

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE COMPUTACIÓN

LICENCIATURA DE ADMINISTRACIÓN DE TECNOLOGÍAS DE INFORMACIÓN

IC8041 - DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES

MeetMeAt

Proyecto de programación
Entrega final

Prof. Andrei Fuentes Leiva

Estudiantes:

Andrey Bolaños Sandoval

Ariela Vargas Vargas


Diego Vásquez Valerio

06 de octubre 2014

Cartago, Costa Rica

1 TABLA DE CONTENIDOS

2	Resumen ejecutivo.....	3
3	Funcionalidades	4
3.1	Administrativas	4
3.2	Cliente	4
3.3	Resumen de funcionalidades	4
4	Requerimientos.....	5
4.1	Requerimientos funcionales	5
4.2	Requerimientos no funcionales	6
5	Interfaz Gráfica, Experiencia y Usabilidad (UI/UX)	7
5.1	Wireframes	7
5.2	Fuentes y Esquema Colores	9
5.3	Mockups.....	10
6	Descripción de diseño de alto nivel	11
6.1	Diagrama de arquitectura conceptual	11
7	Descripción de diseño detallada	12
7.1	Diagrama de despliegue.....	12
7.2	Diagrama de clases	13
7.3	Diagrama de base de datos.....	14
8	herramientas utilizadas.....	15
8.1	Frontend.....	15
8.1.1	Onsen UI.....	15
8.1.2	Angular js.....	15
8.2	BackEnd.....	15
8.2.1	Cakephp	15
8.2.2	Autentificación.....	15
8.2.3	PHP REST SQL	16
8.2.4	Modificaciones al API	17
8.3	Patrones de diseño	18
8.3.1	Patrón de navegación	18
8.3.2	Patrón de navegación secundaria	19
8.3.3	Patrón de formularios	20

8.3.4	Patrón de búsquedas	20
9	Interacción con sistemas externos.....	21
9.1	API Facebook.....	21
9.1.1	Método de login.....	21
9.2	API Foursquare.....	21
9.2.1	Método de obtención de coordenadas geográficas	21
10	Manual de usuario	22
10.1	Inicio de la aplicación	22
10.2	Eventos 	23
10.3	Planificación de un evento.....	24
10.4	Perfil de usuario	25
10.5	Búsqueda de lugares.....	25
11	Pendientes	27

2 RESUMEN EJECUTIVO

MeetmeAt reúne a tus amigos y contactos en una sola aplicación para compartir las actividades sociales que desees realizar. Desde crear una lista de posibles actividades hasta elegir de manera aleatoria el próximo sitio para seguir la fiesta, eso sí, ajustándose a tu presupuesto y ubicación geográfica, para localizar los lugares de mayor proximidad.

Así también, MeetMeAt considera tus gustos, e incluye un filtro de los sitios que desees visitar según tus preferencias de música, ambientes sociales, comidas, entre otros.

Salí de la rutina con MeetMeAt, donde tendrás la facilidad de planear tus actividades sociales.

***Meet
MeAt***

3 FUNCIONALIDADES

Entiéndase el uso de la palabra gestionar como crear, consultar, actualizar y eliminar la entidad respectiva.

3.1 ADMINISTRATIVAS

1. Login/logout de usuarios
2. Gestionar usuarios
3. Gestionar tags
 - i. Los tags son entidades que almacenan datos relacionados con los eventos para clasificarlos por tipos de música, bebidas, comidas, ambiente, entre otros.

3.2 CLIENTE

1. Registrar usuario
 - a. Un usuario se registra a través de facebook.
2. Gestionar eventos
 - a. Al registrarse un usuario este puede crear eventos para planificar sus actividades sociales.
3. Gestionar actividades
 - a. La planificación del evento puede ser dividido en actividades específicas.
 - b. Así también el usuario puede crear un reto para la actividad (atributo reto).
 - c. Para determinar la ubicación de los eventos es necesario registrar los lugares, para ello el atributo venue obtiene de foursquare las coordenadas geográficas.
4. Invitar contactos
 - a. El usuario puede invitar a sus contactos a su evento para que notifiquen si se unen a este o no.

3.3 RESUMEN DE FUNCIONALIDADES

	Backend	Frontend
Login/logout de usuarios	Sí	Sí
Registrar usuario	Sí	Sí
Gestionar tags	Sí	No
Gestionar eventos	Sí	Sí
Gestionar actividades	Sí	Implementación de Foursquare.
Gestionar usuarios	Sí	No. Faltó invitar contactos.

4 REQUERIMIENTOS

Para el diseño de MeetMeAT es necesario definir los requerimientos funcionales y no funcionales que guiarán el desarrollo de la aplicación.

Los requerimientos funcionales indican lo que el sistema deben hacer y no deben hacer. Estas declaraciones de los servicios indican al sistema cómo éste debe reaccionar a entradas y situaciones que se presenten para el flujo continuo del programa.

Por otro lado, los requerimientos no funcionales indican cómo se deben llevar a cabo las funcionalidades en términos de calidad para que la solución final sea aceptada por el usuario.

4.1 REQUERIMIENTOS FUNCIONALES

RF-1	Gestionar usuarios
Funcional	El administrador podrá agregar, consultar, editar, y eliminar un usuario del sistema. Atributos: username, email, password, role, created, modified, status.

RF-2	Gestionar eventos
Funcional	El usuario podrá agregar, consultar, editar, y eliminar un evento de la aplicación. Atributos: title, theme, date, photo, description, budget, startTime, endTime.

RF-3	Gestionar actividades
Funcional	El usuario podrá agregar, consultar, editar, y eliminar una o más actividades asociadas a un evento de la aplicación. Atributos: name, description, budget, status, venue, challenge.

RF-4	Gestionar etiquetas
Funcional	El administrador podrá agregar, consultar, editar, y eliminar una etiqueta asociada a un evento de la aplicación. Las etiquetas son de datos relacionados con música, bebidas, dress code, animación, mood, ambientes. Atributos: name, description.

RF-5	Asignar una actividad a un evento
Funcional	El usuario podrá asignar una o más actividades al evento creado por él en la aplicación.

RF-6	Autenticación de usuarios con facebook
Funcional	Los usuarios pueden tener acceso a la aplicación al loguearse por medio de facebook.

RF-7	Importar los contactos de facebook del usuario
Funcional	La aplicación debe permitir al usuario importar sus contactos de facebook para que estos sean invitados del evento que desee crear.

RF-8	Invitar contactos a un evento
Funcional	El usuario podrá asignar uno o más invitados al evento creado por él.

4.2 REQUERIMIENTOS NO FUNCIONALES

RNF-1	Interrupciones, notificaciones y multitasking
No Funcional	Cuando el usuario responda a la interrupción (como aceptar una llamada o la lectura de un SMS), la aplicación tendrá la capacidad para funcionar normalmente y reanudar el servicio después de que el usuario termina el manejo de la interrupción o después de que prefieren ignorar la interrupción.

