

Instituto Tecnológico de Costa Rica



Escuela de Ingeniería en Computación

Desarrollo de Aplicaciones para Dispositivos Móviles

Prof. Andrei Fuentes
Prof. Jeff Schmidt

Documentación Proyecto Programado #2

Equipo de Trabajo

Segura Benavides, Esteban
IC

Torres Ruiz, Herberth
ATI

Junio, 2014

Tabla de contenido

Resumen ejecutivo	3
Propósito	4
Requerimientos de la Aplicación Móvil	4
Requerimientos del Backend	5
Descripción de diseño de alto nivel	5
Diagrama de Capas de la Aplicación	5
Diagrama de Componentes	6
Descripción detallada	7
Modelo de Dominio	7
Diagrama de Clases	8
Diagrama de Base de Datos	9
Problemas de diseño	10
Interacción con sistemas externos	10
Documentación del API del backend	11
Wireframes para la Idea de la Aplicación:	13
Manual de Usuario del Backend	15
Manual de Usuario de la Aplicación Móvil	18

Resumen ejecutivo

El presente documento describe el diseño de la aplicación móvil como parte del segundo proyecto programado para el curso de *Desarrollo de Aplicaciones para Dispositivos Móviles*.

Cabe destacar que esta es una iteración de la documentación externa y no se trata de un documento estático sino que será actualizado y adaptado conforme se avance en el desarrollo de la aplicación y se maduren más algunos conceptos específicos.

En general la aplicación pretende solucionar una necesidad insatisfecha que tienen los usuarios de transporte de nuestro país ya que a la hora de viajar a sus destino hay gran cantidad de carros que no utilizan toda su capacidad llegando inclusive a viajar conductores solos, lo que provoca mayor afluencia de vehículos y embotellamiento vial.

Para solventar dicho inconveniente surge la idea de desarrollar una aplicación móvil que enlace a los conductores que tienen espacios vacíos en sus vehículos con personas que compartan al menos parcialmente sus lugares de origen y destino, con el fin de que se pueda poner de acuerdo y viajar juntos, opcionalmente a cambio de una contribución.

En el presente documento se abarcan temas de la arquitectura y el diseño de la aplicación mencionada, así como una primera aproximación a lo que será la experiencia e interfaz de usuario de la misma.

Propósito

La aplicación surge para intentar solucionar el problema de la gran cantidad de carros que existe en el país, por medio de un sistema que permita a los usuarios registrados en la aplicación, dar a conocer las rutas por las que transcurre, con el fin de que otros usuarios conozcan de la ruta y si viajan a puntos en común pueda solicitar viajar juntos siempre que el usuario dueño del transporte lo permita.

La aplicación permitirá registrar viajes en carro con sus respectivos asientos disponibles y ofrecer esos asientos a personas que vayan de un origen cercano y a un destino similar, opcionalmente a cambio de un aporte económico para cubrir gastos de gasolina.

Los clientes de la aplicación serían cualquier tipo de persona que necesite viajar a su destino maximizando los recursos y ayudando al medio ambiente.

Requerimientos de la Aplicación Móvil

La aplicación permitirá a los usuarios:

- Registrarse en la aplicación: para esto el sistema solo solicitará correo y contraseña.
- Iniciar sesión con facebook utilizando el API de esta red social.
- Consultar y editar el perfil de usuario: El usuario podrá terminar de incluir información personal que le permita hacer un perfil más robusto.
- Ofrecer un viaje: es decir, si el usuario tiene un automóvil con espacios disponibles podrá ofrecer estos espacios. En este caso la aplicación solicitará:
 - Punto de Origen
 - Punto destino
 - Cantidad de espacios disponibles
 - Costo del viaje
 - Fecha
 - Hora de Partida
- Ver viajes disponibles y filtrarlos por origen y destino
- Solicitar un espacio disponible en un viaje
- Aceptar Solicitudes de Viajes (Reservar)
- Enviar mensajes al otro usuario sobre una reservación
- Cancelar una solicitud
- Cancelar un viaje

Requerimientos del Backend

El backend ofrecerá los endpoints necesarios para que el usuario de la aplicación móvil pueda realizar las funciones descritas anteriormente, sin embargo no es necesaria una página compleja de administración donde se introduzcan datos adicionales.

