Sistema distribuído basado en geolocalización para el registro y asignación de viajes de una empresa de transporte de pasajeros

Instituto Universitario Aeronáutico - Facultad de Ingeniería

Juan Alejandro Giannuzzo - Juan Pablo Caivano

2016

A Mariano, cuyo apoyo y empuje constante permitió que podamos cerrar una etapa. A nuestras familias, quienes nos acompañaron este proceso de forma incondicional. ¡Gracias!

Índice

1. Introducción **4**

2. Objetivo del proyecto **5**

3. Destinatarios **6**

4. Beneficios esperados **7**

5. Estudio Técnico **8**

5.1. Tecnologías . . . 8
5.1.1. Riders app . . . 8
5.1.2. Drivers app . . . 8
5.1.3. WebServices . . . 8
5.1.4. Dashboard / Admin Web . . . 8
5.2. Procesos . . . 9
5.3. Medios . . . 10
5.4. Infraestructura . . . 11
5.5. Rendimientos . . . 12

6. Desarrollo del Trabajo **13**

6.1. Resumen Técnico . . . 13
6.1.1. Web Services . . . 13
6.1.2. Dashboard Web . . . 13
6.1.3. Admin Web . . . 13
6.1.4. Driver Mobile Application . . . 13
6.1.5. Rider Mobile Application . . . 13
6.2. Etapa I - Análisis de Requisitos . . . 15
6.2.1. Reunión Kickoff con el Cliente, Backend . . . 15
6.2.2. Reunión con el Cliente, Requerimientos Raider y Driver app . . . 16
6.3. Etapa II - Diseño de sistema . . . 17
6.3.1. Terminología . . . 17
6.3.2. Aplicación de Rider . . . 18
6.3.3. Aplicación de Driver . . . 31
6.3.4. Aplicaciones Web . . . 41
6.3.5. Base de Datos . . . 68
6.3.6. Milestones y Planificacion . . . 71
6.3.7. Restricciones, Supuestos y Convenciones . . . 74

6.4. Etapa III - Codificiación 74
6.5. Etapa IV - Pruebas 75
6.5.1. Pruebas de desarrollador 75
6.5.2. Pruebas del equipo de calidad 78
6.6. Etapa V - Mantenimiento 80
6.6.1. Manual de Usuario 80

7. Conclusión **86**

8. Bibliografia **87**

1. Introducción

El sistema se implementará a pedido de nuestro cliente, una empresa de transporte de pasajeros gratuito de una playa de Florida con el objetivo de brindar un mejor servicio a los pasajeros a la vez que se optimiza la operativa de la misma. Una parte importante del proyecto esta destinado al aprovechamiento de un recurso crítico del negocio que es la autonomía de los vehículos. Al tratarse de vehículos eléctricos, la autonomía de los mismos es escaza y es necesario optimizar las rutas de manera efectiva para maximizar la cantidad de viajes por carga de batería.

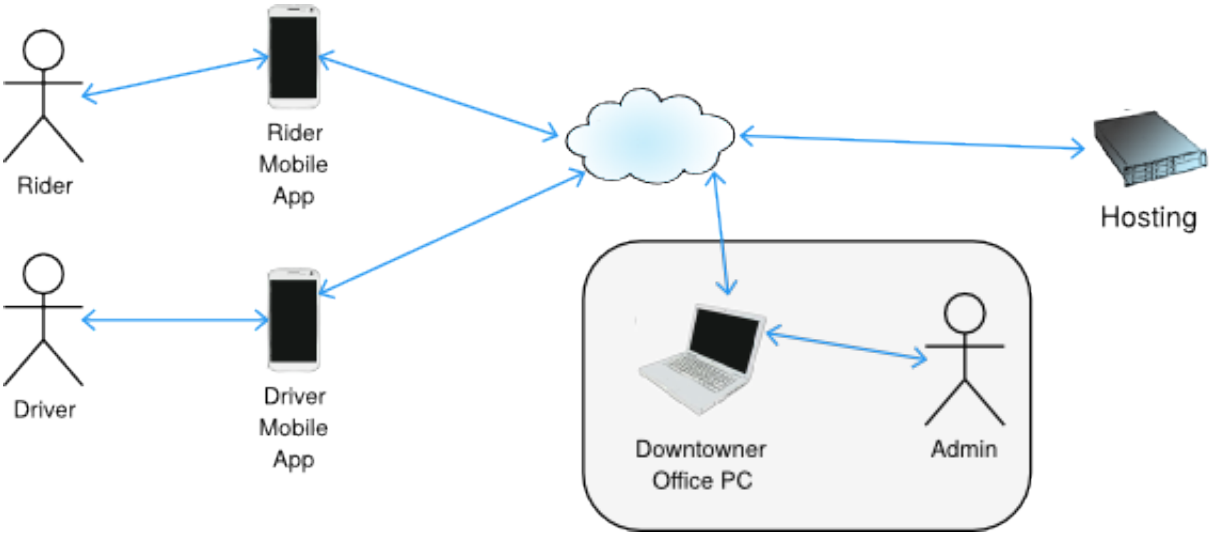
2. Objetivo del proyecto

El objetivo general es la optimización de los procesos vinculados a la administración de viajes de una empresa de transporte de pasajeros. Específicamente para lograr la optimización mencionada el sistema debe ser capaz de:

1. Capturar las solicitudes de viajes de los pasajeros.
2. Asignar y administrar los viajes.
3. Monitorizar a los conductores, vehículos y viajes en forma permanente.

3. Destinatarios

El proyecto se realiza a pedido de una empresa de transporte gratuito destinado a los turistas de una playa de estados unidos. El transporte se realiza en vehículos eléctricos que pueden cargar poca cantidad de pasajeros. Como usuarios de la aplicación podemos encontrar a 3 actores diferentes.



- Por un lado los **Riders**: los turistas de la playa que utilizan el servicio de transporte. Estos forman parte del público en general.
- Por otro lado los **Drivers**: los conductores de los vehículos de transporte de pasajeros. Estos son empleados de nuestro cliente.
- Por último los **Administradores**: que son los que van a gestionar y monitorizar el sistema web desde la central. Estos son empleados de nuestro cliente.

4. Beneficios esperados

- Reducción de costos de mantenimiento de los vehículos de transporte, debido a la optimización de rutas.
- Reducción de tiempos de carga de las baterías de los vehículos, debido a la optimización de rutas.
- Reducción de costos minimizando el tiempo ocioso de los **Drivers**.
- Aumento de control e información por parte de los **Administradores** de la flota, clientes y viajes realizados.
- Aumento en la cantidad de clientes debido a que aumentan las vías de acceso de los **Riders** al servicio (por teléfono y vía la aplicación mobile).

5. Estudio Técnico

5.1. Tecnologías

Se seleccionaron las siguientes tecnologías:

5.1.1. Riders app

- iPhone / Objetive-C
- Apple Map Kit
- Google Geocoding web services API
- Apple Push Notification Services

5.1.2. Drivers app

- iPhone / Objetive-C
- Apple Map Kit
- Google Geocoding web services API
- Apple Push Notification Services

5.1.3. WebServices

- PHP 5.3
- MySQL
- Apple Push Notification Services

5.1.4. Dashboard / Admin Web

- HTML 5
- JavaScript
- Bootstrap
- Google Maps API v3

5.2. Procesos

No aplica

5.3. Medios

No aplica

5.4. Infraestructura

Se presentaron diferentes alternativas, todas ellas orientadas a poder servir tanto el **Web Service** como el **Dashboard / Admin Web**

- Utilizando infraestructura propia
- Utilizando infraestructura de un tercero (física o virtual)
- Utilizando un servicio de hosting

En el caso que se opte por realizarlo con **infraestructura propia**, se prevé la siguiente infraestructura inicial:

- 1 Servidor físico, con un microprocesador de al menos 2 Núcleos, operando a una frecuencia de al menos 1 Ghz, 1 GB de memoria RAM, 2 GB de memoria de swap, 10 GB de espacio de disco.
- Sistema Operativo Ubuntu Server 14.04 LTS, los ciclos de soporte de las versiones LTS son extendidos a cinco años, liberando nuevas versiones LTS cada dos años.
- PHP 5.3
- MySQL 5.1
- Conexión a internet permanente de banda ancha empresarial, con garantía de 1MB de subida y 1MB de bajada.
- Una IP pública accesible desde internet.

Hay que tener en cuenta que esta solución presenta inconvenientes para un proyecto inicial, ya que hay que sumar a los costos propios de hardware, los costos de alojamiento en un data center, así también como los costos de administración de infraestructura.

Si se utilizara la **infraestructura de un tercero**, se deberían cumplir los requisitos de procesamiento ya definidos para la infraestructura propia. Para ello existen dos alternativas, la contratación de un servidor físico dedicado, o la de un VPS [1], siendo las propuestas virtuales mas baratas, pero de menor desempeño. También en este tipo de casos se debería prever los costos de administración de la infraestructura, que normalmente no están cubiertos en los planes de servicios.

Otra alternativa es la de utilizar un servicio especializado en el hosting de aplicaciones PHP, estos desligan del proyecto la complejidad de armar un servidor, Base de Datos, manejo de URLs, monitoreo de servidores, y demás actividades relacionadas al manejo de la infraestructura. Muchos de ellos, ofrecen planes gratuitos que pueden ser utilizados inicialmente, pero con la flexibilidad suficiente de poder comprar nuevos recursos a medida que la demanda aumenta, lo que hacen que sea una solución atractiva y escalable.

Comparando las soluciones propuestas tenemos:

Solución	Costo	Rendimiento	Escalable	Adm de IT	Complejidad
Propia[2]	Alto	Media-Alta	No	No incluida	Alta
Tercero	Medio	Media-Alta	No	No incluida	Alta
Hosting	Bajo-Medio	A demanda	Sí	Incluida	Baja

Por lo que para el inicio del proyecto, se recomienda un servicio de Hosting de aplicaciones frente a las otras alternativas, que ofrecen la posibilidad de poder alojar gratuitamente el software en sus fases iniciales donde los requerimientos de procesamiento, almacenamiento y conectividad son bajos, y cuando sea necesario mayor potencia de cómputo se pueden escalar comprando una mayor cantidad de recursos.

También se deberá contar con Licencia de iOS Developer para la distribución de las aplicaciones iOS (**Driver y Rider**).

[1] Virtual Private Server - Servidor Privado Virtual
[2] Se asume un servidor configurado para trabajar en modalidad Stand Alone, sin ser parte de un cluster.

5.5. Rendimientos

No aplica

6. Desarrollo del Trabajo

6.1. Resumen Técnico

Se implementan cuatro piezas de software principales, las mismas se comunicarán entre sí, y son:

6.1.1. Web Services

Interface común que centraliza los datos e intercomunica al resto de las aplicaciones entre sí. Tiene 2 **funciones principales**:

- Establecer la lógica de negocios
- Gestionar la persistencia de los datos de toda la aplicación

Presenta 4 API diferentes, cada cual posee su propia lógica de negocios, pero comparten una Base de Datos en común.

1. **Rider API**
2. **Driver API**
3. **Dashboard API**
4. **Admin API**

6.1.2. Dashboard Web

Es la aplicación web que se va a utilizar en la central, su objetivo es organizar y monitorizar los viajes. La misma integra un mapa en el que se indica en tiempo real la posición de los vehículos, también otorga los mecanismos visuales necesarios para administrar los pedidos de viajes que se realicen mediante llamadas telefónicas convencionales.

6.1.3. Admin Web

Es la interface web que se utiliza para la administración del sistema. Desde la misma se pueden gestionar a los usuarios, las zonas de operación, promociones, y demás parámetros de la aplicación.

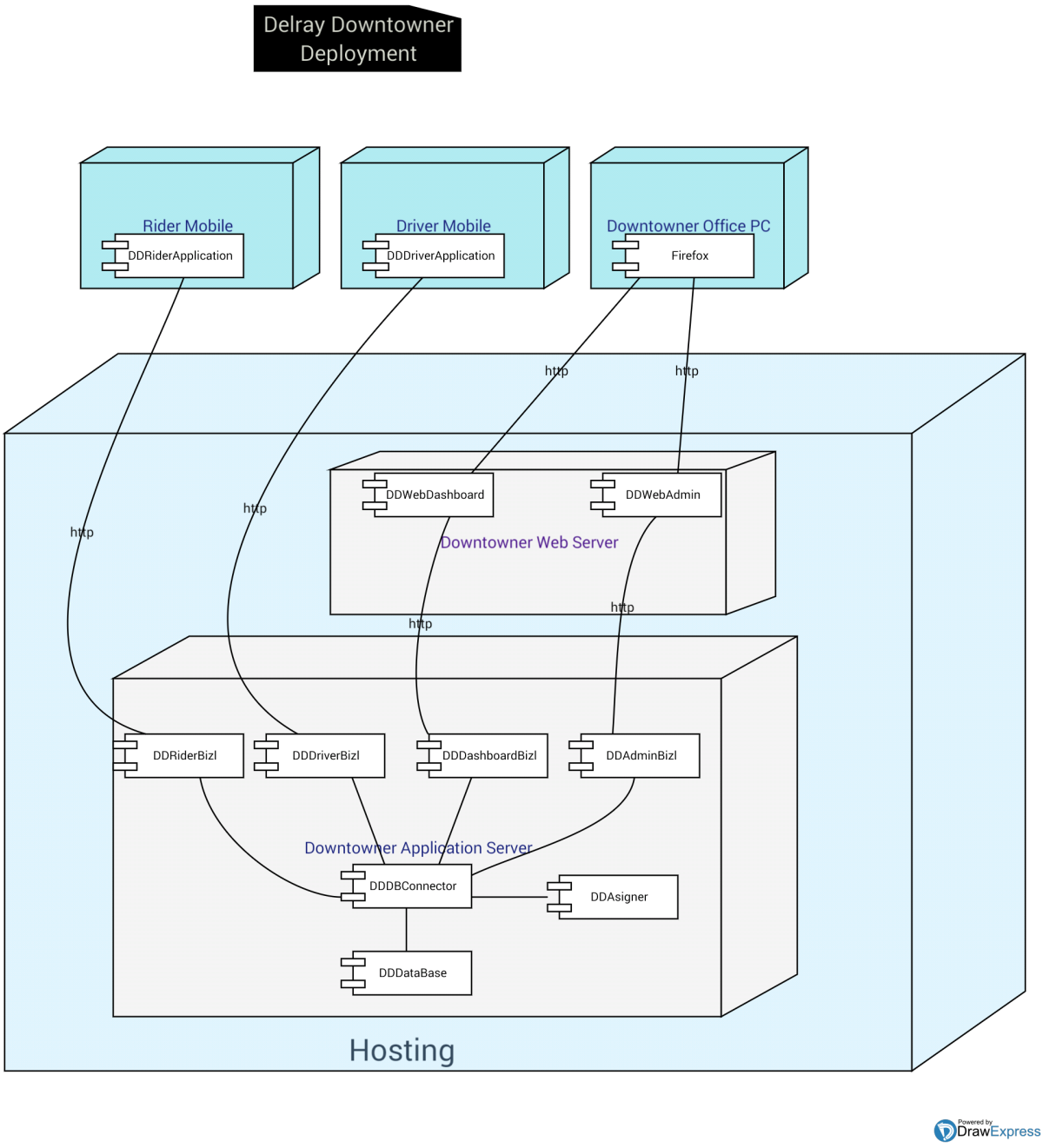
6.1.4. Driver Mobile Application

Es la aplicación *iPhone* que se ejecuta en los teléfonos de cada uno de los vehículos. La misma otorga la opción de identificación de un **driver**, que una vez identificado en la terminal puede navegar los diferentes viajes disponibles en existencia, elegir uno de ellos y tomarlo, calcular las rutas, administrar sus "breaks" (recreos), comunicarse con los pasajeros y administrar su propio perfil.

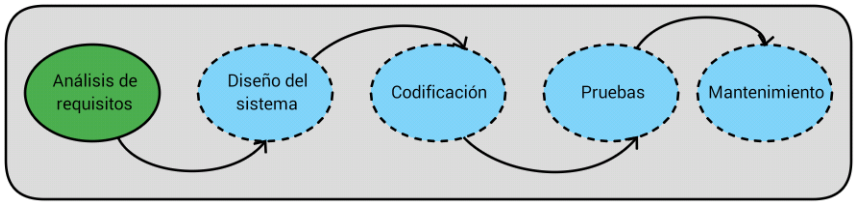
6.1.5. Rider Mobile Application

Esta aplicación es la que se ejecuta en los teléfonos móviles de los turistas que desean utilizar el servicio de transporte. Desde la misma pueden realizar el pedido de viaje, ver las promociones vigentes del sistema, compartir su viaje mediante redes sociales y realizar la administración de su perfil.

La interrelación de las cuatro piezas de software se pude visualizar en el siguiente diagrama de bloques:


Delray Downtowner
Deployment
Rider Mobile
DDRiderApplication
Driver Mobile
DDDriverApplication
Downtowner Office PC
Firefox
http
http
http
http
http
http
DDWebDashboard
DDWebAdmin
Downtowner Web Server
DDRiderBizl
DDDriverBizl
DDDashboardBizl
DDAdminBizl
Downtowner Application Server
DDDBConnector
DDAsigner
DDDataBase
Hosting
Powered by DrawExpress

6.2. Etapa I - Análisis de Requisitos



Debido a las características del proyecto, hemos elegido la metodología de **desarrollo en cascada**, dado que la confección de este sistema tiene requisitos fijos, en esta etapa se logró clarificar con mucha calidad cuáles eran los requerimientos que el cliente necesitaba, y por lo tanto dejaba en claro cómo sería el producto final a presentar.

Para iniciar con la captura de requerimientos, se agenda en primera instancia se agenda una reunión con el cliente, de las cuales obtenemos la siguiente minuta:

6.2.1. Reunión Kickoff con el Cliente, Backend

- Participantes:
  - Juan Alejandro Giannuzzo
  - Juan Pablo Caivano
  - Cliente
- Fecha: 18/08/14
- Objetivo: En base al intercambio de mails realizado con el cliente, se establece una reunión para capturar requisitos a cumplimentar por la interface de Backend.
- Notas:
  - Se identifica la necesidad de tres tipos de usuarios
    * **Riders:** Turistas de la playa que utilian el servicio de transporte, son parte del público en general
    * **Drivers:** Conductores de vehículos de transporte de pasajeros. Empleados del cliente.
    * **Admins:** Gestionan y monitorean el sistema desde la central, también empleados del cliente.
  - Desde el backend (dispatcher), el **Admin** debe visualizar un mapa con el área en la cual se recogen a los **riders**
    * Cuando un **Rider** solicita un viaje:
      · Su *nombre* y *dirección* deben aparecer en el listado en el backend (en un costado)
      · Desde un menú drop down, seleccionar y despachar un **Driver** asignado
      · El **Driver** recive un MMS[3] con el nombre y dirección del **Rider** a ser buscado
      · El **Driver** confirmará, y ambos, **Rider** y **Admin** recibirán un MMS de confirmación
      · Cuando un **Driver** colecta a un **Rider**, desde un botón puede enviar confirmación al **Admin**
    * Todas las locaciones de los **Drivers** deben ser visibles como un punto con su identificador asociado en el mapa, a medida que se desplazan, esto ayudará al **Admin** a decidir qué **Driver** obtiene qué **Rider**.
    * Cuando el pedido de viaje de un **Rider** es confirmado por el **Admin**, se debe mostrar la posición del mismo en el mapa de Backend con un punto diferenciado (por ejemplo, un punto azul).
    * Si un **Rider** no posee la aplicación en su terminal y decide realizar el pedido llamando por teléfono, el **Admin** que reciba el llamado ingresará la carga de datos del **Rider** de forma manual en el sistema, ingresará el número de móvil del mismo, y cuando se guardan los datos se deben disparar las siguientes acciones:

[3] del inglés, **m**ultimedia **m**essaging **s**ervice

- El **Admin** puede asignar el viaje a un **Driver** para que recoja al **Rider**
- Se creará en el servidor el perfil del **Rider**
- Se enviará un SMS[4] al número del **Rider** indicándole que baje la aplicación.

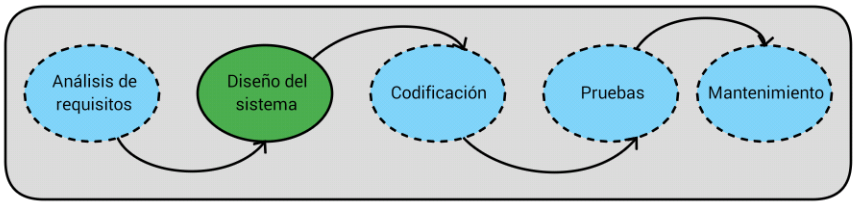
* En el campo dirección del menú de la aplicación, el cliente quiere tener las 10 locaciones más solicitadas del área de forma pre-cargada, las mismas son restaurantes y otras locaciones populares. Las mismas se deberían poder de modificar desde el backend. No se necesitan los datos GPS de las mismas en sí, ya que el **Admin** las ingresaría manualmente desde el backend. Ejemplo de estas locaciones serían McDonalds, Holiday Inn, Burger King. De esta forma, cuando un **Rider** quiere volver a su lugar de origen, simplemente selecciona la locación en la que está sin necesidad de ingresar la dirección (la probabilidad de estar en un lugar popular es alta).

6.2.2. Reunión con el Cliente, Requerimientos Raider y Driver app

- Participantes:
  - Juan Alejandro Giannuzzo
  - Juan Pablo Caivano
  - Cliente
- Fecha: 21/08/14
- Objetivo: En base al intercambio de mails realizado con el cliente, se establecerá una reunión para capturar mayor cantidad de requisitos a cumplimentar.
- Notas:
  - Destinos de los **raiders**
    * El **raider** al momento de ingresar necesita ingresar su destino final, así también como otra información, como es el lugar de recogida, la cantidad de personas en el viaje
  - Locations
    * El sistema deberá mostrar un menú del tipo drop down con las locaciones más utilizadas para recogida, **Top Locations**.
    * Si la locación de un **Rider** (posicionándolo mediante GPS) es una locación dificultosa (por ejemplo porque el restaurant no cuente con espacio de estancionamiento definido frente al local), se le mostrará un mensaje emergente (pop up) al **raider** para dirigirlo a una locación predeterminada, **Ideal Pick Up Location**. Se debe definir si es posible realizar algún mecanismo de cercado por geolocalización, armando grillas que incluyan a determinadas cuadras, y redirigiendo al **raider** a determinadas esquinas o locaciones, hay que verificar si el GPS será lo suficientemente preciso para lograrlo.
  - Se deben definir mecanismos de **alertas sonoras** en los teléfonos de los **drivers**, para un nuevo viaje, un viaje cancelado, etc.
  - No hay necesidad de agregar un botón para rechazar viajes en el teléfono del **driver**

[4] del inglés, **s**hort **m**essaging **s**ervice

6.3. Etapa II - Diseño de sistema

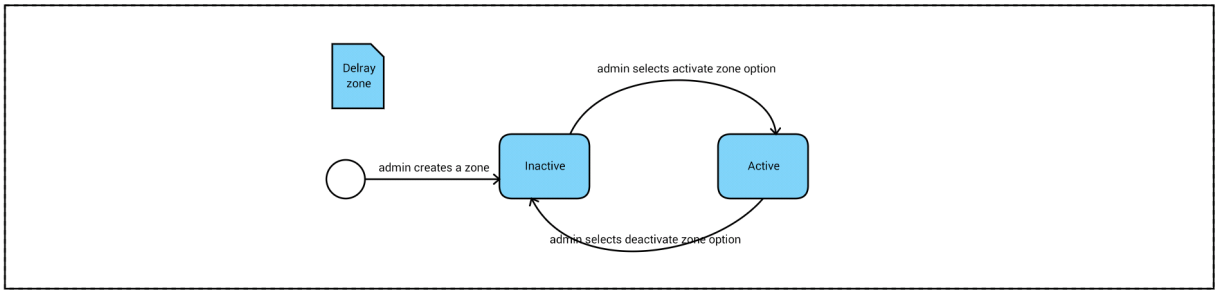


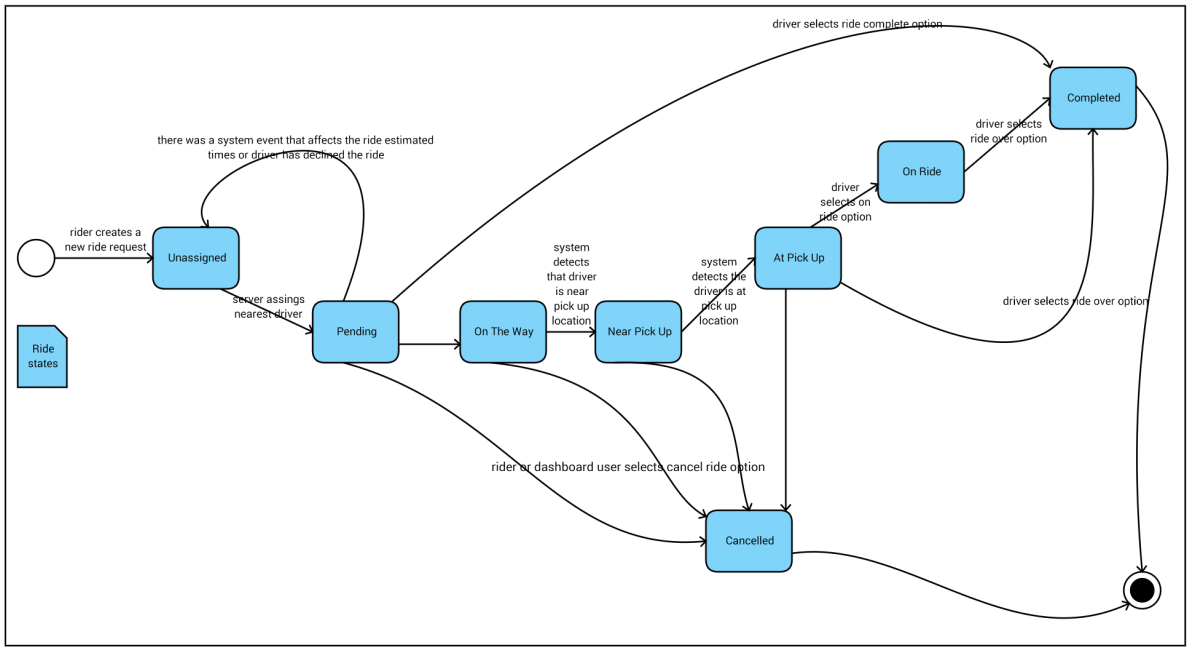
Una vez capturados de la etapa anterior todos los requisitos, tomando como entrada el **documento de requisitos** confeccionado y validado con el cliente, se procede a avanzar a la etapa de **Diseño de Sistema**. En la misma confeccionamos los diferentes diagramas y documentos de especificaciones, mediante los cuales hacemos visibles tanto el funcionamiento como así también la comunicación entre los diferentes componentes del sistema, también se presenta la planificación de funcionalidades por versiones, como el cálculo de tiempo para la implementación.

