



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**

“APLICACIÓN MÓVIL UTILIZANDO LA TECNOLOGÍA ANDROID  
PARA OPTIMIZAR TIEMPO DE BÚSQUEDA DE LUGARES QUE  
VISITAR O EVENTOS QUE REALIZAR EN GUAYAQUIL”

**INFORME DE PROYECTO DE GRADUACIÓN**

**Previa a la obtención del Título de:**

**INGENIERO EN CIENCIAS COMPUTACIONALES  
ESPECIALIZACIÓN SISTEMAS INFORMACIÓN**

**Presentada por:**

**BLANCA ELENA GALLEGOS ZURITA  
MARIUXI NARCISA MORÁN CHANCAY**

**Guayaquil - Ecuador**

**2013**

# AGRADECIMIENTO

*A Dios, que nos ha conservado con vida y salud.*

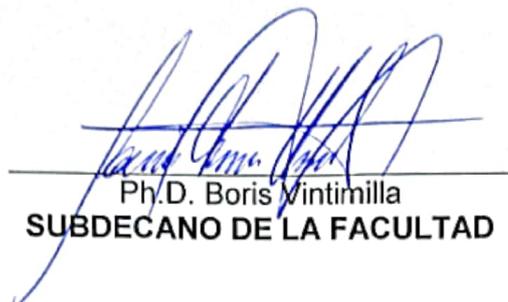
*A nuestros padres, quienes han sido y son un  
pilar fundamental en nuestras vidas.*

*A nuestros profesores, por todos los conocimientos  
y consejos transmitidos*

# DEDICATORIA

*A Dios  
A nuestros padres  
A nuestros familiares  
A nuestros amigos  
A nuestros profesores*

# TRIBUNAL DE SUSTENTACIÓN



---

Ph.D. Boris Vintimilla  
**SUBDECANO DE LA FACULTAD**



---

Dr. Xavier Ochoa Ch.  
**DIRECTOR DE PROYECTO DE  
GRADUACION**



---

Dra. Katherine Chiluzia  
**PROFESORA DELEGADA POR  
LA UNIDAD ACADEMICA**

# DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral"

(Reglamento de exámenes y títulos profesionales de la ESPOL)



Blanca Elena Gallegos Zurita



Mariuxi Narcisa Morán Chancay

# RESUMEN

Este proyecto tiene como propósito brindar a las personas que viven en Guayaquil y a sus visitantes la información debida de los sitios turísticos o eventos que se dan en la ciudad a través de una aplicación móvil, para que puedan conocer y disfrutar de todos los atractivos que ofrece la ciudad.

Además esta aplicación resulta importante para las personas extranjeras, así podrán conocer fácilmente nuestra ciudad.

El documento está dividido en 4 capítulos que comprenden el planteamiento del problema, el análisis de la solución, el diseño de la solución, implementación de la solución con sus respectivas pruebas y posteriormente las conclusiones y recomendaciones.

En el capítulo 1 se expone el planteamiento del problema que el presente proyecto de grado busca solucionar, las plataformas tecnológicas que se van a usar incluyendo los fundamentos teóricos de las mismas.

En el capítulo 2 se establece el análisis de la solución, requerimientos y descripción de la aplicación, los objetivos generales y específicos, y el alcance propuesto para este proyecto.

En el capítulo 3 se detalla el diseño de la aplicación, es decir la arquitectura que se aplica para el desarrollo del sistema; y el diseño de la estrategia que se va a manejar para escoger la mejor solución.

En el capítulo 4 se explica la implementación de la extracción de datos de la web así como el uso del OpenStreetMap en la aplicación Android para la ubicación de los lugares en el mapa, se realizan las pruebas de mediciones que consiste en evaluar los objetivos, el tiempo de consulta y optimización de cada búsqueda que se realice de un evento en concreto.

Finalmente se presentan las conclusiones obtenidas y se plantean recomendaciones para futuros trabajos relacionados.

# ÍNDICE GENERAL

RESUMEN .....	VI
ÍNDICE GENERAL.....	VIII
ÍNDICE DE FIGURAS .....	XI
ÍNDICE DE TABLAS .....	XIV
ABREVIATURAS .....	XV
INTRODUCCIÓN.....	XVI
CAPÍTULO 1 .....	1
PLANTEAMIENTO DEL PROBLEMA Y MARCO TEÓRICO .....	1
1.1. Definición del problema .....	1
1.2. Visión general de la solución propuesta .....	2
1.3. ¿Qué otras soluciones existen?.....	2
1.4. Justificaciones .....	6
CAPÍTULO 2.....	7
ANÁLISIS DE LA SOLUCIÓN .....	7
1.5. Requerimientos de la aplicación web y móvil .....	7
1.5.1. Requerimientos de la aplicación web .....	7
1.5.2. Requerimientos de la aplicación móvil .....	8

1.6.	Descripción de la aplicación móvil .....	9
1.7.	Objetivos del proyecto .....	10
1.7.1.	Objetivo general .....	10
1.7.2.	Objetivos específicos.....	11
1.8.	Alcance.....	11
1.9.	Viabilidad económica de la solución .....	13
CAPÍTULO 3.....		15
DISEÑO DE LA APLICACIÓN WEB Y MÓVIL.....		15
1.10.	Componentes de la aplicación.....	15
1.11.	Arquitectura de la aplicación.....	20
1.11.1.	Diseño lógico .....	20
1.11.2.	Diseño físico .....	23
1.12.	Diseño de estrategia.....	25
1.12.1.	Acceso a la información .....	25
1.12.2.	Almacenamiento de los datos.....	26
1.12.3.	Presentación de los datos .....	27
1.12.4.	Administración de la información en la aplicación web.....	28
CAPÍTULO 4.....		30
IMPLEMENTACIÓN Y PRUEBAS .....		30

1.13. Java Web Scraping.....	30
1.14. Implementación del OpenStreetMap en Android .....	45
1.15. Descripción del plan de pruebas.....	48
1.16. Tiempo de consulta y optimización.....	49
1.17. Evaluación de los objetivos .....	52
1.18. Estadísticas producidas por el sistema.....	52
CONCLUSIONES Y RECOMENDACIONES .....	55
CONCLUSIONES .....	55
RECOMENDACIONES.....	56
APÉNDICE.....	57
BIBLIOGRAFÍA.....	64

# ÍNDICE DE FIGURAS

Figura 1.1 Captura de pantalla de la aplicación AroundMe.....	3
Figura 2.1 Captura de pantalla de la aplicación Goby .....	4
Figura 3.1 Captura de pantalla de la aplicación Yelp.....	5
Figura 4.1 Captura de pantalla de la aplicación MyCityWay.....	6
Figura 1.3 Captura de pantalla del sitio turístico Palacio de Cristal .....	16
Figura 2.3 Captura de pantalla de los eventos publicados en el sitio web de El Universo – Arte y Cultura.....	16
Figura 3.3 Captura de pantalla de la aplicación web - Ingresar Evento .....	18
Figura 4.3 Captura de pantalla de la aplicación web - Ingresar Evento .....	19
Figura 5.3 Captura de pantalla de la aplicación web - Ingresar Evento .....	19
Figura 6.3 Diseño lógico .....	20
Figura 7.3 Modelo lógico aplicación móvil.....	22
Figura 8.3 Diseño lógico aplicación web .....	23
Figura 9.3 Modelo físico aplicación móvil.....	24
Figura 10.3 Modelo físico aplicación web .....	25
Figura 1.4 Java Web Scraping.....	30
Figura 2.4 Captura de pantalla de la aplicación del sitio web de El Universo – Todos los titulares.....	32
Figura 3.4 Código del scrap al sitio web El Universo – Todos los titulares ...	33