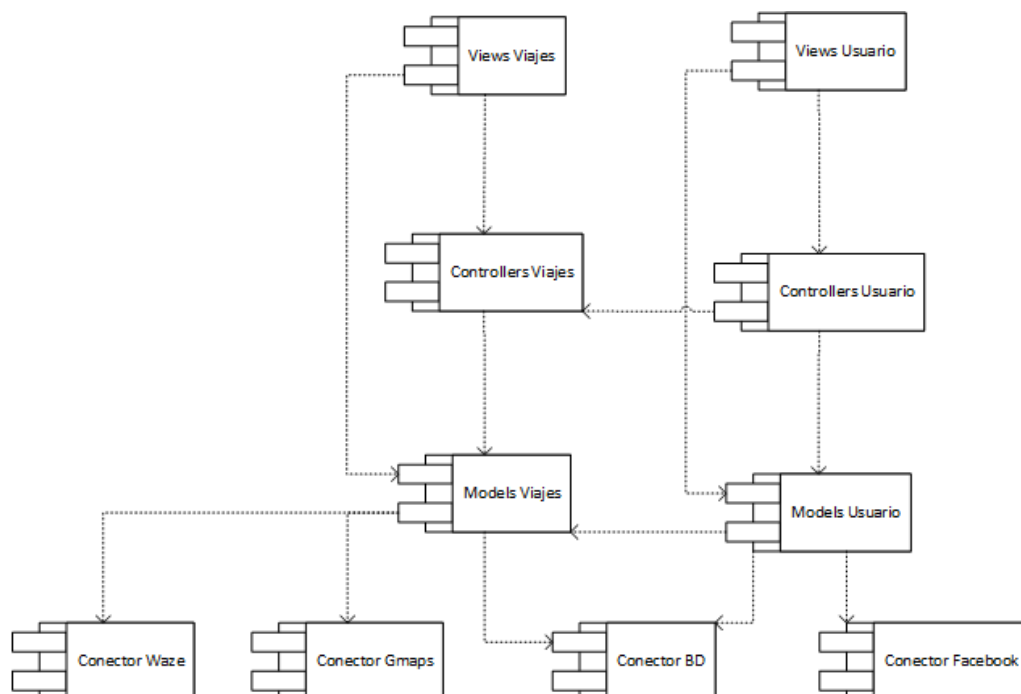
- Registrar y consultar información de Usuarios
- Registrar y consultar información de viajes
- Registrar y consultar información de solicitudes
- Cambiar el estado de una solicitud
- Registrar y consultar mensajes
- Cancelar viajes o solicitudes.

Descripción de diseño de alto nivel

Diagrama de Capas de la Aplicación

La aplicación se estructurará según la arquitectura de Modelo Vista Controlador, además se manejará una capa de acceso a datos en el que se ubicarán las clases encargadas de conectarse a servicios externos y a consultar o almacenar la información.

En el siguiente diagrama se detalla cómo se haría la división de las componentes de la aplicación:

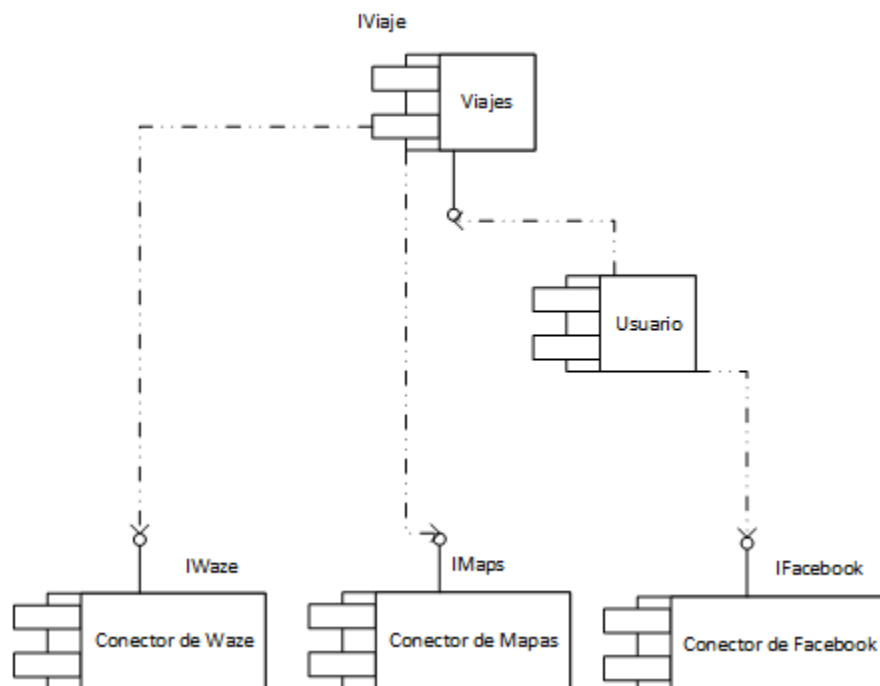


En la aplicación móvil el MVC se realiza de la siguiente manera:

- Las vistas son representadas por archivos de HTML que muestran la interfaz de usuario. Para darle un aspecto más estético se utilizaron librerías de jquery mobile que nos facilitaron la creación de elementos e íconos estándares.
- Los controladores son representados por archivos de Javascript que acceden al componente de los modelos y devuelven los datos a la vista según corresponda y además se encarga de manejar los eventos que el usuario genere.
- El acceso a datos está representado por una librería de Javascript que cumple con el propósito de acceder a la base de datos a través de los web services que el backend expone y devolver los resultados en un formato manejable por el controlador.

Diagrama de Componentes

A continuación se muestra el diagrama de componentes de la aplicación



A continuación una descripción de cada componente

- Componente de Viajes: En este componente se administran los viajes publicados y las solicitudes de los espacios disponibles, además de que le permite al usuario de una solicitud chatear con el que publicó el viaje. El componente se conecta con los mapas y el waze para mostrar las rutas y ubicar los puntos.

- **Componente de Usuarios:** En este componente se administran las cuentas de usuario y el perfil de los mismos. El componente se conecta con el componente de facebook para la autenticación de los usuarios y poder compartir contenido en la red social. Además también hay conexión con el componente de base de datos para mantener el almacén de los mismos.
- **Componente conector con Base de Datos:** Este es el componente general de redundancia de información y permite el acceso a los demás componentes sin que estos tengan conocimiento de la implementación de la base de datos permitiendo cambiar el DBMS sin causar cambios significativos en el resto de la aplicación.
- **Componente conector con Mapas:** Es el componente encargado de implementar la API de Google Maps para permitir a la aplicación mostrar la ubicación de los puntos.
- **Componente conector con Facebook:** Es el componente encargado de implementar la API de facebook permitiéndole al usuario la autenticación y compartir contenido sobre la aplicación en la red social.
- **Componente conector con Waze:** Es el componente que permite abrir la aplicación de Waze para calcular la ruta óptima y viajar al punto de destino.

Descripción detallada

Modelo de Dominio

A continuación se muestran las entidades del dominio y las relaciones que existen entre las mismas.

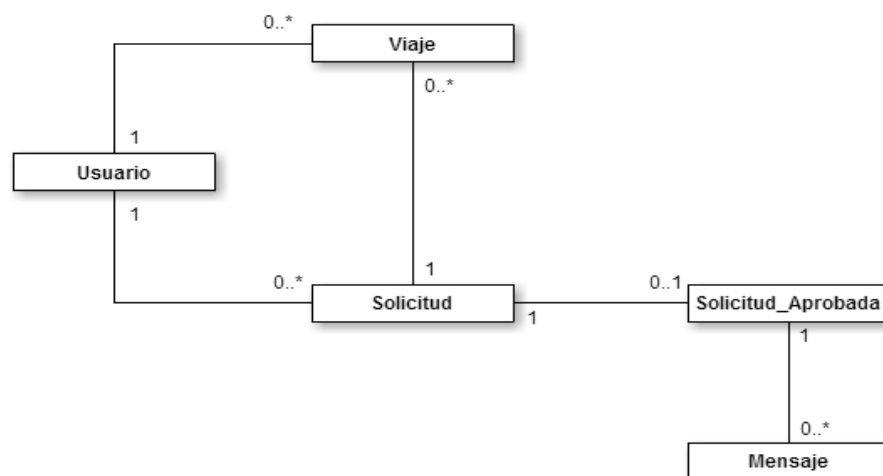


Diagrama de Clases

A continuación se muestra el diagrama general de clases necesarias para la aplicación.