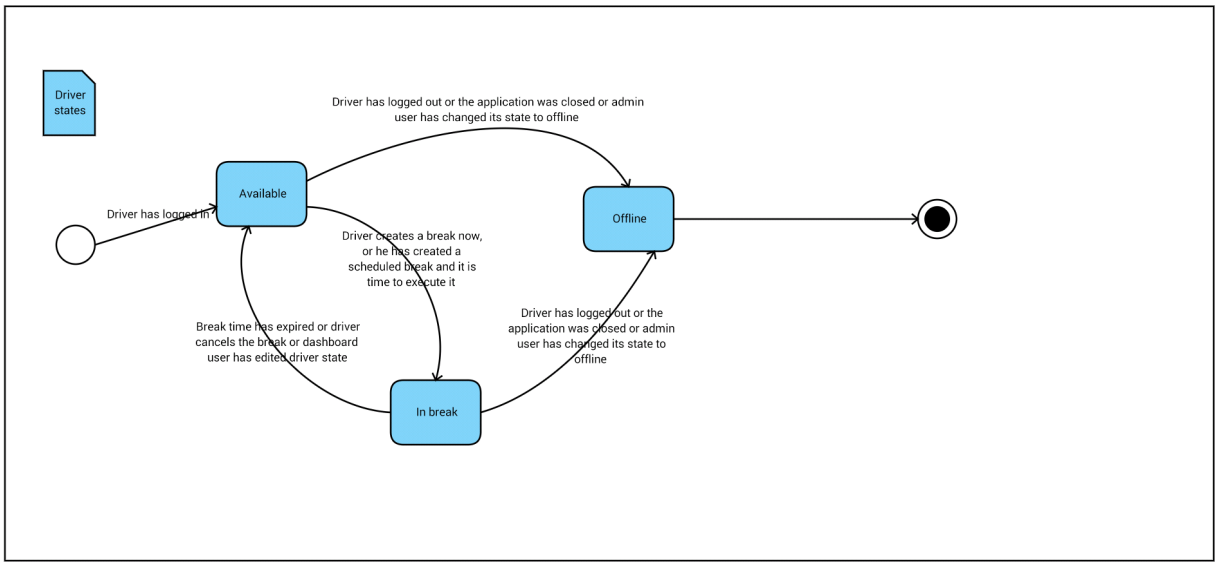
6.3.1. Terminología

- Rider: Es el usuario del sistema de transporte
- Driver: Es el conductor del vehículo
- Zone/Zona: Es la zona en la cual opera el servicio de transporte
- Ride/Viaje: Es un viaje en si.
- Pick-up Point / Punto de Recogida: geoposicionamiento del sitio a donde un usuario pidió el inicio de un viaje.
- Drop-off Point / Punto de llegada: geoposicionamiento del sitio en donde termina el viaje y el usuario desciende del vehículo.

Como elementos principales del sistema competo existen las siguientes entidades, con sus respectivos estados y transiciones:







6.3.2. Aplicación de Rider

Pull Services Los **Pull Services** son servicios específicos de aplicación que estarán ejecutándose localmente en el teléfono mientras la aplicación **Rider** está ejecutándose en primer plano.

- Pull ride service: El estado retornado del viaje es aquel que tenga el Rider (Pasajero). La aplicación realizará un **pull** del viaje desde el **Servidor DelRay** cuando el **Rider** haya solicitado un viaje, y el estado del viaje no sea **"Ride Over"**. Una vez que el viaje alcance el estado **"Over"**, el servicio **Pull** será detenido. La frecuencia de **Pull** será retornada por el servidor DelRay luego de un acceso exitoso (login).

Zones (Zonas) Las direcciones de recogida y entrega serán delimitadas por la Zona DelRay. La Zona estará dividida en Subzonas, Este y Oeste. Estas subzonas pueden estar activas o no. Cuando una zona está inactiva, los viajes en dicha zona no serán permitidos.

Application Sessions (Sesiones de Aplicación) Un Rider puede acceder y permanecer activo solamente con un dispositivo. Si ha accedido usando un dispositivo A, y luego accede usando un dispositivo B, entonces la próxima petición al servidor DelRay desde el dispositivo A será rechazada, y la aplicación en el dispositivo A caducará su sesión navegando nuevamente a la pantalla de acceso Login Screen

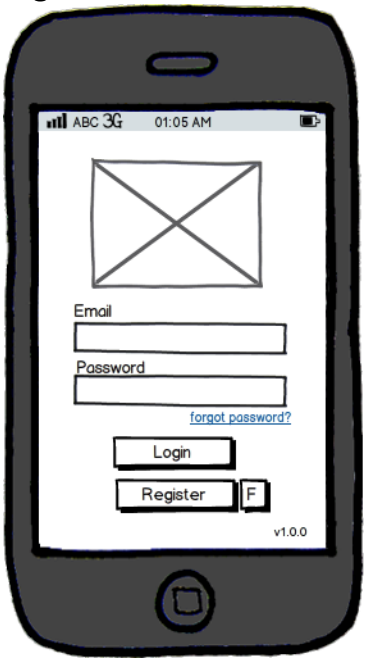
Funcionalidad de autocompletado de Direcciones (Address autocomplete feature)

- La funcionalidad de autocompletado será soportada por la funcionalidad **Google Places Autocomplete**, con sus limitaciones de uso. Ver Restricciones, Supuestos y Convenciones
- Si el servicio de **Google Places Autocomplete** no está disponible, o se ha alcanzado el límite de uso, la función de autocompletado quedará deshabilitada.

Funcionalidad de Promociones (Promotions feature)

- A fin de poder otorgar a DelRay Downtowner y sus directivos una gran flexibilidad en el diseño de los detalles de las promociones, los detalles de las promociones se guardarán como una página web en formato html.
- DelRay Downtowner debe construir la página html de promociones y subir el archivo en el servidor DelRay en una ubicación determinada.
- La aplicación Rider mostrará los detalles de la promoción como una página web.

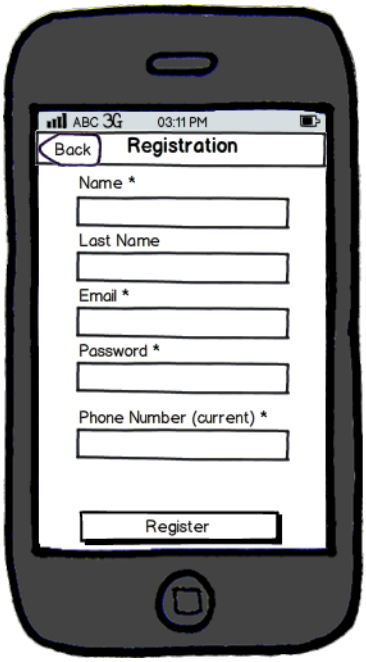
Login Screen



- Forgot Password:
  - El Rider puede recuperar su password presionando el botón **Forgot Password**. El sistema enviará un email al Rider con una nueva password.

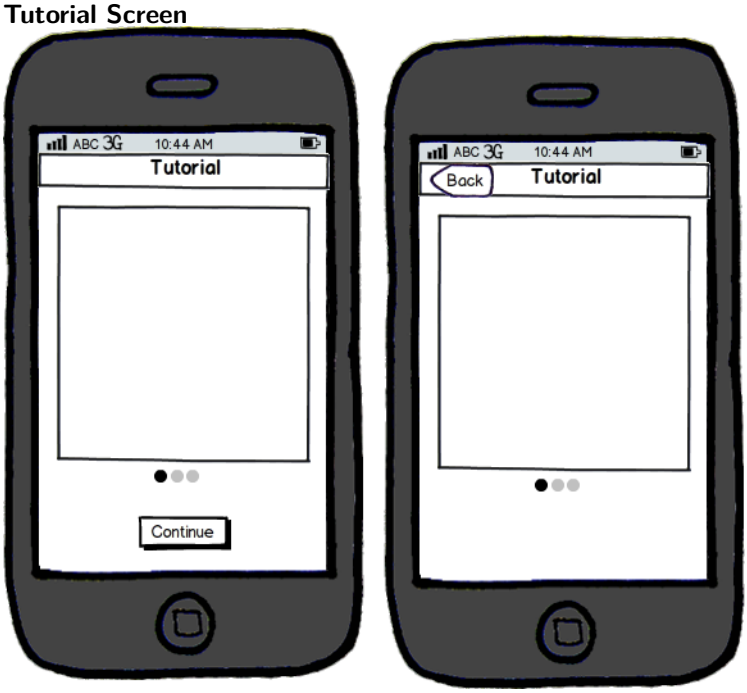
- Login Button:
  - El Rider ingresará sus credenciales y presionará el botón **Login**.
  - Si las credenciales ingresadas son válidas, el servidor enviará los parámetros de configuración de la aplicación y la pantalla Home Screen será mostrada.
  - Si las credenciales ingresadas son inválidas, un mensaje de error aparecerá.
- Register Button:
  - Si el Rider no posee una cuenta DelRay, la aplicación permitirá crear una nueva cuenta presionando el botón **Register**

Register Screen



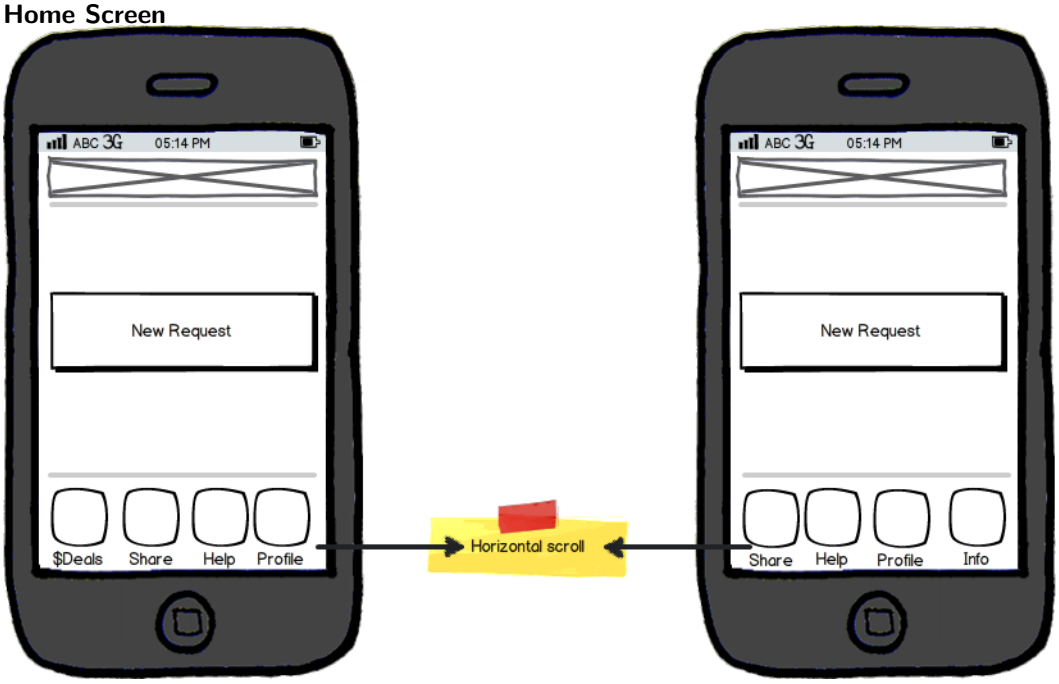
- Register Button
  - El Rider ingresa la información de registro y presiona el botón **Register**
  - Los campos válidos serán:
    * name: cualquier caracter, cadenas vacías no son permitidas.
    * last name: cualquier caracter.
    * email: dirección de correo electrónico de formato válido, cadenas vacías no son permitidas.
    * password: cadena con una longitud mínima de 6 caracteres.
    * address: cualquier caracter. La dirección debe estar dentro de la zona DelRay. Una función de autocompletado aparecerá a medida que el usuario vaya ingresando la dirección.
    * phone number: cualquier número de teléfono de Estados Unidos, cadenas vacías no son permitidas.
  - Si el proceso es exitoso, el servidor enviará a la aplicación los parámetros de configuración y la pantalla Tutorial Screen será mostrada.
- Back Button
  - Navega nuevamente hacia la pantalla Login Screen

Tutorial Screen



- La pantalla se muestra luego de que el registro se realizó de forma correcta.
- Continue: Si es la primera ejecución de la aplicación, se muestra este botón. Cuando el mismo es pulsado, se navega a Home Screen

Home Screen



- Request/Ride Status: Esta área mostrará y actualizará el estado de los pedidos/viajes.

  - Ver Pull Services y Restricciones, Supuestos y Convenciones por las actualizaciones de estado de los viajes.
  - Ver Terminología para las transiciones de estados de los viajes.

- New Request: Cuando es pulsado, una pantalla New Pick Up Request Modal Screen aparecerá.
- Otras Opciones: Deslizar para abrir el menú que contiene las siguientes opciones:

  - Promos: Cuando es pulsado, abre la lista Promotions
  - Share: Cuando es pulsado, abre la sección Share Screen
  - Info: Cuando es pulsado, abre la sección de información de la aplicación Information Screen
  - Profile: Cuando es pulsado, abre la sección Profile Screen
  - Help: Cuando es pulsado, abre la sección Tutorial Screen

Share Screen

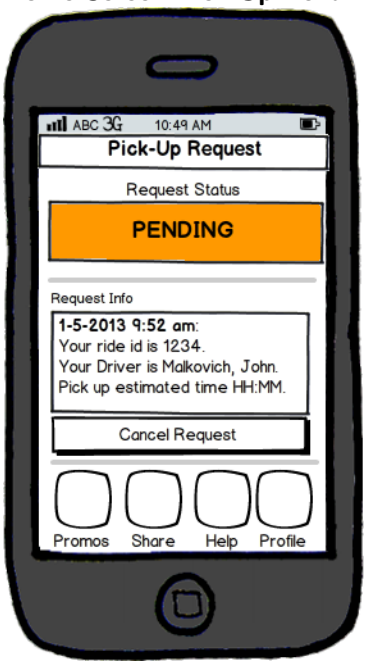
- Si el Raider no autorizó a la aplicación DelRay a compartir información en su muro, entonces la aplicación de Facebook se mostrará para permitir al Rider dichar autorización. Si la autorización es exitosa, un post de facebook se realizará en el muro del Raider.
- Si el Raider ya ha autorizado a la aplicación DelRay a compartir en su muro, entonces la aplicación realizará un post en el muro del Raider y mostrará un mensaje emergente con los resultados del post.
- El texto del mensaje del post será recuperado de los parámetros del sistema.

New Pick Up Request Modal Screen

- Pick-Up: El Rider escribe en el campo de texto la dirección de recogida. Una funcionalidad de autocompletado aparecerá a medida que el Rider vaya escribiendo la dirección.
- Find me: La aplicación llenará el campo de texto con la dirección detectada por el servicio web de geocodificación de Google; cuando una petición de geolocalización es enviada a Google Geocoding, un indicador de carga será mostrado. Cuando la dirección sea retornada por el servicio de Google Geocoding, se mostrará en el campo de texto y el indicador de carga desaparecerá. Luego el Rider podrá editar la dirección si la misma no es lo suficientemente exacta, la edición será acompañada por la función de autocompletado. Si el Rider presiona el **Home Button** o comienza a escribir en el campo de texto antes de que el servicio de Google Geocoding devuelva una dirección, la aplicación cancelará la petición al servicio de geocodificación y ocultará el indicador de carga.
- Home (Pick-Up): Dirección predefinida. La dirección se toma del perfil del Rider (si la dirección no está definida entonces la aplicación solicita al Rider ingresar una dirección en la pantalla de registro). Si la dirección ya fue cargada en el campo de texto correspondiente a **Pick-Up**, al presionar el botón Home se sobreescribirá el campo de texto con la dirección de recogida. Si el Rider presiona el botón Home antes de que el servicio de Google Geocoding retorne una dirección, entonces la petición se cancelará y se esconderá el indicador de carga.
- Additional Info: Cuando sea presionado, un popup aparecerá con un campo de texto. El Rider podrá agregar cualquier información adicional que pueda ayudar al Driver a encontrar la dirección de recogida. Por ejemplo, el Rider puede ingresar el piso y el número de departamento.
- Home (Drop-off):
  - Dirección Predefinida: La dirección se toma del perfil del Rider (si la dirección no está configurada, entonces la solicitará al Rider ingresar dicha dirección en la pantalla de registro). Si el campo de texto correspondiente a Destino (Destination) ya contiene una dirección, la dirección será reemplazada por la dirección Home.
  - Si la dirección de recogida fue obtenida utilizando el botón Home, este botón se deshabilitará.
- Number of passengers (Cantidad de pasajeros): El Rider puede seleccionar la cantidad de pasajeros para el viaje presionando los botones de selección "+" o "-".
- Send Request:
  - Cuando es presionado, la aplicación verifica la información, si la misma es correcta la petición es enviada al servidor, en caso contrario, un mensaje de error aparecerá.

  - Ver asignación de algoritmo de viaje en el documento del servidor.
    * Los valores válidos son:
      · My location: no puede ser vacío, la dirección debe estar dentro de una zona DelRay activa.
      · Number of passengers: no puede ser vacío, el rango de valores válido es entre 1 y 5 (inclusive).
      · Destination: no puede ser vacío, la dirección debe estar dentro de una zona DelRay activa.
      · Ver Terminología para ver la definición de estados de las zonas DelRay.
  - Cancel: Cuando es presionado, navega nuevamente a la pantalla Home Screen

Home Screen Pick Up Pending



- Una vez que la petición de viaje es enviada al servidor DelRay, la pantalla Home Screen mostrará el estado Pendiente, **Pending**, y el estado **pull ride** del servicio del viaje en el servidor DelRay será iniciado.
- Cancel Request:
  - La petición de cancelación de un viaje es enviada al servidor DelRay actualizando el estado del viaje a Cancelado (**Canceled**) y cargando el valor de tiempo estimado de recogida a cero.
  - La pantalla Home Screen será actualizada mostrando el estado **No Request**, la aplicación dejará de forzar actualizaciones de estado del viaje desde el servidor.
- Ver Pull Services y Restricciones, Supuestos y Convenciones para las actualizaciones del estado del viaje.
- Ver Terminología para las transiciones de estado del viaje.

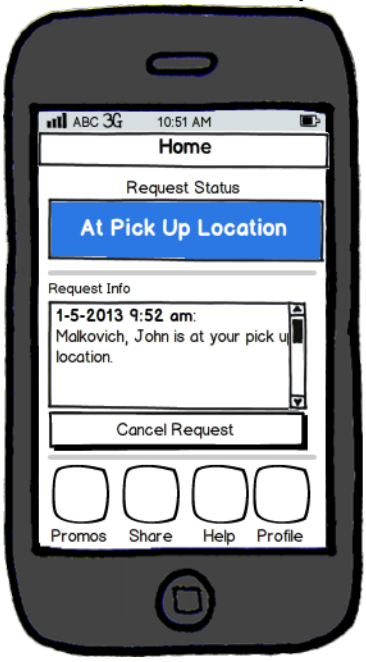
Home screen Pick Up On The Way

- Cuando un Driver de DelRay acepta el viaje, y el Pull Ride Service ha actualizado el viaje, la pantalla Home Screen mostrará el estado **On The Way**.
- La información concerniente al viaje será periódicamente actualizada por el Pull Ride Service, en la sección Request Info.
- Cancel Request:
  - Si el Rider lo confirma, un pedido de cancelación del viaje es enviado al servidor DelRay, actualizando el estado del viaje a Cancelado, y configurando el tiempo estimado de recogida a cero.
  - La pantalla Home Screen se actualizará, mostrando el estado **No Request**, y la aplicación parará de solicitar actualizaciones de estado del viaje al servidor.
- Ver Pull Services y Restricciones, Supuestos y Convenciones para las actualizaciones de estado de un viaje.
- Ver Terminología para las transiciones de estados del viaje.

Home Screen Pick Up Near To Location

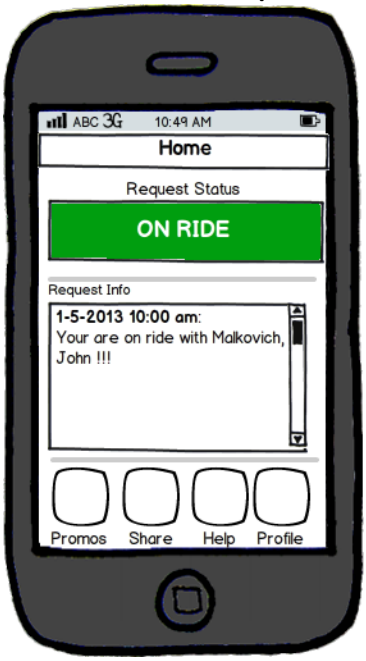
- Una vez que el viaje es actualizado por el servicio **pull rise service**, y que el estado del viaje ha cambiado a **Near Pick Up**, la pantalla Home Screen se actualizará mostrando el estado **Near Pick Up**, indicando que el driver está arribando a la dirección de recogida.
- Un SMS con el mensaje "The driver is close to pick up location" será enviado al Rider.
- Cancel Request:
  - Si el Rider lo confirma, un pedido de cancelación del viaje es enviado al servidor DelRay, actualizando el estado del viaje a Cancelado, y configurando el tiempo estimado de recogida a cero.
  - La pantalla Home Screen se actualizará, mostrando el estado **No Request**, y la aplicación parará de solicitar actualizaciones de estado del viaje al servidor.
- Ver Pull Services y Restricciones, Supuestos y Convenciones para las actualizaciones de estado de un viaje.
- Ver Terminología para las transiciones de estados del viaje.

Home Screen At Pick Up Location



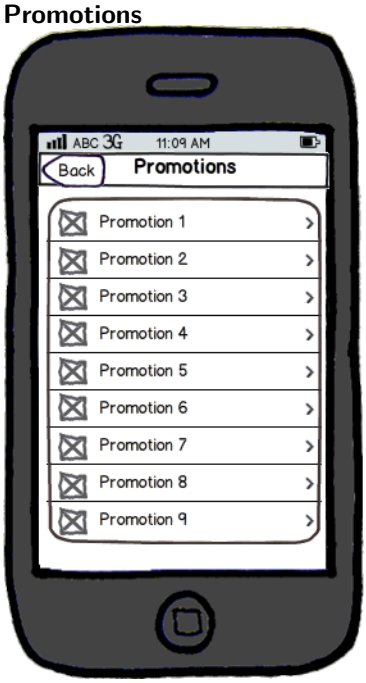
- Una vez que el viaje es actualizado por el servicio **pull rise service**, y que el estado del viaje ha cambiado a **At Pick Up**, la pantalla Home Screen se actualizará mostrando el estado **At Pick Up**, indicando que el driver está en la dirección de recogida.
- Cancel Request:
  - Si el Rider lo confirma, un pedido de cancelación del viaje es enviado al servidor DelRay, actualizando el estado del viaje a Cancelado, y configurando el tiempo estimado de recogida a cero.
  - La pantalla Home Screen se actualizará, mostrando el estado **No Request**, y la aplicación parará de solicitar actualizaciones de estado del viaje al servidor.
- Ver Pull Services y Restricciones, Supuestos y Convenciones para las actualizaciones de estado de un viaje.
- Ver Terminología para las transiciones de estados del viaje.

Home Screen Pick Up On Ride



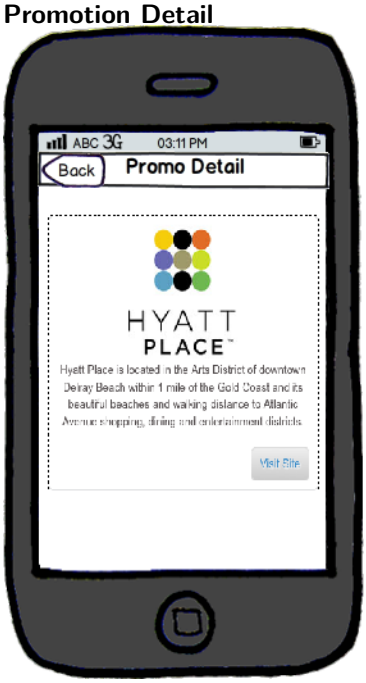
- Una vez que el Rider está en el vehículo, y el Driver ha actualizado el estado del viaje a "On Ride", el estado "On Ride" se mostrará en la pantalla Home Screen, cuando el servicio de estado Pull Ride es ejecutado.
- La información del viaje aparecerá en el campo **Request Info Area**
- Cuando el viaje termine, y el Driver actualice el estado del viaje a **Over**, la pantalla Home Screen se actualizará en el momento que el servicio de estado Pull Ride sea ejecutado, mostrando el estado **No Request Status**, el servicio de **Pull Ride** será finalizado.
- Ver Pull Services y Restricciones, Supuestos y Convenciones para las actualizaciones de estado de un viaje.
- Ver Terminología para las transiciones de estados del viaje.

Promotions



- Un listado de promociones disponibles es mostrado.
  - El listado de promociones es cargado desde el servidor DelRay
  - Si la promoción no tiene una imagen en miniatura, o si la misma no se puede cargar, una imagen en miniatura por defecto será mostrada.
- Cuando el usuario pulsa en un ítem, la aplicación navega a la pantalla Promotion Detail.
- Back Button: Navega a la pantalla Home Screen.

Promotion Detail



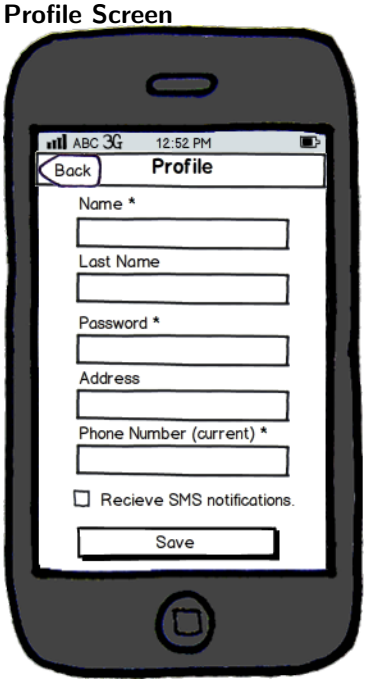
- La página web de la promoción seleccionada es mostrada.
- Back Button: Navega de regreso a la pantalla Home Screen.

Information Screen

- La información de la empresa es mostrada. La lógica para obtener la información de la empresa, es la misma utilizada para obtener lista de promociones en Promotions
- La información de la empresa debe ser cargada en texto plano, no se soporta texto enriquecido.

- Back Button: Navega de regreso a la pantalla Home Screen.

Profile Screen



- El perfil del Rider es mostrado.
- El Rider puede editar su perfil.

- Save Button: Los valores del perfil son enviados al servidor DelRay.
- Back Button: Navega de regreso a la pantalla Home Screen. Si el Rider no presionó el botón Save, entonces todos los cambios sin guardar se perderán.
- Ver la sección Register Screen para valores válidos.

6.3.3. Aplicación de Driver

Pull Services Los **Pull Services** son servicios específicos de aplicación que estarán ejecutándose localmente en el teléfono mientras la aplicación **Driver** está ejecutándose en primer plano.

- Pull Rides Service:
  - Obtiene del servidor DelRay todos los viajes del día actual pertenecientes al Driver.
  - El intervalo de actualización será otorgado por el servidor DelRay, luego de que se haya autenticado exitosamente en la aplicación.

Tracking Service Este servicio se ejecutará localmente en la aplicación Driver, y rastreará la ubicación del Driver utilizando la apo del servicio de localización del teléfono. El servicio de rastreo se encontrará apagado solamente cuando el Driver no tenga una sesión abierta en la aplicación. Obtener una localización precisa depende básicamente de la disponibilidad de satélites gps y del entorno en el que el usuario se encuentre. Puede demorar unos 30 segundos obtener una localización aceptable. Consideraremos como precisión aceptable cuando la precisión está dada por lo siguiente:

- Best for navigation or Best (API del servicio de localización de iOS con la precisión estándar).
- El intérvalo de rastreo será configurado en X metros, esto significa que la aplicación recibirá una nueva actualización de de localización desde el servicio de localización del teléfono cada X cantidad de metros, este valor se obtendrá como un parámetro de sistema cuando se acceda mediante login exitosamente a la aplicación.

Zones (Zonas) Las direcciones de recogida y entrega serán delimitadas por la Zona DelRay. La Zona estará dividida en Subzonas, Este y Oeste. Estas subzonas pueden estar activas o no. Cada vez que un Driver o un Rider ingresen una zona de recogida o entrega, la aplicación validará la misma contra el estado de la zona y los polígonos de zonas definidos en el servidor DelRay.

Funcionalidad de autocompletado de Direcciones (Address autocomplete feature)

- La funcionalidad de autocompletado será soportada por la funcionalidad **Google Places Autocomplete**, con sus limitaciones de uso. Ver Restricciones, Supuestos y Convenciones
- Si el servicio de **Google Places Autocomplete** no está disponible, o se ha alcanzado el límite de uso, la función de autocompletado quedará deshabilitada.