Figura 4.4 Código conexión a un sitio web .....	33
Figura 5.4 Código html del sitio web de El Universo - Arte y Cultura.....	34
Figura 6.4 Código análisis de tags del documento .....	34
Figura 7.4 Información de los eventos extraídos del sitio web de El Universo – Arte y Cultura.....	35
Figura 8.4 Código para extraer datos en cada uno de los eventos de Arte y Cultura .....	36
Figura 9.4 Información de cada evento de Arte y Cultura.....	37
Figura 10.4 Captura de pantalla de los eventos extraídos del sitio web de El Universo – Arte y Cultura.....	38
Figura 11.4 Captura de Pantalla de la aplicación móvil, mientras realiza la carga de información .....	39
Figura 12.4 Código que nos permite analizar el XML .....	40
Figura 13.4 Código para analizar los nodos y realizar la carga de información .....	40
Figura 14.4 Código para mostrar las imágenes asincrónicamente .....	41
Figura 15.4 Captura de pantalla donde se observan los eventos en un listview .....	42
Figura 16.4 Captura de pantalla donde se describe el evento .....	43
Figura 17.4 Captura de pantalla donde se comparten eventos en facebook	44
Figura 18.4 Captura de pantalla de un evento publicado en Facebook .....	44
Figura 19.4 Código del layout en donde se presentará el mapa.....	47

Figura 20.4 Código de la clase Localizacion.java .....	47
Figura 21.4 Captura de pantalla de la ubicación del Museo de Bomberos en OpenStreetMap .....	48
Figura 22.4 Diagrama de barras de los tiempos en que se carga la información de los sitios web, los datos son tomados de la Tabla 1.....	53
Figura 23.4 Gráfico circular en el que se representa el tiempo de carga promedio, datos tomados de la Tabla 1.....	53
Figura 24.4 Diagrama de barras de los tiempos en que se carga la información desde el rss, los datos son tomados de la Tabla 1.....	54
Figura 25.4 Gráfico circular en el que se representa el tiempo de carga promedio, datos tomados de la Tabla 1.....	54

# ÍNDICE DE TABLAS

TABLA 1.4 TIEMPO PROMEDIO DE CARGA DE DATOS – EL UNIVERSO VS SUITES GYE .....	50
TABLA 2.4 TIEMPO PROMEDIO DE CARGA DE DATOS – SITIO WEB VS BD .....	51

# ABREVIATURAS

**HTML**      HyperText Markup Language

**RSS**        Rich Site Summary

**URL**        Uniform Resource Locator

**XML**        Extensible Markup Language

# INTRODUCCIÓN

La tecnología móvil ha atravesado importantes innovaciones en estos últimos tres años, con nuevas plataformas, nuevos servicios y nuevas tendencias de uso. Todo parece apuntar al establecimiento de las tecnologías móviles como elemento fundamental en el uso de Internet.

El número de usuarios de Smartphone está en crecimiento, los sistemas operativos móviles están proliferando y las aplicaciones se han convertido en el centro de la actividad web móvil, tanto es así que muchas empresas han implementado dichos sistemas, lo cual ha generado un campo competitivo en un medio tecnificado por las nuevas tendencias.

Pero hay que tener en cuenta de que no por el hecho de ser una nueva tecnología, esto signifique que para el usuario le sea fácil de usar. Lo que hace un sistema operativo fácil de usar, es el número y la calidad de las aplicaciones disponibles para ese sistema. Si un sistema operativo es muy bueno, pero no tiene aplicaciones, los usuarios no lo van a usar.

Esto es precisamente uno de los puntos fuertes de Android, el crecimiento en cuanto al número y calidad de aplicaciones disponibles.

Pensando en la optimización del tiempo de búsqueda en la web, que una persona en Guayaquil, sea local o extranjera pueda realizar, se ha decidido

desarrollar una aplicación que realice la misma búsqueda de una forma más eficiente, utilizando web scraping (también llamado web harvesting o web data extraction) los usuarios podrán acceder a contenidos publicados en diferentes sitios web correspondientes con lo que está buscando.

A más de eso la tecnología móvil es una forma de acceso mucho más rápida a internet, lo que la hace eficiente e interesante al usar aplicaciones que utilicen este tipo de comunicación.

El primer capítulo describe la problemática en cuanto a la extracción de la información y las plataformas tecnológicas a usarse, esto va hacer el ápice para establecer un análisis de la solución y la descripción de la aplicación obtenida en el segundo capítulo, para llegar al planteamiento de los objetivos generales y específicos, y luego al alcance de los mismos. Con lo propuesto en el capítulo anterior, en el siguiente se procede a bosquejar un diseño de solución y la arquitectura del sistema móvil, que permita validar la mejor estrategia para la correcta implementación del mismo, lo cual va a dar como resultado la investigación contenida en el último capítulo donde se explica cómo fue implementado el sistema, y finalmente las conclusiones y recomendaciones obtenidas darán origen a nuevos modelos que irán perfeccionando el desarrollo de los aplicativos móviles.

# CAPÍTULO 1

## 1. PLANTEAMIENTO DEL PROBLEMA Y MARCO TEÓRICO

### 1.1. Definición del problema

Dado a que las personas que viven o que visitan Guayaquil, no tienen acceso a toda la información de los sitios turísticos o eventos que se dan en la ciudad en una sola fuente o que esta información se la adquiere de diferentes medios como periódicos, revistas, televisión, etc., muchos no pueden participar de ciertos eventos. Además, la información en muchos casos no es muy directa o no muy explicativa para lo que el usuario desea saber, a veces podemos encontrar errores en la información que no brindan al usuario una mayor confiabilidad de la información que busca.

Por otro lado, uno de los problemas más relevantes es el tiempo requerido para realizar dichas búsquedas en los diferentes medios antes mencionados, siendo esto poco eficiente para el usuario.

## 1.2. Visión general de la solución propuesta

Se ha pensado en una aplicación móvil que brinde a la comunidad guayaquileña y a los visitantes la información debida de una manera fácil, exacta y rápida, acorde a lo que ofrece la aplicación. Paralelamente se desarrollara un sitio web aliado, al cual llamaremos web **GeYE** el cual será alimentado por medio de varias fuentes de datos para llegar a la información correcta, sin dejar que el usuario consulte ningún otro medio externo, brindándole al usuario confiabilidad en su búsqueda y con el objeto de que puedan conocer y disfrutar de todos los sitios turísticos y eventos que se agenda en la ciudad.

Su mayor utilidad está dirigida a personas extranjeras o visitantes, pues les ayuda a elaborar un itinerario de actividades que deseen realizar durante su estadía en Guayaquil.

## 1.3. ¿Qué otras soluciones existen?

Actualmente existen aplicaciones móviles que nos ayudan a encontrar lugares turísticos, por ejemplo tenemos:

- **AroundMe**: Le permite encontrar rápidamente la información sobre su entorno; es decir, identifica rápidamente su posición y le permite elegir el banco, bar, estación de servicio,

hospital, hotel, película, teatro, restaurante, supermercado y taxi más cercanos.

La aplicación tiene como autores a los siguientes, Marco Pifferi, Edoardo Luciani, Duncan Wilcox, Annarita Enrico. En el siguiente link encontraremos mucha más información de AroundMe. <http://www.aroundmeapp.com/>

En la Figura 1.1 se presenta la captura de pantalla de la aplicación AroundMe.



Figura 1.1 Captura de pantalla de la aplicación AroundMe

- **Goby:** Le ayuda a encontrar cosas interesantes y creativas que puede hacer en su tiempo libre.