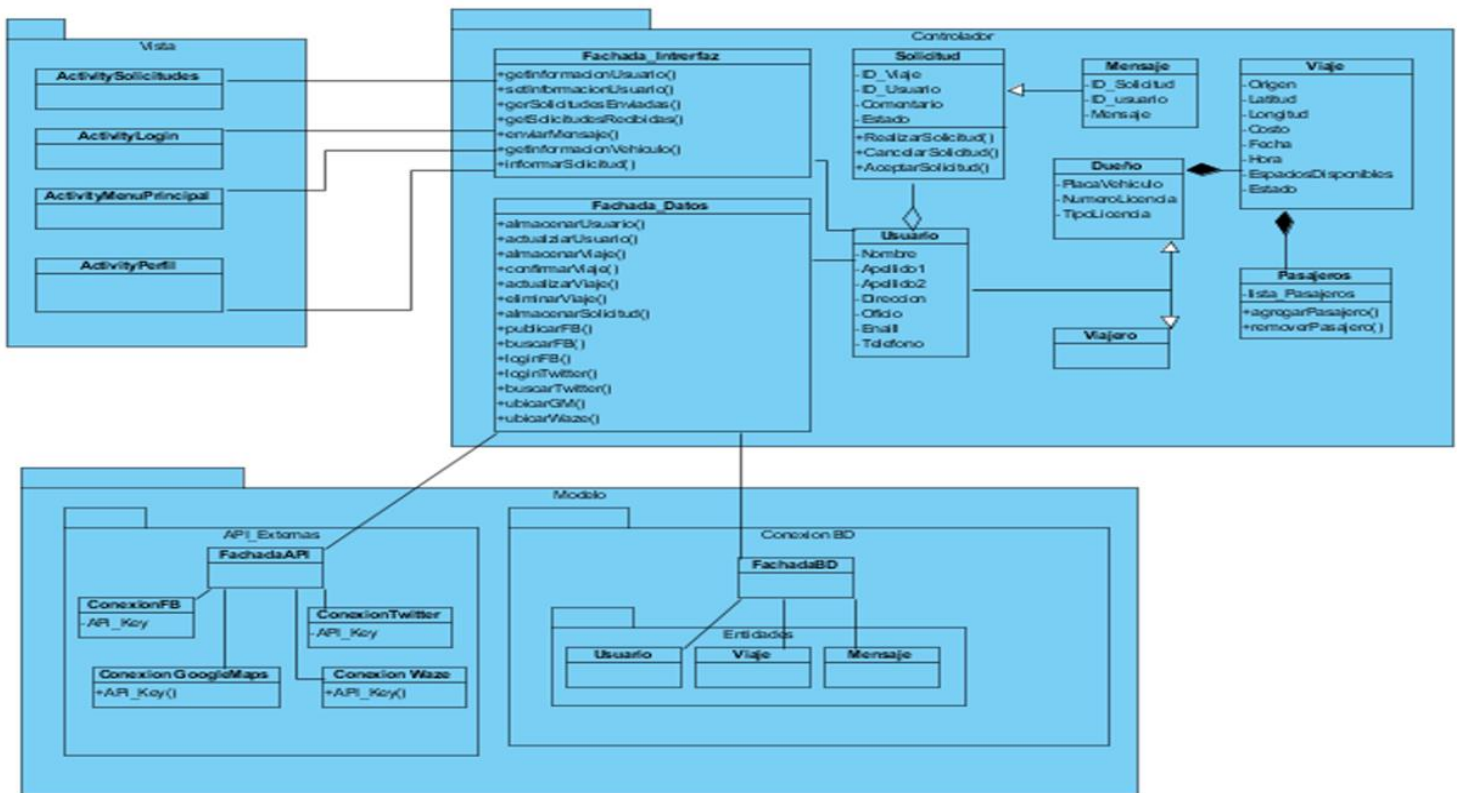