RNF-2	Resolución de pantalla y factores de forma
No Funcional	La aplicación debe ser capaz de ajustarse a la resolución de pantalla y factores de forma de los dispositivos móviles.

RNF-3	Tiempos de procesamiento
No Funcional	El tiempo aceptable para realizar funciones claves o datos de exportación/importación es de 10 segundos.

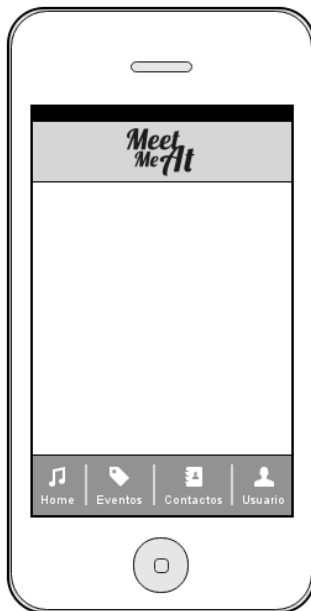
RNF-4	Consultas y tiempos de reportes
No Funcional	Las consultas o tiempos de reportes con sistemas externos deben considerar los tiempos de consulta aceptables (máximo 1 minuto), proporcionados por una API.

RNF-5	Funcionamiento de la aplicación considerando la ubicación
No Funcional	La ubicación geográfica, los requisitos de conexión y las restricciones de una red local prevalecen, por tanto deben ser considerados para lograr alcanzar las funcionalidades de la aplicación.

5 INTERFAZ GRÁFICA, EXPERIENCIA Y USABILIDAD (UI/UX)

5.1 WIREFRAMES

Versión original de la interfaz de la aplicación.



Master



Vista Inicial



Home



Eventos



Eventos



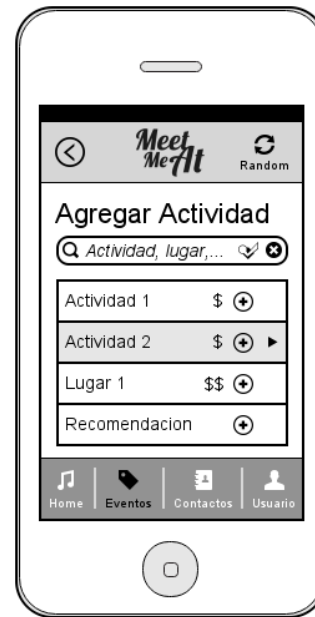
Eventos



Eventos



Eventos



Actividad



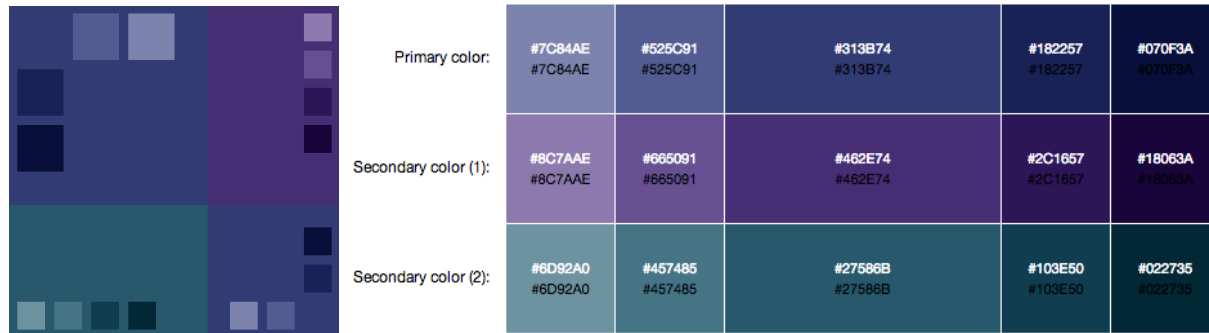
Contactos



Usuario

5.2 FUENTES Y ESQUEMA COLORES

El diseño visual seguirá la tendencia “azul” de las aplicaciones sociales. Se ha comprobado que este color se relaciona psicológicamente a conceptos como: comunicación, confianza y seguridad.¹ Se desea que MeetMeAt simplifique estos conceptos en la creación de eventos.

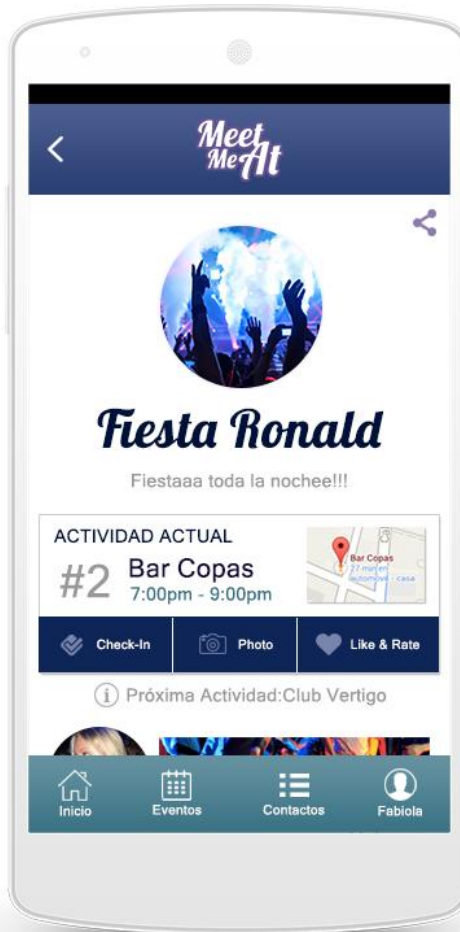


Link: <http://paletton.com/#uid=53T0u0klIllaFw0g0qFg0w0aF>

	Tipo Letra	Tamaño	Estilos	Color
Logo	Lobster	--	Bold	#070F3A / #7C84AE
Titulos 1	Sans, Droid Sans	18px	Bold	#070F3A
Títulos 2	Sans, Droid Sans	14px	Bold	#182257
Contenido	Sans, Droid Sans	12px	Normal	#000000
Formulario	Sans, Droid Sans	12px	Normal	#000000
Errores	Sans, Droid Sans	12px	Normal	#FF0000

¹ TOMADO DE “SOCIAL MEDIA GIANTS – WHY THEY USE BLUE AS THEIR PRIMARY BRANDING COLOUR”
[HTTP://WWW.ALICIAOWAN.COM/SOCIAL-MEDIA-AND-DIGITAL-MARKETING/STRATEGY-AND-ADVICE/SOCIAL-MEDIA-GIANTS-BRANDING](http://www.aliciacowan.com/social-media-and-digital-marketing/strategy-and-advice/social-media-giants-branding)

