

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

29 de setiembre 2014

Cartago, Costa Rica

1 TABLA DE CONTENIDOS

2	Resumen ejecutivo.....	3
3	Funcionalidades	4
3.1	Administrativas	4
3.2	Cliente	4
4	Requerimientos.....	5
4.1	Requerimientos funcionales	5
4.2	Requerimientos no funcionales	6
5	Interfaz Gráfica, Experiencia y Usabilidad (UI/UX)	8
5.1	Wireframes	8
5.2	Fuentes y Esquema Colores	10
5.3	Mockups.....	11
6	Descripción de diseño de alto nivel	12
6.1	Diagrama de despliegue.....	13
6.2	Diagrama de clases	14
6.3	Diagrama de base de datos.....	15
7	Descripción detallada.....	16
7.1	Frontend.....	16
7.1.1	Onsen UI.....	16
7.1.2	Angular js.....	16
7.2	BackEnd.....	16
7.2.1	Cakephp	16
7.2.2	Autenticación	16
7.2.3	PHP REST SQL	17
7.3	Patrones de diseño	19
7.3.1	Patrón de navegación	19
7.3.2	Patrón de navegación secundaria	20
7.3.3	Patrón de formularios	21
7.3.4	Patrón de búsquedas	21
8	Interacción con sistemas externos.....	23
8.1	API Facebook.....	23
8.1.1	Método de login.....	23

8.2	API Foursquare.....	23
8.2.1	Método de obtención de coordenadas geográficas	23

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.
4. Gestionar ubicación del lugar
 - a. Para determinar la ubicación de los eventos es necesario registrar los lugares, para ellos la entidad Venues obtiene de foursquare las coordenadas geográficas de latitud y longitud.
5. Invitar contactos
 - a. El usuario puede invitar a sus contactos a su evento para que notifiquen si se unen a este o no.

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.

RF-4	Gestionar ubicación del lugar
Funcional	El usuario podrá agregar, consultar, editar, y eliminar la ubicación de un lugar en relación con un evento de la aplicación. Los datos para agregar la ubicación son obtenidos por medio de uso del API de Foursquare. Atributos: latitude, longitude.

RF-5	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-6	Asignar una actividad a un evento
Funcional	El usuario podrá asignar una o más actividades al evento creado por él en la aplicación.

RF-7	Autenticación de usuarios con facebook
Funcional	Los usuarios pueden tener acceso a la aplicación al loguearse por medio de facebook.

RF-8	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-9	Invitar contactos a un evento
Funcional	El usuario podrá asignar uno o más invitados al evento creado por él.

RF-10	Calcular presupuesto del evento
Funcional	El sistema deberá calcular el presupuesto total del evento según la suma del costo de las actividades, si existen actividades relacionadas al evento.

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.

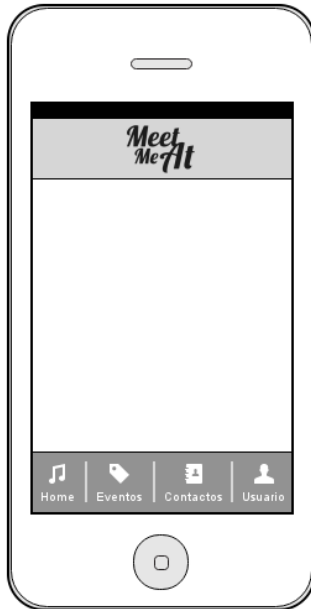
RNF-4	Consultas y tiempos de reportes
No Funcional	Las consultas o tiempos de reportes con sistemas externos deben considerar los tiempos de consulta aceptables (máximo 1 minuto), proporcionados por una API.

RNF-5	Funcionamiento de la aplicación considerando horarios de disponibilidad
No Funcional	La aplicación debe estar disponible 24/7.