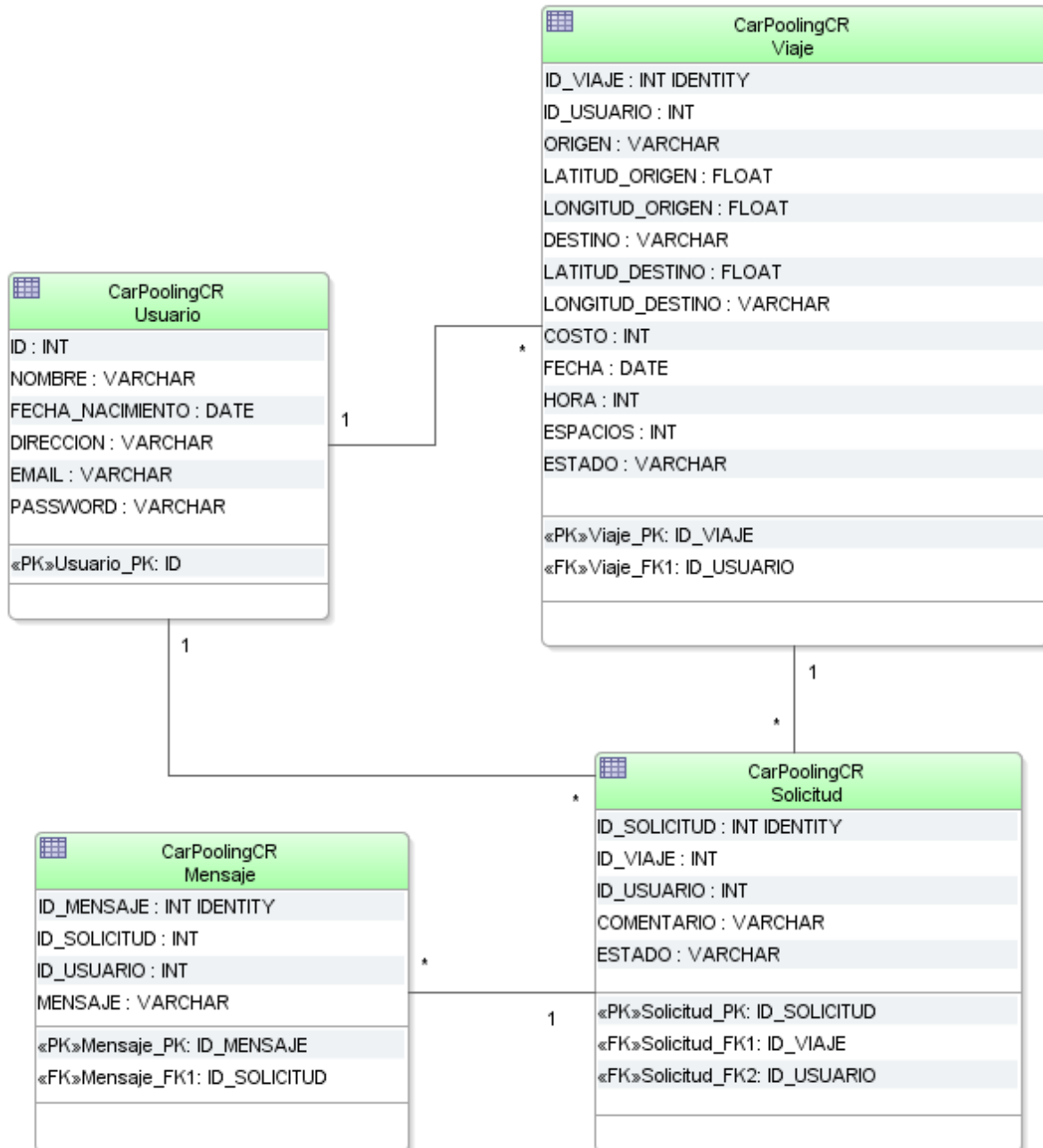