Desarrolladores de Goby, Mark Watkins, Vince Russo, Michael Stonebraker, Mujde Pamuk. En el siguiente link

encontraremos información importante de Goby

<http://www.goby.com/page/mobile>

En la Figura 2.1 se presenta la captura de pantalla de la aplicación Goby.



Figura 2.1 Captura de pantalla de la aplicación Goby

- **Yelp:** Busca lugares para comer, beber, comprar y divertirte; y a parte de todo esto, lee los comentarios de una activa comunidad local que lo sabe todo.

La presente aplicación fue desarrollada por el siguiente grupo de trabajo **yelpTeam**, para obtener más información de la aplicación, ponemos a consideración la siguiente dirección web <http://www.yelp.com/yelpmobile> .

En la Figura 3.1 se presenta la captura de pantalla de la aplicación Yelp.



Figura 3.1 Captura de pantalla de la aplicación Yelp

**MyCityWay:** Es un portal de descubrimiento interactivo. Revoluciona la forma de vivir, trabajar, jugar y visitar cada ciudad. Se puede tener información acerca de restaurantes, menús, tours, etc. Útil para vecinos y visitantes. Fue implementado por, Puneet Mehta, Sonpreet Bhatia, Archana Patchirajan, Steve Rowe y su sitio web es el siguiente; <http://www.mycityway.com/> .

En la Figura 4.1 se presenta la captura de pantalla de la aplicación MyCityWay.



Figura 4.1 Captura de pantalla de la aplicación MyCityWay

## 1.4. Justificaciones

Si bien es cierto, este tipo de aplicaciones existen en otros lugares, tienen ciertas deficiencias como: información no actualizada, retardo en las consultas y restricciones en el acceso a la información. En nuestro país no contamos con este tipo de aplicaciones, por lo tanto será de mucha ayuda para la ciudadanía en el aspecto social, cultural y lo más importante en la optimización de su tiempo de búsqueda, por lo que la aplicación busca ser la solución a dichos problemas.

# CAPÍTULO 2

## 2. ANÁLISIS DE LA SOLUCIÓN

El objetivo del estudio de este capítulo es conocer y comprender ampliamente el funcionamiento del sistema y su correcto desarrollo económicamente hablando.

### 2.1. Requerimientos de la aplicación web y móvil

#### 2.1.1. Requerimientos de la aplicación web

La aplicación web **GeYE** deberá contar con una serie de requisitos que le permitan servir como una de las fuentes de información para la aplicación móvil.

Los parámetros de ingresos de datos deben ser reales y exactos de tal forma que el requerimiento sea atendido con mayor facilidad.

Esta aplicación web **GeYE** deberá usar un hosting externo para el correspondiente acceso a ella desde la web.

Si alguna persona está interesada en que su evento aparezca entre las búsquedas que el usuario realice, tendrá un espacio en el cual se necesitará de algunos parámetros de registro del evento y luego solicitar el ingreso del mismo en el sistema, de tal forma que aparezca en su perfil.

Requerirá de un espacio publicitario que será promovido por los restaurantes y cafeterías que quisieran ser parte de la aplicación web **GeYE** y posteriormente de la aplicación móvil, que tendrá un costo dependiendo del ranking que desee dicha empresa.

De esta forma la aplicación web **GeYE** se convierte en nuestra fuente de información más directa de los eventos a realizarse en la ciudad.

### **2.1.2. Requerimientos de la aplicación móvil**

Si hablamos de una aplicación móvil como tal, lo primero que podemos imaginarnos es el acceso a internet que necesita para funcionar y no solo eso también en las limitaciones del hardware en el cual será instalada la aplicación.

En lo que tiene que ver con el hardware nos enfocamos en el procesador, debido a que es una aplicación que va a generar extracción de gran cantidad de datos a través de un parseador RSS, por lo cual necesita un procesador de mínimo 832MHz de capacidad.

## 2.2. Descripción de la aplicación móvil

Teniendo en cuenta los argumentos presentados en el Capítulo 1, acerca del problema que representa para los usuarios no tener a primera mano la información correspondiente a los sitios turísticos o eventos a dónde asistir en la ciudad en sus ratos libres, se plantea la aplicación móvil utilizando la tecnología Android para optimizar los tiempos de búsqueda de lugares que visitar o eventos que realizar en Guayaquil, también la aplicación pretende incentivar el turismo haciéndolo de una manera fácil y rápida.

La aplicación móvil contará con tres componentes, los cuales serán:

- **Atracciones:** Proveerá la mayor cantidad de atracciones turísticas que tiene la ciudad con un amplio contenido de información que de una descripción clara y visual de la misma.

- **Eventos:** la aplicación extraerá desde la web **GeYE** información de los eventos que se van a realizar hoy y semanalmente en la ciudad.
- **Restaurantes y Cafeterías:** Permitirá a los ciudadanos conocer y disfrutar de los mejores restaurantes y cafeterías de la ciudad.

Para acceder a los diferentes sitios turísticos de Guayaquil, conocer y disfrutar de lo que se realice dentro de la ciudad dependerá de lo que el usuario desee hacer. Usando la aplicación móvil podrá observar una descripción del sitio o evento, un mapa de ubicación además tendrá la opción de compartirlo en redes sociales para difundirlo.

## **2.3. Objetivos del proyecto**

### **2.3.1. Objetivo general**

El presente trabajo tiene como objetivo general aprovechar la información disponible en diferentes sitios web dedicados al turismo, actividades, eventos realizados en la ciudad de Guayaquil, a fin de que los ciudadanos o turistas puedan consultar toda esta información en una sola aplicación.

### 2.3.2. Objetivos específicos

Se han planteado los siguientes objetivos:

- **En la aplicación móvil** el usuario podrá acceder a información detallada de los sitios turísticos y eventos que se den en Guayaquil.

Presentar un mapa de ubicación de los sitios y eventos que se realice en la ciudad.

Compartir el evento de interés en las redes sociales.

- **En la aplicación web** el usuario podrá crear eventos y espacios publicitarios, permitiendo realizar la consulta de los mismos.

Módulo administrador que se encarga de parametrizar la información de los sitios y eventos en un formato rss.

## 2.4. Alcance

La profundidad del proyecto es básicamente proporcionar al usuario un ambiente informativo e interactivo de realizar actividades en la ciudad, además de compartir dichas actividades en las diferentes redes sociales, por medio de su aplicación móvil.

Con la aplicación móvil lograremos realizar las siguientes tareas:

- Realizar una actualización automática al final del mes de la información de los sitios turísticos de la ciudad, este mecanismo de actualización de datos realizará la acción de actualizar los sitios escogidos como turísticos en la base de datos de la aplicación móvil.
- Los sitios turísticos serán presentados directamente desde la base de datos de la aplicación móvil.
- Los eventos se presentarán directamente desde el contenido rss de la aplicación web **GeYE** y estos se actualizarán en la base de datos de la aplicación móvil cada vez que el usuario consulte sus eventos, el proceso localiza el evento nuevo y lo guarda. Esta actualización se realiza con el fin de obtener la base de datos del teléfono correctamente actualizada.
- Presentar la ubicación de los sitios turísticos y/o eventos en un mapa.
- Debido a los procesos de actualización, la base de datos de la aplicación móvil estará correctamente actualizada lo que permitirá que la aplicación funcione con o sin acceso a Internet.

- Restricciones físicas de la plataforma y hardware: la aplicación funciona correctamente en la versión de android 2.3.5 GINGERBREAD con un api **minSdkVersion** de 10 y sus versiones posteriores.