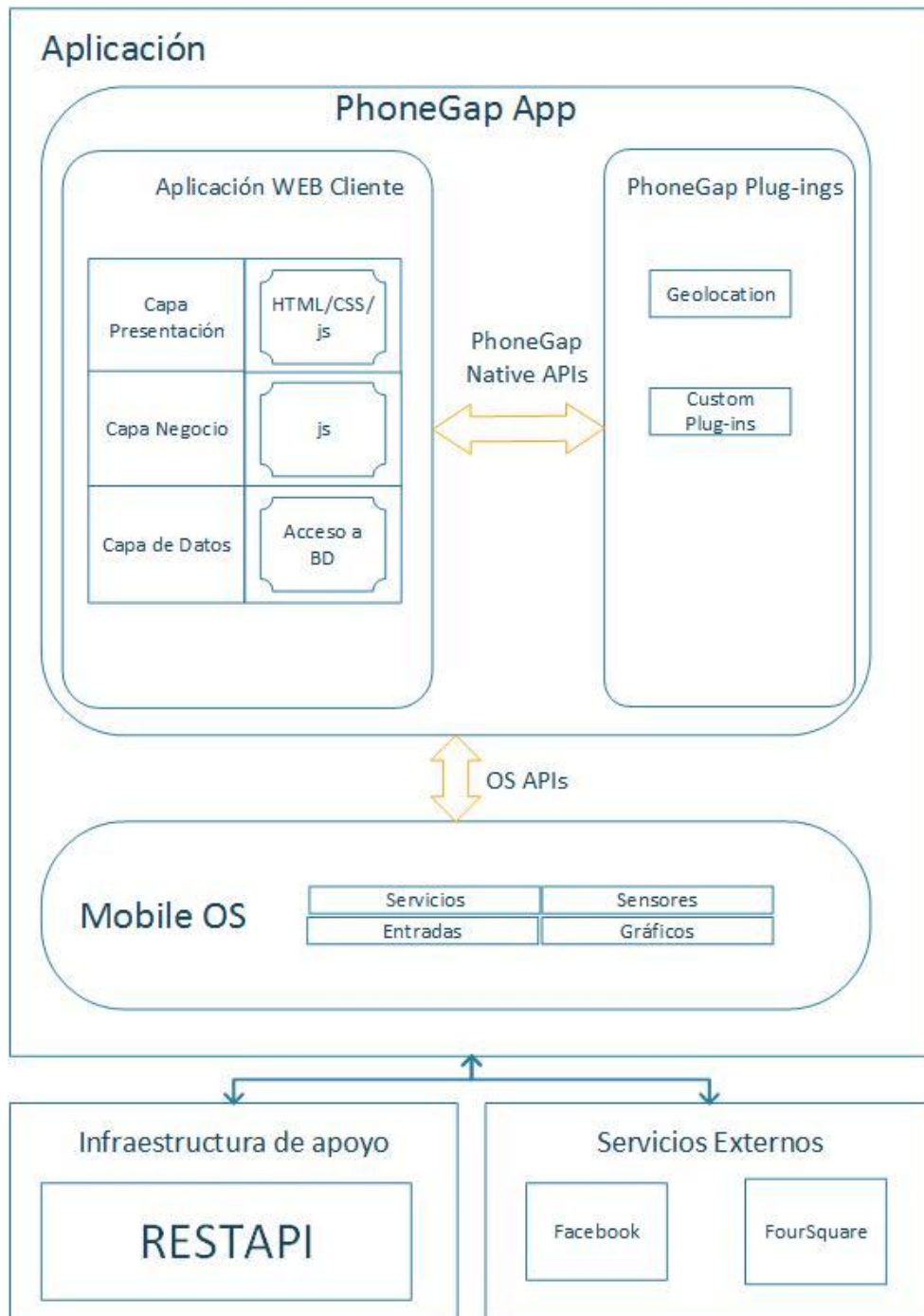
5.3 MOCKUPS



6 DESCRIPCIÓN DE DISEÑO DE ALTO NIVEL

6.1 DIAGRAMA DE ARQUITECTURA CONCEPTUAL

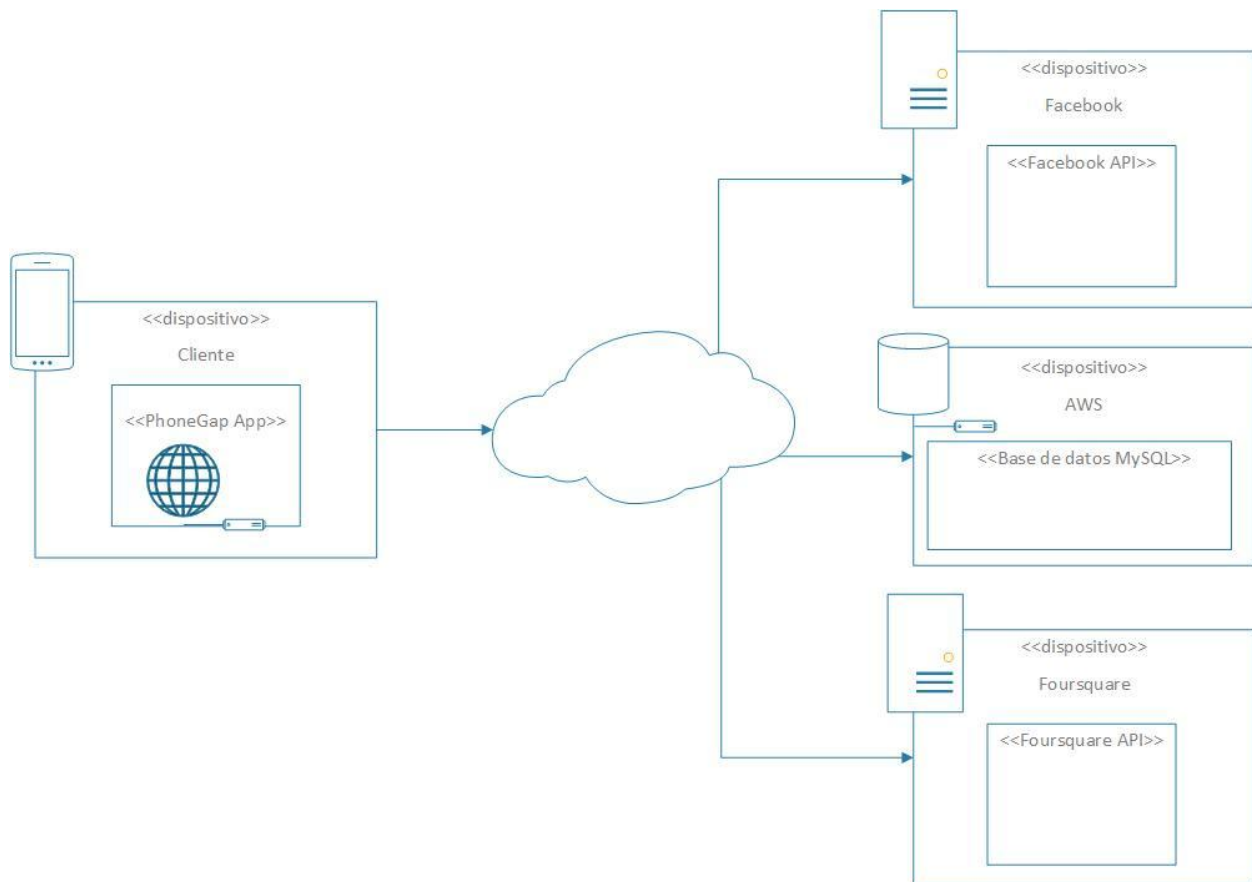
La aplicación se va a desarrollar utilizando el framework PhoneGap tomando ventaja de nuestros conocimientos de HTML, css y java script. La base de datos se va a hospedar utilizando los servicios proporcionados por Amazon.com con AWS.



