formulario para crear una transacción. 22](#_l0kc0dxnnhp6)

[● Lista.html: muestra un listado de todas las transacciones. 22](#_8gwiemywcc44)

[● Detalles.html: vista de detalle de una sola transacción. 22](#_74lp4zkcljcu)

[● Borrar.html: confirma eliminación de la transacción. 22](#_kl7ubnq0d7fd)

[Usuarios 22](#_fv8i50bmx4s9)

[● Formulario.html: formulario para crear una usuario. 22](#_s6lebxga30ig)

[● Lista.html: muestra un listado de todas los usuarios. 22](#_kud3z29yz2mg)

[● Detalles.html: vista de detalle de una sola usuario. 22](#_foi6dybo0fr6)

[● Borrar.html: confirma eliminación del usuario. 22](#_qltlv0eiyg4j)

[Viajes 22](#_wcxqy5uumafk)

[● Formulario.html: formulario para crear una viaje. 23](#_1po5ipfemexy)

[● Lista.html: muestra un listado de todas los viajes. 23](#_71puljm0yxnq)

[● Detalles.html: vista de detalle de una sola viaje. 23](#_e4ku67dk0fjx)

[● Borrar.html: confirma eliminación del viaje. 23](#_v0flpp93eimz)

[Index 23](#_ajal64gzsioq)

[Documentación Estándares de Seguridad 24](#_mx5bee543p7n)

# Documentación Tablas y relaciones entre Tablas

## Estación

Id int

Nombre varchar(255)

Ubicación varchar(255)

Medio\_id int

Guarda estaciones de origen/destino con id, nombre, ubicación, y vínculo a un medio de transporte.

## Medio Transporte

Id int

Tipo varchar(255)

Nombre varchar(255)

Lista los tipos de transporte (colectivo, tren, subte…) con id, tipo y nombre.

## Recarga

Id int

Fecha Hora Datetime(6)

Monto double

Tarjeta id int

Registra cada operación de recarga de saldo: fechaHora, monto, y la tarjeta asociada.

## Tarjeta

Id int

Codigo varchar(255)

Fecha Emisión Datetime(6)

Saldo double

Estado varchar(255)

Usuario id int

Almacena datos de cada tarjeta SUBE: código, fechaEmisión, saldo, estado y propietario.

## Transacción

Id int

Importe double

Tipo varchar(255)

Fecha Hora Datetime(6)

Recarga id int

Viaje id int

Tarjeta id int

Historial de movimientos (recarga o viaje), con importe, tipo, fecha y referencias a recarga, viaje y tarjeta.

## Usuario

Id int

Nombre varchar(255)

Apellido varchar(255)

Email varchar(255)

Contrasenia varchar(255)

Fecha Alta Datetime(6)

Rol varchar(255)

Dni int

Tipo dni varchar(255)

Genero varchar(255)

Telefono varchar(255)

Datos de los pasajeros: nombre, apellido, email, contraseña, fechaAlta, rol, documento, tipo documento, género y teléfono.

## Viaje

Id int

Fecha Hora Datetime(6)

Estación origen id int

Estación destino id int

Tarjeta id int

Usuario id int

Medio transporte id int

Cada uso de tarjeta en transporte: fechaHora, estaciones de origen/destino, usuario, tarjeta y medio de transporte.

# 

# Documentación componentes software backend

## Controllers

### Global Advice Controller

Gestiona roles de usuario correspondientes para habilitar acceso a determinadas URL.

### Home Controller

Gestiona la ruta raíz ("/") sin necesidad de autenticación, proporcionando por normativa general el archivo index.html.

### Estación Controller

Ofrece los endpoints CRUD para las estaciones de transporte: listar, crear, editar, ver detalles y borrar estaciones..

### Information Controller

Proporciona rutas para mostrar información de soporte, en nuestra aplicación solo se aplica con “conseguila.html”.

### Login Controller

Gestiona el login a la aplicación y el registro de nuevos usuarios.