RNF-6	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



Master



Vista Inicial



Home



Eventos



Eventos



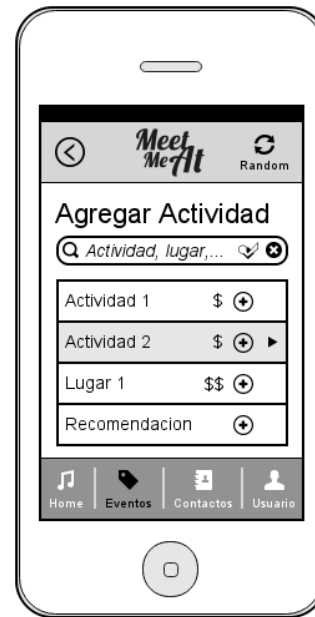
Eventos



Eventos



Eventos



Actividad



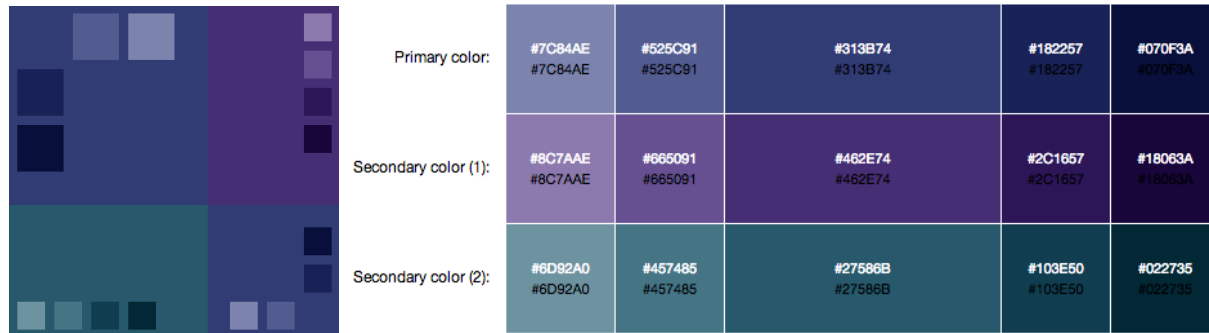
Contactos



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.