Diagrama de Base de Datos

A continuación se muestra el diagrama general de base de datos con las tablas necesarias para el almacenamiento de la información. En este caso se utilizará una BD relacional.



Problemas de diseño

- Rich client/thin client: La aplicación será un rich client ya que mantendrá la lógica de negocio y la información para conectarse a los servicios externos mientras que el servidor se encargará únicamente del almacenamiento de los datos.
- Dispositivos a Soportar: Se dará soporte a dispositivos de diferentes sistemas operativos utilizando un modelo multiplataforma lo que permite genera aplicaciones en Android, iOS, Windows Phone, etc.
- Conectividad: La aplicación necesitará de acceso a internet ya sea por wi-fi o por redes móviles ya que se trabajará con un perfil de usuario en línea, además de que se enlazará con facebook, twitter, waze y google maps.

Interacción con sistemas externos

Para el desarrollo del programa se utilizarán una conexión a las interfaces que brindan Facebook, Waze y Google Maps, y posteriormente se podrán agregar conexiones con otras redes sociales o servicios externos según requerimientos del negocio.

Para una descripción detallada de las APIs mencionadas se podrá visitar los siguientes enlaces:

Facebook SDK for javascript:

<https://developers.facebook.com/docs/javascript>

API de Waze:

<https://www.waze.com/es-419/about/dev>

API de Google Maps para javascript: biblioteca externa

<https://developers.google.com/maps/documentation/javascript/>

Documentación del API del backend

Api de Usuarios:

En los métodos de PUT y POST los parámetros de creación del objeto se pasan en el cuerpo de la solicitud.

Controller Method	Description	URI	HTTP Method
GetUsuarios	Gets all Usuarios.	api/Usuario	GET
GetUsuario	Finds a Usuario by ID.	api/Usuario/ <i>id</i>	GET
PutUsuario	Updates a Usuario.	api/Usuario/ <i>id</i>	PUT
PostUsuario	Creates a new Usuario.	api/Usuario	POST
DeleteUsuario	Deletes a Usuario.	api/Usuario/ <i>id</i>	DELETE

Api de Viajes:

En los métodos de PUT y POST los parámetros de creación del objeto se pasan en el cuerpo de la solicitud.

Controller Method	Description	URI	HTTP Method
GetViajes	Gets all Viajes.	api/Viaje	GET
GetViaje	Finds a Viaje by ID.	api/Viaje/ <i>id</i>	GET
PutViaje	Updates a Viaje.	api/Viaje/ <i>id</i>	PUT
PostViaje	Creates a new Viaje.	api/Viaje	POST
DeleteViaje	Deletes a Viaje.	api/Viaje/ <i>id</i>	DELETE

Api de Solicitudes:

En los métodos de PUT y POST los parámetros de creación del objeto se pasan en el cuerpo de la solicitud.

Controller Method	Description	URI	HTTP Method
GetSolicitudes	Gets all Solicitudes.	api/Solicitud	GET
GetSolicitud	Finds a Solicitud by ID.	api/Solicitud/ <i>id</i>	GET
PutSolicitud	Updates a Solicitud.	api/Solicitud/ <i>id</i>	PUT
PostSolicitud	Creates a new Solicitud.	api/Solicitud	POST
DeleteSolicitud	Deletes a Solicitud.	api/Solicitud/ <i>id</i>	DELETE