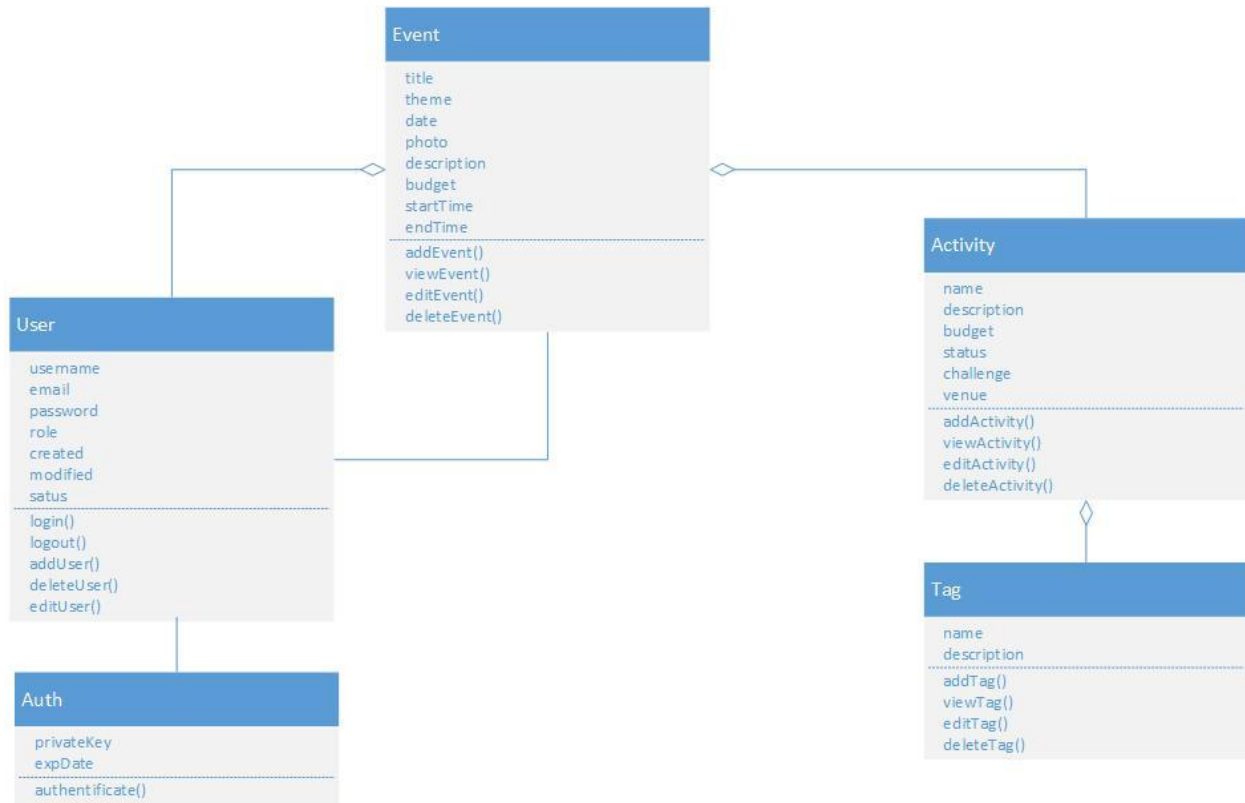
7 DESCRIPCIÓN DE DISEÑO DETALLADA

7.1 DIAGRAMA DE DESPLIEGUE

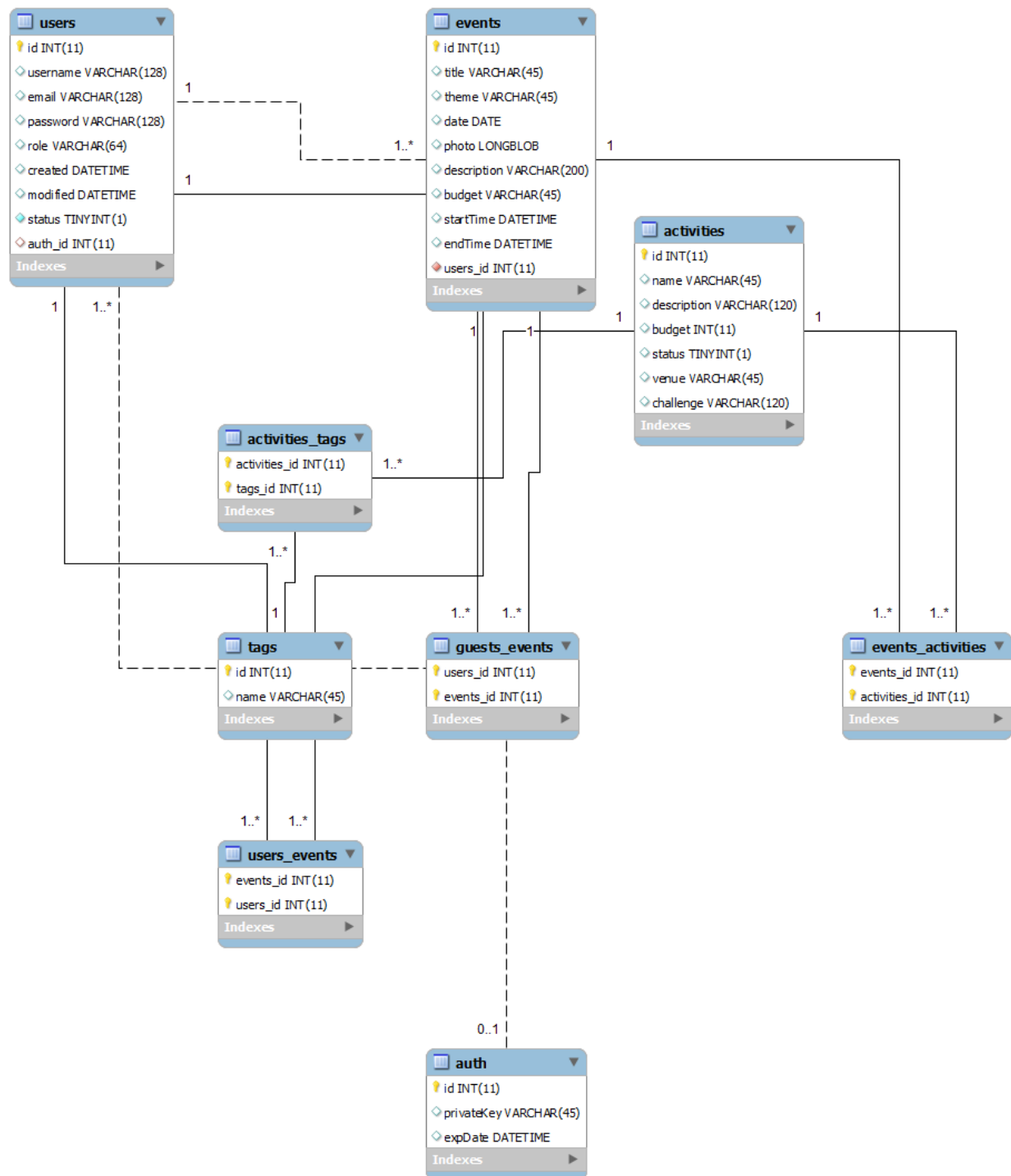
Este diagrama representa los dispositivos involucrados en el funcionamiento del sistema, así como los ambientes de ejecución para nuestra aplicación. El dispositivo móvil interactúa con el servidor que contiene la base de datos hospedada en Amazon AWS por medio de la red pública. Dicho dispositivo también interactúa con los servidores de facebook y de foursquare para obtener información de los contactos y lugares cercanos.