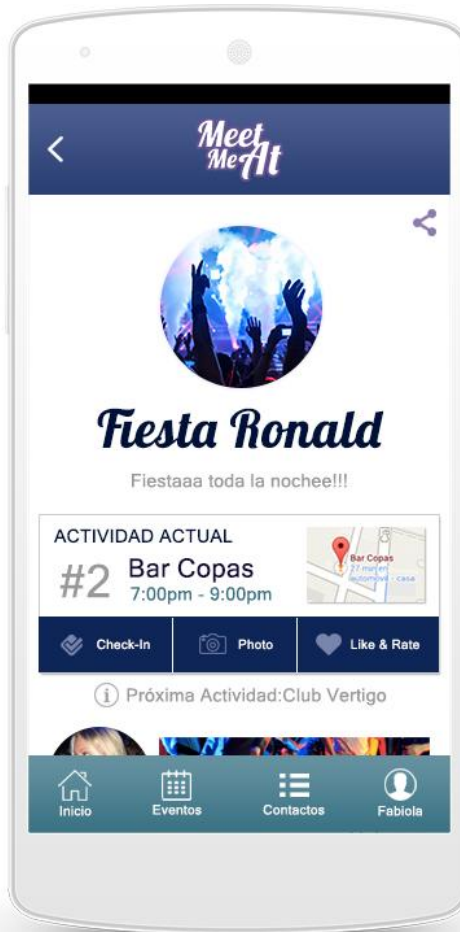


Link: <http://paletton.com/#uid=53T0u0klIllaFw0g0qFg0w0aF>

	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

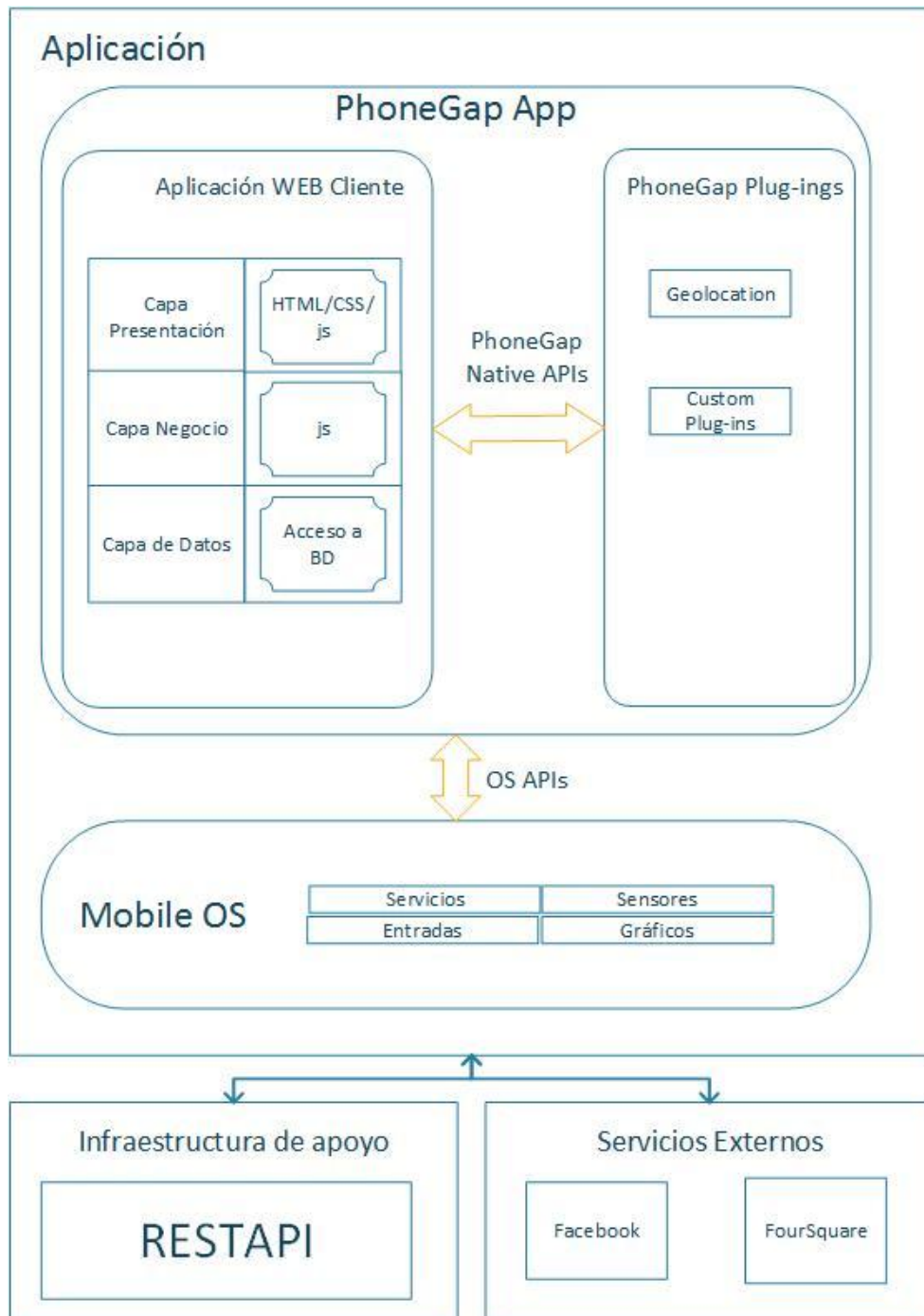
¹ TOMADO DE “SOCIAL MEDIA GIANTS – WHY THEY USE BLUE AS THEIR PRIMARY BRANDING COLOUR”
[HTTP://WWW.ALICIAOWAN.COM/SOCIAL-MEDIA-AND-DIGITAL-MARKETING/STRATEGY-AND-ADVICE/SOCIAL-MEDIA-GIANTS-BRANDING](http://www.aliciacowan.com/social-media-and-digital-marketing/strategy-and-advice/social-media-giants-branding)