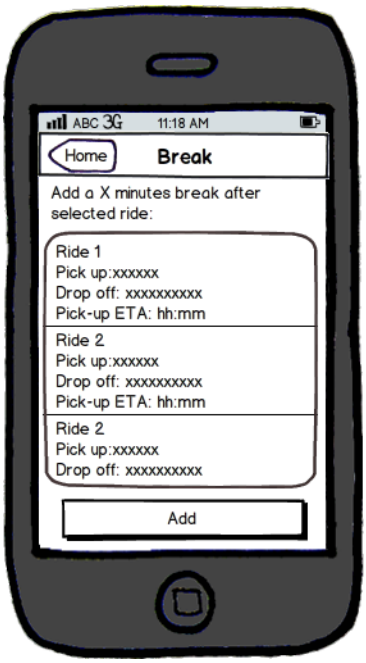
Login Screen (Driver)

- Forgot Password:
  - El Driver puede recuperar su password presionando el botón **Forgot Password**. El sistema enviará un email al Driver con una nueva password.
- Login Button:
  - El Driver ingresará sus credenciales y presionará el botón **Login**.
  - Si las credenciales ingresadas son válidas, el servidor enviará los parámetros de configuración de la aplicación y la pantalla Home Screen (Driver) será mostrada. El estado del Driver cambiará a **Available** en el servidor DelRay.
  - Si las credenciales ingresadas son inválidas, un mensaje de error aparecerá.

Home Screen (Driver)

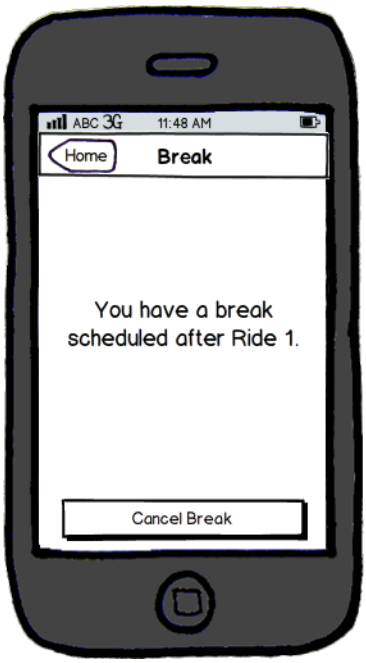
- La aplicación mostrará un indicador de carga hasta que la primera respuesta del **Pull Rides Service** retorne.
- Botón Offline:
  - Cuando es pulsado, la aplicación realiza un logout, para todos los procesos de pull y los servicios de rastreo, actualiza el estado del Driver a Offline en el servidor DelRay, y navega a la pantalla Login Screen (Driver), sus viajes son actualizados. Ver Terminología para las transiciones de estado de los viajes.
- Pending Rides:
  - Cuando se presiona, la aplicación navega a la pantalla Pending Rides
  - El botón contiene un ícono que indica el número de viajes pendientes. Dicho ícono es actualizado de acuerdo a la cantidad de viajes pendientes recividos mediante el **Pull Rides Service**.
- New Ride: Cuando es pulsado, navega a la pantalla New Ride Modal Screen
- Break: La aplicación navega a la pantalla Break Screen
  - Ver Terminología para las transiciones de estado del Driver
- Today's Rides List:
  - Lista que contiene los viajes completados.
  - Cuando se pulsa la fila de un viaje completado, la aplicación navega a Accepted Ride Detail Ride status is in Completed State

Break Screen



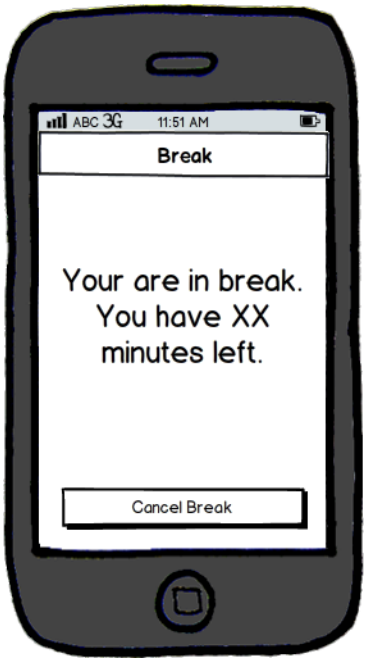
- Si existe una pausa de descanso programada, la pantalla Break Scheduled Screen se mostrará.
- Botón *Home*:
  - La aplicación navegará hacia atrás, a la pantalla Home Screen (Driver)
- Botón *Add*:
  - Cuando es pulsado, se muestra un diálogo de confirmación. Si el Driver confirma la pausa, la aplicación agendará una pausa de *X* cantidad de minutos luego del viaje seleccionado. Luego la aplicación navega a Home Screen (Driver).
  - La duración en minutos de la pausa es obtenida de los parámetros del sistema luego de que se realiza un login exitoso a la aplicación.

Break Scheduled Screen



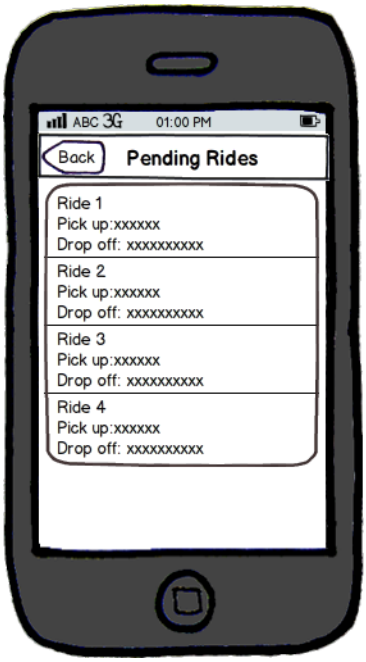
- Botón *Home*:
  - La aplicación navega de nuevo a la pantalla Home Screen (Driver)
- Botón *Cancel Break*:
  - Cuando es presionado, aparece un diálogo de confirmación. Si el Driver confirma la cancelación de la pausa, la aplicación cancelará la pausa agendada. Luego la aplicación navegará a Home Screen (Driver).

In Break Screen



- Cuando la aplicación detecta que es tiempo de ejecutar una pausa agendada y el Driver no tiene un viaje en estado *On The Way, Near Pick Up u On Ride*, se mostrará esta pantalla. Un mensaje indicando el inicio de la pausa se enviará al servidor DelRay, el estado del Driver se actualizará a *In Break* y todos los viajes pendientes del Driver se cambiarán al estado *Unassigned*, el agloritmo de asignación de viajes será ejecutado.
- Cuando la aplicación detecta que es tiempo de ejecutar una pausa agendada y el Driver tiene un viaje en estado *On The Way, Near Pick Up u On Ride*, la pausa no será ejecutada hasta que el viaje actual no sea cancelado o haya terminado.
- Cuando la aplicación verifica que el tiempo asignado a la pausa ya ha sido consumido, un mensaje indicando la finalización de la pausa será enviado al servidor DelRay y el estado del Driver se cambiará a *Online*.
- Botón *Cancel Break*:
  - Cuando es presionado, se muestra un mensaje de confirmación. Si el Driver confirma la cancelación de la pausa, la aplicación cancelará la misma, enviará un mensaje al servidor DelRay informando la finalización, y el estado del Driver se actualiará a *Online*. La aplicación navegará a la pantalla Home Screen (Driver).

Pending Rides



- Una lista con todos los viajes pendientes del día asignados al Driver será mostrada, ordenada como una cola fifo[5]. La lista se mostrará de acuerdo al **Pull Ride Service**. También será actualizada cuando el Driver cancele el viaje en el Accepted Ride Detail Ride status is not in Completed State

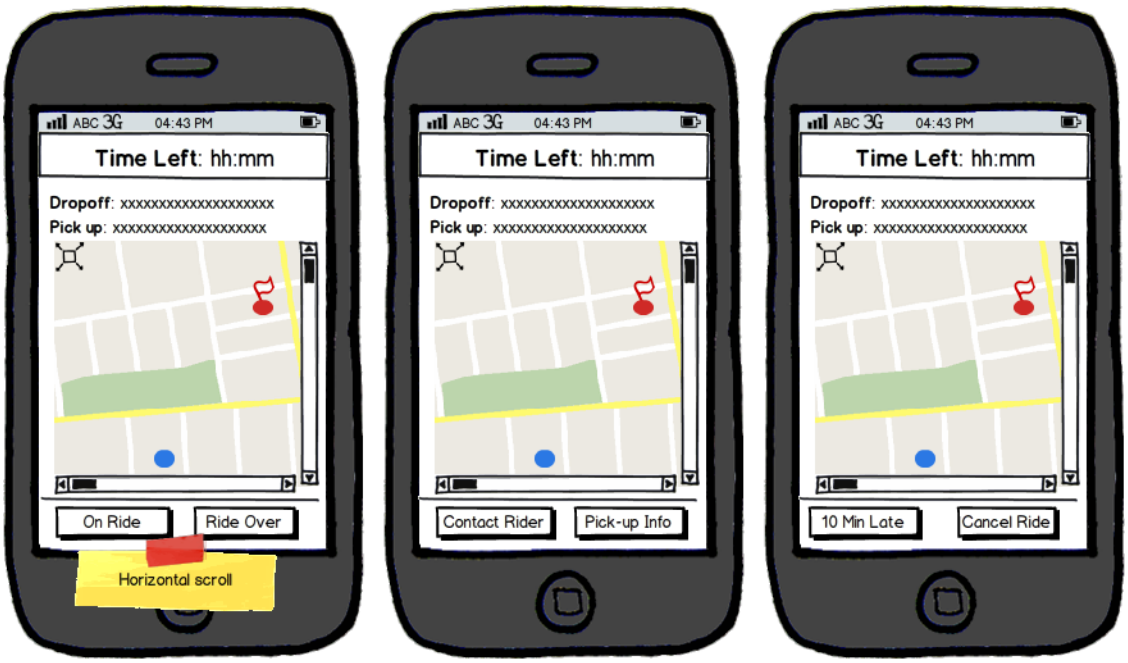
- Cuando se presione sobre una celda de un viaje, la aplicación navegará a la pantalla Pending Ride Detail.
- Botón *Back*: Navega nuevamente a la pantalla Home Screen (Driver)

Pending Ride Detail

[5]del inglés, **f**irst **i**n **f**irst **o**ut

- Pick up estimated time: El tiempo estimado de recogida para el viaje
- Mapa: La aplicación mostrará un mapa de Google Maps con la localización de recogida, la localización de entrega, y la posición actual del Driver. El mapa permite desplazamiento y zoom. En la posición superior izquierda del mapa, se mostrará un ícono que al ser presionado, destacará mediante acercamiento (zoom) la posición de recogida, entrega y localización del Driver.
- Accept Ride:
  - Solamente estará disponible si el viaje accedido era el que se encontraba en la posición superior de la lista fifo de viajes pendientes.
  - La aceptación es enviada al servidor DelRay, actualiando el estado del viaje a *On The Way*.
  - La aplicación navegará a Accepted Ride Detail Ride status is not in Completed State
- Skip Ride:
  - Cuando es presionado, el viaje evadido es enviado al servidor DelRay y el algoritmo de asignación es ejecutado.
  - La aplicación solicita una actualización del viaje y navega a la pantalla Pending Rides.

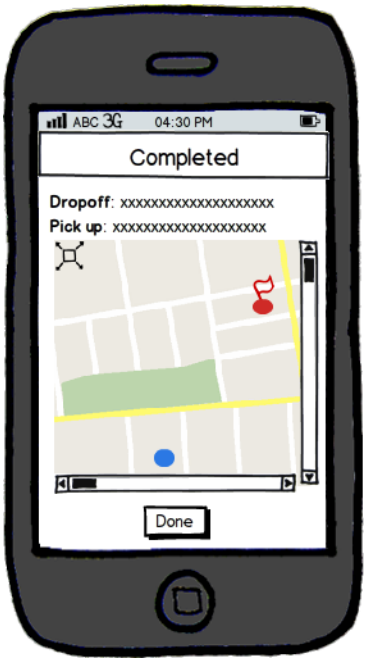
Accepted Ride Detail Ride status is not in Completed State



- Cuando la locación actual del Driver está cerca de la dirección de recogida, el servidor DelRay enviará una notificación de cercanía.
- Time Left: Mostrará el tiempo estimado en alcanzar la dirección de recogida. Es un timer decremental cuyo valor inicial es igual al tiempo estimado de arrivo.
- Mapa: La aplicación mostrará un mapra Google Map ubicado en la dirección de recogida, la dirección de destino y la locación actual del Driver. La locación del Driver es actualizada de acuerdo a la información provista por el sistema de rastreo. El mapa permite desplazamiento y acercamiento. En la esquina superior izquierda se mostrará un ícono que permitirá realizar acercamiento a las direcciones de recogida, destino y posición actual del Driver.
- Botón *On Ride*: Cuando es pulsado la aplicación actualizará el estado del viaje en el servidor DelRay a *On Ride*. El botón *On Ride* será deshabilitado y el botón *Rider Over* será habilitado.

- Botón *Rider Over*: Cuando sea presionado, la aplicación actualizará el estado del viaje a *Completed* en el servidor DelRay, la aplicación luego navegará a la pantalla Home Screen (Driver). El viaje aparecerá en el listado de los viajes diarios como completado.
- Botón *Contact Rider*:
  - La pantalla Contact Rider Modal Screen aparece.
  - El botón solamente se encuentra habilitado si el estado del viaje es *On Ride*.
  - El botón solamente se encuentra habilitado si el viaje no es creado por el Rider.
- *Pick-up Info*:
  - Cuando es presionado la aplicación mostrará un mensaje emergente con información adicional del campo *pick-up additional info*, referente a la dirección de recogida.
  - Si no hay información adicional, el botón estará deshabilitado.
- Botón *X Min Late*: Cuando es presionado un mensaje de confirmación es mostrado. Si el Driver lo confirma, X minutos adicionales se agregarán al tiempo estimado del viaje actual, y el servidor DelRay ejecutará el algoritmo de asignación de viajes. Este botón se encontrará solamente habilitado si el viaje está en estado *On The Way* o *Near Pick Up*.
- Los minutos a adicionar se obtendrán como parámetros del sistema.
- Botón *Cancel Ride*:
  - Cuando es presionado, el viaje será cancelado y dicha cancelación será enviada al servidor, actualizando el estado del viaje a *Canceled* y configurando el tiempo estimado de recogida a cero.
  - La aplicación navegará de vuelta a la pantalla Home Screen (Driver), y el viaje cancelado será retirado de la lista de viajes pendientes.
  - Este botón se encontrará habilitado solamente si el estado del viaje es *On The Way, Near Pick Up* o *At Pick Up*.

Accepted Ride Detail Ride status is in Completed State

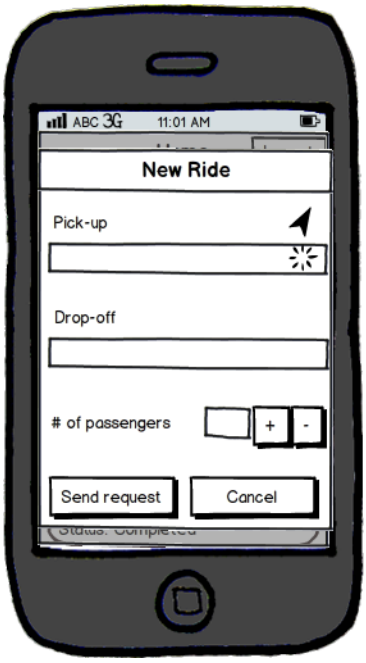


- Botón *Done*:
  - Cuando es pulsado, la aplicación navega de vuelta a la pantalla Home Screen (Driver).

Contact Rider Modal Screen

- *I'm Outside*: Un mensaje indicando que se está afuera, en la dirección de recogida es enviado al Rider. El mensaje se envía al servidor DelRay, y luego este es entregado como un SMS a traveś de una pasarela SMS.
- *Direct Call*: El Driver puede llamar al Rider para major información.
- *Cancel*: La pantalla *Contact Rider Modal Screen* desparece y se muestra Pending Ride Detail.

New Ride Modal Screen



- *Pickup*: El Driver escribe en el cuadro de texto la dirección de recogida. Una función de auto completado aparecerá a medida que se vaya escribiendo. La dirección actual en la que se encuentra el Driver se muestra por defecto.

- *Finde me*: La aplicación llenará el cuadro de texto con la dirección actual, usando el servicio de geo posicionamiento Google Geocoding, cuando una petición sea devulta por el servicio web Google Geocoding, se mostrará en el cuadro de texto y el indicador de carga se ocultará. El Driver podrá luego editar la dirección si la misma no es lo suficientemente precisa. Una función de autocompletado aparecerá a medida que el Driver ingrese la dirección. Si el Driver comienza a escribir en el cuadro de texto antes que el servicio de geolocalización devuelva una dirección, entonces la aplicación cancelará la llamada al servicio web Google Geocoding y esconderá el indicador de carga.
- *Number of passengers*: El Driver podrá seleccionar la cantidad de pasajeros para el viaje presionando los botones selectores "+" o "-".
- *Drop off*: El Driver ingresará la dirección de entrega, una función de autocompletado aparecerá a medida que el Driver vaya ingresando la dirección.
- *Send Ride*:
  - Cuando es presionada, la aplicacion verificará la información, si la información es correcta, la petición será enviada al servidor DelRay, en caso que sea incorrecta, aparecerá un mensaje de error.
  - Los valores correctos serán:
    * *Pickup*: no puede ser vacío, la dirección debe estar dentro de una zona DelRay activa.
    * *Number of passengers*: no puede ser vacío, el rango de valores válidos es de 1 a 5 (inclusive).
    * *Drop off*: no puede ser vacío, la dirección debe estar dentro de una zona DelRay activa.
    * Ver Terminología para los estados de las zonas DelRay.

6.3.4. Aplicaciones Web

Zonas

- Se va a definir una Zona de operaciones dentro de la cual va a ser posible recoger a un rider y depositarlo.
- La Zona de operaciones va a estar dividida en 2 Sub-Zonas a saber, Zona Este y Zona Oeste.
- Las Sub-Zonas pueden estar activadas o desactivadas;
- Los Viajes que tenga el punto de recogida o el punto de llegada dentro de una Zona inactiva serán rechazados.

Asignación de Viajes

- Algoritmo de asignación de viajes:
  - El algoritmo de asignación esta basado en un ejemplo enviado por el cliente
  - El servidor va a asignar el mejor Driver para un Viaje no asignado, basado en el mínimo tiempo estimado. Una vez asignado, el estado del Viaje pasara a Pendiente.
  - Algoritmo de asignación de viajes:
    * Para cada Conductor:
      · Obtener la lista de viajes asignados.
      · Obtener la ubicacion actual del conductor.
      · Ordenar los viajes por prioridad y tiempo.
      · Si el viaje esta en estado "Pending", "On The Way", "Near Pick-up", "at Pick-up": calcular un tiempo estimado desde la posicion actual del conductor al siguiente punto de recogida (primer viaje en la lista ordenada [R0]) [T0].
      · Si el viaje esta en estado "On Ride": Calcular un tiempo estimado desde la posicion actual del conductor hasta el siguiente punto de llegada (primer viaje en la lista ordenada [R0]) [T0].
      · Si el viaje esta en estado "Pending", "On The Way", "Near Pick-up", "at Pick-up": obtener el tiempo estimado entre el punto de recogida hasta el punto de llegada del primer viaje de la lista ordenada (previamente calculado) [R0] [T1].
      · Calcular el tiempo estimado entre el punto de llegada de [R0] y el punto de recogida del siguiente viaje en la lista ordenada [R1], [T2].

        - Obtener el tiempo estimado entre el punto de recogida y el punto de llegada de [R1], previamente calculado, [T3].
        - Repetir este procedimiente hasta el ultimo viaje de de la lista ordenada [Rn].
        - Sumar todos los tiempos: T0 + T1 +T2 + ... + Tn
    * El viaje sera asignado al conductor con menor tiempo estimado.

- Algoritmo de asignación de viajes, restricciones y supuestos:
  - El sistema no va a soportar viajes con múltiples puntos de recogida o destino. Ej.: 3 personas piden un viaje, pero uno de ellos desea bajarse del vehículo en un destino diferente al de los otros 2
- Un Pasajero o un usuario del Dashboard pide un nevo viaje:
  - Cuando un pasajero o un usuario del Dashboard crean un nuevo viaje el sistema lo registrar como un viaje en estado No Asignado / Unassigned.
  - El viaje es asignado automáticamente a un conductor siguiendo el algoritmo de asignación de viajes descripto anteriormente. Cuando el viaje es asignado a un conductor, su estado pasara a ser Pendiente / Pending.
- Un conductor crea un nuevo viaje:
  - El estado del viaje recientemente creado sera On Ride / En viaje.
  - El conductor puede crear un nuevo viaje solamente si todos los viajes asignados a el están en estado Pendiente / Pending o si no tiene ningún viaje asignado. Si el conductor esta en medio de un viaje y levanta un nuevo pasajero en la calle, solamente podrá crear el nuevo viaje, para el nuevo pasajero recién cuando el el viaje actual finalice.
  - Si un conductor crea un nuevo viaje, los viajes que tenga asignados y en estado Pendiente, se les des asignarán y el servidor volverá a correr el algoritmo de asignación de viajes.
- Lista de viajes pendientes de un conductor:
  - El conductor tendrá una lista de viajes pendientes. Esta lista estará ordenada como una cola (primero en entrar primero en salir)
  - El conductor puede aceptar solo el viaje que esta primero de esta lista. Aceptar el viaje conlleva cambiar el estado del viaje de Pending / Pendiente a On the Way / En Camino.
  - El conductor pude saltearse el primer viaje de la lista. El mismo cambiará a estado Unassigned / No Asignado y el servidor volverá a correr el algoritmo de asignación de viajes.
- El conductor solicita un descanso:
  - El conductor puede solicitar u descanso programado.
  - El conductor añadirá 10 minutos al tiempo estimado de arribo del siguiente viaje.
  - Si el conductor añade estos 10 minutos, todos los viajes asignados a el pasaran a estado Unassigned y el servidor volverá a correr el algoritmo de asignación de viajes.
- El conductor esta en el punto de recogida y el pasajero tarda mucho en llegar:
  - Si el conductor espera mas de X cantidad de minuto en el punto de recogida y luego pasa el viaje a estado "On Ride", todos los viajes pendientes del conductor pasaran a estado "No Asignado" y el servidor correrá de nuevo el algoritmo de asignación.
- El conductor esta "Fuera de Linea":
  - Un conductor puede estar "Fuera de Linea" por uno de los siguientes motivos:
    * El conductor se "deslogueo" de la aplicacion.
    * El Administrador edl sistema cambio el estado de un conductor a "Fuera de Linea"
  - Cuando un usuario entra al estado "Fuera de Linea" todos los viajes que tenga asignados pasaran a estado "No asignado" y el servidor volverá a correr el algoritmo de asignación de viajes.
- El Conductor, el Pasajero o el Administrador cancela un viaje:

  - Todos los viajes pendientes del conductor serán movidos al estado "No asignado" y se volverá a correr el algoritmo de asignación de viajes en el servidor.

- Se cambia el punto de recogida desde el Dashboard:
  - Un usuarios del Dashboard puede cambiar el punto de recogida si el viaje esta en estado "Pendiente" o "En Camino"
  - Todos los viajes del conductor cambiaran al estado "No asignado" y se volverá a correr el algoritmo de asignación de viajes.
- Se cambia el punto de destino desde el Dashboard:
  - Un usuario del Dashboard puede cambiar el punto de destino de un viaje siempre que el mismo se encuentre en estado "Pendiente", "En Camino", "Cerca del Punto de Recogida", "En el Punto de Recogida" o "En Viaje".
  - Todos los viajes del conductor cambiaran al estado "No asignado" y se volverá a correr el algoritmo de asignación de viajes.
- Notar que el algoritmo de asignación de viajes se volverá a ejecutar cada vez que ocurra un cambio en algún viaje que pueda afectar los tiempos estimados de recogida o de llegada.

Funcionalidad de Autocompletado de Direcciones

- La funcionalidad de Autocompletado de Direcciones va a ser implementada mediante "Google Places Autocomplete", con sus respectivas limitaciones. Ver Restricciones, Supuestos y Convenciones
- Si el servicio de Autocompletado de Google Places no esta disponible, o si se ha superado el limite previsto para el mismo, la funcionalidad de autocompletado sera desactivada.

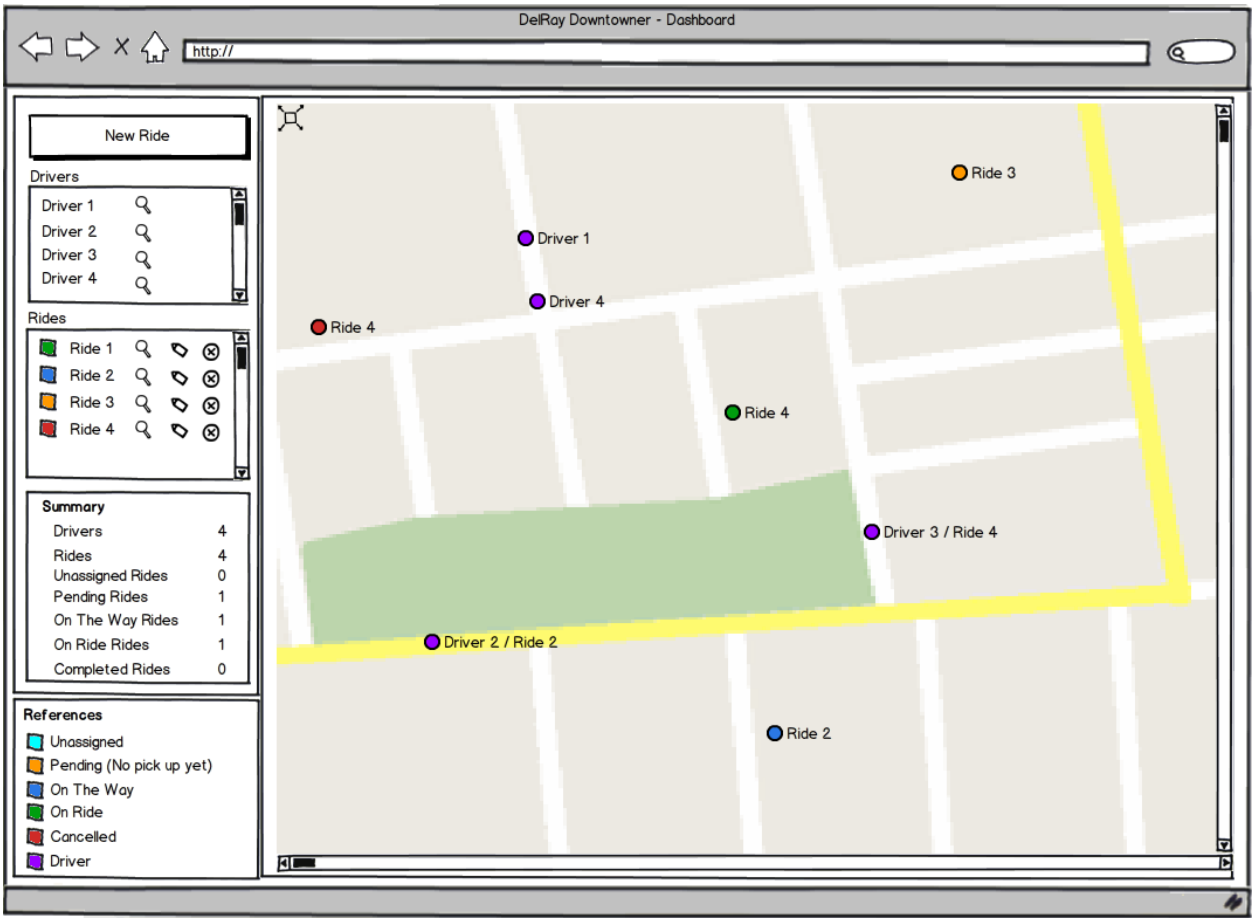
Funcionalidad de Promociones

- Con el objetivo de proveer al administrador del sistema una alta flexibilidad a la hora de diseñar los componentes visuales de las promociones, la misma contendrá el código fuente de una pagina HTML.
- El cliente es el responsable de diseñar, crear y proveer el código HTML de las promociones al sistema.
- La aplicación de "Rider" mostrara el contenido de la promoción como una pagina web embebida.