7.2 DIAGRAMA DE CLASES



7.3 DIAGRAMA DE BASE DE DATOS



8 HERRAMIENTAS UTILIZADAS

A continuación se describen las herramientas utilizadas para desarrollar la aplicación MeetMeApp.

8.1 FRONTEND

8.1.1 Onsen UI

Onsen UI permite desarrollar aplicaciones híbridas y web. En el caso de las aplicaciones híbridas se utiliza la línea de comandos Cordova/PhoneGap. (Asial Corporation, 2014)

Uso de plugin: <https://github.com/katzer/cordova-plugin-hidden-statusbar-overlay>.

8.1.2 Angular js

AngularJS es un framework JavaScript lanzado en 2012 por Miško Hevery, un empleado de Google, comenzó a trabajar en AngularJS en 2009. Este se distribuye como un archivo de JavaScript, y se puede agregar a una página web con una etiqueta `<script>`, extendiendo atributos HTML con las Directivas, y uniendo datos al HTML con Expresiones. (Google, 2014)

8.2 BACKEND

8.2.1 Cakephp

CakePHP es un framework o marco de trabajo que facilita el desarrollo de aplicaciones web, utilizando el patrón de diseño MVC (Modelo Vista Controlador). Es de código abierto y se distribuye bajo licencia MIT. Al igual que Ruby On Rails, CakePHP facilita al usuario la interacción con la base de datos mediante el uso de ActiveRecord. Además hace uso del patrón Modelo Vista Controlador. (Wikipedia, 2014)

En el caso práctico de la aplicación MeetMeApp se crearon los siguientes modelos, controladores, y vistas.

- Modelos: User.php, Event.php, Activity.php, Venue.php, Tag.php.
- Controller: UsersController.php, EventsController.php, ActivitiesController.php, VenuesController.php, TagsController.php.
- Vistas: en el caso de las vistas solo se crearon unas específicas para User (add.ctp, edit.ctp, login.ctp, index.ctp), las demás hacen uso de scaffolding.

8.2.2 Autenticación

Para la autenticación se escogió utilizar un método similar al utilizado por AWS de amazon. Nuestra adaptación parte del supuesto que un usuario se loguea en nuestro sistema y cuando es su primera vez se le asigna una llave publica y una privada. Cuando un usuario hace una consulta a nuestro backend se debe concatenar la llave pública y unos datos generados a partir de las llaves.

Los datos se generan de esta forma en php:

```
hash_hmac( 'sha256' , $PublicKey . $PrivateKey, $PrivateKey)
```

Cuando el backend obtiene dicha información de la consulta se busca cual es la llave privada que corresponde a esa llave publica y se combinan utilizando el mismo algoritmo. Luego de esto se comparan los dos datos generados y en caso de que sean iguales se permite la consulta.

8.2.3 PHP REST SQL

Es una interfaz HTTP REST para bases de datos MySQL escrito en PHP. A dicha interfaz se le configura la base de datos y ella sola se encarga dependiendo del URL al que se llame así muestra o crea información. Permite por ejemplo hacer peticiones GET para obtener datos de las tablas así como filas en específico.

A continuación se presentan unas imágenes de ejemplo:

Tables in database "andreybs_meetmeat"

- [activities](#)
- [activities_tags](#)
- [auth](#)
- [events](#)
- [events_activities](#)
- [guests_events](#)
- [tags](#)
- [users](#)
- [users_events](#)
- [venues](#)

Records in table "events" Record #3

- [2](#)
 - [4](#)
 - [5](#)
 - [7](#)
 - [8](#)
 - [27](#)
 - [31](#)
 - [32](#)
 - [33](#)
 - [34](#)
 - [35](#)
 - [36](#)
 - [3](#)
- | | |
|--------------------|---------------------|
| id | 3 |
| title | Prueba 2 |
| theme | Rock & Roll |
| date | 2014-09-04 |
| photo | |
| description | I feeeel good :) |
| budget | 12 |
| startTime | 2014-09-04 19:00:00 |
| endTime | 2014-09-05 02:00:00 |
| users_id | 3 |

Esta interface tiene la ventaja de que permite agregar un .json al final de la consulta y de esta manera cambia el tipo de respuesta a un archivo con formato .json.

```
[{"field": "id", "value": "3"}, {"field": "title", "value": "Prueba 2"}, {"field": "theme", "value": "Rock & Roll"}, {"field": "date", "value": "2014-09-04"}, {"field": "photo", "value": null}, {"field": "description", "value": "I feeeel good :)"}, {"field": "budget", "value": "12"}, {"field": "startTime", "value": "2014-09-04 19:00:00"}, {"field": "endTime", "value": "2014-09-05 02:00:00"}, {"field": "users_id", "value": "3"}]
```

8.2.4 Modificaciones al API

Se modificó el api de manera que permitiera devolver consultas personalizadas por ejemplo un evento es importante en nuestra lógica obtener todas las actividades asociadas.