5.3 MOCKUPS



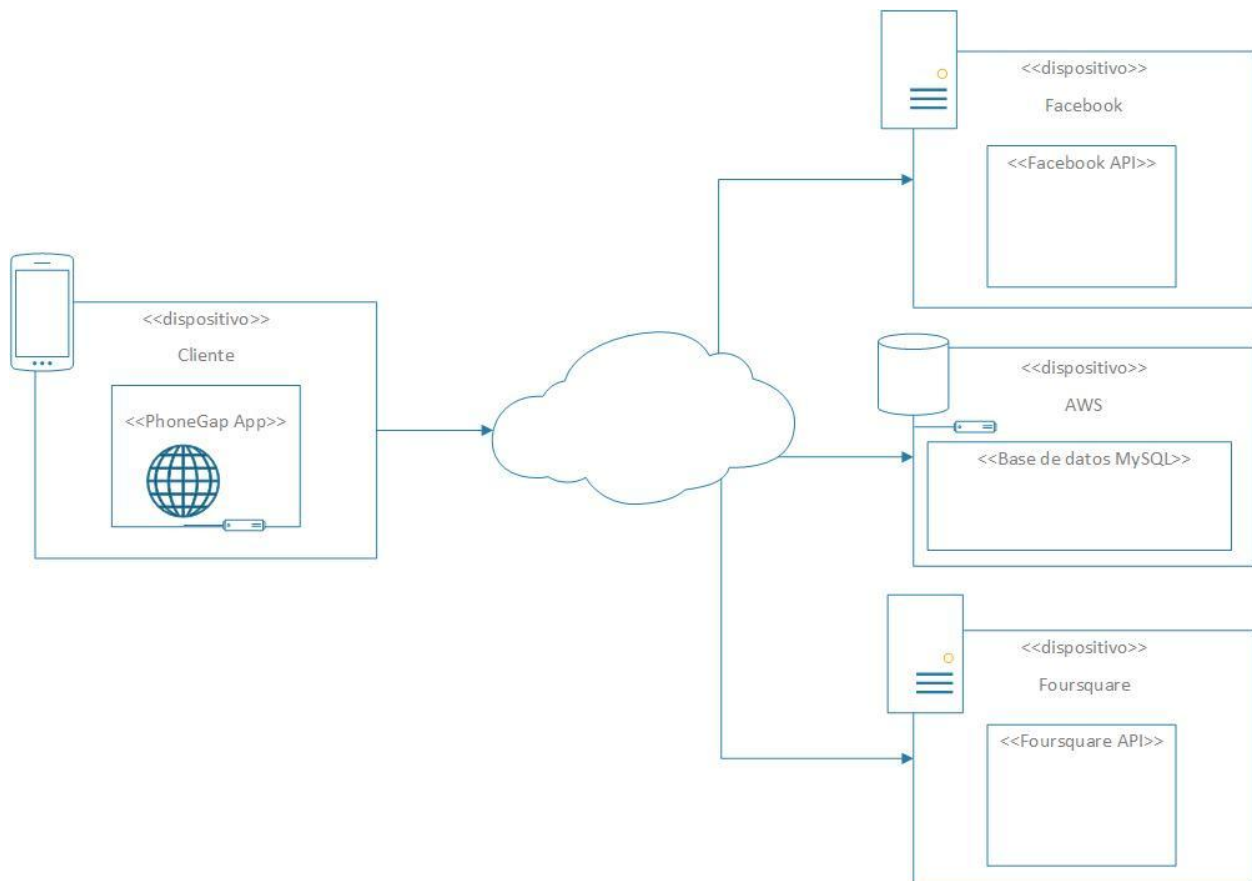
6 DESCRIPCIÓN DE DISEÑO DE ALTO NIVEL

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.