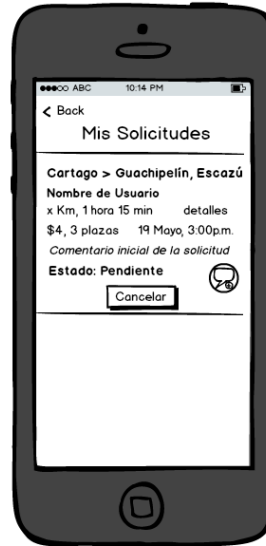
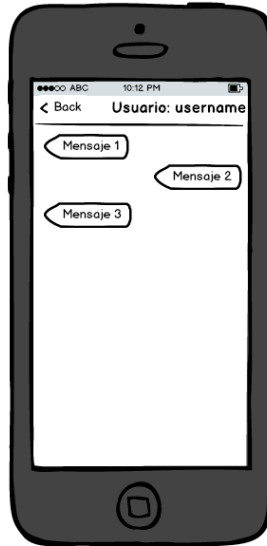
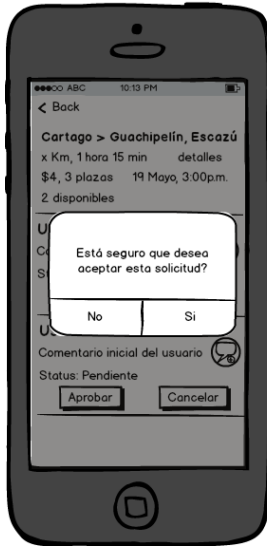
Api de Mensajes:

En los métodos de PUT y POST los parámetros de creación del objeto se pasan en el cuerpo de la solicitud.

Controller Method	Description	URI	HTTP Method
GetMensajes	Gets all Mensajes.	api/Mensaje	GET
GetMensaje	Finds a Mensaje by ID.	api/Mensaje/ <i>id</i>	GET
PutMensaje	Updates a Mensaje.	api/Mensaje/ <i>id</i>	PUT
PostMensaje	Creates a new Mensaje.	api/Mensaje	POST
DeleteMensaje	Deletes a Mensaje.	api/Mensaje/ <i>id</i>	DELETE

Wireframes para la Idea de la Aplicación:



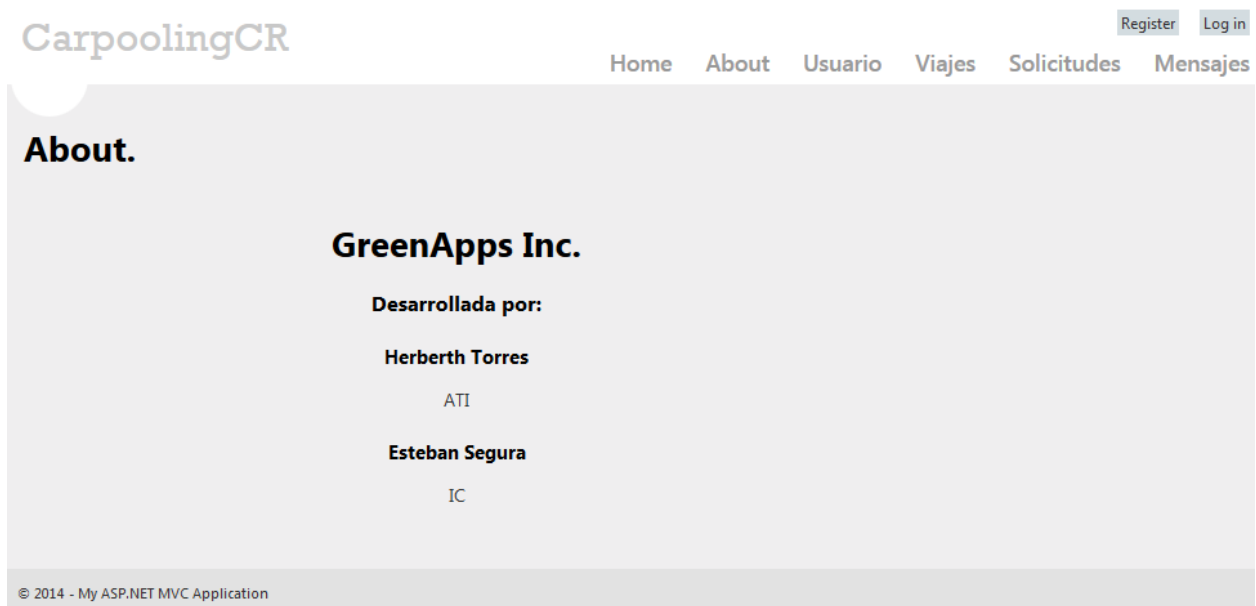


Manual de Usuario del Backend

Esta es la pantalla principal del backend, en esta se muestra una descripción de la Aplicación así como el menú principal para entrar a las otras opciones.