### Medio Transporte Controller

Ofrece los endpoints CRUD para las estaciones de transporte: listar, crear, editar, ver detalles y borrar medios de transporte..

### Recarga Controller

Ofrece los endpoints CRUD para las estaciones de transporte: listar, crear, editar, ver detalles y borrar recargas.

### Tarjeta Controller

Ofrece los endpoints CRUD para las estaciones de transporte: listar, crear, editar, ver detalles y borrar tarjetas.

### Transacciones Controller

Ofrece los endpoints CRUD para las estaciones de transporte: listar, crear, editar, ver detalles y borrar transacciones.

### Usuario Controller

Ofrece los endpoints CRUD para las estaciones de transporte: listar, crear, editar, ver detalles y borrar usuarios.

### Viaje Controller

Ofrece los endpoints CRUD para las estaciones de transporte: listar, crear, editar, ver detalles y borrar viajes.

## DTOs

### Estación DTO

Data Transfer Object de una estación de transporte, utilizada para optimizar el funcionamiento de la aplicación y mejorar la eficiencia del flujo de la información necesaria.

### Medio Transporte DTO

Data Transfer Object de un medio de transporte, utilizada para optimizar el funcionamiento de la aplicación y mejorar la eficiencia del flujo de la información necesaria.

### Recarga DTO

Data Transfer Object de una recarga, utilizada para optimizar el funcionamiento de la aplicación y mejorar la eficiencia del flujo de la información necesaria.

### Recarga Form DTO

Data Transfer Object de una estación de transporte, utilizada para optimizar el funcionamiento de la aplicación y mejorar la eficiencia del flujo de la información necesaria. Una especificación utilizada para incorporar detalles específicos del usuario, se utiliza para prevenir traer todos los datos del usuario en conjunto.

### Tarjeta DTO

Data Transfer Object de una tarjeta, utilizada para optimizar el funcionamiento de la aplicación y mejorar la eficiencia del flujo de la información necesaria.

### Tarjeta Form DTO

Data Transfer Object de una tarjeta, utilizada para optimizar el funcionamiento de la aplicación y mejorar la eficiencia del flujo de la información necesaria. Una especificación utilizada para incorporar detalles específicos del usuario, se utiliza para prevenir traer todos los datos del usuario en conjunto.

### Tarjeta Form Comprar DTO

Data Transfer Object de una tarjeta, utilizada para optimizar el funcionamiento de la aplicación y mejorar la eficiencia del flujo de la información necesaria. Una especificación utilizada para incorporar detalles específicos del usuario, se utiliza para prevenir traer todos los datos del usuario en conjunto, en este caso se utiliza aún mayor información que el caso anterior, esta separación aumenta la modularización y facilita el mantenimiento..

### Transacción DTO

Data Transfer Object de una transacción, utilizada para optimizar el funcionamiento de la aplicación y mejorar la eficiencia del flujo de la información necesaria.

### Usuario DTO

Data Transfer Object de un usuario, utilizada para optimizar el funcionamiento de la aplicación y mejorar la eficiencia del flujo de la información necesaria.

### Viaje DTO

Data Transfer Object de un viaje, utilizada para optimizar el funcionamiento de la aplicación y mejorar la eficiencia del flujo de la información necesaria.

## Entity

### Estación

Clase Java que mapea la tabla Estación, con atributos como id, nombre, ubicación y la clave foránea al medio de transporte.

### Medio Transporte

Clase Java que mapea la tabla Medio Transporte, con atributos como id, tipo, nombre.

### Recarga

Clase Java que mapea la tabla Recarga, con atributos como id, fechaHora, monto, y clave foránea a la tarjeta asociada.

### Tarjeta

Clase Java que mapea la tabla Tarjeta, con atributos como id, codigo, fechaEmision, saldo, estado, y clave foránea a usuario propietario.

### Transacción

Clase Java que mapea la tabla Transacción, con atributos como id, importe, tipo, fechaHora y clave foránea a recarga, viaje y tarjeta.