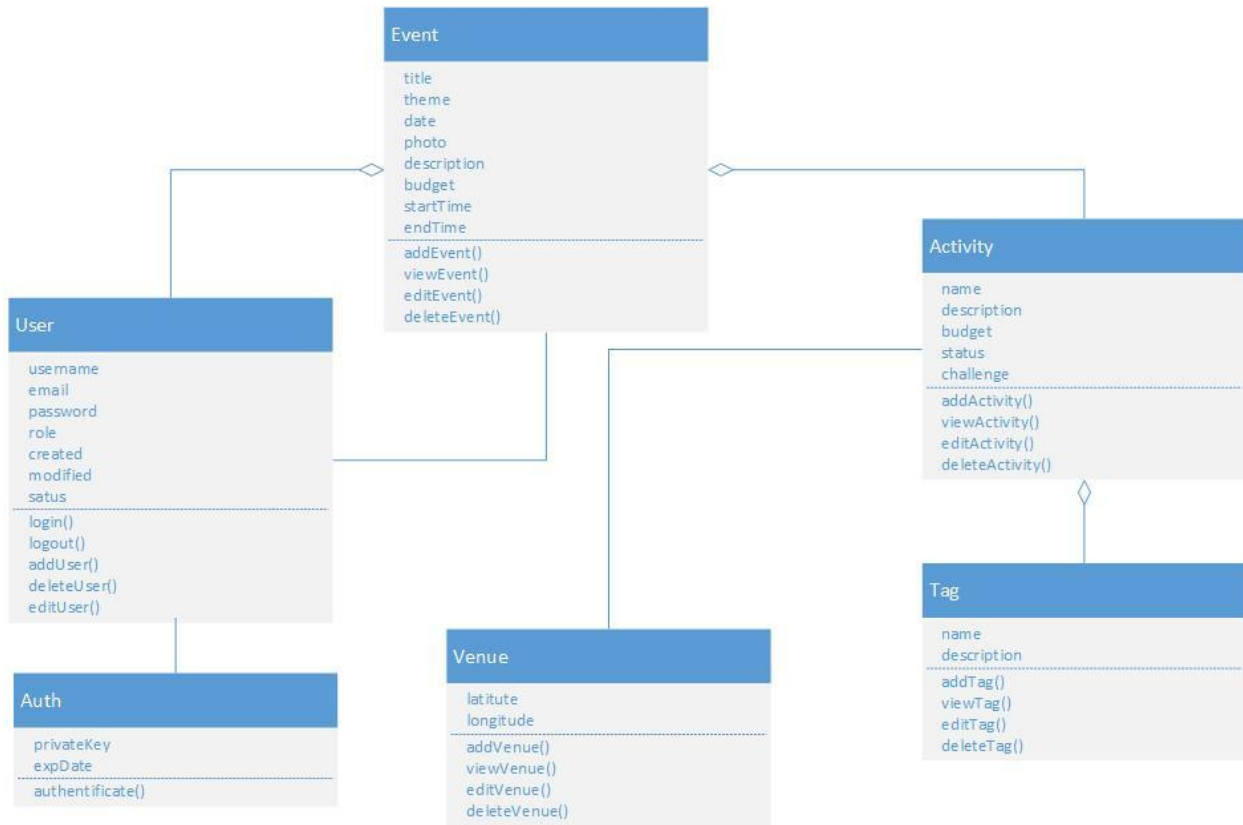


6.1 DIAGRAMA DE DESPLIEGUE

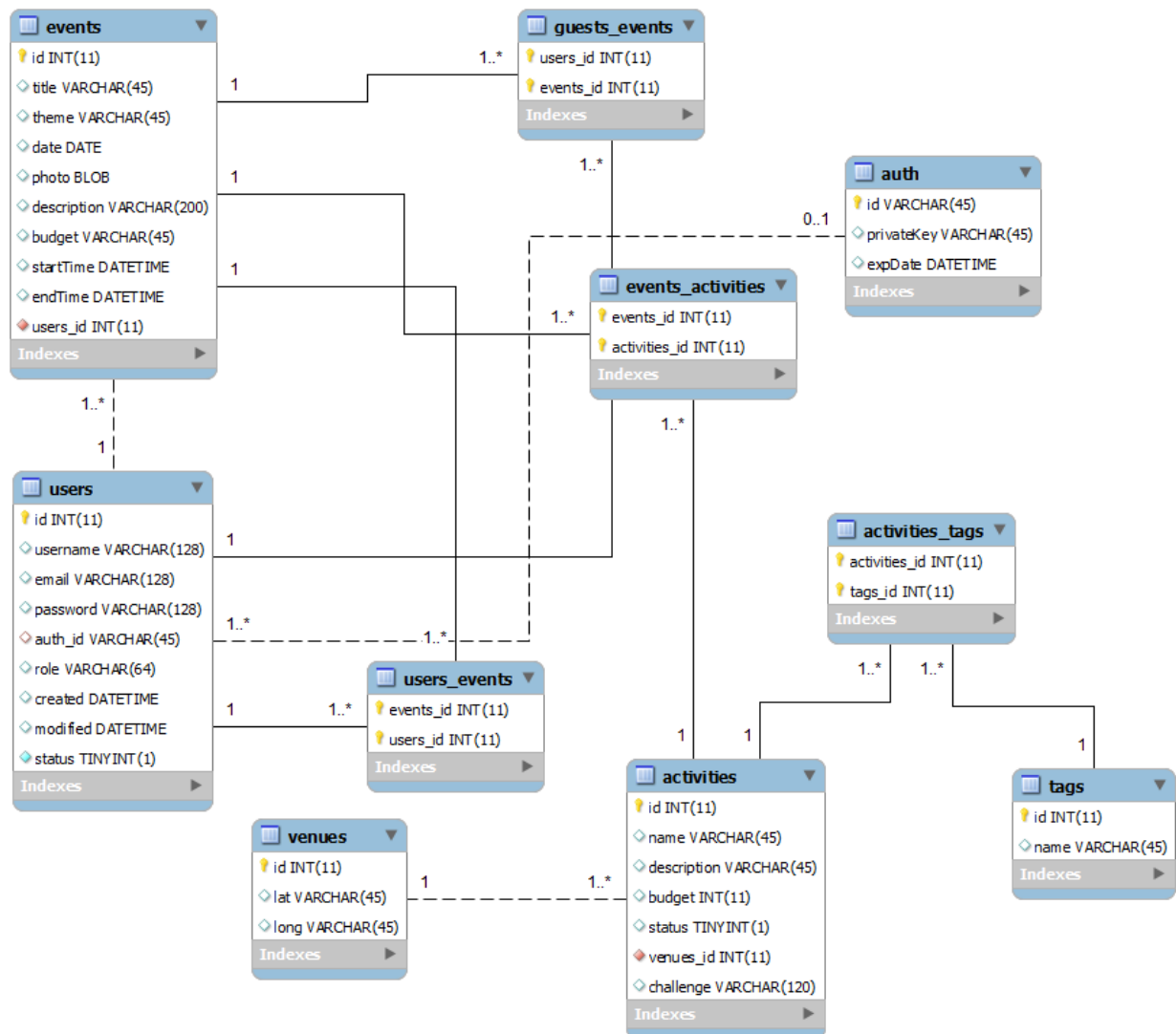
Este diagrama representa los dispositivos involucrados en el funcionamiento del sistema, así como los ambientes de ejecución para nuestra aplicación. El dispositivo móvil interactúa con el servidor que contiene la base datos hospedada en Amazon AWS por medio de la red pública. Dicho dispositivo también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