Por ejemplo:

Record #3

```

    id      3
    title   Prueba 2
    theme   Rock & Roll
    date    2014-09-04
    photo
description I feeeel good :)
    budget  12
    startTime 2014-09-04 19:00:00
    endTime  2014-09-05 02:00:00
    users_id [{"field":"id","value":"3"}, {"field":"username","value":"Ari"},
              {"field":"email","value":"arivargas17@gmail.com"}, {"field":"password","value":"koala123"},
              {"field":"role","value":null}, {"field":"created","value":null}, {"field":"modified","value":null},
              {"field":"status","value":"1"}, {"field":"auth_id","value":[{"field":"id","value":"1000"},
              {"field":"privateKey","value":"GSAUDUASH75667565"},
              {"field":"expDate","value":null}]}]
    Actividades [{"field":"activities_id","value":[{"field":"id","value":"2"},
              {"field":"name","value":"activity"}, {"field":"description","value":"blablabla"},
              {"field":"budget","value":"89983"}, {"field":"status","value":"0"},
              {"field":"venues_id","value":[{"field":"id","value":"1"}, {"field":"lat","value":"85.63"},
              {"field":"long","value":"88.63"}]}, {"field":"challenge","value":null}]}],
              {"field":"activities_id","value":[{"field":"id","value":"3"},
              {"field":"name","value":"activity3"}, {"field":"description","value":"hola"},
              {"field":"budget","value":"245424"}, {"field":"status","value":"0"},
              {"field":"venues_id","value":[{"field":"id","value":"1"}, {"field":"lat","value":"85.63"},
              {"field":"long","value":"88.63"}]}, {"field":"challenge","value":null}]}]
```

8.3 PATRONES DE DISEÑO

8.3.1 Patrón de navegación

8.3.1.1 Springboard



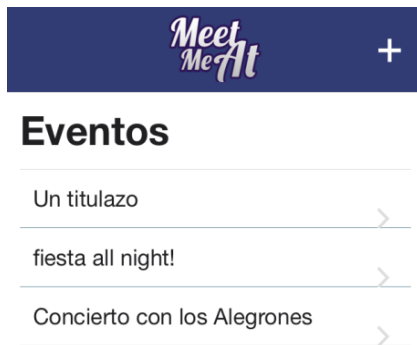
El patrón springboard se ejemplifica con uso de los botones inferiores para el ingreso a la aplicación.

8.3.1.2 Tabs



El uso de los tabs: Home, Eventos, Buscar, Amigos y Perfil son implementdas para facilitar el flujo de las funcionalidad de la aplicación para el usuario.

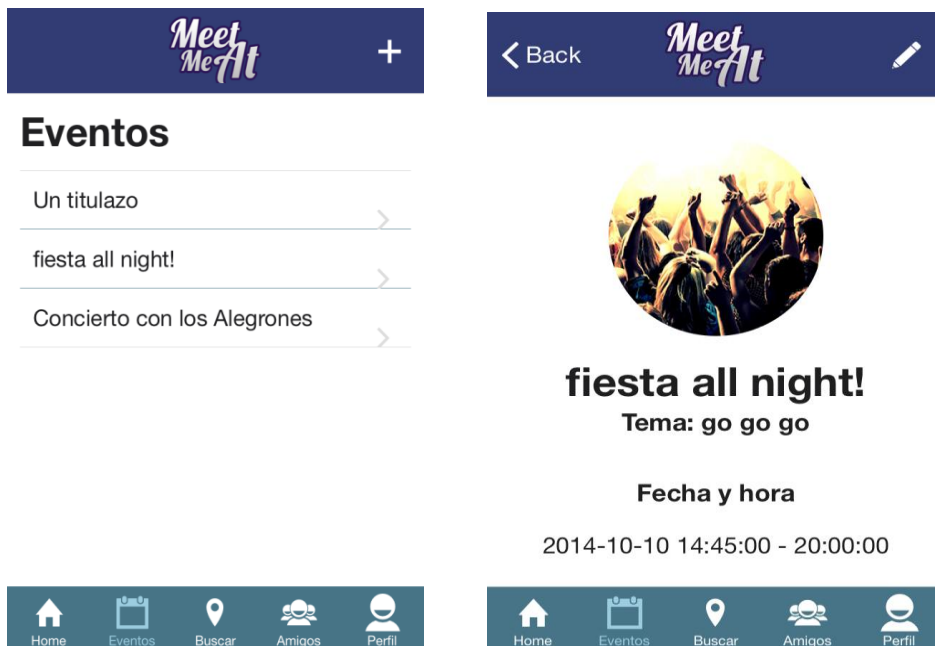
8.3.1.3 Listview



El patrón listview es necesario para enlistar una cantidad de eventos de manera ordenada los cuales el usuario pueda identificar, y en caso de requerir conocer más de este, dar clic y tener los detalles.

8.3.2 Patrón de navegación secuandaria

8.3.2.1 Master-detail



El patrón master and detail enlist a serie de eventos los cuales pueden ser accedidos para en caso conocer más de estos, mediante el clic y así tener los detalles de un evento.

8.3.3 Patrón de formularios

8.3.3.1 Formularios

The screenshot shows the 'Crear Evento' form. At the top is a dark blue header with a white back arrow and the 'Meet MeAT' logo. Below the header, the title 'Crear Evento' is in bold. The form consists of several input fields: 'Titulo' (Title), 'Tema' (Topic), 'Fecha' (Date) with a dropdown arrow, 'Inicia' (Start) with a dropdown arrow, and 'Fin:' (End) with a dropdown arrow. At the bottom, there is a 'Foto:' label and a row of five icons: Home, Eventos, Buscar, Amigos, and Perfil.

The screenshot shows the 'Editar Evento' form. It has the same header as the 'Crear Evento' form. The title 'Editar Evento' is in bold. The form fields are: 'Titulo' (Title) with the text 'fiesta all night!', 'Tema' (Topic) with the text 'go go go', 'Fecha' (Date) with a dropdown arrow showing 'Oct 10, 2014', 'Inicia' (Start) with a dropdown arrow showing '2:45 PM', and 'Fin:' (End) with a dropdown arrow showing '8:00 PM'. At the bottom, there is a 'Foto:' label and a row of five icons: Home, Eventos, Buscar, Amigos, and Perfil.

Los formularios son utilizados para ingresar información de manera ordenada, según un flujo de información coherente para el usuario, donde se permita entender el propósito del formulario. En este caso, el formulario se implementa para gestionar un evento.