Con la aplicación web lograremos realizar las siguientes tareas:

- Permitir la creación de eventos personalizados por medio de un formulario, los mismos que después se verán reflejados en la aplicación móvil.
- Creación de un archivo RSS para manejar los eventos personalizados.
- Creación del espacio publicitario por medio de un formulario que permita el ingreso de los datos, según las conveniencias del usuario.
- Creación de un archivo rss para el manejo de la publicidad.
- Editar los eventos obtenidos desde la web.
- Editar los sitios turísticos.

## **2.5. Viabilidad económica de la solución**

La mayoría de las aplicaciones para solventarse económicamente tienden a vender el aplicativo, ya sea el caso de Android Market o

simplemente con una buena publicidad en su página web lo pueden lograr, siempre que la aplicación sea de interés para el usuario.

Nuestro proyecto proporcionará una versión libre y otra completa que tendrá un costo, la versión libre contendrá la parte turística de la ciudad, y en la versión completa se agregará el módulo eventos y el espacio publicitario, luego las actualizaciones que se realicen estarán disponibles para que el usuario invierta en ellas.

En el espacio publicitario los usuarios podrán comprar una publicidad de acuerdo a un valor de inversión que determinará la prioridad de la misma, dichas publicidades serán vistas en la aplicación móvil.

Esto permitirá generar dinero el cual servirá para solventar los servicios utilizados y mantener un buen nivel de los mismos con actualizaciones y nuevas mejoras de los aplicativos.

# CAPÍTULO 3

## 3. DISEÑO DE LA APLICACIÓN WEB Y MÓVIL

### 3.1. Componentes de la aplicación

La aplicación móvil se compone en los siguientes tres módulos:

- **Módulo Atracciones**: El cual mostrará los diferentes sitios turísticos que conocer en Guayaquil, dichos lugares son alimentados por un lector RSS situado en la web **GeYE**. Por medio de la extracción de esta información y el correcto procesamiento de la misma se genera la presentación de dichos sitios.

En la Figura 1.3 se presenta la captura de pantalla del sitio turístico Palacio de Cristal – <http://www.visitaguayaquil.com>



Figura 1.3 Captura de pantalla del sitio turístico Palacio de Cristal

- **Módulo Eventos:** Muestra los eventos en dos diferentes formas, la primera me dice cuáles son los eventos y/o actividades que hoy están sucediendo y el segundo son los eventos que se van a realizar en la semana.

En la Figura 2.3 se presenta la captura de pantalla de los eventos publicados en el sitio web de El Universo – [http://www.eluniverso.com/arte\\_y\\_cultura](http://www.eluniverso.com/arte_y_cultura)



Figura 2.3 Captura de pantalla de los eventos publicados en el sitio web de El Universo – Arte y Cultura

- **Módulo Restaurantes y Cafeterías:** Genera un espacio donde el usuario podrá ver los lugares de la gastronomía de Guayaquil, el usuario observara primero la publicidades que esten mejor rankeadas por el sistema, en base a ciertos acuerdos generados en la página web **GeYE** .

En la página web tenemos los siguientes módulos:

- **Módulo Eventos:** Si el usuario es visitante podrá publicar eventos, esto lo llevará a cabo llenando un formulario que le permita el ingreso de dichos eventos.

El formulario permitirá el ingreso de los siguientes campos: Nombre del evento, Categoría del evento (el cual va a tener que escoger entre 6 categorías: Arte y cultura, Diversión Familiar, Música, Viajes, Deportes, Cafeterías y Restaurantes), Descripción, Lugar (las categorías tienen lugares asociados, de los cuales tenemos todos los datos), si el lugar del evento que el usuario está creando no está entre las categorías, se procede a ingresar un nombre del lugar, una dirección y una referencia del lugar mediante coordenadas proporcionadas por google maps, de acuerdo a la marca señalada por el usuario en el mapa, Fecha, Hora, Imagen (jpg, png) y el Ranking (que tendrán valores del 1-10), el mismo que se escoge y valorará la posición del evento.

En la Figura 3.3 se presenta la captura de pantalla de la aplicación web en el ingreso del evento.

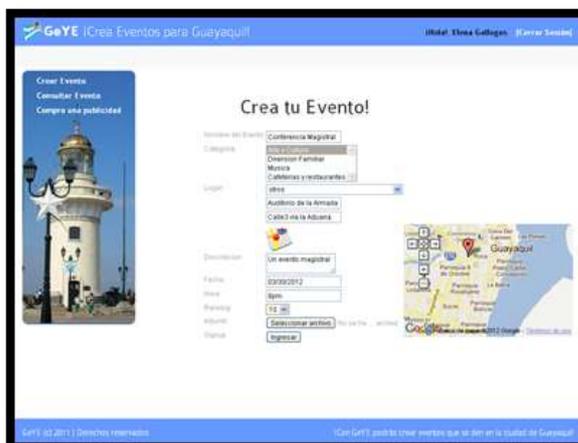


Figura 3.3 Captura de pantalla de la aplicación web - Ingresar Evento

- **Módulo Publicitario:** Proporciona un ambiente apropiado al usuario ya que por medio del formulario puede ingresar su publicidad más personalizada, por medio de los siguientes parámetros: nombre del local, descripción del local, dirección del local, coordenadas, ranking, y una imagen correspondiente al local a registrarse.

En la Figura 4.3 se presenta la captura de pantalla de la aplicación web en el ingreso de la publicidad.



Figura 4.3 Captura de pantalla de la aplicación web - Ingresar Evento

En la Figura 5.3 se presenta la captura de pantalla de la aplicación web.



Figura 5.3 Captura de pantalla de la aplicación web - Ingresar Evento

## 3.2. Arquitectura de la aplicación

### 3.2.1. Diseño lógico

Para un mejor entendimiento de los módulos que conforman la arquitectura lógica, presentamos el siguiente diseño.

En la Figura 6.3 se presenta el diseño lógico de la aplicación.

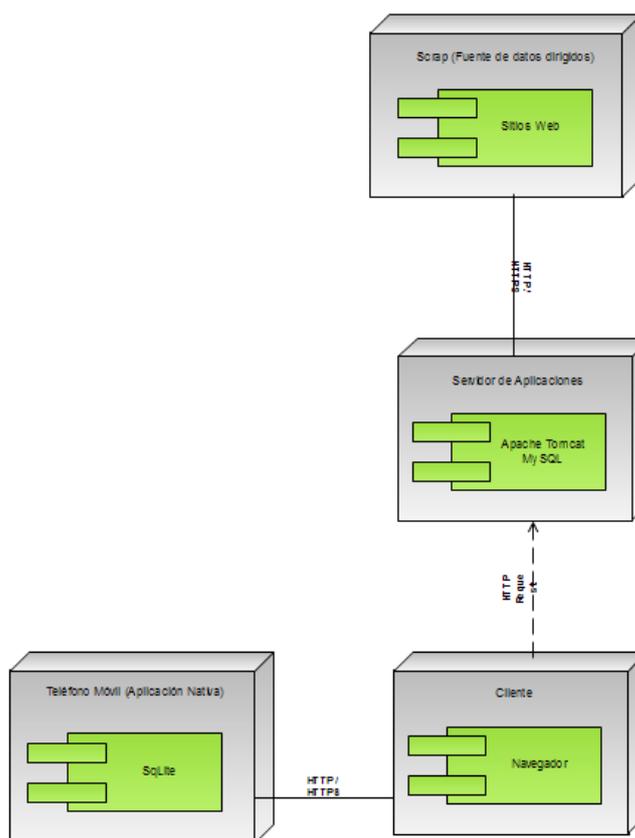


Figura 6.3 Diseño lógico

En la figura observamos lo siguiente:

- **Teléfono Móvil:** El dispositivo móvil se conecta vía HTTPS hacia el sitio web GeYE con el propósito de alimentarse de un archivo rss generado por medio del raspado de datos hecho a los sitios web escogidos para la obtención de la información.
- **Scrap/Fuente de datos dirigidos:** Es en la página web **GeYE** donde se realiza el scrap de la información la misma que estará alojada a un servidor de aplicaciones que permite realizar dicho raspado.
- **Servidor de Aplicaciones:** Estará basado en el software de servidor de aplicaciones Tomcat y sobre él se desplegará nuestra aplicación web **GeYE**. Aquí también estará nuestra base de datos MySQL. Este servidor enviará peticiones (request) para obtener la información de las páginas escogidas para realizar la correspondiente obtención de la información.