### Usuario

Clase Java que mapea la tabla Usuario, con atributos como id, nombre, apellido, fechaAlta, email, rol, dni, tipoDni, genero, teléfono.

### Viaje

Clase Java que mapea la tabla Viaje, con atributos como id, fechaHora, y clave foránea a estación origen. estación destino, medio de transporte, tarjeta y usuario..

## Repository

### Estación Repository

Interfaz que extiende JpaRepository<Estación, Long>, encargada de las operaciones de persistencia (crear, leer, actualizar, borrar) sobre la entidad Estación.

### Medio Transporte Repository

Interfaz que extiende JpaRepository<MedioTransporte, Long>, encargada de las operaciones de persistencia (crear, leer, actualizar, borrar) sobre la entidad Medio Transporte.

### Recarga Repository

Interfaz que extiende JpaRepository<Recarga, Long>, encargada de las operaciones de persistencia (crear, leer, actualizar, borrar) sobre la entidad Recarga.

### Tarjeta Repository

Interfaz que extiende JpaRepository<Tarjeta, Long>, encargada de las operaciones de persistencia (crear, leer, actualizar, borrar) sobre la entidad Tarjeta.

### Transaccion Repository

Interfaz que extiende JpaRepository<Transaccion, Long>, encargada de las operaciones de persistencia (crear, leer, actualizar, borrar) sobre la entidad Transaccion.

### Usuario Repository

Interfaz que extiende JpaRepository<Usuario, Long>, encargada de las operaciones de persistencia (crear, leer, actualizar, borrar) sobre la entidad Usuario.

### Viaje Repository

Interfaz que extiende JpaRepository<Viaje, Long>, encargada de las operaciones de persistencia (crear, leer, actualizar, borrar) sobre la entidad Viaje.

## Service

### Estación Service

Define las operaciones de negocio como crear, borrar, listar todos, obtener por ID, relacionadas a la clase Estación.

### Medio Transporte Service

Define las operaciones de negocio como crear, borrar, listar todos, obtener por ID, relacionadas a la clase Medio Transporte.

### Recarga Service

Define las operaciones de negocio como crear, borrar, listar todos, obtener por ID, relacionadas a la clase Recarga.

### Tarjeta Service

Define las operaciones de negocio como crear, borrar, listar todos, obtener por ID, relacionadas a la clase Tarjeta.

### Transacción Service

Define las operaciones de negocio como crear, borrar, listar todos, obtener por ID, relacionadas a la clase Transaccion.

### Usuario Service

Define las operaciones de negocio como crear, borrar, listar todos, obtener por ID, relacionadas a la clase Usuario.

### Viaje Service

Define las operaciones de negocio como crear, borrar, listar todos, obtener por ID, relacionadas a la clase Viaje.

## Service Implementation

### Estación Service Implementation

Implementación de los métodos de operaciones de negocio definidos previamente en la interfaz de Estacion Service.

### Medio Transporte Service Implementation

Implementación de los métodos de operaciones de negocio definidos previamente en la interfaz de Medio Transporte Service.

### Recarga Service Implementation

Implementación de los métodos de operaciones de negocio definidos previamente en la interfaz de Recarga Service.

### Tarjeta Service Implementation

Implementación de los métodos de operaciones de negocio definidos previamente en la interfaz de Tarjeta Service.

### Transaccion Service Implementation

Implementación de los métodos de operaciones de negocio definidos previamente en la interfaz de Transaccion Service.

### Usuario Service Implementation

Implementación de los métodos de operaciones de negocio definidos previamente en la interfaz de Usuario Service.

### Viaje Service Implementation

Implementación de los métodos de operaciones de negocio definidos previamente en la interfaz de Viaje Service.

## Testing

### Estación Test

Clase de prueba que se encarga de validar la lógica y los endpoints de la entidad Estación.