6.2 DIAGRAMA DE CLASES



6.3 DIAGRAMA DE BASE DE DATOS



7 DESCRIPCIÓN DETALLADA

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

7.1 FRONTEND

7.1.1 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)

7.1.2 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)

7.2 BACKEND

7.2.1 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.

7.2.2 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.

7.2.3 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 feeeeel 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"}]
```

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 feeeeel 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}]]

7.3 PATRONES DE DISEÑO

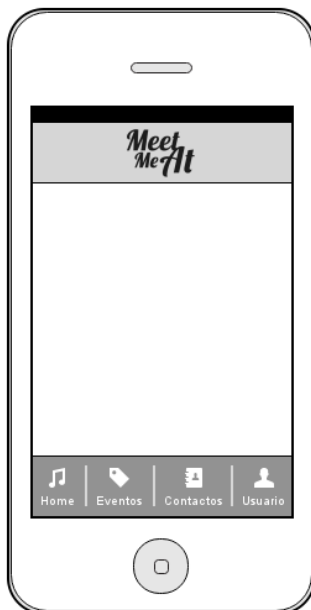
7.3.1 Patrón de navegación

7.3.1.1 *Springboard*



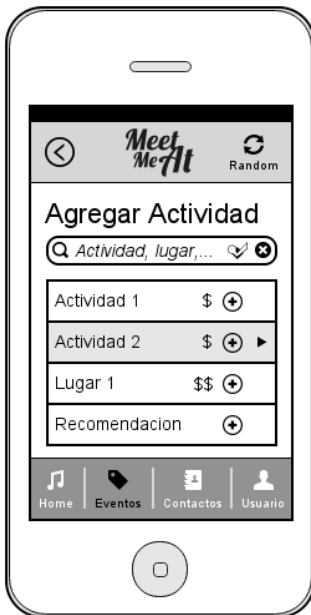
Vista Inicial

7.3.1.2 *Tabs*



Master

7.3.1.3 Listview



Actividad

7.3.2 Patrón de navegación secuandaria

7.3.2.1 Master-detail



Eventos

7.3.3 Patrón de formularios

7.3.3.1 Formularios



Eventos

7.3.4 Patrón de búsquedas

7.3.4.1 Búsqueda explícita



Contactos

8 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.

8.1 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.

8.1.1 Método de login

(descripción, url, params, método http)

8.2 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.

8.2.1 Método de obtención de coordenadas geográficas

(descripción, url, params, método http)