8.3.4 Patrón de búsquedas

8.3.4.1 Búsqueda explícita

The screenshot shows the '¿Donde vamos?' search screen. At the top is a dark blue header with the 'Meet MeAT' logo and a clock icon. Below the header, the title '¿Donde vamos?' is in bold. Underneath, it says 'Powered by Foursquare'. There is a search bar with a magnifying glass icon, a dropdown menu showing '250mts', and a search button with a magnifying glass icon. Below the search bar is a map showing a location marked with a star. The map includes labels for 'Ontario Dr', 'Blackhawk', 'N De Anza', 'N Blaney', and 'Merritt'. At the bottom, there is a row of five icons: Home, Eventos, Buscar, Amigos, and Perfil.

La búsqueda explícita se utiliza en la barra de búsqueda para encontrar un lugar, implementando el uso de gps con Foursquare.

9 INTERACCIÓN CON SISTEMAS EXTERNOS

La aplicación tiene interacción con otros sistemas externos para permitir diversas funcionales. Para conocer esos sistemas se realiza una breve descripción de los web services utilizados, explicando los métodos del API que debieron ser implementados.

9.1 API FACEBOOK

Facebook ofrece un entorno de software para crear aplicaciones y servicios que accedan a datos de la red social. El API permite funcionalidades como la autenticación y plugin sociales, los cuales serán utilizados para conocer la información del perfil del usuario, foto de perfil, así como para obtener una lista de sus amigos y sus nombres.

9.1.1 Método de login

Para este método se utilizó el post del autor Christophe Coenraets, Cordova/PhoneGap Facebook Integration without Plugin.

Descripción: openfb. Esta solución implementa el OAuth para conectarse a Facebook haciendo solicitudes HTTP directas al API, donde se recuperan y envían datos.

```
openFB.api (
  {
    method: 'POST',
    path: '/me/feed',
    params: {
      message: 'Testing the Facebook Graph API'
    },
    success: successHandler,
    error: errorHandler
  });
```

9.2 API FOURSQUARE

Foursquare brinda acceso a su base de datos de lugares, así como la posibilidad de interactuar con sus usuarios. Este API nos permite buscar lugares en un área determinada, conectarse con amigos, saber cuando un usuario hace un check-in e incluso obtener datos globales. Este API será implementado para recomendar lugares de interés a sus actividades.

9.2.1 Método de obtención de coordenadas geográficas

Descripción: FoursquareCall.php

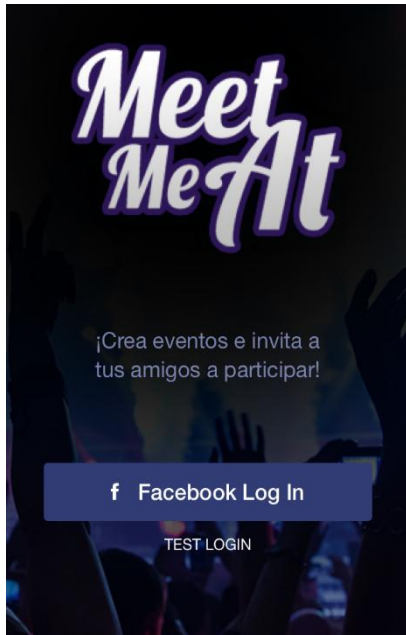
Este servicio nos retorna cuales son los lugares de interes o venues cercanos a un punto geográfico. Recibe latitud y longitud, y devuelve un JSON con todos los lugares cercanos.

Url: <http://meetmeat.andreybolanos.com/call/FourSquareCall.php?lat=9.927297&long=-84.037523>

10 MANUAL DE USUARIO

10.1 INICIO DE LA APLICACIÓN

- 1) Registro de usuario: para registrarse en la aplicación el usuario debe conectarse vía Facebook.



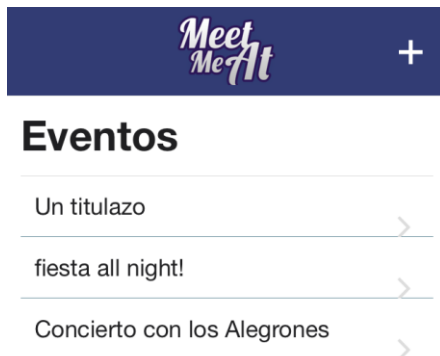
- 2) Visualización de información: la pantalla del home después de loguearse le muestra lo que está sucediendo para el día actual.



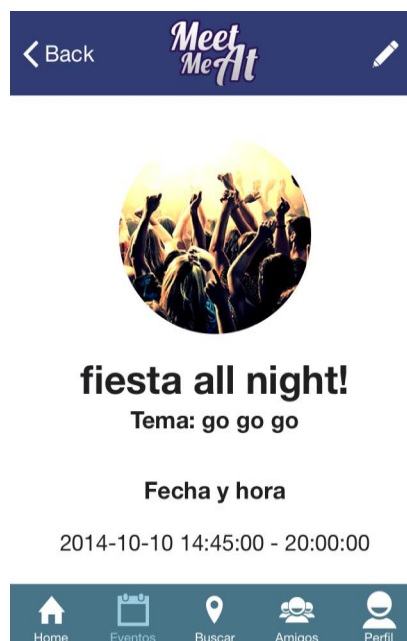
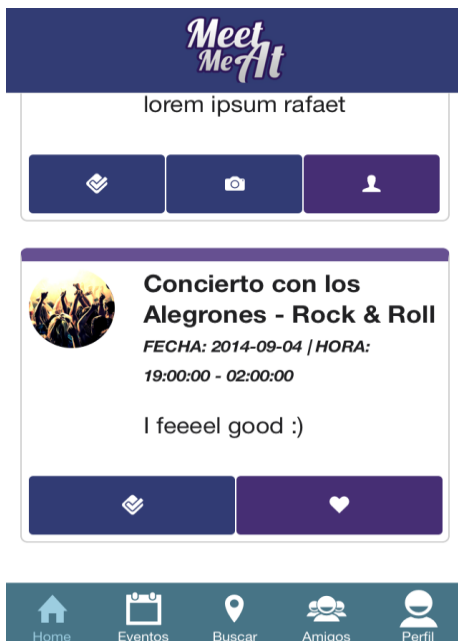
10.2 EVENTOS




- 3) El home muestra la lista de eventos que tienen relación con el usuario.