Login Screen

- Botón *Forgot password*:
  - El usuario puede recuperar su contraseña de acceso presionando este botón. El sistema enviará un correo electrónico al usuario con una nueva contraseña.
- Botón *Login*:
  - El usuario ingresa sus credenciales y presiona dicho botón.
  - Si las credenciales del usuario son válidas, el server navegará a la pantalla Dashboard Screen si el usuario tiene un rol de Dashboard, si tiene rol de Administrador navegará a la pantalla Admin Home Screen.
  - Si las credenciales del usuario no son válidas, se mostrará un mensaje de error.

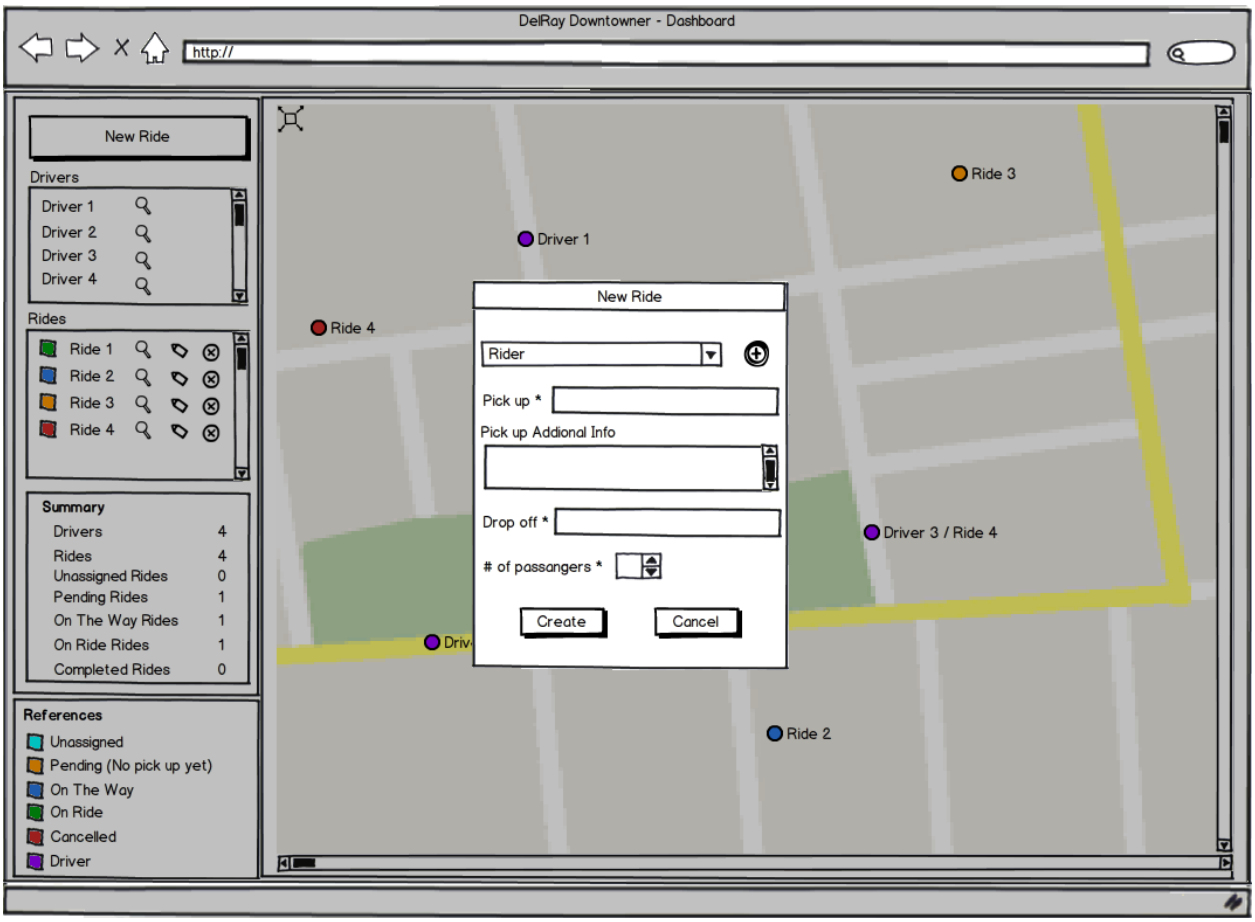
Dashboard Screen



- Mapa:
  - El mapa es actualizado cada una x cantidad de segundos, o cuando el usuario refresca la pantalla del navegador utilizando la tecla F5.
  - El usuario puede realizar scroll y zoom sobre el mapa. Cuando se presiona el ícono superior izquierdo, se realizará un zoom en el mapa hasta que tanto el conductor como el pasajero quepan en la vista.
  - Los viajes en estado *Cancelled* o *Completed* no se mostrarán en el mapa.
  - Los conductores que no estén logueados en la aplicación (cuyo estado sea *Offline*) no se mostrarán en el mapa.
  - Presionado sobre el ícono de un conductor (*Driver*) se mostrará un diálogo popup con el estado del *Driver* (*Available*, *In Break*), el identificador del viaje actual asignado, la dirección donde se encuentra, la dirección de recogida (*pick-up*) y la dirección de destino (*drop-off*).
  - Presionando sobre el ícono de un pasajero (*Rider*) se mostrará un diálogo popup con el estado del viaje, las direcciones de recogida y destino, y el Id del conductor asignado.
- Botón *New Ride*:
  - Cuando es presioando se muestra el popup Dashboard New Ride Screen.
- Lista *Drivers*:
  - Todos los conductores se muestran ordenados por estado: *Online*, *In Break*, *Offline*
  - Presionando sobre el ID del *Driver* el mapa se centrará en él (Solamente se realizará sobre los drivers que no estén en estado *Offline*).

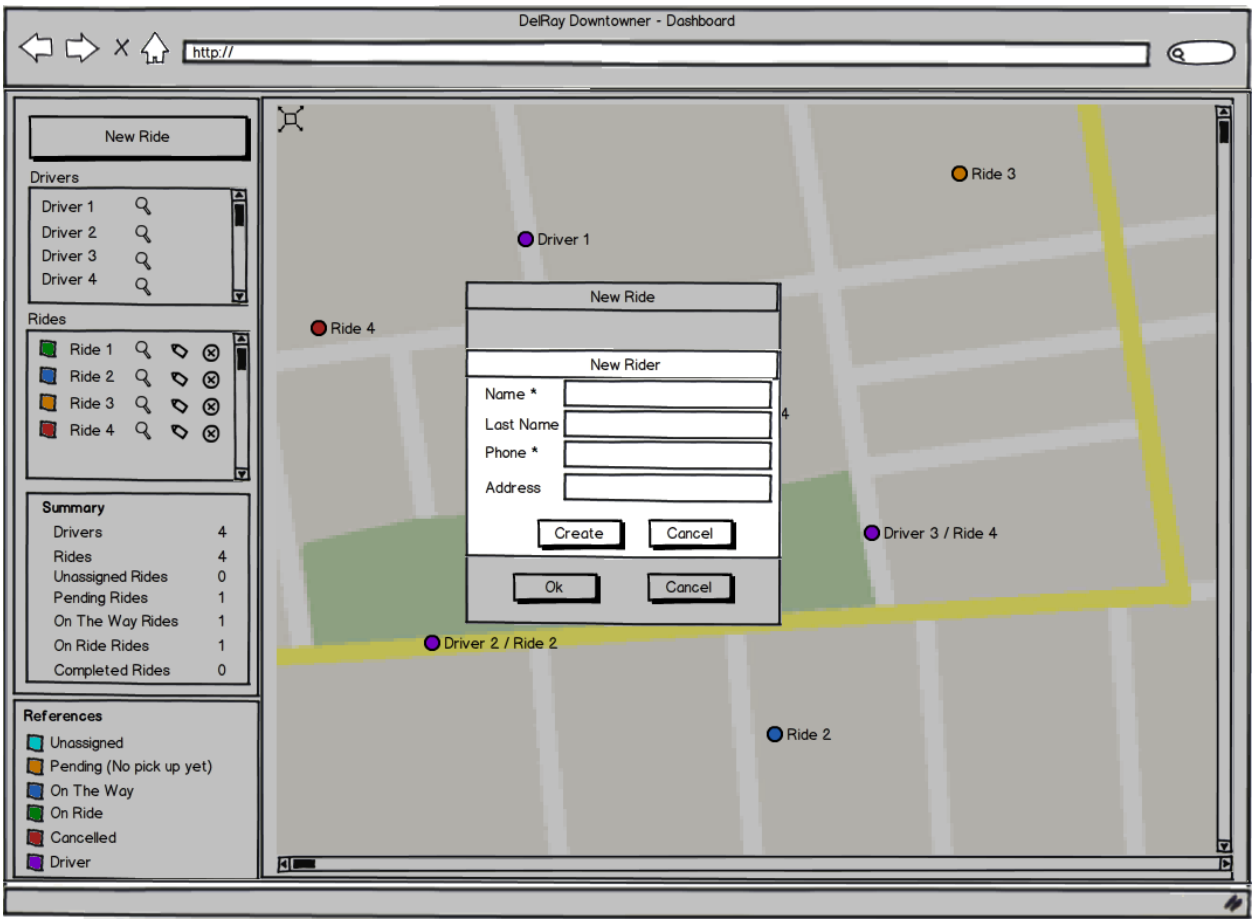
- Botón *Driver Details*:
  - Cuando es presionado se muestra el popup Dashboard Driver Details Screen.
- Lista *Rides*:
  - Se muestran todos los viajes ordenados por estado y fecha de creación.
  - Presionando sobre el identificar de un viaje, el mapa se centrará en él (aplica solamente sobre viajes que no estén en estado *Canceled* o *Completed*).
- Botón *Ride Detail*:
  - Cuando es presionado se muestra el popup Dashboard Edit Ride Screen.
- Botón *Cancel Ride*:
  - Cuando es presionado un diálogo de confirmación aparece.
  - Si el usuario confirma la cancelación, el viaje cambiará de estado a *Cancelled*, sin importar el estado en el que se encuentre dicho viaje.
- Summary:
  - El sumario es actualizado cada x cantidad de segundos, o bien cuando el usuario fuerza el refresco del navegador presionando la tecla F5.
    * El campo *Driver* mostrará la cantidad de **Drivers** que no estén deslogueados.
    * El campo *Rides* mostrará un contador con todos los viajes del día.

Dashboard New Ride Screen



- Rider Combo Box:
  - Lista el nombre y apellido de los conductores registrados in orden alfabético, tomando el campo nombre.
  - Si no hay conductores registrados, el combo box se mostrará vacío.
- Botón New Rider:
  - Cuando es presionado el popup Dashboard New Rider Screen se muestra.
- Number of passangers:
  - El usuario puede seleccionar el número de pasajeros para el viaje.
- Pick up:
  - El usuario ingresa en el cuadro de text la dirección de recogida. Una función de autocompletado asistirá al usuario a medida que escribe la dirección. Estará deshabilitada si el estado del viaje es *On Ride*.
- Pick up Additional Info:
  - El usuario puede editar la información adicional de recogida.
- Drop off:
  - El usuario ingresará la dirección de destino, una función de autocompletado asisitirá al usuario a medida que ingresa la información.
- Botón *Create*:
  - Cuando es presionado, la aplicación valida la información. Si la información es correcta los valores del viaje son guardados, el diálogo de confirmación se oculta y un refresco del dashboard es solicitado. En caso contrario, un mensaje de error aparece.
    * Los valores válidos son:
      · Pick up: No puede ser vacío, la dirección debe estar dentro de una zona DelRay activa.
      · Number of passengers: No puede ser vacío, el rango de valores válidos es entre 1 y 5 inclusive.
      · Drop off: No puede ser vacío, la dirección debe estar dentro de una zona DelRay activa.
      · Pick Up y Ride ETA: El usuario debe ser capaz de poder agregar o sustraer hasta 30 minutos. El Valor de ETA no puede ser menor a 5 minutos.
- Botón *Cancel*:
  - Cuando es presionado, el popup se oculta.

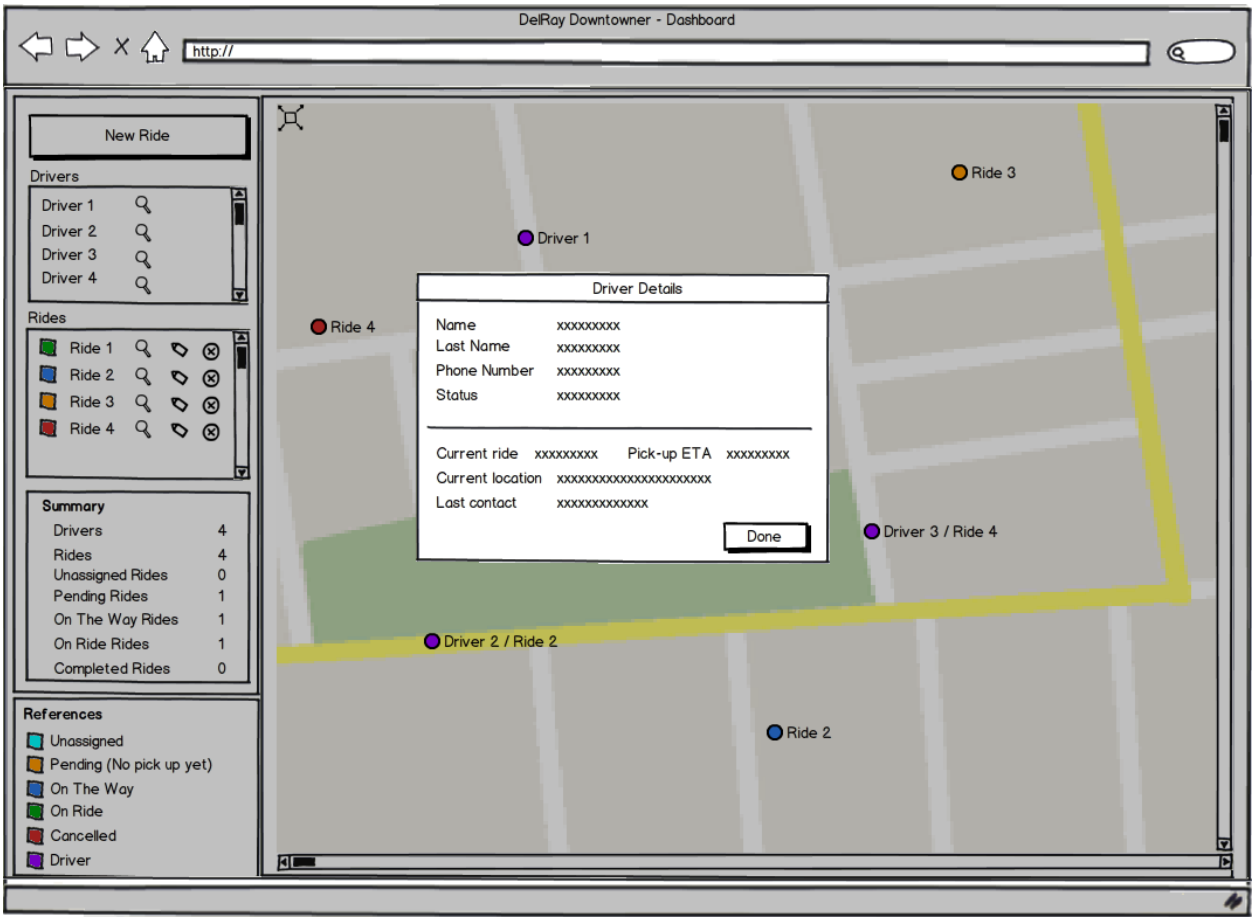
Dashboard New Rider Screen



- Name:
  - El usuario escribe en el textbox el nombre del *Rider*
- Last Name;
  - El usuario escribe en el textbox el apellido del *Rider*
- Phone Number:
  - El usuario escribe en el textbox el número de teléfono del *Rider*
- Address:
  - El usuario escribe en el textbox la dirección del *Rider*. Una función de autocompletado asistirá la entrada de la dirección a medida que se vaya escribiendo.
- Botón *Create*
  - Cuando es presionado, la aplicación valida los datos. Si los datos son correctos, el nuevo *Rider* es guardado y el popup se oculta. Si no es válido, se muestra un mensaje de error.
    * Los *valores válidos* son:
      · Name: Cualquiera caracter, cadenas vacías no son válidas.
      · Last Name: Cualquier caracter.
      · Phone Number: Cualquier número de teléfono de Estados Unidos, no se permiten cadenas vacías.
      · Address: Cualquier caracter. La dirección debe estar dentro de una zone DelRay activa.

- Botón *Cancel*
  - Cuando es presionado, el popup se oculta.

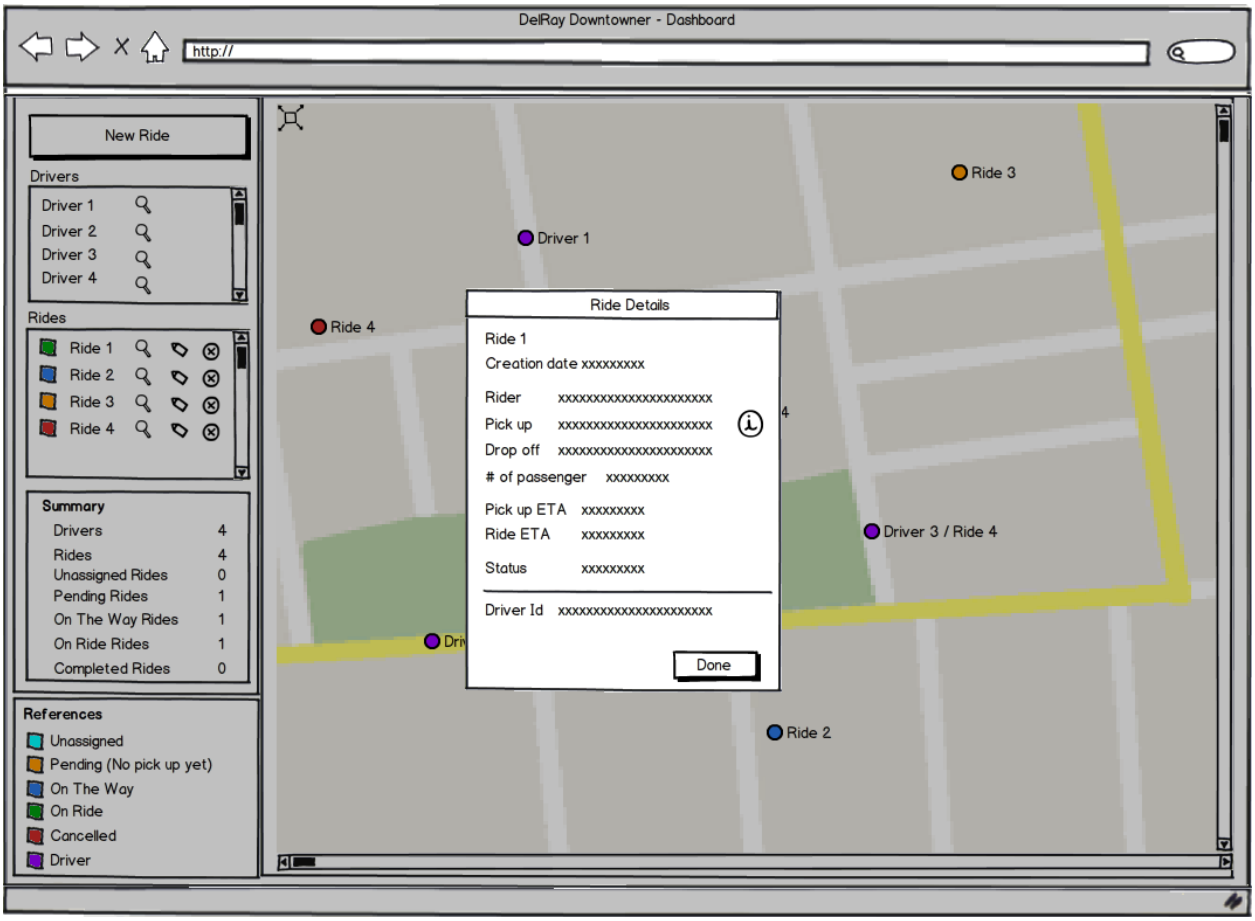
Dashboard Driver Details Screen



- Name:
  - El nombre del conductor.
- Last Name:
  - El apellido del conductor.
- Phone Number:
  - El número de teléfono del conductor.
- Status: El estado actual del conductor.
- Current Ride:
  - El identificador del viaje actual.
- Pickup ETA:
  - El tiempo estimado que se demorará en recojer a los pasajeros.
  - Si el estado del viaje es *On Ride*, *Completed*, o *Cancelled* este campo no es mostrado.
- Last contact:

  - El día y hora en que la aplicación del *Driver* fue contactada por última vez.
- Botón *Done*:
  - Cuando es presionado, el popup se oculta.

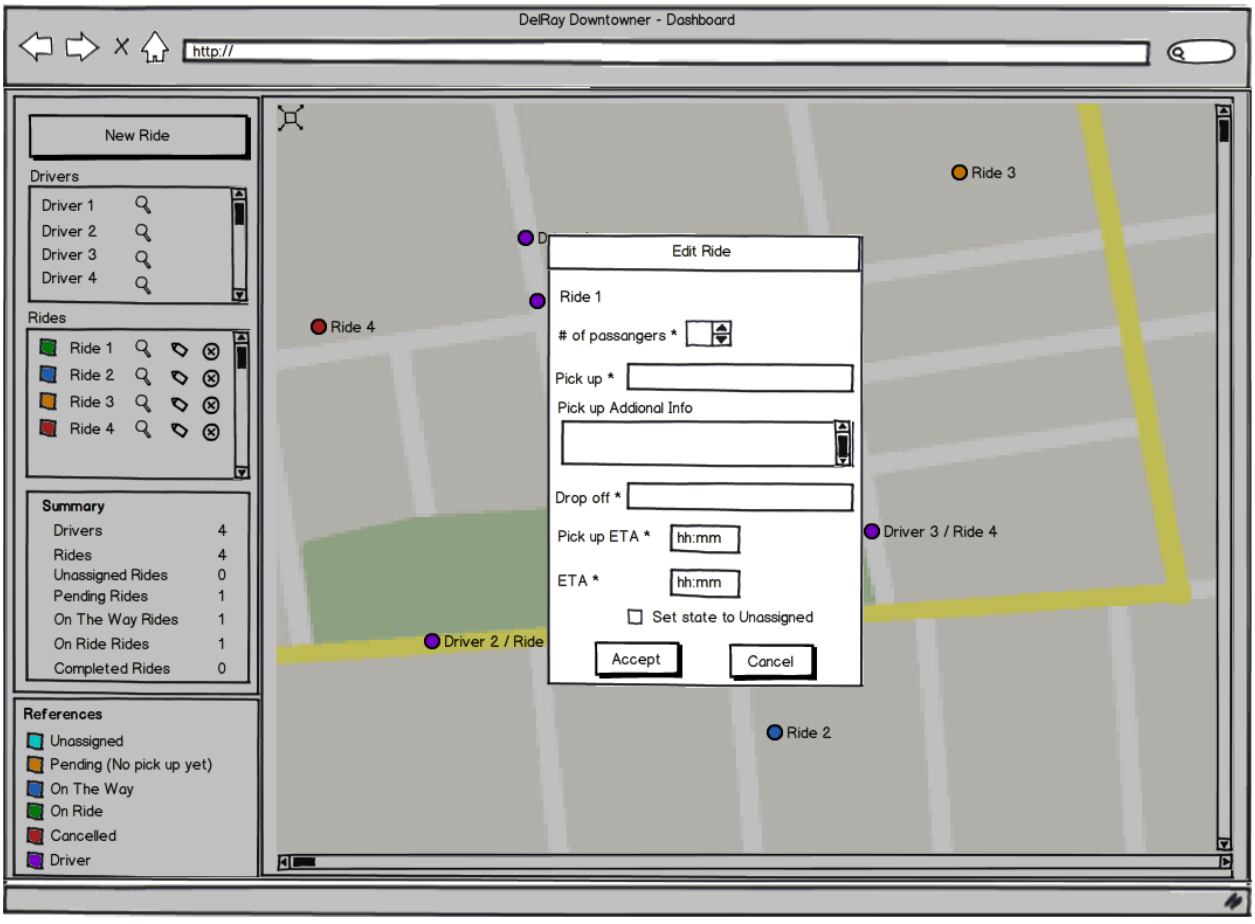
Dashboard Ride Details Screen



- Creation date:
  - La fecha y hora en la que el viaje fue creado.
- Rider:
  - El nombre y apellido del *Rider*.
- Pick up:
  - La dirección de recogida.
  - El botón *info* es habilitado solamente si el viaje tiene información adicional.
  - Presionar el botón *info* la información adicional del viaje se mostrará en un diálogo popup.
- Drop off:
  - La dirección de destino del viaje.
- Number of passangers:
  - El número de pasajeros para el viaje.

- Pick up ETA:
  - El tiempo estimado que se demorará en recojer a los pasajeros.
- Ride ETA:
  - El tiempo estimado de duración del viaje.
- Status:
  - El estado del viaje.
- Driver Id:
  - El identificador del *Driver* asignado al viaje.
- Botón *Done*:
  - Cuando es presionado el diálogo popup se oculta.