### Medio Transporte Test

Clase de prueba que se encarga de validar la lógica y los endpoints de la entidad Medio Transporte.

### Recarga Test

Clase de prueba que se encarga de validar la lógica y los endpoints de la entidad Recarga.

### Tarjeta Test

Clase de prueba que se encarga de validar la lógica y los endpoints de la entidad Tarjeta.

### Test Runner

Clase de prueba que se encarga de validar la lógica y los endpoints de todas las entidades Test del paquete, listando todas aquellas que fallen imprimiendo por consola las fallas.

### Transacción Test

Clase de prueba que se encarga de validar la lógica y los endpoints de la entidad Transaccion.

### Usuario Test

Clase de prueba que se encarga de validar la lógica y los endpoints de la entidad Usuario.

### Viaje Test

Clase de prueba que se encarga de validar la lógica y los endpoints de la entidad Viaje.

# 

# Documentación componentes software web

## Static

Carpeta predeterminada de Spring Boot en el contexto web. Abarca todos los archivos web (archivos CSS, JS, fuentes, etc.).

### Css

Subcarpeta dentro de static dedicada exclusivamente a archivos css.

#### Conseguila-style.css

Estilos generales para el formato de la página “conseguila.html”, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### Estaciones-style.css

Estilos generales para el formato de las páginas de la carpeta estaciones, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### General-style.css

Estilos generales para el formato de las páginas globales (body, navbar, footer, etc), aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### Index-style.css

Estilos generales para el formato de la página principal index.html, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### Login-style.css

Estilos generales para el formato de las páginas login,, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### Medios-transporte-style.css

Estilos generales para el formato de las páginas de la carpeta medios transporte, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### Recargas-style.css

Estilos generales para el formato de las páginas de la carpeta recargas, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### Register-style.css

Estilos generales para el formato de las páginas register,, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### Tarjetas-style.css

Estilos generales para el formato de las páginas de la carpeta tarjetas, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### Transacciones-style.css

Estilos generales para el formato de las páginas de la carpeta transacciones, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### Usuarios-style.css

Estilos generales para el formato de las páginas de la carpeta usuarios, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

#### Viajes-style.css

Estilos generales para el formato de las páginas de la carpeta viajes, aumentando la modularización y mejorando el mantenimiento y legibilidad del proyecto al tener presente las separaciones de dichos módulos web.

### Images

Se almacenan todos los archivos de formato imagen (.png, .jpg, .jpeg, etc), generalmente se tratará sobre logos o íconos de la aplicación.

### Js

Contiene archivos de JavaScript que proporcionan lógica del lado del cliente, renderizando en el navegador, evitando realizar la consulta al backend de la aplicación.

## Templates

### Estaciones

Crud de estaciones de transporte:

#### Formulario.html: formulario para crear una estación..

#### Lista.html: muestra un listado de todas las estaciones.

#### Detalles.html: vista de detalle de una sola estación.

#### Borrar.html: confirma eliminación de la estación.

### Fragments

Componentes reutilizables que se incluyen en otras páginas mediante la sintaxis de Thymeleaf th:insert o th:replace.

#### Head.html: define los importes de los .css, el título, etc.

#### Navbar.html: barra de navegación principal, con enlaces URL correspondientes.

#### Footer.html: pie de página compartido derechos de autor.

#### Contacto.html: sección de información de contacto, número de teléfono para soporte.

### Información

#### Conseguila.html: página acerca del procedimiento para obtener una tarjeta SUBE.

### Login

#### Login.html: formulario de acceso a la página.

#### Register.html: formulario para ingresar como un usuario nuevo.

### Medios-transporte

Crud de medios de transporte:

#### Formulario.html: formulario para crear un medio de transporte.

#### Lista.html: muestra un listado de todas los medios de transporte

#### Detalles.html: vista de detalle de un solo medio de transporte .

#### Borrar.html: confirma eliminación del medio de transporte.

### Recargas

Crud de recarga:

#### Formulario.html: formulario para crear una recarga.

#### Lista.html: muestra un listado de todas las recargas.

#### Detalles.html: vista de detalle de una sola recarga.

#### Borrar.html: confirma eliminación de la recarga.

### Tarjetas

Crud de tarjeta:

#### Formulario.html: formulario para crear una tarjeta.

#### Lista.html: muestra un listado de todas las tarjetas.

#### Detalles.html: vista de detalle de una sola tarjeta.

#### Borrar.html: confirma eliminación de la tarjeta.

#### Formulario-comprar.html: procedimiento alternativo de compra de tarjeta por medio de un link y condiciones específicas en torno a datos del usuario solicitados.

#### Buscar.html / Buscar-tarjeta.html: formularios para búsqueda de tarjetas por código o documento del usuario.

### Transacciones

Crud de transacción:

#### Formulario.html: formulario para crear una transacción.

#### Lista.html: muestra un listado de todas las transacciones.

#### Detalles.html: vista de detalle de una sola transacción.

#### Borrar.html: confirma eliminación de la transacción.

### Usuarios

Crud de usuario:

#### Formulario.html: formulario para crear una usuario.

#### Lista.html: muestra un listado de todas los usuarios.

#### Detalles.html: vista de detalle de una sola usuario.

#### Borrar.html: confirma eliminación del usuario.

### Viajes

Crud de viaje:

#### Formulario.html: formulario para crear una viaje.

#### Lista.html: muestra un listado de todas los viajes.

#### Detalles.html: vista de detalle de una sola viaje.

#### Borrar.html: confirma eliminación del viaje.

### Index

* Index.html: página base principal de la aplicación, con todos los servicios correspondientes.

# 

# Documentación Estándares de Seguridad

Se utiliza el componente PasswordEncryptor para encriptar automáticamente todas las contraseñas de todos los usuarios registrados dentro de nuestra base de datos. Este procedimiento se ejecuta cada vez que se inicializa la base de datos, ya sea desde un servidor o de manera local para la realización de pruebas. Al tratarse de información sensible, el fin es asegurar que la información de los usuarios no sea filtrada.

*@Component*

public class PasswordEncryptor {

private final UsuarioService usuarioService;

private final PasswordEncoder encoder = new BCryptPasswordEncoder(10);

public PasswordEncryptor(UsuarioService usuarioService) {

this.usuarioService = usuarioService;

}

*@PostConstruct*

public void init() {

usuarioService.findAll().forEach(usuario -> {

usuario.setContrasenia(encoder.encode(usuario.getContrasenia()));

usuarioService.crear(usuario);

});

}

}

Implementación del componente WebSecurityConfig para incorporar dentro del proyecto el @Bean BCryptPasswordEncoder, el cual nos habilitará un método de encriptación estándar para aplicar a nuestros correspondientes datos sensibles.

*@Configuration*

public class WebSecurityConfig {

public void addViewControllers(ViewControllerRegistry registry) {

registry.addViewController("/index").setViewName("index");

}

*@Bean*

public BCryptPasswordEncoder passwordEncoder() {

return new BCryptPasswordEncoder();

}

}

El componente GlobalControllerAdvice tiene la finalidad de implementar un sistema de verificación sobre roles de los usuarios, cada vez que se ingrese a una URL, se verificará el rol del usuario actual, logrando prevenir el acceso por parte de usuarios no autorizados a zonas sensibles de la aplicación, haciéndose pasar por administradores por ejemplo.

*@ControllerAdvice*

public class GlobalControllerAdvice {

*@Autowired*

private UsuarioService usuarioService;

*@ModelAttribute*

public void addUserToModel(HttpSession session, Model model) {

Long userId = (Long) session.getAttribute("userId");

if (userId != null) {

Usuario user = usuarioService.findById(userId);

model.addAttribute("currentUser", user);

if (user.getRole() == "USUARIO") {

model.addAttribute("isHost", true);

}

}

}

}