También presentamos los modelos de la base de datos:

- **Modelo Lógico Aplicación Móvil:** El presente modelo lógico describe los elementos principales de la aplicación móvil y como estos interactúan para lograr los objetivos propuestos. Nos permite desarrollar una visión general de la aplicación y crear un plan de acción que explique cómo funciona el aplicativo.

En la Figura 7.3 se presenta el modelo lógico de la aplicación móvil.

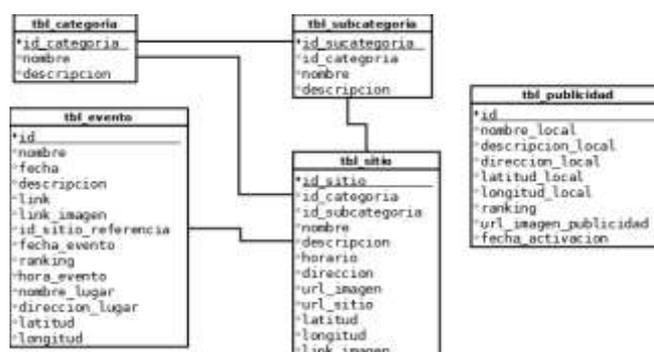


Figura 7.3 Modelo lógico aplicación móvil

El modelo propuesto tiene como entidad principal al sitio, debido a que la relación de un sitio con las demás entidades es de uno a muchos, esto permite en la aplicación móvil tener una referencia de ubicación dado el caso que fuere, evento, publicidad o en el caso del sitio propiamente dicho.

- **Modelo Lógico Aplicación Web:** En general los modelos identifican los elementos o entidades que intervienen en el desarrollo de una aplicación, en el presente modelo nos permite organizar la idea general y planificar estrategias de desarrollo en la aplicación.

En la Figura 8.3 se presenta el modelo lógico de la aplicación web.

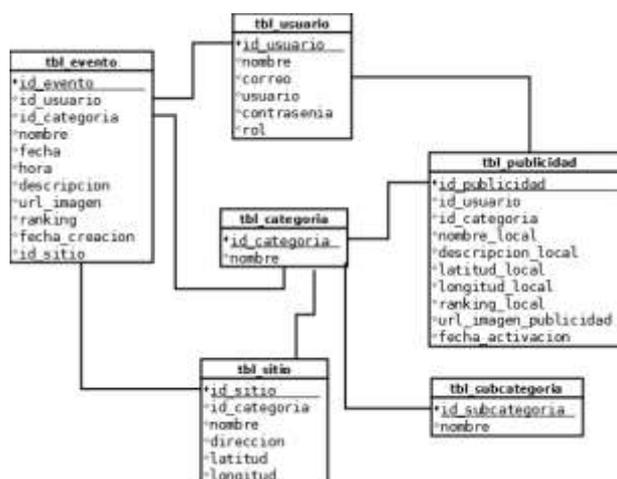


Figura 8.3 Diseño lógico aplicación web

### 3.2.2. Diseño físico

Para el mejor entendimiento global del funcionamiento de la aplicación, planteamos los siguientes diseños:

- **Diseño Físico Aplicación Móvil:** como se muestra en la figura, el usuario accede a la aplicación a través de un dispositivo móvil con sistema operativo Android, y éste a su vez se conecta mediante internet a los sitios web configurados al servidor de aplicaciones.

En la Figura 9.3 se presenta el modelo físico de la aplicación móvil.

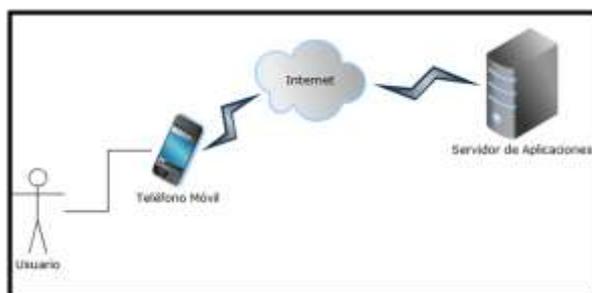


Figura 9.3 Modelo físico aplicación móvil

- **Diseño Físico Aplicación Web:** El usuario accede a su sesión de la aplicación web **GeYE**, y podrá crear eventos, publicidad.

Un administrador también podrá acceder a la aplicación, éste podrá extraer información de las páginas web de El Universo y suitesguayaquil, a

través de operaciones de web-scraping, con el fin de que la aplicación móvil se alimente de ella por medio de un canal rss.

En la Figura 10.3 se presenta el modelo físico de la aplicación web.

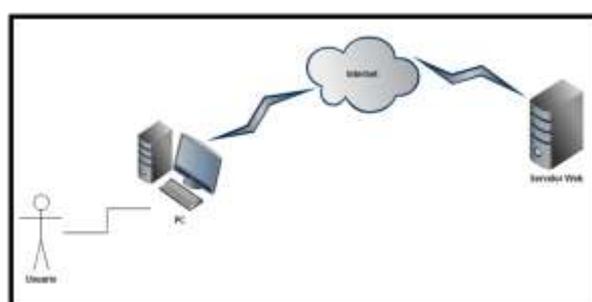


Figura 10.3 Modelo físico aplicación web

### 3.3. Diseño de estrategia

Para cada una de las aplicaciones se ha determinado seguir las siguientes estrategias:

#### 3.3.1. Acceso a la información

Utilizando una estrategia de acceso a los datos en el cual se ponen en consideración una serie de aspectos.

El aspecto clave, consiste en que el usuario pueda usar la aplicación móvil en el momento que desee, sin

necesidad de tener acceso a internet, dado que la información se registra en la base de datos de la aplicación móvil cada vez que el usuario ve sus eventos y publicidades, debido a que accede directamente a la página web **GeYE** en internet, en este escenario el usuario podrá acceder a dicha información en cualquier momento sin necesidad de estar conectado, esto hace que la aplicación móvil sea interactiva en todo momento.

La recopilación de los datos es diaria, debido a que los eventos pueden ser creados en cualquier momento en la aplicación web **GeYE** y esto permitirá tener al usuario informado todo el tiempo en la aplicación móvil. La información estará en un 100% en línea debido a que el usuario obtiene sus eventos directamente de la página web **GeYE**, solo en el caso cuando el usuario no tenga acceso a internet se le presentará los eventos registrados en la base de datos de la aplicación móvil.

### **3.3.2. Almacenamiento de los datos**

Para obtener un mejor rendimiento de la aplicación móvil se ha pensado en una estrategia de carga de información en la base de datos del dispositivo móvil. La carga de

datos se realiza por medio de lectores rss que nos proporciona la aplicación web **GeYE** a la aplicación móvil, es decir la información clave que la aplicación debe suministrar al usuario corresponde a lo que el archivo rss tanto de eventos como de publicidad contengan en el momento en que la aplicación móvil se conecta a Internet, desde un estado anterior sin conexión.