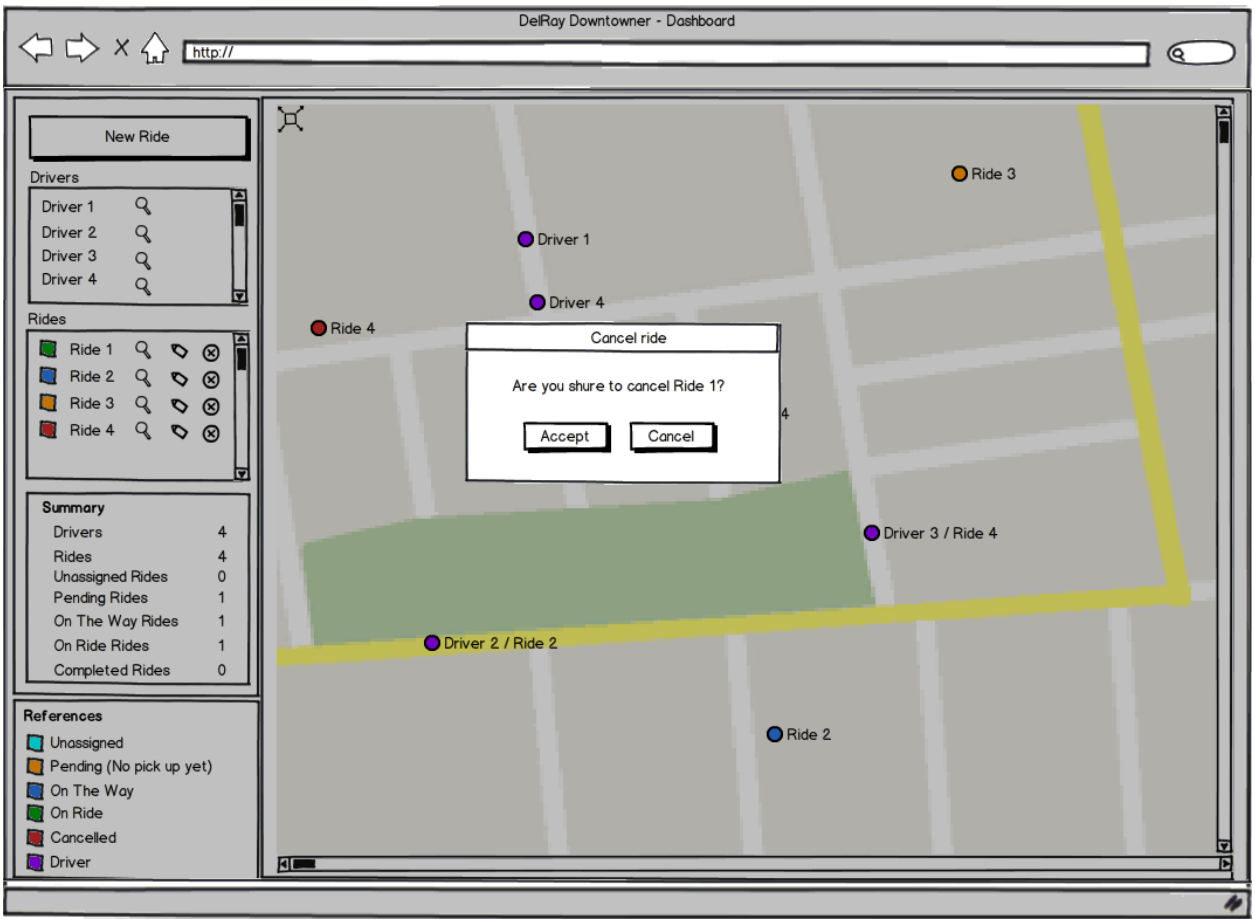
Dashboard Edit Ride Screen



- Un viaje se puede editar solamente si no está en estado *Completed*.
- Ride:
  - El identificador del viaje
- Number of passangers:

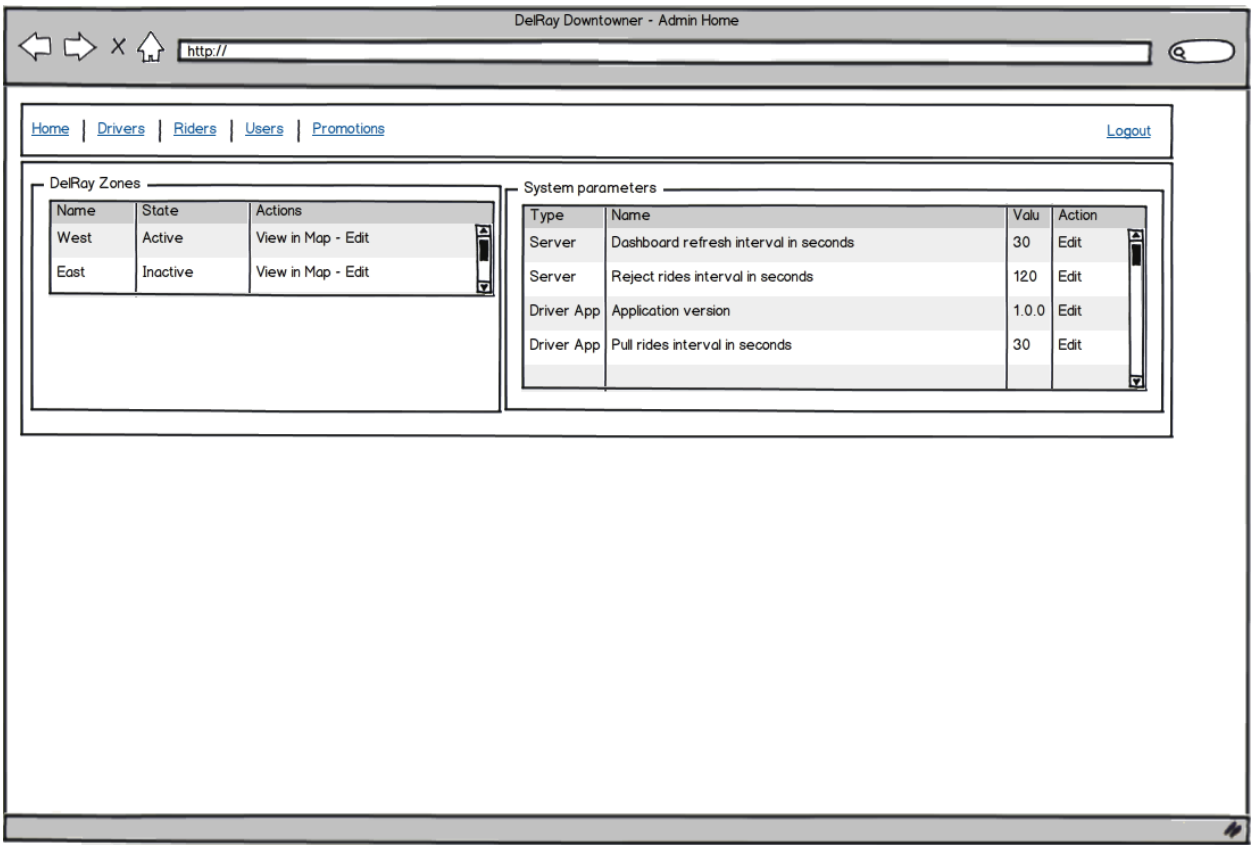
  - El usuario puede editar la cantidad de pasajeros para el viaje.
- Pick up:
  - El usuario ingresa en el cuadro de text la dirección de recogida. Una función de autocompletado asistirá al usuario a medida que escribe la dirección. Estará deshabilitada si el estado del viaje es *On Ride*.
- Pick up Additional Info:
  - El usuario puede editar la información adicional de recogida.
- Drop off:
  - El usuario ingresará la dirección de destino, una función de autocompletado asisitirá al usuario a medida que ingresa la información.
- Pick up ETA:
  - El tiempo estimado que se demorará en la recogida del Rider. Este valor estará deshabilitado si el viaje está en estado *On Ride* o *On Pick Up Location*.
- Ride ETA:
  - El valor estimado que se demorará en llegar a destino.
- Checkbox Status:
  - Este checkbox solamente estará habilitado si el viaje no está en estado *On Ride* o *Completed* (ver Terminología para información referente a los estados del viaje).
- Botón *Accept*:
  - Cuando es presionado, la aplicación valida la información. Si la información es correcta los nuevos valores del viaje son guardados, el diálogo de confirmación se oculta y un refresco del dashboard es solicitado. En caso contrario, un mensaje de error aparece.
    * Los valores válidos son:
      · Pick up: No puede ser vacío, la dirección debe estar dentro de una zona DelRay activa.
      · Number of passengers: No puede ser vacío, el rango de valores válidos es entre 1 y 5 inclusive.
      · Drop off: No puede ser vacío, la dirección debe estar dentro de una zona DelRay activa.
      · Pick Up y Ride ETA: El usuario debe ser capaz de poder agregar o sustraer hasta 30 minutos. El Valor de ETA no puede ser menor a 5 minutos.
- Botón *Cancel*:
  - Cuando es presionado, el popup se oculta.

Dashboard Cancel Ride Screen



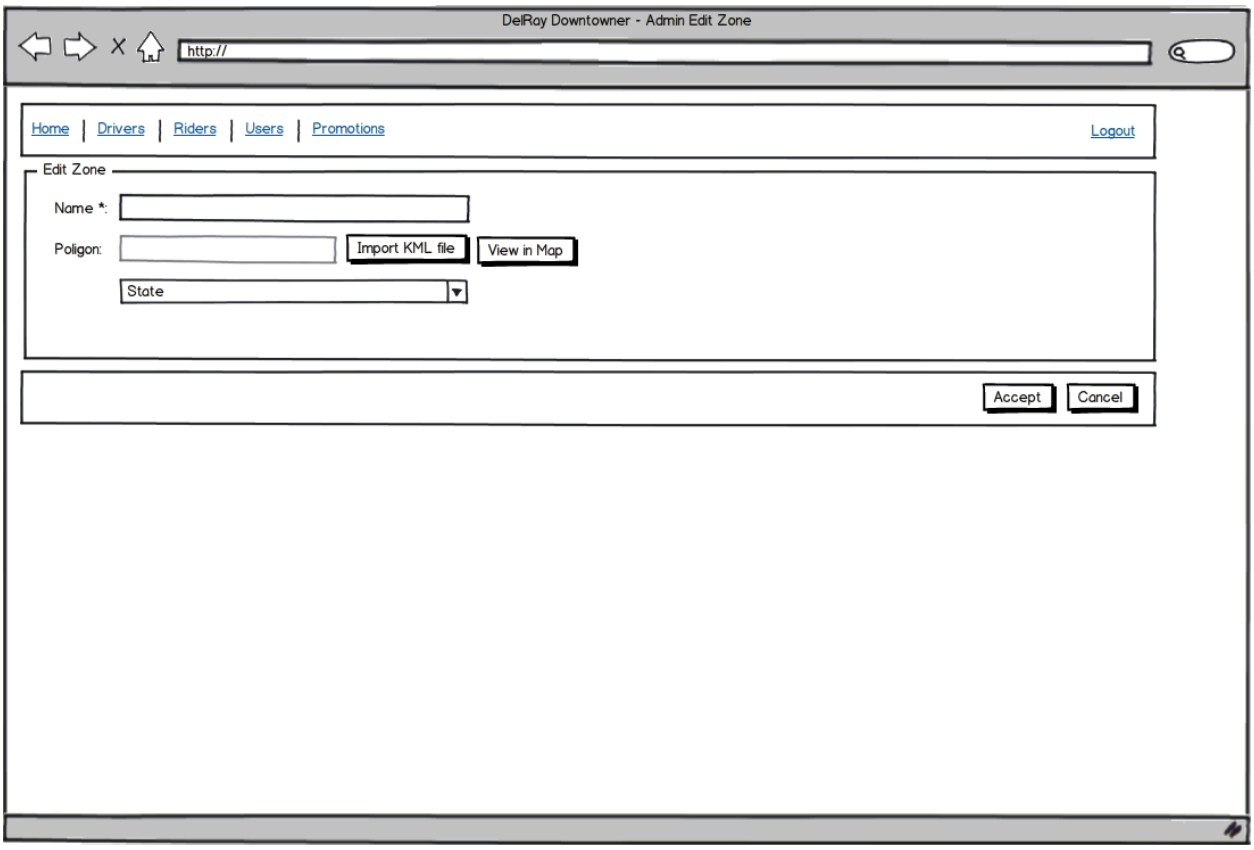
- Botón *Accept*:
  - Cuando es presionado, el viaje se cancela actualizando su estado a *Cancelled* y cargando el valor *Ride Pick Up estimated time* a cero. El diálogo de confirmación se cierra y el dashboard se actualiza.
- Botón *Cancel*:
  - Cuando es presionado, el diálogo de confirmación se oculta.

Admin Home Screen



- DelRay Zones:
  - Muestra un listado con las zonas Este y Oeste.
  - Presionando en *View in Map* un navegador es abierto motrando el polígono delimitador de zona en Google Maps.
  - Presionando en *Edit* la aplicación navega a la pantalla Admin Edit Zone Screen.
- System parameters:
  - Muestra un listado de los parámetros del sistema. Presionando en *Edit* la aplicación navega a la pantalla Admin Edit System Parameter Screen.

Admin Edit Zone Screen

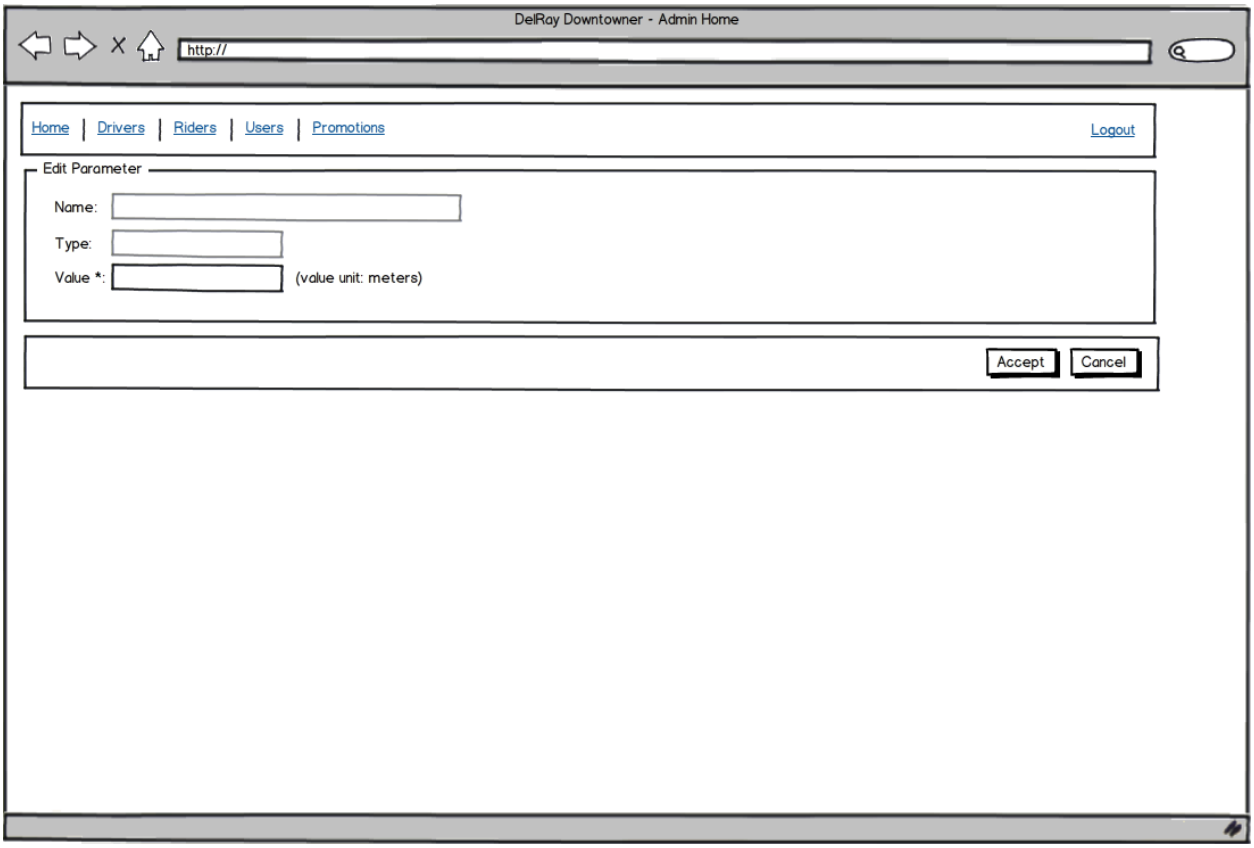


- Name:
  - El nombre de la zona actual es mostrado. El usuario puede ediar el nombre de la zona.
- Poligon:
  - Muestra el nombre actual del archivo de polígono delimitador de la zona. No puede ser editado.
- Import KML file:
  - Importa el archivo que define el poligono que delimita una zona. Este archivo debe ser generado con Google Earth y exportado en formato KML.
- View in Map:
  - Un nuevo navegador es abierto con Google Maps, mostrando el polígono que delimita la zona.
- Status:
  - Permite cambiar el estado de la zona al valor Activo o Inactivo (*Active* o *Inactive*).
- Botón Aceptar:
  - Al presionarlo muestra un diálogo de confirmación. Si el usuario confirma dicho diálogo, la aplicación valida los datos, si son correctos, guarda los nuevos valores y envía un SMS de notificación a los Drivers y a los Riders, solamente los Riders que tengan activada la notificación por SMS en la aplicación y hayan utilizado el servicio las últimas 48 horas serán notificados; la aplicación navegará luego a la pantalla Admin Home Screen. Si los datos no son válidos, se mostrará un mensaje de error. En caso que el usuario no confirme, el diálogo de confirmación se ocultará.
  - Los valores válidos son:
    * Name: no puede ser vacío, debe ser de valor único.

    * KML file: el servidor validará que el fichero sea de formato KML.
    * Ver Terminología para información de los estados de las zonas.

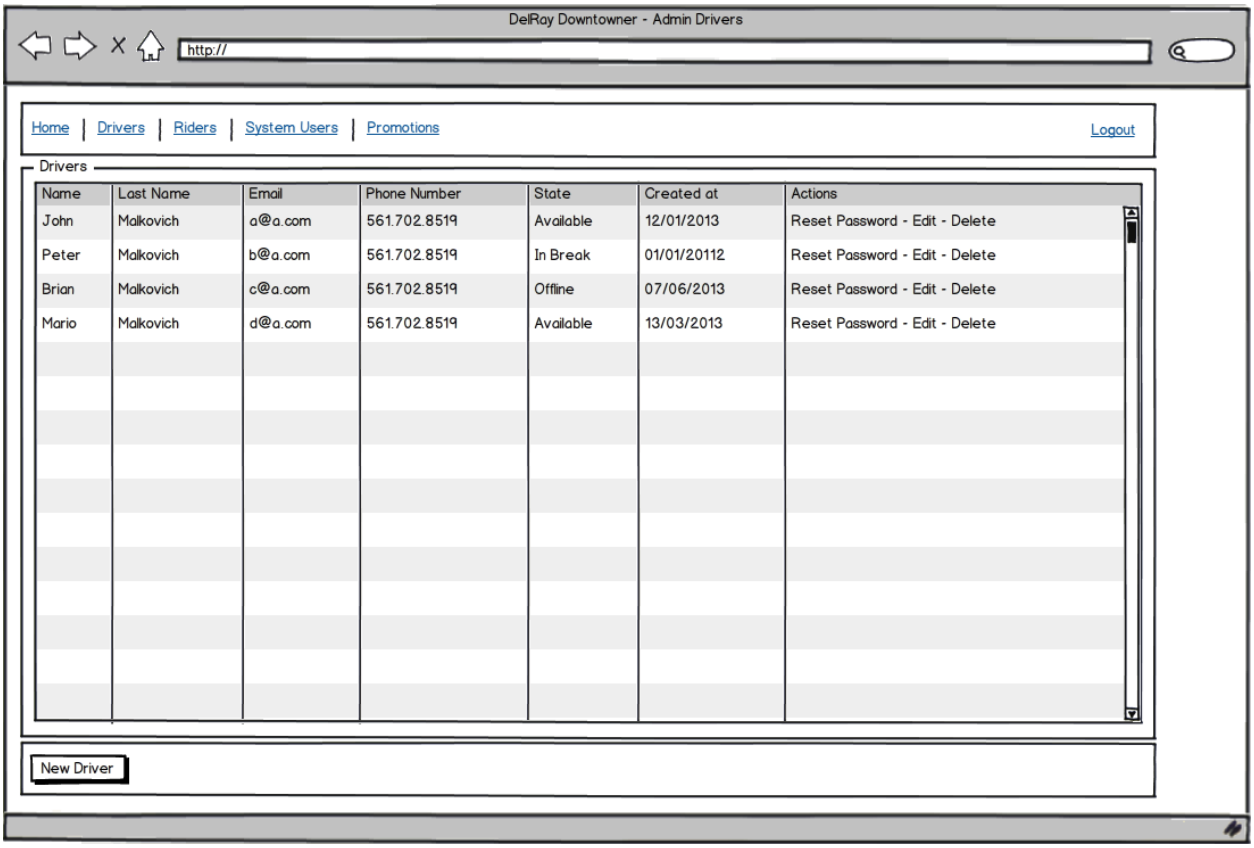
- Botón Cancel:
  - Cuando es presionado, la aplicación navega a la pantalla Admin Home Screen.

Admin Edit System Parameter Screen



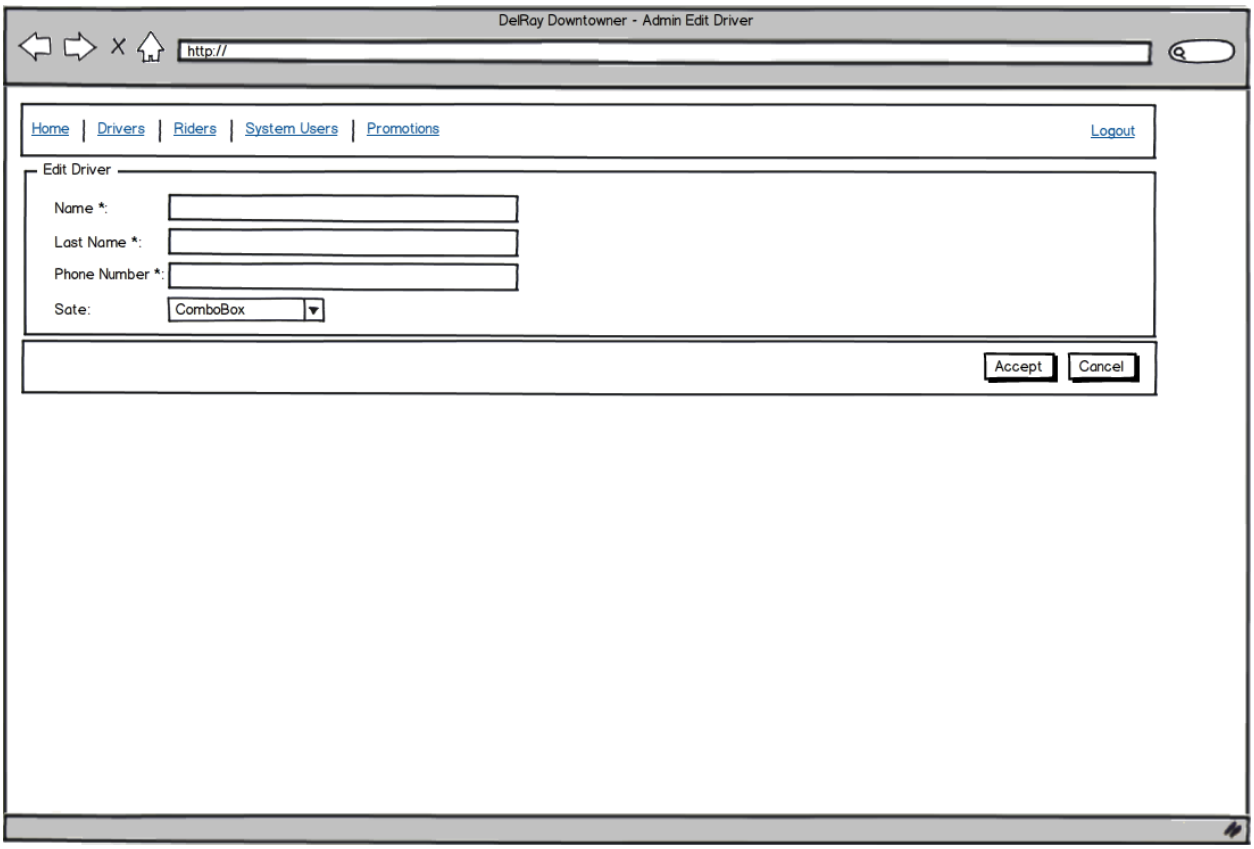
- Name:
  - El parámetro nombre se muestra, no puede ser editado.
- Type:
  - El parametro tipo se muestra, no puede ser editado.
- Value:
  - El valor actual del parámetro se muestra. El usuario puede ingresar un nuevo valor.
- Botón Aceptar:
  - Cuando es presionado un diálogo de confirmación se muestra. Si el usuario lo acepta el sistema valida los datos. Si los datos son correctos, el nuevo parámetro cargado en *Value* es guardado y la aplicación navega a la pantalla Admin Home Screen. En caso que los parámetros no sean válidos, se mostrará un mensaje de error. Si el usuario no realiza confirmación el cuadro de diálogo se oculta.
- Botón Cancel:
  - Cuando es presionado, la aplicación navega a Admin Home Screen.

Admin Drivers Screen



- Drivers:
  - Un listado con los *Drivers* se mostrará ordenado por el estado (Available, In Break, Offline).
- Reset Password:
  - Presionando en **Reset Password** se envía un correo electrónico al *Driver* con una nueva contraseña.
- Edit:
  - Presionando en **Edit** la aplicación navega a la pantalla Admin Edit Driver Screen.
- Delete:
  - Presionando en **Delete** un diálogo de confirmación se mostrará. Si el usuario lo acepta, el *Driver* será borrado.
- New Driver:
  - Presionándolo, la aplicación navega a la pantalla Admin New Driver Screen.

Admin Edit Driver Screen

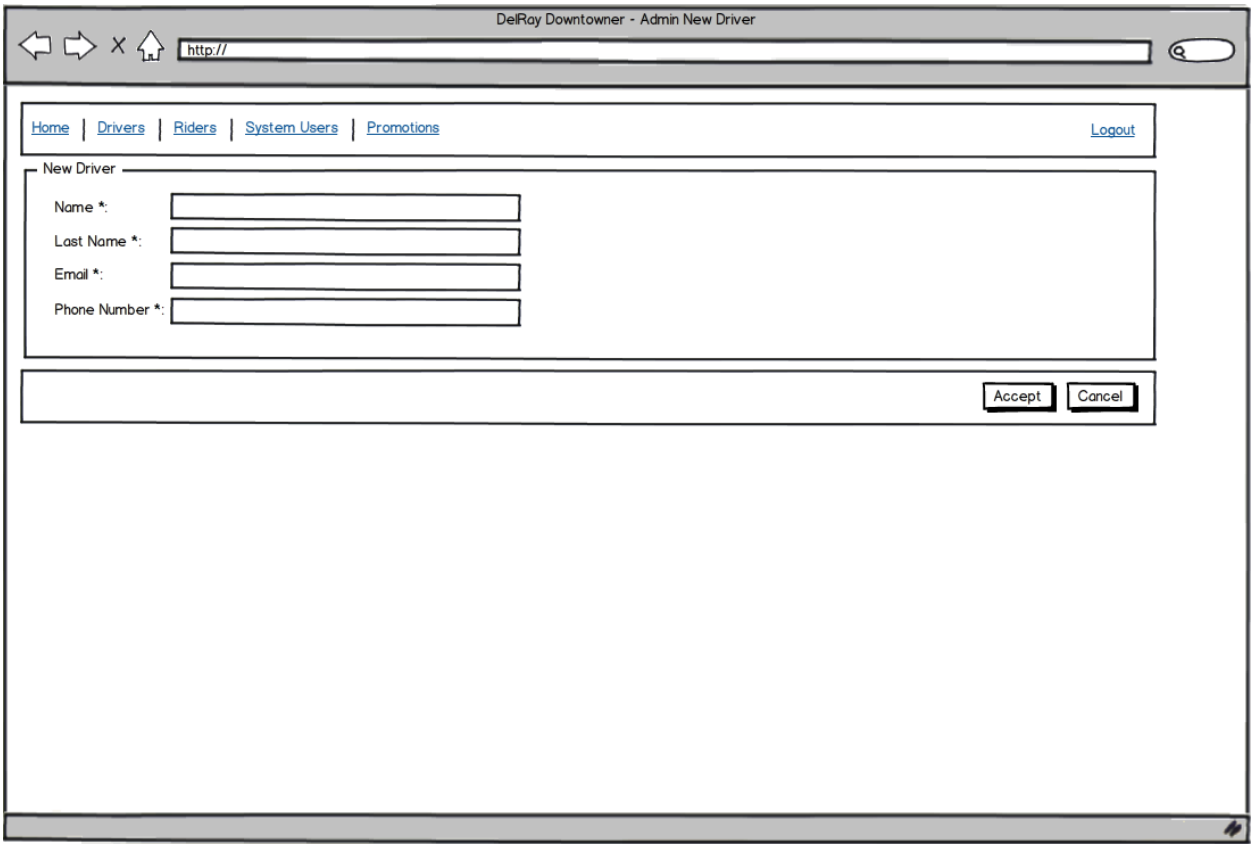


- Name:
  - Muestra el valor actual cargado como Nombre del Driver. El usuario puede ingresar un nuevo valor.
- Last Name:
  - Muestra el valor actual cargado como Apellido del Driver. El usuario puede ingresar un nuevo valor.
- Phone Number:
  - Muestra el valor actual cargada como número de teléfono del Driver. El usuario puede ingresar un nuevo valor.
- Estado:
  - Muestra el estado cargado para el Driver. El usuario puede cambiar el estado solamente al valor **Offline**. Al realizar el cambio de estado, la siguiente acción que la aplicación Driver realiza contra el servidor DelRay será rechazada, el servidor entonces ordenará a la aplicación móvil Driver a cerrar la sesión abierta (Logout).
- Botón Accept:
  - Cuando es presionado muestra un diálogo de confirmación. Si el usuario lo confirma, la aplicación valida los datos, si los mismos son correctos los valores del nuevo *Driver* son guardados y la aplicación navega a la pantalla Admin Drivers Screen; en caso que no confirme, un mensaje de error es mostrado. Si el usuario no realiza ninguna acción, el diálogo de confirmación se oculta.
  - Los *valores* se consideran *válidos* si:
    * Name and Last Name: No es una cadena vacía, se aceptan valores duplicados para nombre y apellido.

    * Email: No es una cadena vacía, puede ser cualquier dirección de correo electrónico, su valor debe ser único.
    * Phone Number: cualquier número de teléfono de Estados Unidos, valores vacíos no son permitidos.

- Cancel Button:
  - Cuando se presiona la aplicación navega a la pantalla Admin Drivers Screen.