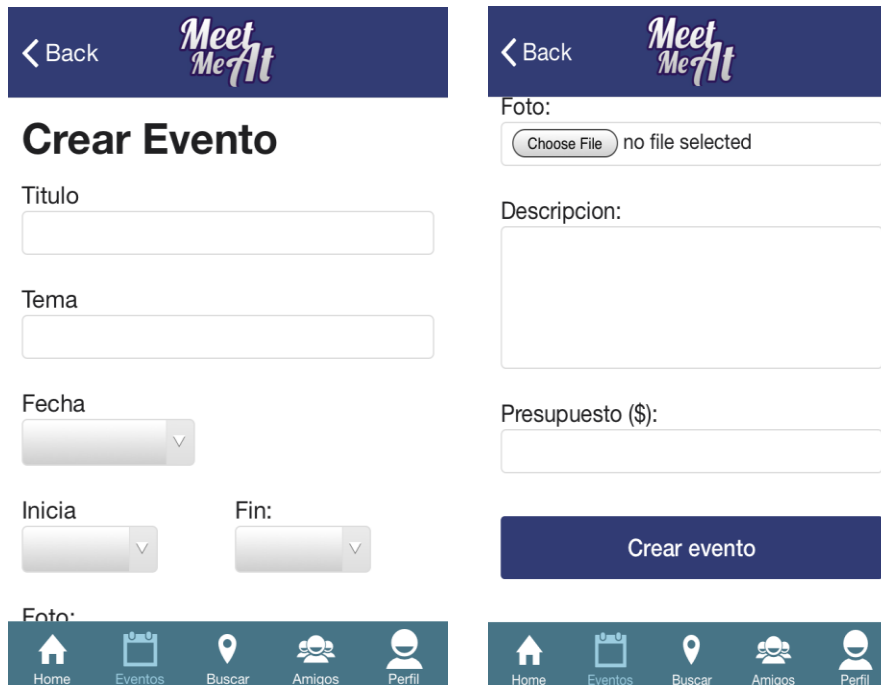


- 4) Mostrar un evento: de la lista que aparece el usuario selecciona un evento y se le despliega la información.



10.3 PLANIFICACIÓN DE UN EVENTO

- 5) Registro de un evento: para registrar un evento es se debe dar clic en  e ingresar título, tema, fecha, hora de inicio y fin, una foto si así se desea, la descripción del evento y el presupuesto si aplica.



Crear Evento

Titulo

Tema

Fecha

Inicia

Fin:

Foto:

Presupuesto (\$):

Crear evento

Home Eventos Buscar Amigos Perfil

- 6) Editar un evento: para editar un evento se da clic sobre el evento para actualizar la información que se desee.



10.4 PERFIL DE USUARIO

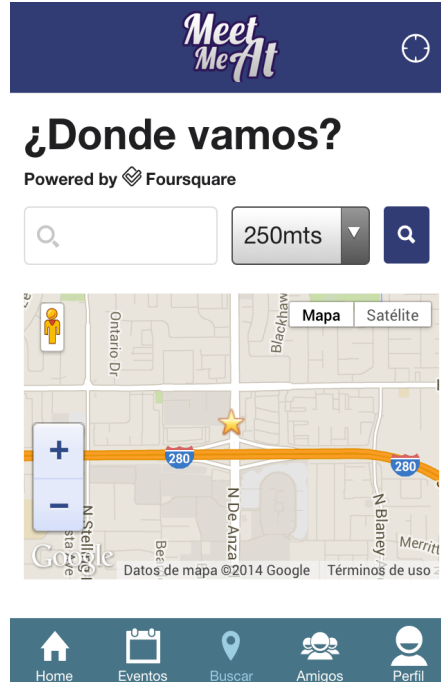
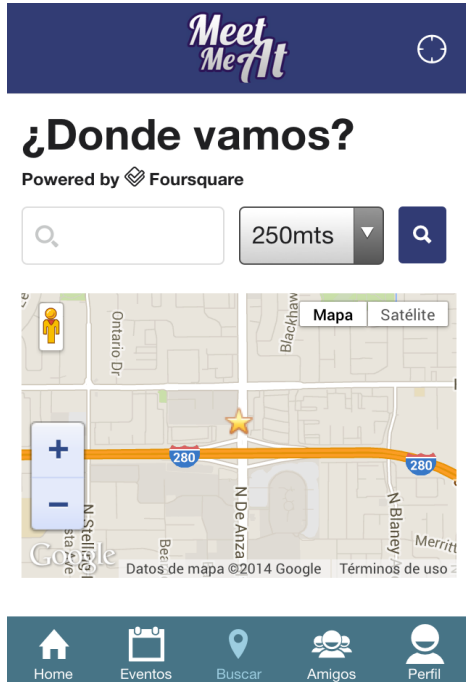
7) Para visualizar el perfil de usuario se accede al botón de perfil



10.5 BÚSQUEDA DE LUGARES

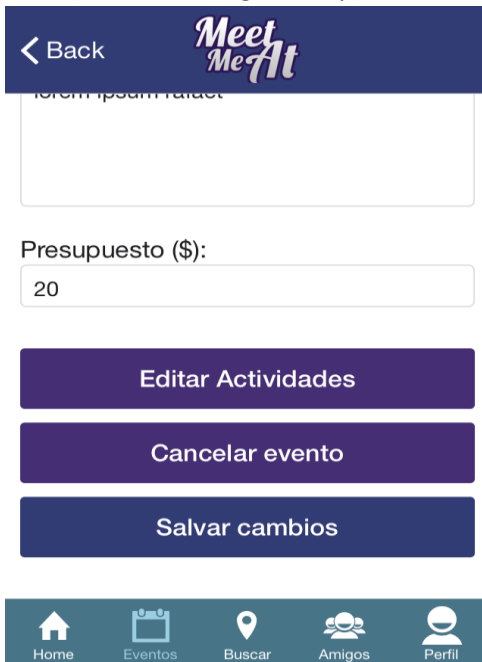
8) La búsqueda de lugares se realiza mediante el botón de buscar
ahí se escribe el texto y se da clic sobre la lupa.






11 PENDIENTES

- 1) Tags: implementación de la interfaz.
Problema con implementación tomando en consideración relaciones entre dos entidades de N..M (HasAndBelongToMany)
- 2) Actividades de un evento: implementación de la interfaz.
Problema con implementación tomando en consideración relaciones entre dos entidades de N..M (HasAndBelongToMany)



The screenshot shows the 'Meet Me At' app interface for editing event activities. At the top, there is a dark blue header with a back arrow and the text '< Back' on the left, and the 'Meet Me At' logo on the right. Below the header is a large white rectangular area for text input. Underneath this area is a label 'Presupuesto (\$):' followed by a text input field containing the number '20'. Below the input field are three stacked buttons: 'Editar Actividades' (purple), 'Cancelar evento' (purple), and 'Salvar cambios' (dark blue). At the bottom of the screen is a navigation bar with five icons: a house for 'Home', a calendar for 'Eventos', a location pin for 'Buscar', two people for 'Amigos', and a person profile for 'Perfil'.

- 3) Invitar amigos 
Problema: Facebook solo devuelve amigos que también usan el APP, por lo tanto actualmente es vacía y se requiere una solución alterna.