### **3.3.3. Presentación de los datos**

Se ha creado un modo de optimización de tiempo por medio del manejo de la presentación de los eventos que el usuario vea a diario en la aplicación móvil, la información es presentada en tiempo real, por medio de una clase en android que implementa una tarea que nos permite ir viendo y registrando la información mientras se cargan las imágenes, esto se da en el momento que el usuario tiene acceso a la página web **GeYE**.

Mientras que la estrategia para cuando no tenemos acceso a internet nos serviremos de la base de datos SQLite que viene incluida en la plataforma Android, debemos saber que el proceso de registrar la información

en la base de datos, solo realiza el registro de texto como nombre del evento, descripción, lugar, hora, latitud, longitud, etc, y con respecto a la imagen, la aplicación móvil tomará una imagen modelo tanto para evento como publicidad, esto hará que el usuario no recargue la memoria de su teléfono y permitirá disfrutar más rápidamente de la información presentada.

#### **3.3.4. Administración de la información en la aplicación web**

La administración de la información de los eventos obtenidos en el proceso del web scraping en la página web **GeYE**, es necesario que sean correctamente administrados por un recurso humano que realice la función de depuración de los datos de tal forma que la información esté, limpia, correcta y clara para el usuario.

Esta información es verificada por el administrador del sitio todos los días en las mañanas de tal forma que el archivo rss se actualice para ser correctamente obtenido por el usuario de la aplicación móvil. Es necesaria la manipulación de la información dado a que existe información relevante como lugar, fecha y hora del

evento, que se encuentra inscrita en el texto de la información raspada y para una mejor presentación es importante tener esta información en variables manipulables, de tal forma que para el usuario este proceso sea transparente y a su vez connote una presentación única de la información.

# CAPÍTULO 4

## 4. IMPLEMENTACIÓN Y PRUEBAS

En este capítulo se detallará el desarrollo de la implementación, la cual ha sido dividida en:

### 4.1. Java Web Scraping

Web scraping es una técnica de ingeniería inversa de una página HTML en donde toma contenido de los sitios web y los transforma en otros formatos adecuados para ser usados en otros contextos.

En la Figura 1.4 se presenta el modelo de java web scraping.



Figura 1.4 Java Web Scraping

La idea es llegar a un tipo de información estructurada en un formato útil y entendible para una aplicación. Para lograr lo antes mencionado necesitamos que la información sea suministrada por

el usuario, conociendo la estructura de la web, el desarrollador provee de la información de los tags que necesitamos para la extracción de la información requerida, como título, descripción, imagen en lo cual, lo que se quiere es tomar esa información desordenada y con ayuda previa suministrada (por un humano) lograr estructurarla en un formato que nos proporciona los resultados que esperamos.

Para realizar un correcto scrap de los datos es necesario conocer primero cual es el formato que se nos presenta, dependiendo de la estructura (HTML, XML) de los sitios el scrap o raspado de datos será más o menos complicado de realizar en algunos casos no se podrá concretar dicha acción.

Para hacer el scrap a los sitios web, utilizamos una biblioteca de Java llamada Jsoup, la cual nos proporciona un API para la extracción y manipulación de datos utilizando DOM y CSS, con lo cual nos permite hacer lo siguiente:

- raspar y analizar código HTML de una URL
- encontrar y extraer los datos, utilizando recorrido DOM o selectores CSS
- manipular los elementos HTML, atributos y texto

Para el presente proyecto vamos realizar el scrap al sitio web del diario El Universo y Suites Guayaquil; específicamente a sus titulares.

En la Figura 2.4 se presenta la captura de pantalla del sitio web de El Universo – [http://www.eluniverso.com/Todos\\_los\\_titulares/](http://www.eluniverso.com/Todos_los_titulares/)



Figura 2.4 Captura de pantalla de la aplicación del sitio web de El Universo – Todos los titulares

Estando en Todos los titulares, nos enfocaremos en Arte y Cultura.

En la Figura 3.4 se presenta el código del scrap.[3]

```

public static ArrayList<CargarEventos> getListaTitulares() throws IOException {
    ArrayList<CargarEventos> list_titulares_artcult = new
    ArrayList<CargarEventos>();

    Date fecha_actual = new Date();
    SimpleDateFormat formato = new SimpleDateFormat("yyyy/MM/dd/");
    String cadena_fecha = formato.format(fecha_actual);
    String url_universo = "http://www.eluniverso.com/";
    String fecha = cadena_fecha;
    String html = "1/1399/Todos_los_titulares.html";

    String url = url_universo.concat(fecha).concat(html);

    Connection con = Jsoup.connect(url);
    con.userAgent("Mozilla/5.0 (Windows NT 5.1) AppleWebKit/534.24 (KHTML,
    like Gecko) Chrome/11.0.696.65 Safari/534.24");

    Document doc = null;
    doc = con.get();

    if (doc != null) {
        Elements eventos = doc.select("dt:containsOwn(Arte y cultura) > dd
        > a");
        for (Element text_event : eventos) {
            CargarEventos e_evento = new CargarEventos();
            e_evento.setTitulo(text_event.text());
            e_evento.setLink(text_event.attr("abs:href"));
            list_titulares_artcult.add(e_evento);
        }
        return list_titulares_artcult;
    }

    return null;
}

```

Figura 3.4 Código del scrap al sitio web El Universo – Todos los titulares

Tenemos lo siguiente, primero se realiza una conexión a dicho sitio web mediante su URL con el método connect() de la librería Jsoup y luego con el método get() se obtiene el documento del markup HTML, este código se presenta en la figura 4.4.

```

Connection con = Jsoup.connect(url);
con.userAgent("Mozilla/5.0 (Windows NT 5.1) AppleWebKit/534.24 (KHTML, like Gecko)
Chrome/11.0.696.65 Safari/534.24");
Document doc = null;
doc = con.get();

```

Figura 4.4 Código conexión a un sitio web

Dentro del documento; que es código HTML, podemos observar por las etiquetas que Arte y Cultura está contenida en una tabla.

En la Figura 5.4 se presenta el código html de sitio web de El Universo – Arte y Cultura.

```

<dl class="Box">
  <dt>Arte y cultura</dt>
  <dd>
    <a name="lknews" href="/2012/06/24/1/1380/bibliotecarios-pais-reunen-universidad-santa-maria.html" title="Bibliotecarios del país se reúnen en la Universidad Santa María">Bibliotecarios del país se reúnen en la Universidad Santa María</a>
  </dd>
  <dd>
    <a name="lknews" href="/2012/06/24/1/1380/mariano-aguilera-cambia-modalidad.html" title="El Mariano Aguilera cambia de modalidad">El Mariano Aguilera cambia de modalidad</a>
  </dd>
  <dd>
    <a name="lknews" href="/2012/06/24/1/1380/la-ciudad-perros-edicion-especial.html" title="'La ciudad y los perros', en edición especial">'La ciudad y los perros', en edición especial</a>
  </dd>
  <dd>
    <a name="lknews" href="/2012/06/24/1/1380/cita-hablar-sobre-novela-aura.html" title="Una cita para hablar sobre la novela 'Aura'">Una cita para hablar sobre la novela 'Aura'</a>
  </dd>
  <dd>
    <a name="lknews" href="/2012/06/24/1/1380/voces-arpegios-museo-musica.html" title="'Voces y arpegios', en el Museo de la Música">'Voces y arpegios', en el Museo de la Música</a>
  </dd>
  <dd>
    <a name="lknews" href="/2012/06/24/1/1380/guayaquileño-orquesta-pluricultural.html" title="Guayaquileño en orquesta pluricultural">Guayaquileño en orquesta pluricultural</a>
  </dd>
</dl>