Admin New Driver Screen

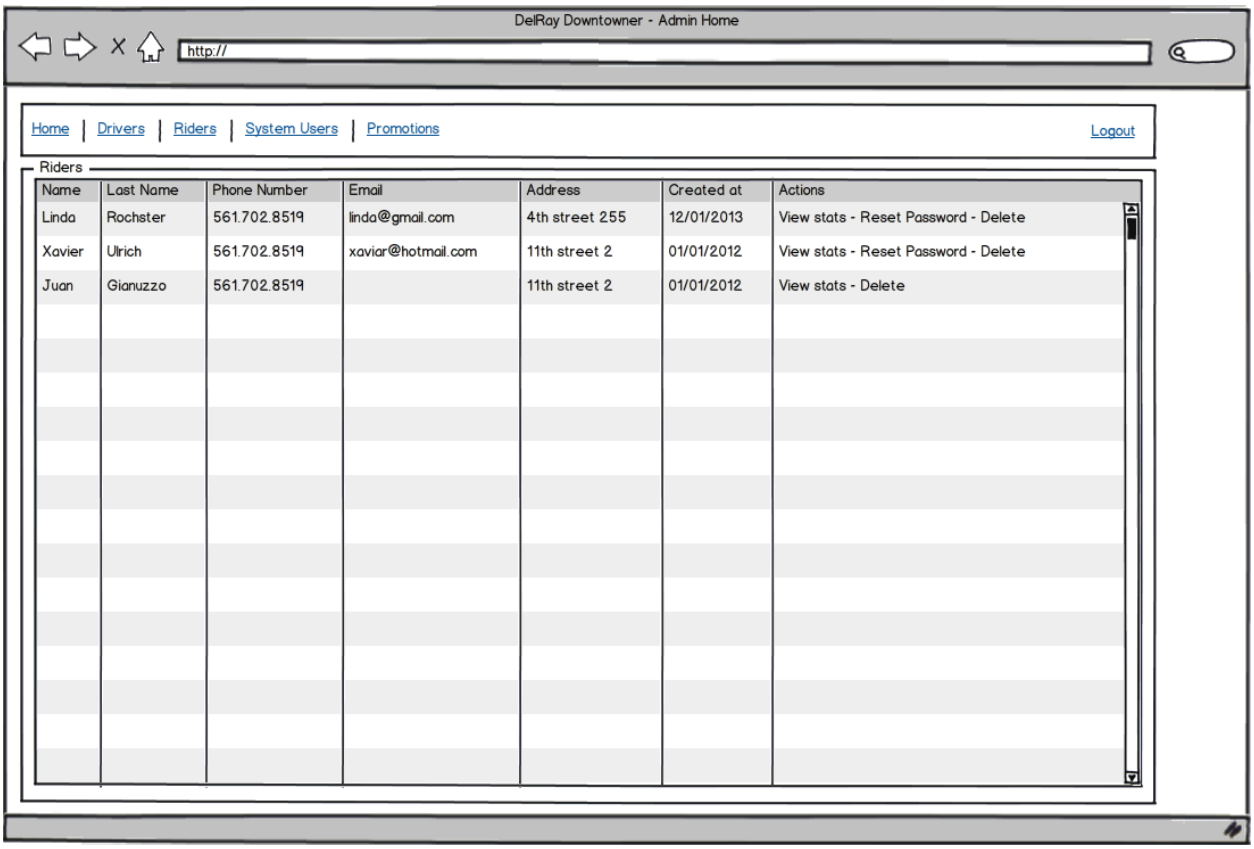


- Name:
  - El usuario puede ingresar el nombre del Driver.
- Last Name:
  - El usuario puede ingresar el apellido del Driver.
- Email:
  - El usuario puede ingresar correo electrónico del Driver.
- Phone Number:
  - El usuario puede ingresar el número de teléfono del Driver
- Accept Button:
  - Cuando es presioando, se muestra un diálogo de confirmación. Si el usuario lo confirma, la aplicación valida los datos ingresados. Si los datos son válidos, el nuevo Driver es creado con estado *Disponible*. La aplicación navega luego a la pantalla Admin Drivers Screen. En caso que los datos sean inválidos, se mostrará un mensaje de error. Si el usuario no realiza ninguna confirmaciòn, el díalogo de confirmación se oculta.

  - Los *valores* se consideran *válidos* si:
    * Name and Last Name: No es una cadena vacía, se aceptan valores duplicados para nombre y apellido.
    * Email: No es una cadena vacía, puede ser cualquier dirección de correo electrónico, su valor debe ser único.
    * Phone Number: cualquier número de teléfono de Estados Unidos, valores vacíos no son permitidos.

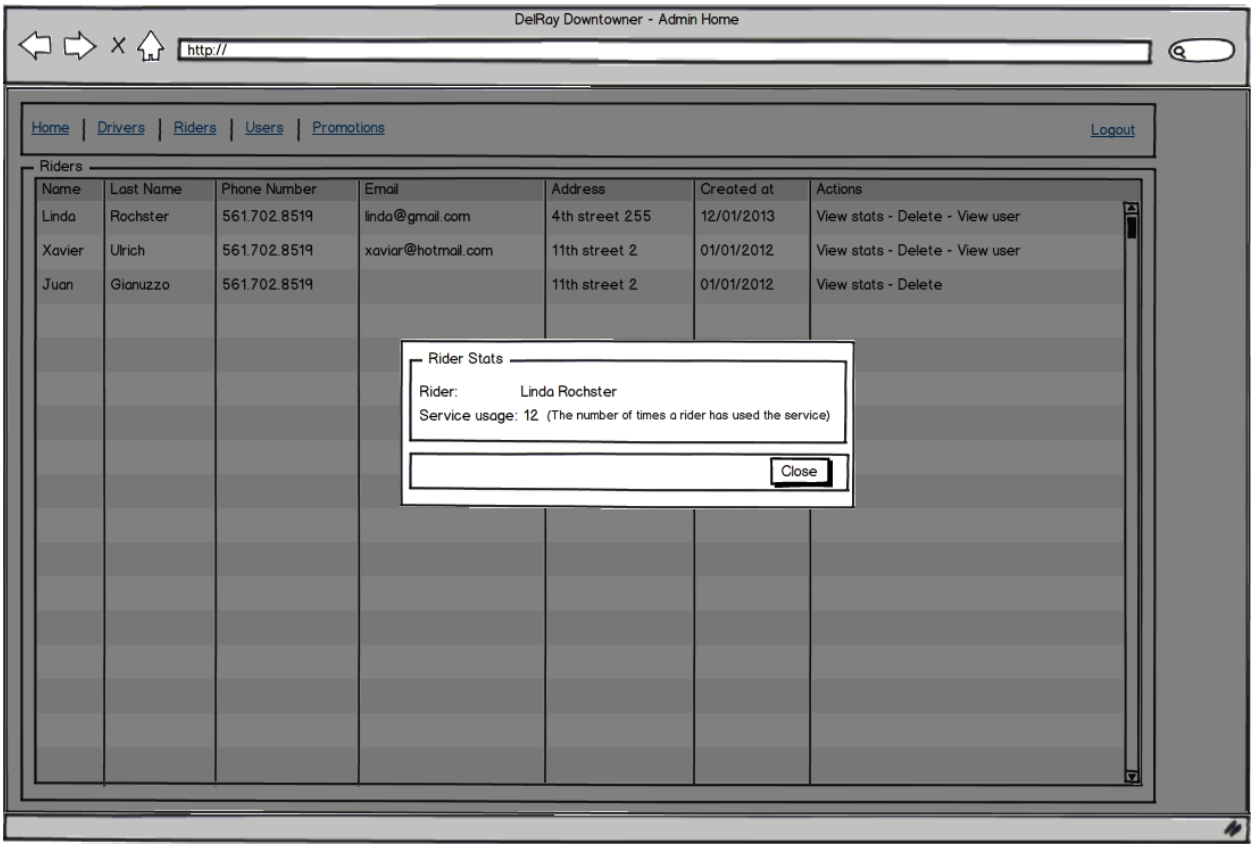
- Cancel Button:
  - Cuando se presiona la aplicación navega a la pantalla Admin Drivers Screen.

Admin Riders Screen



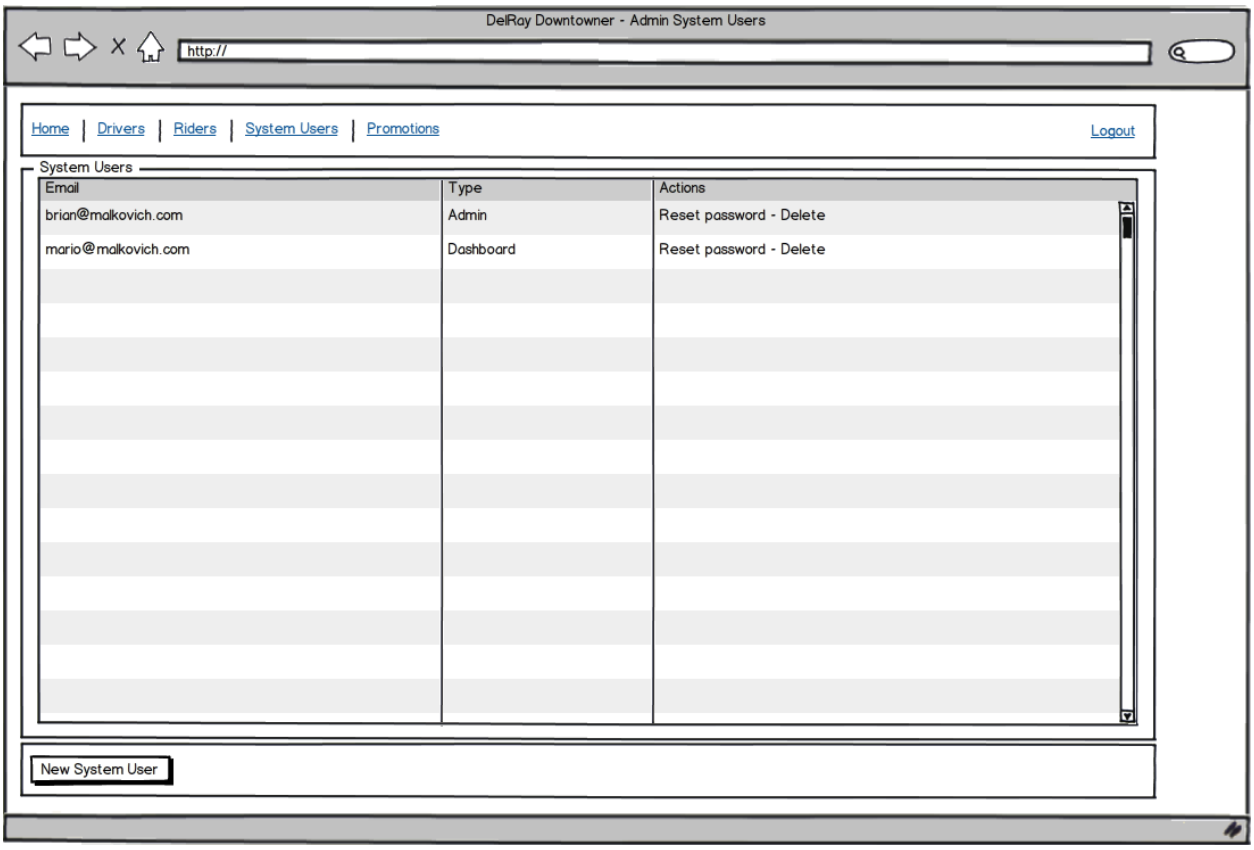
- Riders:
  - Un listado de *riders* se mostrará ordenado por Nombre y Apellido.
- View Stats:
  - Presionando en View Stats, la aplicación mostrará la pantalla Admin View Rider Stats Screen.
- Reset password:
  - Al presionarlo se enviará un correo electrónico al Driver con una nueva contraseña.
- Delete:
  - Presionando en Delete se mostrará un diálogo de confirmación. Si el usuario lo acepta, el *Rider* será borrado.

Admin View Rider Stats Screen



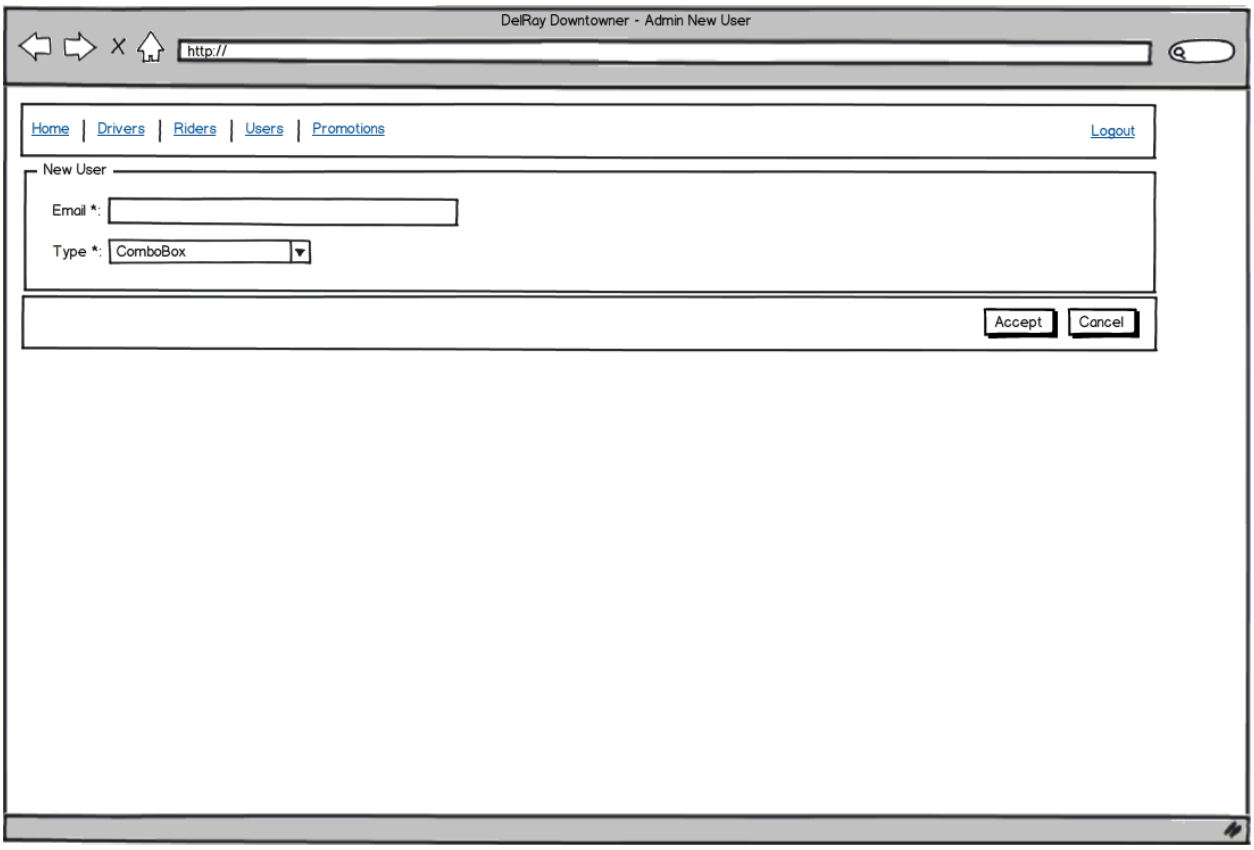
- Rider:
  - El Nombre y Apellido del pasajero
- Service usage:
  - Muestra el número de veces que un pasajero ha usado el servicio. Solamente se contabilizan los viajes que han llegado a un estado completado.
- Close:
  - La ventana de tipo popup es cerrada.

Admin System Users Screen



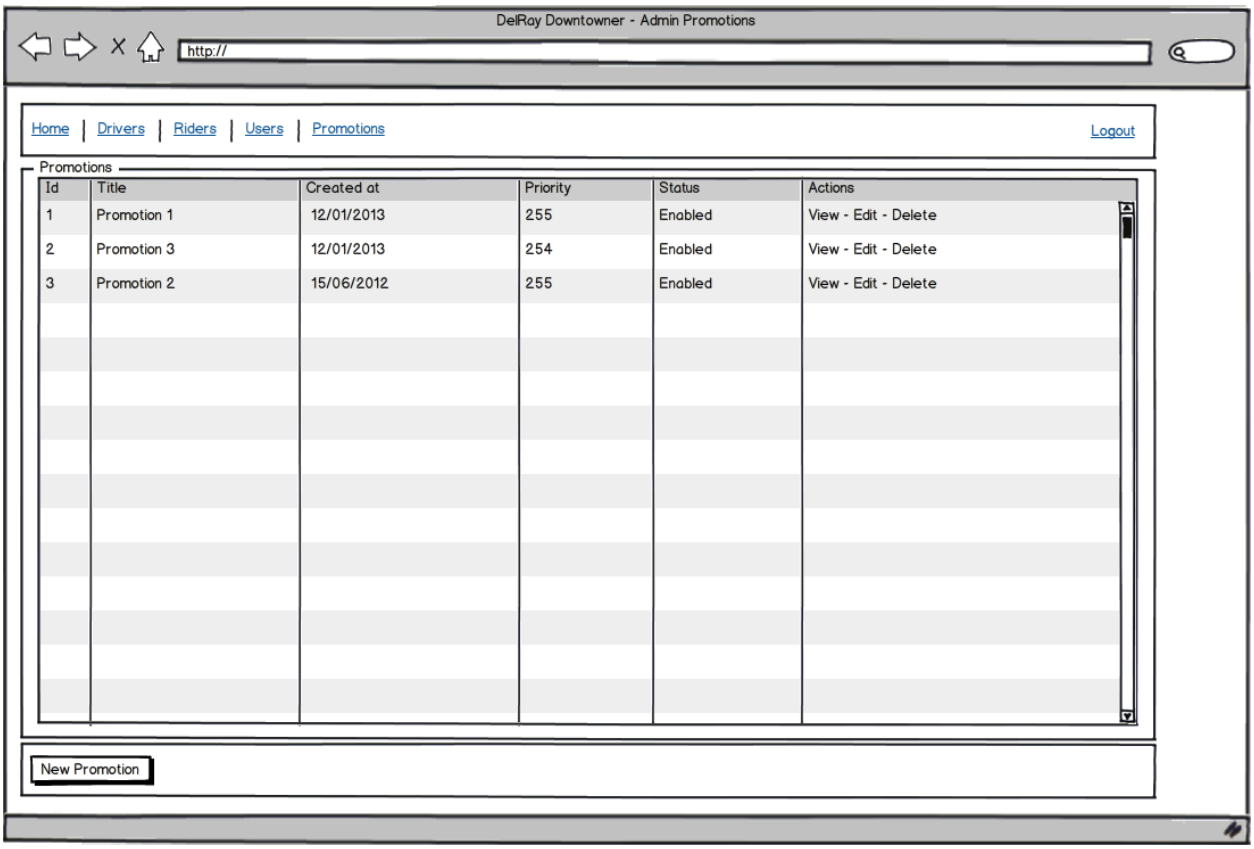
- System Users:
  - Un listado de todos los usuarios del sistema será mostrado, ordenados por su id.
- Delete:
  - Al ser presionado un diálogo de confirmación es mostrado. Si el usuario lo confirma, el usuario es borrado.
- Reset password:
  - Presionándolo, una nueva contraseña es generada y enviada al e-mail del Usuario del Sistema.
- New system user:
  - Presionándolo la aplicación navega a la pantalla Admin New System User Screen.

Admin New System User Screen



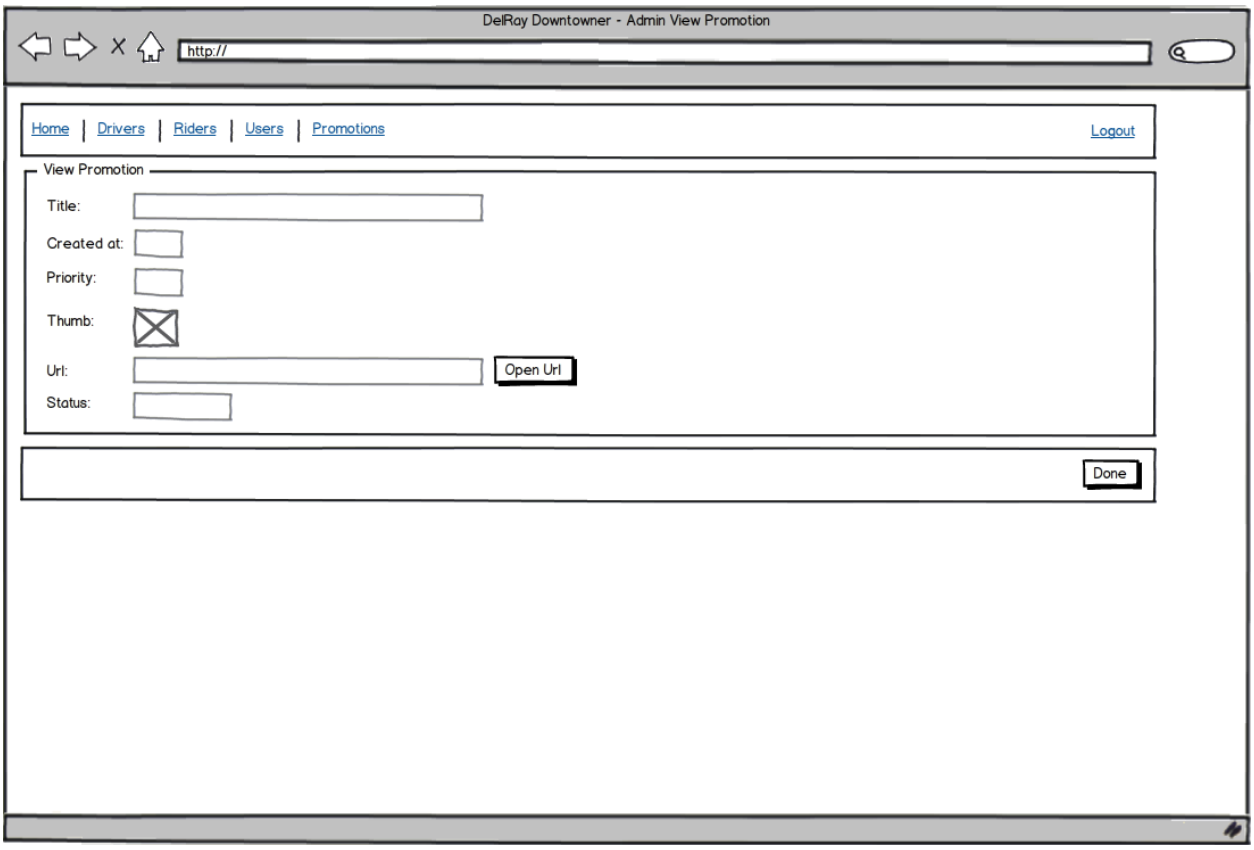
- Email
  - El usuario puede ingresar su email.
- Type
  - El usuario puede seleccionar los tipos **Admin** o **Dashboard**
- Accept button
  - Cuando es presionado, se muestra un diálogo de confirmación. Si el usuario confirma, la aplicación valida la información. Si la información se valida correctamente el nuevo Usuario de Sistema es creado y la aplicación navega a la pantalla Admin System Users Screen. En caso contrario, un mensaje de error es mostrado. Si el usuario no realiza confirmación, el diálogo de confirmación es mostrado.
  - Los valores considerados como válidos son:
    * Email: Cualquier dirección de e-mail, un string vacío no es permitido. Este campo debe ser único.
- Cancel button
  - Cuando es presionado la aplicación navega a Admin System Users Screen.

Admin Promotions Screen



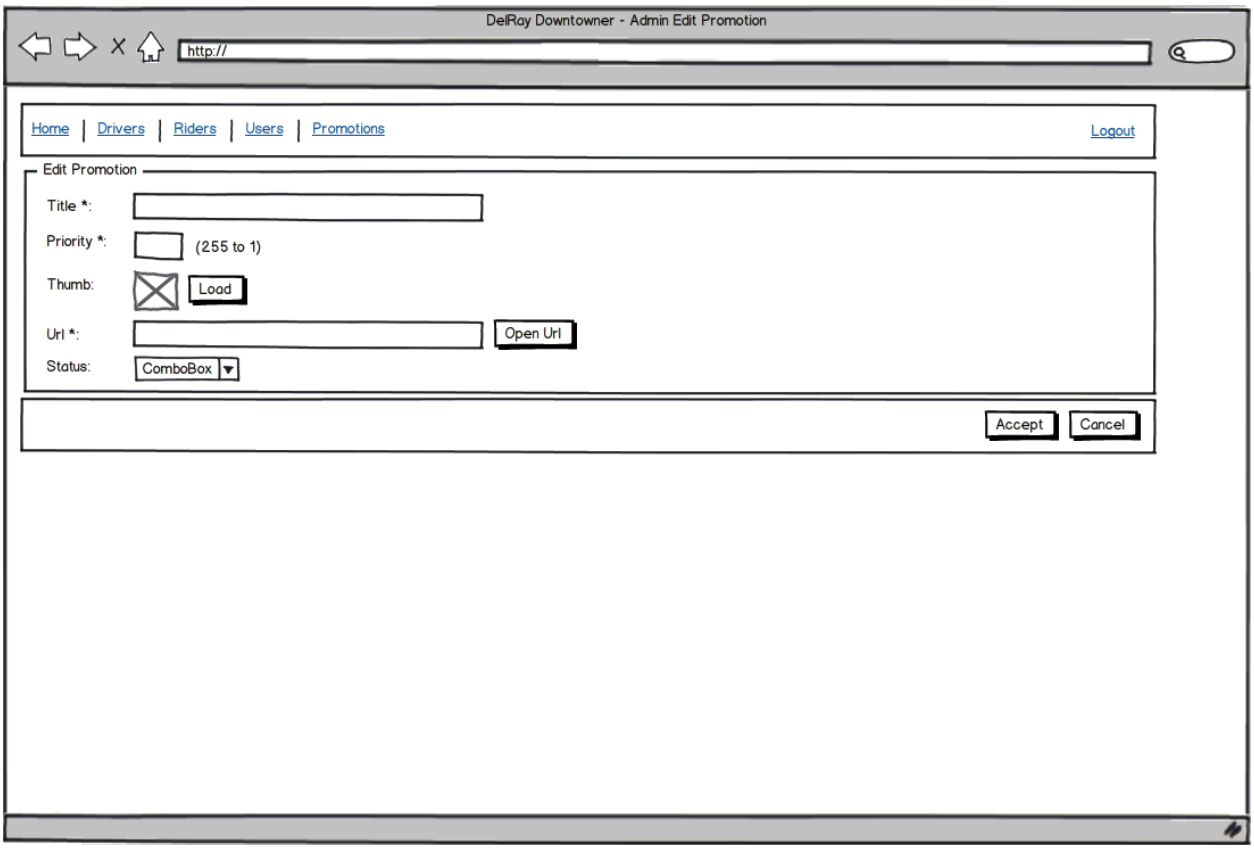
- Promotions
  - Se mostrará una lista de **Promociones** ordenadas por su fecha de creaciòn, en orden descendente.
- View
  - Presionando en **View** la aplicación navegará a la pantalla Admin View Promotion.
- Edit
  - Presionando en **Edit**, la aplicación navegará a la pantalla Admin Edit Promotion.
- Delete
  - Presionando en **Delete** se muestra un diálogo de confirmación. Si el usuario acepta, la Promoción es borrada y la versión de la lista de promociones es incrementada.
- New promotion:
  - Presionando en **New Promotion** la aplicación navega a la pantalla [Admin New Prommotion][]

Admin View Promotion



- Title
  - Muestra el título de la promoción
- Priority
  - Muestra la prioridad de la promoción
- Thumb
  - Muestra una imagen en miniatura de la promoción URL
  - Muestra la URL de la promoción
- Open URL
  - Cuando se realize un click sobre mismo un nuevo navegador mostrará el html de la promoción
- Status
  - Muestra el estado de la promoción.
- Done Button
  - Cuando es presionado la aplicación navega a la pantalla Admin Promotions Screen

Admin Edit Promotion

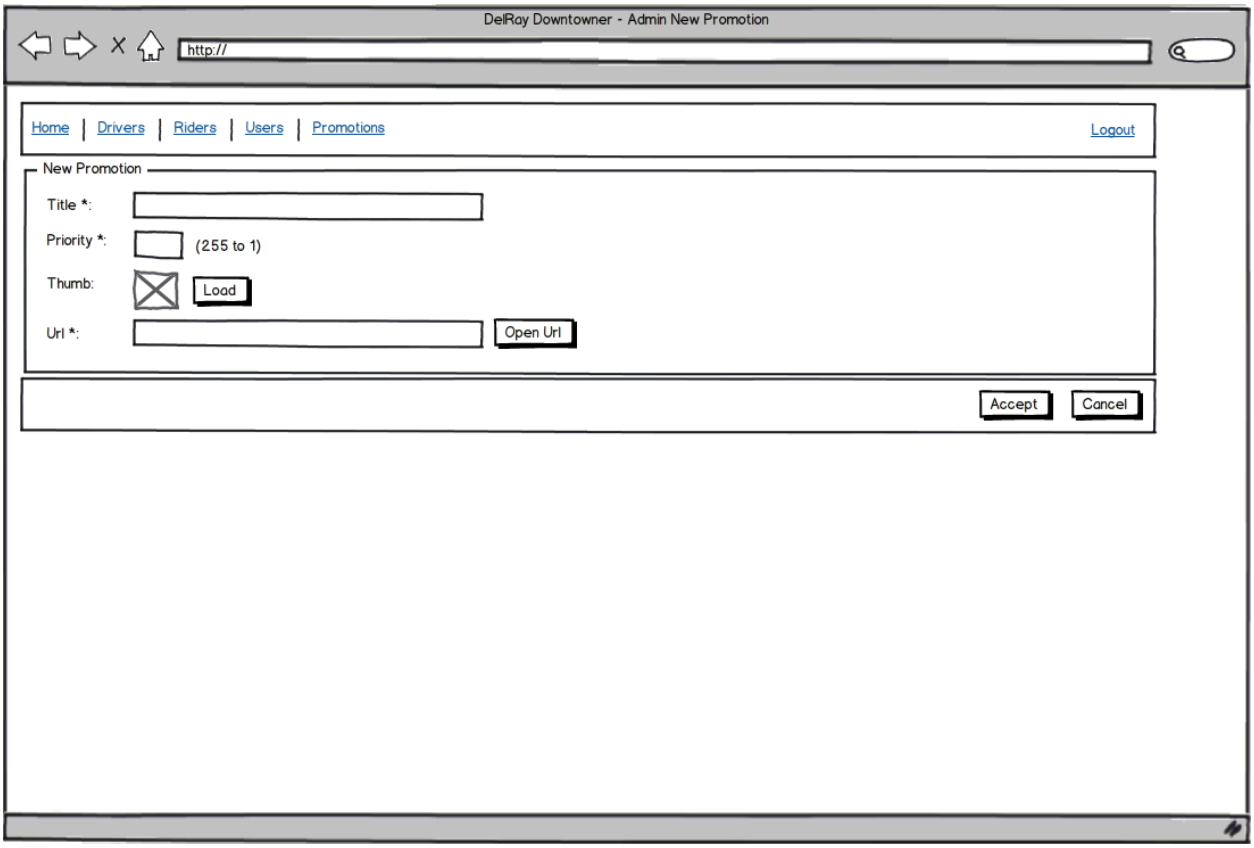


- Title
  - El título de la promoción
- Priority
  - La prioridad de la promoción
- Thumb
  - Una imagen en miniatura de la promoción
- Load
  - Se mostrará un cuadro de diálogo para elegir la imagen en miniatura de la promoción. Cuando el usuario seleccione la imagen un indicador de carga se mostrará.
- URL
  - La URL de la promoción
- Open URL
  - Cuando se realize un click sobre mismo un nuevo navegador mostrará el html de la promoción
- Accept Button
  - Cuando se realice un click sobre dicho botón, un diálogo de confirmación será mostrado. Si el usuario confirma, la aplicación valida la información, si la información es correcta una nueva promoción es creada con estado **Deshabilitado**, la aplicación luego navega a Admin Promotions Screen; en caso que no sea correcta, un mensaje de error aparecerá. Si el usuario no confirma, el cuadro de confirmación se ocultará.