Esta es la ventana de “Acerca de...” de la aplicación con la información de los desarrolladores.



Esta es la ventana de gestión de usuarios en la que se puede visualizar los usuarios creados, crear usuarios nuevos, editar los datos de los usuarios y borrar algún usuario de la base de datos.

CarpoolingCR

RegisterLog in

HomeAboutUsuarioViajesSolicitudesMensajes

Index

[Create New](#)

NOMBRE	FECHA_NACIMIENTO	DIRECCION	EMAIL	PASSWORD	
Herberth Torres	7/7/1991 12:00:00 AM	Cartago, Cartago	herberthtorres1@gmail.com	123456	Edit Details Delete
Esteban Segura Benavides	3/17/1992 12:00:00 AM	Costa Rica, Alajuela	esteban.1703@hotmail.com	123456	Edit Details Delete

© 2014 - My ASP.NET MVC Application

Esta es la ventana de gestión de Viajes en la que se puede visualizar los viajes creados, crear viajes nuevos, editar los datos de los viajes y borrar algún viaje de la base de datos.

CarpoolingCR

RegisterLog in

HomeAboutUsuarioViajesSolicitudesMensajes

Index

[Create New](#)

ID_VIAJE	NOMBRE	ORIGEN	LATITUD_ORIGEN	LONGITUD_ORIGEN	DESTINO	LATITUD_DESTINO	LONGITUD_DESTINO	COSTO	FECHA
1	Esteban Segura Benavides	Hostal Plaza Iglesias	9.86252	-83.923954	Hotel Indigo San Jose Forum	9.925259	-84.079099	2000	6/29/2014 12:00:00 AM
3	Esteban Segura Benavides	Estación del Pacífico	9.9247838	-84.0818119	Angeles	9.928637	-84.0834625	2500	6/30/2014 12:00:00 AM
4	Esteban Segura Benavides	Oriental	9.8666667	-83.9166667	Occidental	9.8625864	-83.9254615	2000	6/30/2014 12:00:00 AM
5	Herberth Torres	Taller de Hojalateria Dymain	9.856688	-83.920881	Clínica Aguilar	9.862577	-83.920633	4500	6/29/2014 12:00:00 AM

Esta es la ventana de gestión de Solicitudes en la que se puede visualizar las solicitudes creadas, crear solicitudes nuevas, editar los datos de las solicitudes y borrar alguna solicitud de la base de datos.

CarpoolingCR

Register

Log in

HomeAboutUsuarioViajesSolicitudesMensajes

Index

[Create New](#)

ID_SOLICITUD	ORIGEN	NOMBRE	COMENTARIO	ESTADO	
1	Hostal Plaza Iglesias	Herberth Torres	vamos juntos!	Rechazada	Edit Details Delete
5	Oriental	Esteban Segura Benavides	Mi primera solicitud :')	Pendiente	Edit Details Delete
6	Estación del Pacífico	Esteban Segura Benavides	mi segunda solicitud :)	Pendiente	Edit Details Delete
7	Taller de Hojalateria Dymain	Esteban Segura Benavides	lleveme	Pendiente	Edit Details Delete
8	Estadio Nacional	Herberth Torres	Mae lleveme	Pendiente	Edit Details Delete

© 2014 - My ASP.NET MVC Application

Esta es la ventana de gestión de Mensajes en la que se puede visualizar los mensajes creados, crear mensajes nuevos, editar los datos de los mensajes y borrar algún mensaje de la base de datos.

CarpoolingCR

Register

Log in

HomeAboutUsuarioViajesSolicitudesMensajes

Index

[Create New](#)

ID_MENSAJE	COMENTARIO	ID_USUARIO	MENSAJE1	
1	vamos juntos!	1	Hola todo bien?	Edit Details Delete
2	vamos juntos!	4	Hola pvl	Edit Details Delete
4	vamos juntos!	1	Nuevo mensaje	Edit Details Delete

© 2014 - My ASP.NET MVC Application

Manual de Usuario de la Aplicación Móvil