```

Figura 5.4 Código html del sitio web de El Universo - Arte y Cultura

Para el análisis de cada uno de los elementos de la tabla el método select() nos ubica en los tags del documento, y por medio del argumento que puede ser desde un tag hasta una expresión regular, la misma que nos ubica las referencias contenidas en Arte y Cultura, en la figura 6.4 podemos ver el código.[4]

```

Elements eventos = doc.select("dt:contains(Arte y cultura) - dt > a");
for (Element text_event : eventos) {
  CargarEventos e_evento = new CargarEventos();
  e_evento.setTitulo(text_event.text());
  e_evento.setLink(text_event.attr("abs:href"));
  list_titulares_artcult.add(e_evento);
}

```

Figura 6.4 Código análisis de tags del documento

Entonces se obtiene el nombre de la referencia y su respectiva dirección URL, éstos vendrían a ser los Eventos de la categoría Arte y Cultura, en la figura 7.4 podemos ver los datos extraídos.

```

TITULO Bibliotecarios del país se reúnen en la Universidad Santa María
URL http://www.eluniverso.com/2012/06/24/1/1380/bibliotecarios-pais-reunen-
universidad-santa-maria.html
TITULO El Mariano Aguilera cambia de modalidad
URL http://www.eluniverso.com/2012/06/24/1/1380/mariano-aguilera-cambia-
modalidad.html
TITULO 'La ciudad y los perros', en edición especial
URL http://www.eluniverso.com/2012/06/24/1/1380/la-ciudad-perros-edicion-
especial.html
TITULO Una cita para hablar sobre la novela 'Aura'
URL http://www.eluniverso.com/2012/06/24/1/1380/cita-hablar-sobre-novela-
aura.html
TITULO 'Voces y arpegios', en el Museo de la Música
URL http://www.eluniverso.com/2012/06/24/1/1380/voces-arpegios-museo-
musica.html
TITULO Guayaquileño en orquesta pluricultural
URL http://www.eluniverso.com/2012/06/24/1/1380/guayaquileno-orquesta-
pluricultural.html
TITULO Jacob Ormaza y su núcleo musical
URL http://www.eluniverso.com/2012/06/24/1/1380/jacob-ormaza-nucleo-
musical.html
TITULO El noruego Per Barclay inunda una sala de arte con 7.000 litros de aceite
negro
URL http://www.eluniverso.com/2012/06/24/1/1380/noruego-per-barclay-inunda-
sala-arte-7000-litros-aceite-negro.html

```

**Figura 7.4 Información de los eventos extraídos del sitio web de El Universo – Arte y Cultura**

Dado a que necesitamos tener la información de cada uno de los eventos obtenidos en el cuadro anterior, debemos realizar el scrap en las urls de los eventos.

En la figura 8.4 vemos el código para extraer los datos de cada uno de los eventos.

```

public static ArrayList<CargarEventos>
getListaEventosTitulares(ArrayList<CargarEventos> ce) throws IOException{

    ArrayList<CargarEventos> lista_eventos = new ArrayList<CargarEventos>();
    String url = new String();

    for(int i=0; i<ce.size(); i++){
        url = ce.get(i).getLink();
        Connection con = Jsoup.connect(url);
        con.userAgent("Mozilla/5.0 (Windows NT 5.1; AppleWebKit/534.24
        HTML; like Gecko; Chrome/41.0.696.55 Safari/534.24");
        con.timeout(10*1000); //Tiempo de espera de la conexión con el
        //servidor: +1minuto

        Document doc = con.get();

        if(doc!=null){
            try{
                Elements agenda = doc.select("#TextoPrint");
                String text_descrip = agenda.text().replaceAll("%<
                CargarEventos e_evento = new CargarEventos();
                e_evento.setTitulo(ce.get(i).getTitulo());
                e_evento.setLink(url);
                e_evento.setDescripcion(text_descrip);

                Elements media = doc.select("#Res [src]");
                if(!media.isEmpty()){
                    for (Element src : media)
                        if (src.tagName().equals("img"))
                            e_evento.setLink_imagen(src.attr("abs:src"));
                }
                else
                    e_evento.setLink_imagen("../ES/Imagenes/icon.png");

                lista_eventos.add(e_evento);

            }catch(Exception e){
                //e.printStackTrace();
            }
        }
    }
    return lista_eventos;
}
}

```

**Figura 8.4 Código para extraer datos en cada uno de los eventos de Arte y Cultura**

La información obtenida al ejecutar el procedimiento anterior corresponde a la información de detalle del evento como descripción e imagen.

En la figura 9.4 podemos ver el resultado presentado al ejecutar el código anterior.



Figura 9.4 Información de cada evento de Arte y Cultura

En la parte del administrador al ejecutar la aplicación carga los eventos obtenidos desde el sitio web de El Universo y Suites Guayaquil, y los deja listos para que el administrador procese la información de acuerdo la estructura requerida, en la figura 10.4 se sustenta lo antes dicho.[2]



Figura 10.4 Captura de pantalla de los eventos extraídos del sitio web de El Universo – Arte y Cultura

El web scrap realizado a los sitios de Guayaquil de la misma forma que los eventos, lo realiza el administrador una vez al mes verificando si un sitio ha cambiado o no.

### **Data XML Parser**

Para una mejor estabilidad y coherencia de la información se procedió a desarrollar un modelo de estructuración rss, archivo suministrado por la pagina web GeYE, por lo cual la aplicación móvil se alimentara de dicho archivo por medio de un proceso de parseador de los datos que extrae la información que está siendo presentada al usuario.

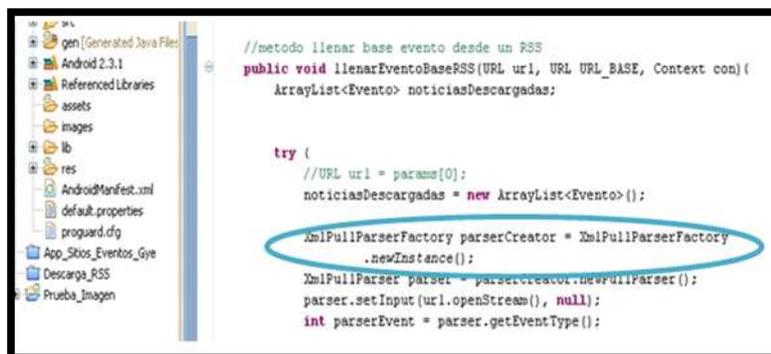
Todo el proceso de análisis del rss se realiza en el momento que el usuario accede a la información de evento y publicidad.

En la figura 11.4, se presenta la aplicación móvil mientras está cargando información.[1]



**Figura 11.4** Captura de Pantalla de la aplicación móvil, mientras realiza la carga de información

Para ello se utiliza el objeto `XmlPullParserFactory` que nos permite analizar el XML definido, como lo podemos observar en la figura 12.4.



```

//metodo llenar base evento desde un RSS
public void llenarEventoBaseRSS(URL url, URL URL_BASE, Context con){
    ArrayList<Evento> noticiasDescargadas;

    try {
        //URL url = params[0];
        noticiasDescargadas = new ArrayList<Evento>();

        XmlPullParserFactory parserCreator = XmlPullParserFactory
            .newInstance();
        XmlPullParser parser = parserCreator.newPullParser();
        parser.setInput(url.openStream(), null);
        int parserEvent = parser.getEventType();
    }
}

```

Figura 12.4 Código que nos permite analizar el XML

Luego verificando una serie de condiciones llegamos a los nodos y a su vez se realiza la carga de los valores para los objetos que se van a mapear en la actividad de la aplicación móvil.