  - Los **valores** que se contemplan como **válidos** serán:
    * Title: Una cadena vacía no es permitida. El nombre debe ser único
    * Priority: No puede ser vacío, un valor entre 255 y 1 es permitido (siendo 255 el de más alta prioridad, y 1 el de más baja).
    * Thumb: no puede ser vacío. Una miniatura por defecto será mostrada si está vacía.
    * URL: no puede ser vacía.

- Cancel Button:
  - Cuando es realizado un click en él, la aplicación navega a la pantalla Admin Promotions Screen. Si una imagen fue subida, la misma se borrará del servidor Delray Server.

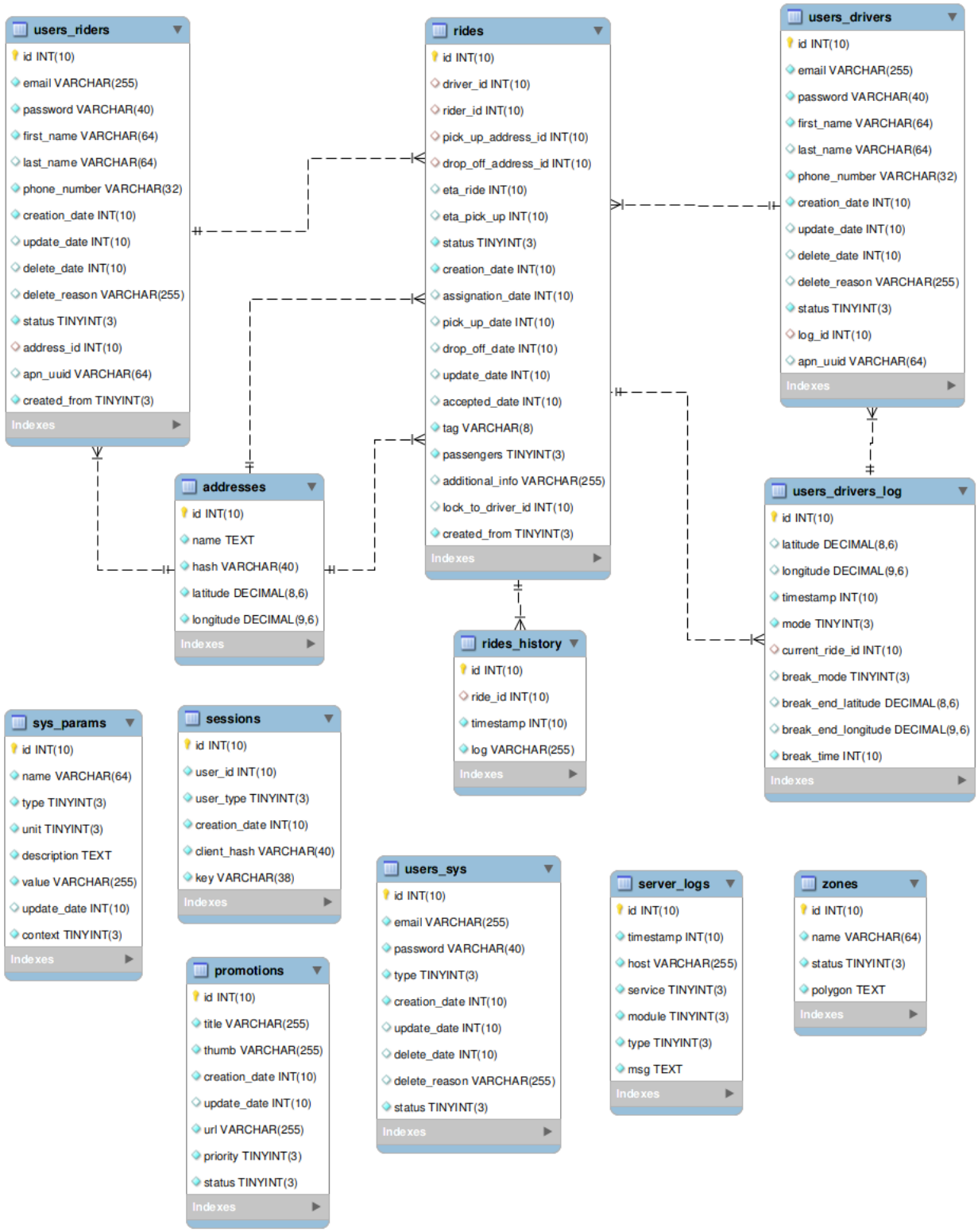
Admin New Promotion



- Title
  - El título de la promoción
- Priority
  - La prioridad de la promoción
- Thumb
  - Una imagen en miniatura de la promoción
- Load
  - Se mostrará un cuadro de diálogo para elegir la imagen en miniatura de la promoción. Cuando el usuario seleccione la imagen un indicador de carga se mostrará.

- URL
  - La URL de la promoción
- Open URL
  - Cuando se realize un click sobre mismo un nuevo navegador mostrará el html de la promoción
- Accept Button
  - Cuando se realice un click sobre dicho botón, un diálogo de confirmación será mostrado. Si el usuario confirma, la aplicación valida la información, si la información es correcta una nueva promoción es creada con estado **Deshabilitado**, la aplicación luego navega a Admin Promotions Screen; en caso que no sea correcta, un mensaje de error aparecerá. Si el usuario no confirma, el cuadro de confirmación se ocultará.
  - Los **valores** que se contemplan como **válidos** serán:
    * Title: Una cadena vacía no es permitida. El nombre debe ser único
    * Priority: No puede ser vacío, un valor entre 255 y 1 es permitido (siendo 255 el de más alta prioridad, y 1 el de más baja).
    * Thumb: no puede ser vacío. Una miniatura por defecto será mostrada si está vacía.
    * URL: no puede ser vacía.
- Cancel Button:
  - Cuando es realizado un click en él, la aplicación navega a la pantalla Admin Promotions Screen. Si una imagen fue subida, la misma se borrará del servidor Delray Server.

6.3.5. Base de Datos



- session

  - key: Llave de la sesión. Debe ser un uuid válido.
  - userId: Identificador único de usuario.
  - userType: Tipo de Usuario: 0=>Admin 1=>Dashboard 2=>Driver 3=>Rider
  - clientHash: Una firma (hash) que identificará al cliente fisico.
  - id: Identificador de sesión único.
  - creationDate: Fecha de creación de la sesión.

- user_sys
  - deleteDate: Fecha en la que fue borrado el usuario.
  - email: Correo electrónico del usuario de sistema.
  - type: Tipo de Usuario: 0=>Admin 1=>Dashboard.
  - deleteReason: Razón por la cual fue borrado el usuario.
  - password: contraseña del usuario de sistema.
  - id: identificador único.
  - creationDate: Fecha de creación del usuario.
  - updateDate: Última actualización del Usuario.
  - status: Estado del Usuario: 0=>Enabled 1=>Disabled 2=>Blocked.

- user_driver_log
  - breakMode: Tipo de la pausa: 0=>None 1=>Now 2=>Scheduled.
  - mode: Modo actual del conductor: 0=>Online 1=>In Break 2=>Offline.
  - breakTime: Horario de Recreo.
  - breakEndLatitude: La latitud en grados. El valor es positivo cuando se indican posiciones al norte respecto del ecuador. Valores negativos indican posiciones al sur del mismo.
  - latitude: La latitud en grados. El valor es positivo cuando se indican posiciones al norte respecto del ecuador. Valores negativos indican posiciones al sur del mismo.
  - id: identificador único de Log.
  - timestamp: Marca temporal de la última actualización.
  - breakEndlongitude: Longitud en grados. Las mediciones son relativas al meridiano cero (0), con valores positivos indicando posiciones al este del meridiano cero, y valores negativos indicando posiciones al Oeste.
  - currentRideId: Identificador único del Conductor actual (FK).
  - longitude: Longitud en grados. Las mediciones son relativas al meridiano cero (0), con valores positivos indicando posiciones al este del meridiano cero, y valores negativos indicando posiciones al Oeste.

- user_rider
  - addressId: Identificador único de dirección (FK).
  - phoneNumber: Número de teléfono del conductor.
  - createdFrom: Creado desde: 0=>Web Dashboard 1=>Rider Mobile Application.
  - deleteReason: Razón por la cual el usuario fue borrado.
  - apnUuid: Identificador del Apple Push Notification Service.
  - id: Identificador único de la tabla usuario.
  - creationDate: Cuando el usuario fue creado.
  - updateDate: Última actualización del usuario.
  - deleteDate: Cuando el usuario fue borrado.
  - firstName: Nombre del usuario (pasajero).
  - email: dirección de correo electrónico del pasajero.
  - password: Contraseña del pasajero.
  - lastName: Apellido del pasajero.
  - claimed: Indica si la cuenta fue reclamada.
  - status: Estado del pasajero: 0=>Enabled 1=>Disabled 2=>Blocked.

- ride_history
  - id: Identificador único de la tabla.

  - rideId: Identificador único del pasajero.
  - log: Descripción del log (bitácora).
  - timestamp: Marca temporal de la bitácora.

- user_driver
  - logId: Identificador único de la bitácora del conductor (FK).
  - phoneNumber: Número de teléfono del conductor.
  - deleteReason: Motivo por el que el usuario fue borrado.
  - apnUuid: Identificador del Apple Push Notification Service.
  - id: Identificador único de la tabla.
  - creationDate: Fecha de creación del usuario.
  - updateDate: Fecha de la última actualización del usuario.
  - deleteDate: Fecha de borrado del usuario.
  - firstName: Nombre del Conductor.
  - email: correo electrónico del conductor.
  - password: Contraseña del conductor.
  - lastName: Apellido del conductor.
  - status: Estado del conductor: 0=>Enabled 1=>Disabled 2=>Blocked (Habilitado, Deshabilitado, Bloqueado).

- ride
  - additionalInfo: Información adicional del viaje.
  - assignationDate: Fecha en la que el viaje fue asignado a un conductor.
  - pickUpDate: Cuando el Conductor llegará a la locación de recojida.
  - riderId: Identificador único del pasajero (Tabla Rider).
  - createdFrom: Creado desde: 0=>Web Dashboard 1=>Rider Mobile Application 2=>Driver Mobile Application.
  - dropOffDate: Fecha de llegada a destino.
  - id: Identificador único de la tabla.
  - creationDate: Fecha de creación del viaje.
  - updateDate: Fecha de la última actualización del viaje.
  - etaRide: Tiempo estimado de llegada (Para recojida o destino).
  - dropOffAddressId: Identificador único de la dirección de destino.
  - passengers: Número de pasajeros.
  - status: Estado actual del viaje: 0=>Unassigned 1=>Pending 2=>On The Way 3=>Near Pick-up 4=>At Pick-up 5=>On Ride 6=>Completed 7=>Cancelled.
  - pickUpAddressId: Identificador único de la dirección de recogida.
  - tag: Tag de identificación descriptivo del viaje. Formato: DDHHMMSS.
  - etaPickUp: Tiempo estimado de llegada (Desde la localización actual del conductor hasta la dirección de recogida).
  - acceptedDate: Fecha en la que el conductor aceptó el viaje.
  - driverId: Identificador único del conductor.

- server_log
  - service: El nombre del servicio desde donde la bitácora es creada.
  - type: Identificador del tipo de entrada.
  - id: Identificador único de la bitácora del servidor.
  - module: Nombre del módulo desde donde la bitácora es creada.
  - timestamp: Marca temporal de cuando la entrada en la bitácora fue creada.
  - msg: Mensaje.
  - host: El hostanme desde donde la bitácora fue creada.

- sys_param
  - name: Nombre del parámetro.

  - value: Valor del parámetro.
  - type: Tipo de dato del parámetro.
  - unit: Unidad de dato del parámetro.
  - id: Identificador único del parámetro.
  - context: Contexto del parámetro : 0=>Global 1=>Server 2=>WebAdmin 3=>WebDashboard 4=>Driver 5=>Rider.
  - updateDate: Fecha de la última actualización.
  - description: Descripción del parámetro.

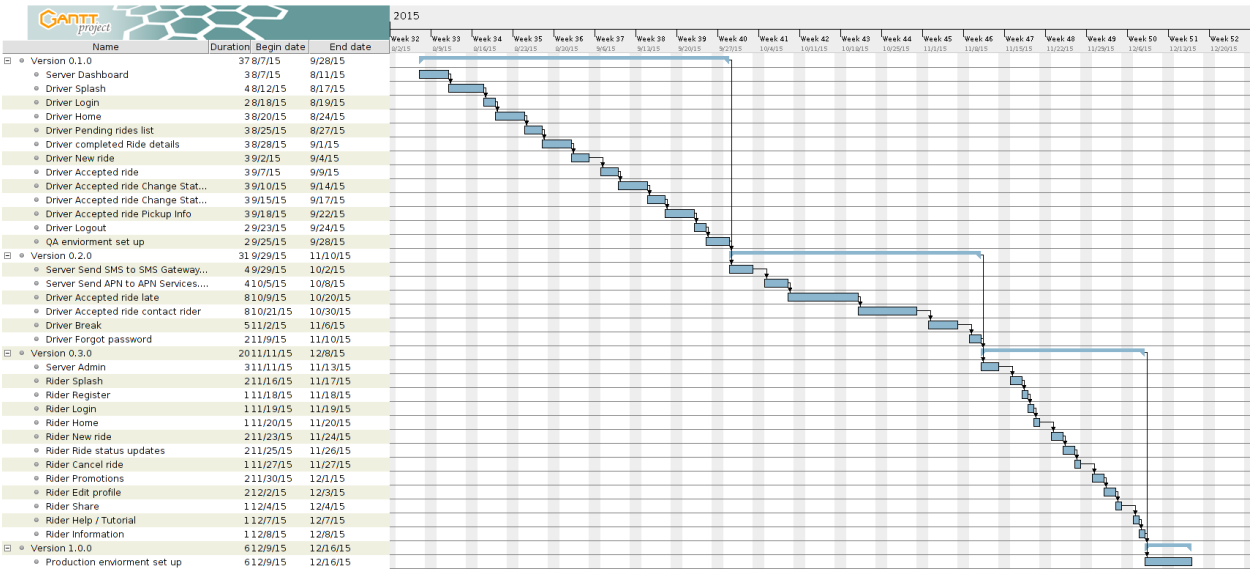
- promotion
  - url: URL de la promoción.
  - thumb: URL a la imagen miniatura de la promoción (32x32 pxs).
  - status: Estado de la promoción: 0=>Enabled 1=>Disabled 2=>Test.
  - priority: Orden de prioridad de la promoción, 0 a 255, (valores más altos indican más prioridad).
  - id: Identificador único de la promoción.
  - creationDate: Fecha de creación de la promoción.
  - updateDate: Fecha de última actualización de la promoción.
  - title: Título de la promoción.

- zone
  - name: Nombre de la zona.
  - id: Identificador único de la zona.
  - status: Estado de la zona: 0=>Active 1=>Inactive.
  - polygon: Definición del polígono, latitud, longitud, latitude, longitud ...

- address
  - name: Nombre de la dirección, compatible con las direcciones de Google Places.
  - id: Identificador único de dirección.
  - latitude: La latitud en grados. Valores positivos indican latitud norte con respecto al ecuador, valores negativos indican latitud sud.
  - hash: Firma (hash) para el nombre.
  - longitude: La longitud en grados. Las mediciones son relativas al meridiano de greenwich. Valores positivos están extendidos al este del meridiano, valores negativos al oeste.

6.3.6. Milestones y Planificacion

El seguimiento del proyecto fue conforme los tiempos que se estimaron, el producto final fue desarrollado en cuatro versiones, siendo las primeras tres versiones (desde 0.1.0 hasta 0.3.0) sobre las que se realizó la codificación del producto, llegando a su versión final en producción con la versión 1.0.0, según los tiempos expresados en el siguiente gráfico Gantt.



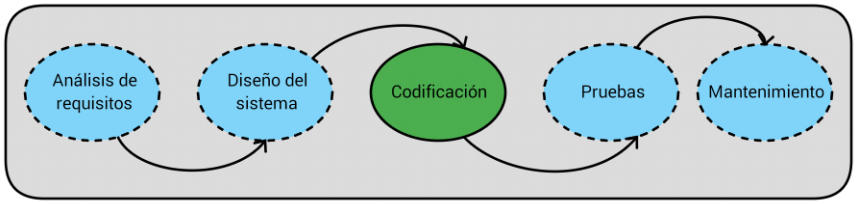
Funcionalidad	0.1.0	0.2.0	0.3.0	1.0.0
Server Dashboard	X			
Server Admin			X	
Server Send SMS to SMS Gateway. (Drivers only)		X		
Server Send APN to APN Services. (Riders and Drivers)	X			
Driver Splash	X			
Driver Login	X			
Driver Home	X			
Driver Pending rides list	X			
Driver completed Ride details	X			
Driver New ride	X			
Driver Accepted ride	X			
Driver Accepted ride Change Status to On Ride	X			
Driver Accepted ride Change Status to Completed	X			
Driver Accepted ride Pickup Info	X			
Driver Logout	X			
Driver Accepted ride late		X		
Driver Accepted ride contact rider		X		
Driver Break		X		
Driver Forgot password		X		
Rider Splash			X	
Rider Register			X	
Rider Login			X	
Rider Home			X	
Rider New ride			X	
Rider Ride status updates			X	
Rider Cancel ride			X	
Rider Promotions			X	
Rider Edit profile			X	
Rider Share			X	
Rider Help / Tutorial			X	
Rider Information			X	
QA enviorment set up	X			

Funcionalidad	0.1.0	0.2.0	0.3.0	1.0.0
Production enviorment set up				X

6.3.7. Restricciones, Supuestos y Convenciones

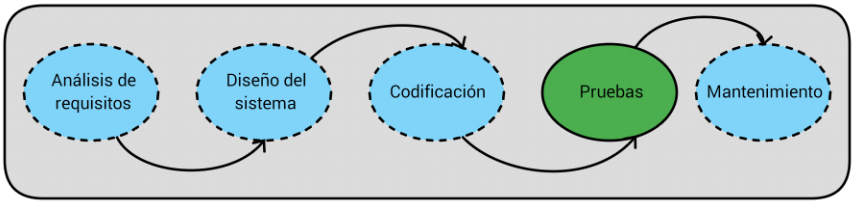
- Servicios de geolocalización: La aplicación va a usar la API de Google Geocoding web services. Esto significa que no se van a poder usar los mapas nativos de iOS por que estaríamos violando los términos y condiciones de Google Maps API y Google Geocoding web service. Para poder usar los servicios web de Google es necesario registrar una API KEY. Con esta API KEY el administrador del sistema va a poder controlar las estadísticas de uso de los servicios de google por parte de la aplicación. El servicio de Geocoding de Google tiene, para una cuenta gratuita, un limite de 2500 llamadas por día. Para ver mas detalles acerca de como este limite es calculado, visitar https://developers.google.com/maps/documentation/geocoding.
- Google Places Autocomplete: la aplicación va a hacer uso del web service de Google Places Autocomplete para poder auto completar las direcciones. Para poder usar este servicio es necesario registrar una API KEY con la cual los administradores de sistema van a poder controlar las estadísticas de uso de este servicio por parte de la aplicación. El web service de Google Places tiene un limite de 1000 llamadas diarias para una cuenta gratuita. Para conocer mas detalles acerca de como se calcula este limite visitar: https://developers.google.com/places/policies.
- Consumo de batería y plan de datos: debido a que la aplicación Rider hace uso intensivo del plan de datos y los servicios de geolocalización es necesario tratar de minimizar el impacto que esto va a tener en el consumo de la batería. Para poder lograr esto se van establecer las limitaciones necesarias en los los tiempos máximo y la frecuencia con la que se va a acceder a estos servicios.
- Cada vez que exista un error del lado del servidor, o un error en la conexión de datos, la aplicación va a mostrar un mensaje con la leyenda: "Services not available at the moment."
- Cuando la aplicación de Rider esté en segundo plano, las consultas al servidor se van a detener debido a que existe una limitación por parte del sistema operativo para poder correr código de usuario en segundo plano. Cuando el Driver esté cerca del lugar de recogida, se va a enviar un mensaje mediante APN a la aplicación Rider para la notificación.
- Apple Push Notifications (APN): APN es un servicio provisto por Apple. Los servidores de la aplicación se limitan a enviar un mensaje a los servidores de APN pidiendo que se le envíe una notificación al usuario. El que esta notificación llegue al teléfono del usuario es responsabilidad de Apple. Los servidores de APN no son 100 % confiables, pueden estar caídos o saturados.
- La aplicación Rider no es una aplicación de seguimiento de usuarios, no se va a guardar información no necesaria para el estricto funcionamiento del negocio de transporte.
- La aplicación de Rider no va a funcionar si el teléfono no tiene conexión a Internet.

6.4. Etapa III - Codificiación



El código fuente del producto por contrato de confidencialidad fue entregado al cliente y se encuentra bajo su custodia, por lo que el mismo no se adjunta al presente trabajo. En esta etapa se implementó en código el sistema diseñado, utilizando las tecnologías que se habían propuesto con anterioridad.

6.5. Etapa IV - Pruebas



Esta etapa se puede dividir en 2 sub-etapas. La primera es la relacionada con las pruebas automatizadas del lado del desarrollo (pruebas de desarrollador) y las pruebas funcionales efectuadas por el equipo de calidad.

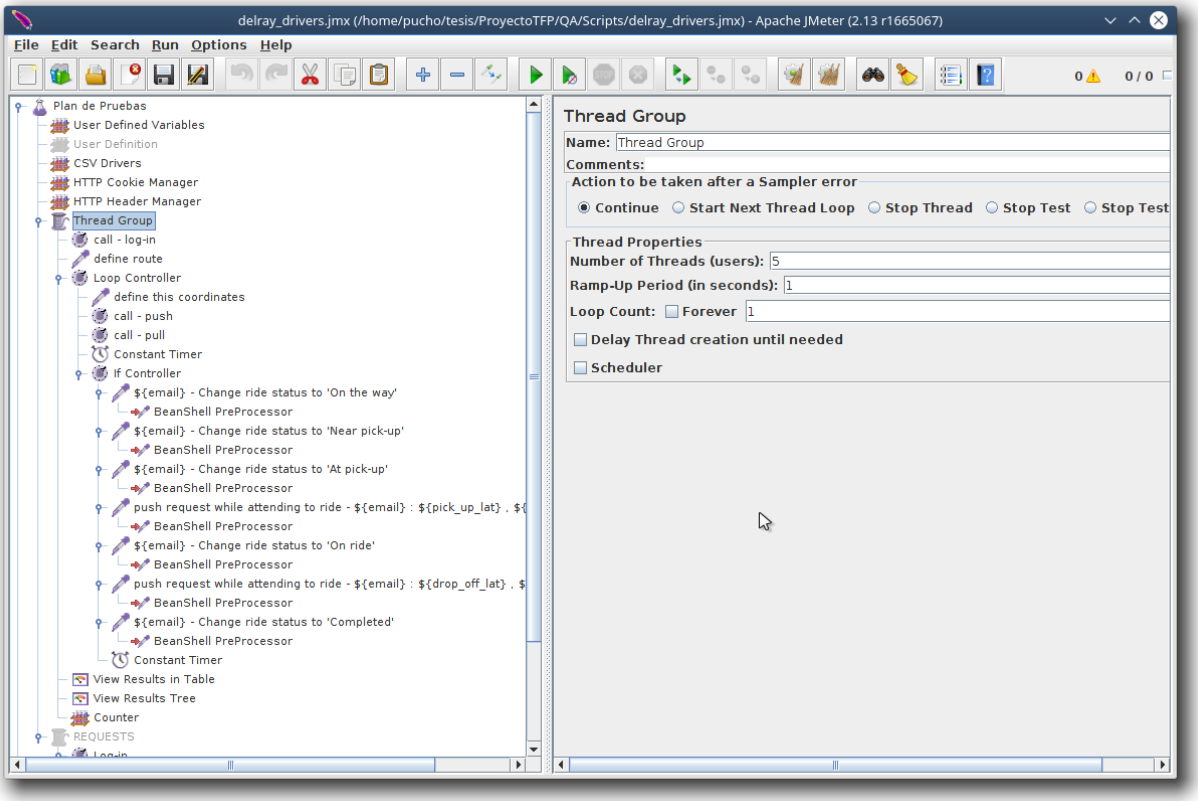
6.5.1. Pruebas de desarrollador

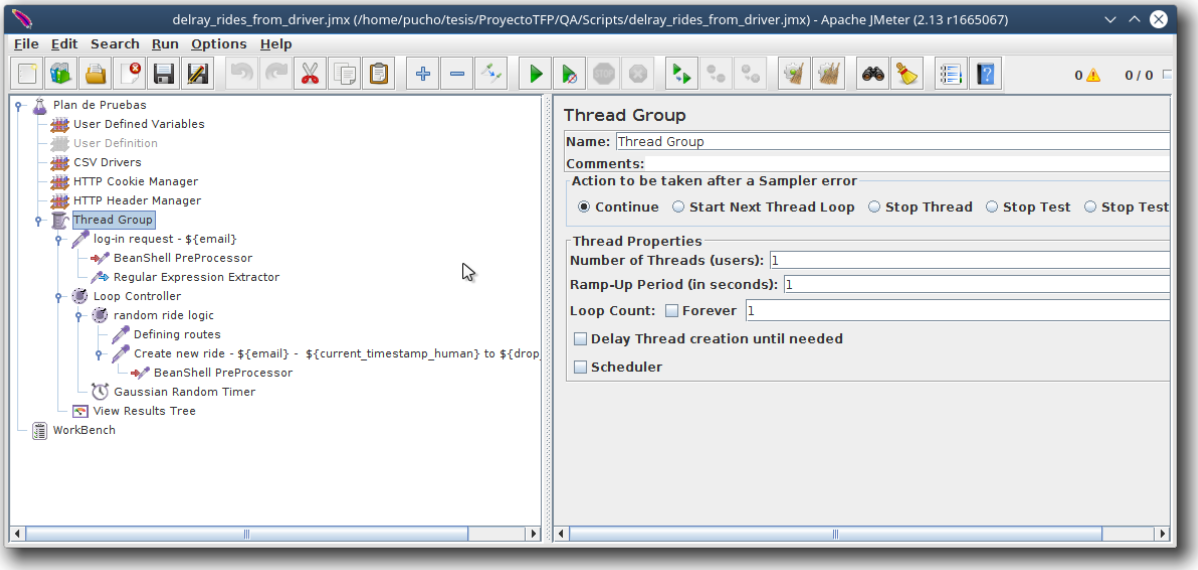
Si bien no se adoptó una filosofía de Desarrollo Guiado por Pruebas, consideramos que las pruebas unitarias son muy útiles en el desarrollo de software. Para el desarrollo del proyecto, se escribieron casos unitarios de prueba en PHP sin ayuda de ninguna librería externa destinados a asegurar la consistencia de las reglas de negocios en el núcleo mas interno de los servidores (bases de datos, objetos de negocios y transaccionales.). Un ejemplo de los reportes generados por estos scrips es el siguiente:

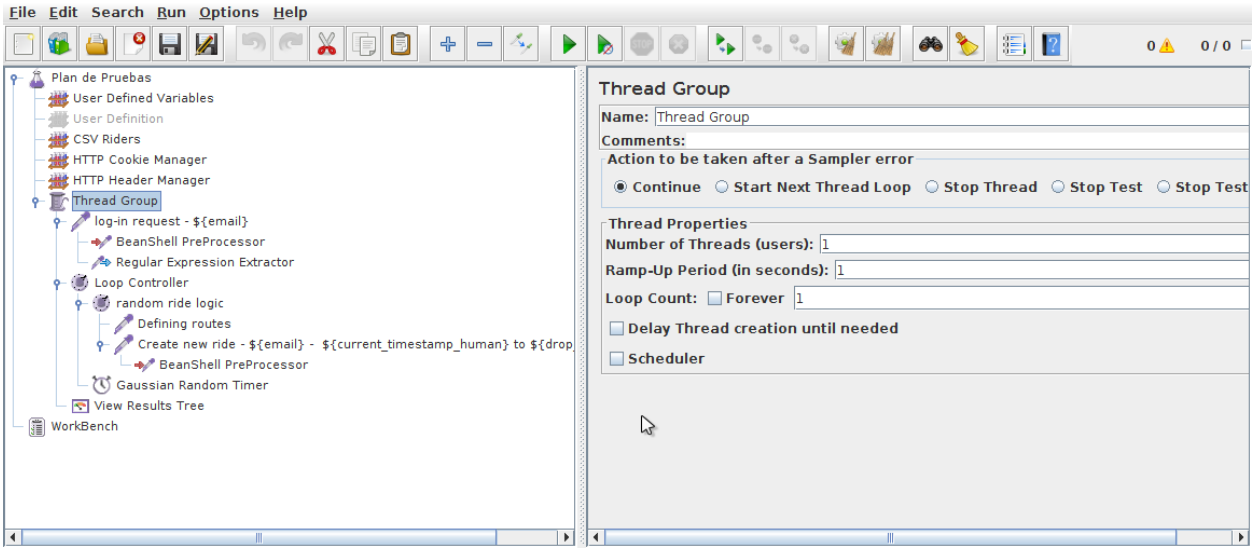
```
BEGIN MAPPER DATABASE SCHEMAS TEST:
----------------------------------
testBTAddress...OK!
testBTPromotion...OK!
testBTRideHistory...OK!
testBTRide...OK!
testBTSession...OK!
testBTServerLog...OK!
testBTSysParam...OK!
testBTUserDriver...OK!
testBTUserDriverLog...OK!
testBTUserRider...OK!
testBTUserSys...OK!
testBTZone...OK!

BEGIN MAPPER RIDER TEST:
------------------------
testBTRiderChangeProfile...OK!
testBTRiderChangeRideStatus...OK!
testBTRiderGetZones...OK!
testBTRiderGetPromotions...OK!
testBTRiderLogin...OK!
testBTRiderLogout...OK!
testBTRiderNewRide...OK!
testBTRiderPull...OK!
testBTRiderRegistration...OK!
testBTRiderResetPassword...OK!
testBTRiderSetAPN...OK!

BEGIN MAPPER DRIVER TEST:
-------------------------
testBTDriverBreak...OK!
testBTDriverChangePassword...OK!
```

```
testBTDriverChangeRideStatus...OK!
testBTDriverGetZones...OK!
testBTDriverLate...OK!
testBTDriverLogin...OK!
testBTDriverLogout...OK!
testBTDriverNewRide...OK!
testBTDriverPull...OK!
testBTDriverPush...OK!
testBTDriverResetPassword...OK!
testBTDriverSendAPNOutside...OK!
testBTDriverSendSMS...OK!
testBTDriverSetAPN...OK!

BEGIN MAPPER DASHBOARD TEST:
----------------------------
testBTWebDashboardChangeRideStatus...OK!
testBTWebDashboardEditRide...OK!
testBTWebDashboardGetInfo...OK!
testBTWebDashboardGetLogs...OK!
testBTWebDashboardGetZones...OK!
testBTWebDashboardLogin...OK!
testBTWebDashboardLogout...OK!
testBTWebDashboardNewRide...OK!
testBTWebDashboardNewRider...OK!
testBTWebDashboardPull...OK!
testBTWebDashboardResetPassword...OK!
testBTWebDashboardSearchRider...OK!
```

Asimismo, se escribieron casos Unitarios de prueba para cada endpoint de las API de servicios web (API de driver, API de rider y API de dashboard), estos últimos casos fueron escritos en JMeter, y terminaron confeccionando pruebas de integración, y ayudaron a mantener la calidad de los mismos según se fueron desarrollando, modificando y manteniendo. Por ultimo, a fin de facilitar la ejecución de las pruebas Funcionales, se desarrollaron scripts en JMeter que simulan diferentes actores del sistema (drivers y riders) interactuando con el mismo en simultaneo.


delray_drivers.jmx (/home/pucho/tesis/ProyectoTFP/QA/Scripts/delray_drivers.jmx) - Apache JMeter (2.13 r1665067)
File Edit Search Run Options Help
Plan de Pruebas
User Defined Variables
User Definition
CSV Drivers
HTTP Cookie Manager
HTTP Header Manager
Thread Group
call - log-in
define route
Loop Controller
define this coordinates
call - push
call - pull
Constant Timer
If Controller
${email} - Change ride status to 'On the way'
BeanShell PreProcessor
${email} - Change ride status to 'Near pick-up'
BeanShell PreProcessor
${email} - Change ride status to 'At pick-up'
BeanShell PreProcessor
push request while attending to ride - ${email} : ${pick_up_lat} , ${
BeanShell PreProcessor
${email} - Change ride status to 'On ride'
BeanShell PreProcessor
push request while attending to ride - ${email} : ${drop_off_lat} , $
BeanShell PreProcessor
${email} - Change ride status to 'Completed'
BeanShell PreProcessor
Constant Timer
View Results in Table
View Results Tree
Counter
REQUESTS
Thread Group
Name: Thread Group
Comments:
Action to be taken after a Sampler error
Continue Start Next Thread Loop Stop Thread Stop Test Stop Test
Thread Properties
Number of Threads (users): 5
Ramp-Up Period (in seconds): 1
Loop Count: Forever 1
Delay Thread creation until needed
Scheduler


delray_rides_from_driver.jmx (/home/pucho/tesis/ProyectoTFP/QA/Scripts/delray_rides_from_driver.jmx) - Apache JMeter (2.13 r1665067)
File Edit Search Run Options Help
Plan de Pruebas
User Defined Variables
User Definition
CSV Drivers
HTTP Cookie Manager
HTTP Header Manager
Thread Group
log-in request - ${email}
BeanShell PreProcessor
Regular Expression Extractor
Loop Controller
random ride logic
Defining routes
Create new ride - ${email} - ${current_timestamp_human} to ${drop
BeanShell PreProcessor
Gaussian Random Timer
View Results Tree
WorkBench
Thread Group
Name: Thread Group
Comments:
Action to be taken after a Sampler error
Continue Start Next Thread Loop Stop Thread Stop Test Stop Test
Thread Properties
Number of Threads (users): 1
Ramp-Up Period (in seconds): 1
Loop Count: Forever 1
Delay Thread creation until needed
Scheduler



6.5.2. Pruebas del equipo de calidad

Para esta etapa tomamos cada una de las distintas versiones, y comparando el resultado obtenido contra el análisis de requerimientos. Una vez descubiertas las fallas se procede a su reparación. Aquí se involucra nuevamente el cliente, a quien se le envían las aplicaciones para su aprobación final.

Para la etapa de pruebas se utilizaron los servicios tercerizados de personas dedicadas a Pruebas Funcionales, principalmente, y algunos casos de Pruebas Automatizadas para comprobar el correcto funcionamiento del algoritmo de Asignación.

Para el seguimiento de los Bugs, se utilizo la herramienta MantisBT. Se confecciona el siguiente resumen luego de sucesivas iteraciones de calidad, el detalle de cada uno de los bugs tratados se puede descargar de la siguiente URL: https://drive.google.com/file/d/0B28RBoqC76Q-R05UOUI4LVh0N0k/view?usp=sharing

- Cantidad de bugs por proyecto

Proyecto	open	resolved	closed	total
Delray Downtowner	1	0	0	1
» Delray Downtowner Admin	0	0	3	3
» Delray Downtowner Dashboard	0	0	38	38
» Delray Downtowner Driver iOS	0	4	17	21
» Delray Downtowner Rider iOS	0	1	7	8
» Delray Downtowner Server	0	6	14	20
» Delray Downtowner Support	8	15	1	24

- Cantidad de Bugs por estado

Estado	open	resolved	closed	total
new	1	-	-	1
feedback	1	-	-	1
assigned	7	-	-	7

Estado	open	resolved	closed	total
resolved	-	26	-	26
closed	-	-	80	80

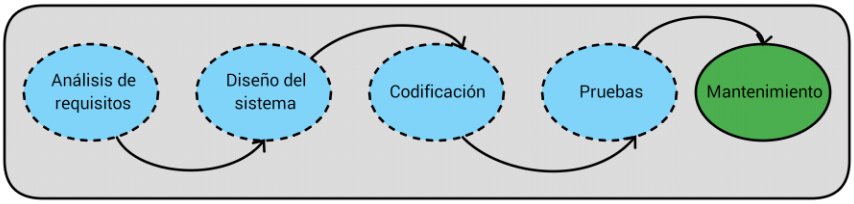
- Cantidad de Bugs por severidad

By Severity	open	resolved	closed	total
feature	1	1	2	4
tweak	0	0	1	1
minor	7	23	54	84
major	1	2	23	26

- Cantidad de Bugs por categoría

By Category	open	resolved	closed	total
Business Logic	9	17	17	43
Functional	0	2	21	23
Integration	0	4	11	15
New feature	0	3	0	3
User Interface	0	0	31	31

6.6. Etapa V - Mantenimiento



Se deja el sistema funcionando en producción, abierto al público. Se detectan y analizan los problemas que surjan en el entorno real, y se toman las medidas necesarias para que el sistema cumpla con todos los requisitos relevados en la **Etapa I** del modelo. Se resuelven los bugs que quedaron abiertos en la etapa anterior, y adicionalmente se confecciona un Manual de Usuario para el cliente, indicando cómo se debe operar para realizar la creación de viajes desde las diferentes interfaces, así como también cómo realizar la alta de una promoción. También se incluye en dicho manual un punto de solución rápida de problemas (Troubleshooting).

6.6.1. Manual de Usuario

El manual de usuario, dado que el cliente final es de habla inglesa, y el desarrollo fue también codificado en dicho idioma, se presenta en dicho idioma para ser fiel al entregable comprometido.

Creating Rides From Dashboard Application

1. If the Rider is not registered within the application you can register a Rider from "New Rider" button.

2. If the Rider is registered, then when typing the Rider name in the Rider text box an autocomplete feature is shown. You will need to select one Rider from the autocompele feature list.

3. Type the pick-up address or place. While you type an autocomplete feature is shown. You will need to select one address or place from the autocompele feature list.

4. Type the drop-off address or place. While you type an autocomplete feature is shown. You will need to select one address or place from the autocompele feature list.

5. Select the number of passangers.

6. Type any additional info. This value is optional.

7. Press the "Create" button.

The system will create a ride in "Unassigned" status. If there are online Drivers, when the assignation algorithm is executed by the system, the ride will be assigned to a Driver. Once the Ride is assigned you will see it on the Dashboard Ride's list, map and a "New ride assigned" push notification will be sent to the Driver.

Creating Rides From Rider Application

At the "New Request" screen.
1. Type the pick-up address or place. While you type an autocomplete feature is shown. You will need to select

one address or place from the autocompele feature list.
2. Type the drop-off address or place. While you type an autocomplete feature is shown. You will need to select one address or place from the autocompele feature list.
3. Select the number of passangers.
4. Type any additional info. This value is optional.
5. Press the "Create" button.

The system will create a ride in "Unassigned" status. If there are online Drivers, when the assignation algorithm is executed by the system, the ride will be assigned to a Driver. Once the Ride is assigned the following updates are made:
1. You will see it on the Dashboard Ride's list and map.
2. A "New ride assigned" push notification will be sent to the Driver.
3. The Rider will see a Ride status updated on the Rider application.

Creating Rides From Driver Application

At the "New Ride" screen.
1. Type the pick-up address or place. While you type an autocomplete feature is shown. You will need to select one address or place from the autocompele feature list.
2. Type the drop-off address or place. While you type an autocomplete feature is shown. You will need to select one address or place from the autocompele feature list.
3. Select the number of passangers.
4. Press the "Create" button.

The system will create a ride in "On Ride" status and all of the Driver's pending rides will be reassigned.

Creating Deals

Creating a Deal is a three step process:

1. Design and build the Deal web page.
2. Upload the web page to Delray Downtowner server.
3. Create a Deal in DelRay Downtowner Web Admin application.

- 1. Design and build the Deal web page.
  Deals must be designed using HTML taking into account the destination drawing canvas height and width. The Rider mobile application has a Deal drawing canvas height and width of 308x402 pixels for iPhone non retina screen.
  A basic Deal HTML file example:

```
<!DOCTYPE html>
<html>
<head></head>
<body style="background-color: transparent;">
<div style="width: 270px; background-color: #fff; border-radius: 6px;
border: 1px solid #DFE0E1; padding: 10px; margin: 10px 0px;
text-align: center;">
    <img style="width: 250px;"
         src="http://www.delraydowntowner.com/wp-content/uploads/2013/04/hyattPlace280.jpg">
    <br>
```

```
    Hyatt Place is located in the Arts District of downtown Delray Beach within
    1 mile of  the Gold Coast and its beautiful beaches and walking distance
    to Atlantic Avenue shopping, dining and entertainment districts.
    <br>
    <a target="_blank"
       href="http://delraybeach.place.hyatt.com/en/hotel/home.html">Visit Site</a>
</div>
</body>
</html>
```

Figura 1: Basic deal example

- 2. Upload the web page to Delray Downtowner server.

- 1. Get Filezilla FTP Client. Download Filezilla Client from: https://filezilla-project.org/download.php?type=client
  When the download has finnished install it and run it.
- 2. Login to Delray Downtowner server. Open Filezilla and enter the following login information:
  At "Host" type: ftp.ord1-1.websitesettings.com
  At "Username" type: ddpromotions
  At "Password" type: pRuwu4ux
  Click on "Quickconnect".
  Filezilla will connect to the promotions path.

- 3. Upload the web page file.
  Using Filezilla:
  * Select your directory where you have the Deal web page HTML file (see Filze image "Your computer directories.").
  * Select the Deal HTML file (see Filze image "Your files inside the above directory.") and drag it to the left pane (see Filze image "Uplaoded Deals HTML files.").

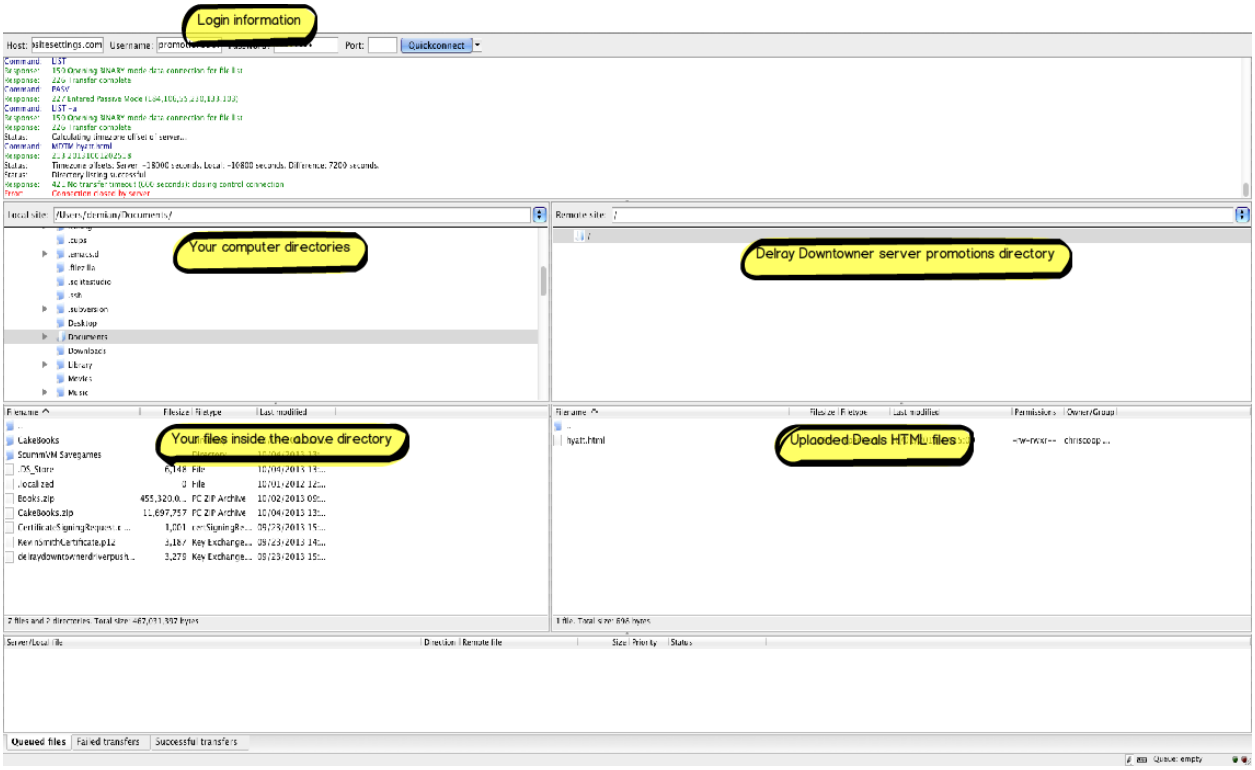


Figura 2: Filezilla

- 3. Create a Deal in DelRay Downtowner Web Admin application.
  Login to Delray Downtowner Web Admin application and select the Deals tab (see Web Admin figure.).
  - Click on "New Deal".
  - At the New Deal popup fill the following information (see Create Deal figure.):
    * The Deal title will be displayed on the Rider Deals list sreen.
    * Thumbnail: A URL from where to get the Deal thumb image.

* Url: The Deal web page HTML file. If you have uploaded a Deal web page HTML file into the Delray Downtowner server the url will be as follow: http://www.downtowner-application.com/web/content/promotions/<your deal web page file name>.html
* Priority: The Deal priority. Beign 0 the lowest and 255 the heighest. Deals with 255 priority will be shown at the top of the Rider's application Deals list.

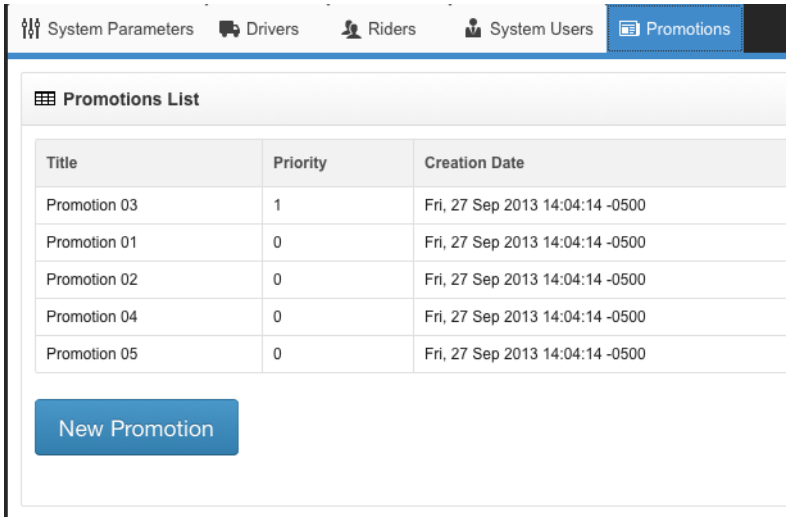


Figura 3: Web Admin

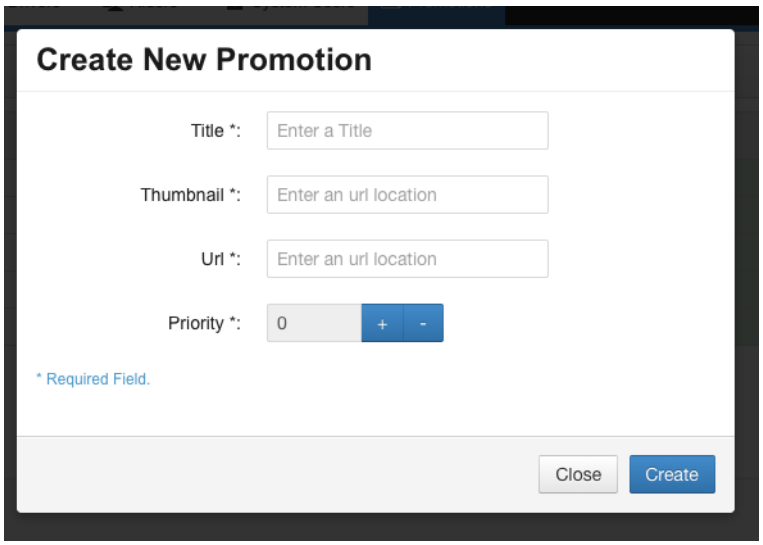


Figura 4: Create Deal

Troubleshooting

- 1. Driver has closed the application and he has an accepted Ride (An accepted ride is a ride that is in the follwing stattus: "On The Way", "Near Pickup", "At Pickup", "On Ride"):
  The Driver will be logged out, all of his "Pending" rides will be reassigned and his current ride will be not updated. If his current ride needs to be reassigned the Driver can perform one of the two following steps:

  - 1. Login again into the Driver application and select the "Ride Over" option from the "Accepted Ride" screen.
  - 2. Call Downtowner base and ask to the Dashboard controller to maually unassign the ride from the "Edit Ride" screen.

- 2. Driver cannot accept a Ride.
  The Driver can perform a logout and then a login and wait for new rides. The previous Ride that the Driver couldnt accept should be reassigned by the system on the logout action.

- 3. Driver cannot logout.
  The Driver must call to Downtowner base and ask to the Admin to set his status to "Offline" from the Edit Driver screen.

- 4. All Driver rides must be reassigned.
  You can perform this task in two ways:
  1. The Dashboard controller must set to "Unassigned" status all of the Driver's rides.
  2. The Dashboard controller must set to "Unassigned" the current Driver accepted ride if he has any.
  The Downtowner admin must set the driver status to "Offline".

- 5. Admin needs to delete a Driver with an accepted ride.
  The Dashboard controller must set to "Unassigned" or cancel the current Driver accepted ride.
  The Downtowner admin deletes de Driver.

- 6. Admin needs to delete a Rider with a requested ride.
  If the ride is not assigned, the system will cancel the Ride.
  If the ride is assigned:
  The Dashboard controller must cancel the Rider's ride.
  The Downtowner admin deletes the Rider.

- 7. Pick-up or Drop-off place are not found by autocomplete.
  Enter the place street address instead.

- 8. Pick-up or Drop-off street address are not found by autocomplete.
  Enter a nearby address.

- 9. The system is always reporting the Pick-up or Drop-off address as invalid.
  This is beacuse of:
  One of them is out of a Downtowner zone.
  One of them was not been selected from the autocomplete list.
  One of them was selected from the autocomplete list but was modified afterwards by the user typing in the address text input.
  To troubleshoot this you can:
  Use a nearby address.
  Make shure you select the address from the autocomplete list.

7. Conclusión

Luego del desarrollo de este trabajo, y teniendo en cuenta el Objetivo General inicialmente planteado, podemos concluir que el objetivo ha sido cumplido, se confeccionó un sistema informático capaz de administrar de forma optimizada, gracias a la algoritmia de recorridos, la adminsitración de viajes de la empresa de transporte. Esto fue posible dado que el sistema tenía las capacidades de:

- Capturar las solicitudes de viajes de los pasajeros:
  - Se cumplió mediante la implementación de la aplicación Rider Mobile App, trajo aparejado un aumento en la cantidad de clientes debido a que dicha aplicación sumó una vía de acceso al servicio que antes no existía.
- Asignar y administrar los viajes:
  - Este objetivo se logró a travez del WebService que procesa la logística de los viajes que ingresan mediantes las aplicaciones de Pasajeros y Conductores, y es entregada a los Administradores.
- Monitorizar a los conductores, vehículos y viajes en forma permanente
  - La creación de un Dashboard permitió que los Administradores se vieran beneficiados en un aumento de la información y control de los viajes.

Por el tipo de sistema de desarrollo acordado con el cliente, no contamos con métricas cuantificables que nos permitan medir la magnitud referidas a costos de mantenimiento de vehículos, reducciones de tiempo de carga de las baterías y reducción de tiempos ociosos de los conductores, la finalización del contrato al obtener la conformidad del cliente sí nos indicó que los objetivos fueron alcanzados.

8. Bibliografia

- https://developers.google.com/maps/documentation/geocoding.
- https://developers.google.com/places/policies
- https://developers.google.com/maps/documentation/distancematrix
- https://developers.google.com/places/policies
- https://developer.apple.com/library/ios/navigation/
- http://es.wikipedia.org/wiki/Desarrollo_en_cascada