En la figura 13.4 se presenta el código que se utiliza para analizar los nodos.



```

evento.setDescription(textohtml);
}

if (tag.equalsIgnoreCase("nombre_empresa")) {
    evento.setNombre_empresa(parser.nextText());
}

if (tag.equalsIgnoreCase("direccion_empresa")) {
    evento.setDireccion_empresa(parser.nextText());
}

if (tag.equalsIgnoreCase("latitud")) {
    evento.setLatitud(Float.parseFloat(parser.nextText()));
}

if (tag.equalsIgnoreCase("longitud")) {
    evento.setLongitud(Float.parseFloat(parser.nextText()));
}
}
}

```

Figura 13.4 Código para analizar los nodos y realizar la carga de información

Luego la aplicación realiza la presentación de las imágenes por evento de forma asíncrona, obteniendo la información del texto primero y luego ir obteniendo las imágenes. Esto se logra gracias al uso de una clase que implementa un hilo de ejecución paralelo llamada AsyncTask, como podemos ver en la figura 14.4.

```

1 TareaDescargaImagen extends AsyncTask<List<Evento>, String, Drawable>{
2
3     @Override
4     protected Drawable doInBackground(List<Evento>... arg0) {
5         List<Evento> eventos = arg0[0];
6         int imagenesSinCargar = eventos.size();
7         while(imagenesSinCargar > 0){
8             imagenesSinCargar = eventos.size();
9             for(Evento n:eventos){
10                if(n.getImagen()==null || n.getImagen().getIntrinsicHeight() <= 0){
11                    try{
12                        n.loadImagen(n.getTituloImagen());
13                        publishProgress("");
14                    }catch(Exception e){
15                        n.setImagen(getResources().getDrawable(R.drawable.ic_launcher));
16                    }
17                }else{
18                    imagenesSinCargar --;
19                }
20            }
21            try {
22                Thread.sleep(1000);
23            } catch (InterruptedException e) {
24            }
25        }
26    }
27 }

```

**Figura 14.4 Código para mostrar las imágenes asincrónicamente**

Cuando termina la actualización de eventos la aplicación accede a la información por medio de la interfaz del móvil que contiene un listview para la presentación de las opciones, como se muestra en la figura 15.4.

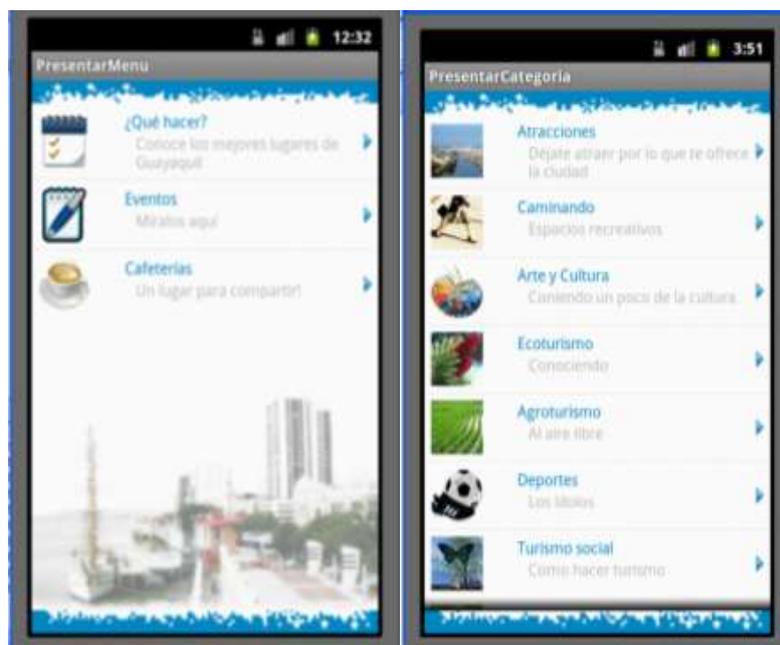


Figura 15.4 Captura de pantalla donde se observan los eventos en un listview

En la siguiente figura 16.4 se muestra la descripción de un sitio en particular, la descripción del lugar, si existen eventos en ese lugar y si están activos.



Figura 16.4 Captura de pantalla donde se describe el evento

La publicación en facebook se realiza por medio de una aplicación de la misma llamada GeYE, creada en facebook developer para obtener el api de interconexión entre las dos aplicaciones y hacer posible el proceso del acceso a la cuenta, como lo podemos ver en la figura 17.4.



**Figura 17.4** Captura de pantalla donde se comparten eventos en facebook  
 Inmediatamente luego de acceder a la cuenta de facebook podemos ver nuestro evento postead, como vemos en la figura 18.4.



**Figura 18.4** Captura de pantalla de un evento publicado en Facebook

## 4.2. Implementación del OpenStreetMap en Android

El OpenStreetMap (OSM) es un proyecto colectivo de datos, que proporciona mapas de calles generados por los usuarios. OSM sigue el modelo de producción entre pares o más conocida como colaboración en masa que creó Wikipedia, y su objetivo es crear un conjunto de datos de mapa que sea gratis, editable, y con licencia bajo nuevos esquemas de derecho de autor.

Para una mejor localización de los eventos presentados en la aplicación móvil, se ha optado utilizar OpenStreetMap para poder visualizar en el mapa los diferentes sitios de la ciudad de Guayaquil y además mostrar los lugares donde se llevarán a cabo los eventos registrados, para poder lograr lo antes mencionado se ha escogido utilizar la librería osmdroid que nos permite interactuar con la data de OpenStreetMap y además de eso crear lugares específicos que no estén registrados en el mapa.

Como toda información que se necesita extraer de la web, necesitamos obtener una instancia a la librería del OpenStreetMap, debemos crear una conexión de acceso a

internet, para lo cual se debe agregar las siguientes líneas al archivo manifest que es un archivo de configuración para crear aplicaciones android.

- android.permission.ACCESS\_COARSE\_LOCATION
- android.permission.ACCESS\_FINE\_LOCATION
- android.permission.ACCESS\_WIFI\_STATE
- android.permission.ACCESS\_NETWORK\_STATE
- android.permission.INTERNET
- android.permission.WRITE\_EXTERNAL\_STORAGE

A continuación, se crea un layout o diseño en donde queremos que aparezca el mapa, a este layout le agregamos la etiqueta `org.osmdroid.views.MapView` para que se visualice el mapa.

En la figura 19.4, se presenta el código del layout en donde se presentará el mapa.

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent" >
  <org.osmdroid.views.MapView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:enabled="true"
    android:clickable="false"
    android:id="@+id/mapa"
  >
</LinearLayout>
```

Figura 19.4 Código del layout en donde se presentará el mapa

Se crea una clase llamada Localizacion.java que contiene un objeto MapView, al mismo que le indicamos el id del mapa definido en el layout y a su vez proporcionarle unas coordenadas que dependerá del sitio al cual se necesita referenciar, también se ha creado un objeto ItemizedOverlayWithFocus<OverlayItem> el cual se sitúa en el mapa indicándole el punto que hace referencia al sitio y además muestra el nombre del sitio y su descripción, esto lo vemos en la figura 20.4.

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.mapa);

    mV_mapa = (MapView) findViewById(R.id.mapa);
    mV_mapa.getController().setZoom(15);

    Coordenada coordenada = new Coordenada();
    int id_coordenada =
    coordenada.obtenerIdCoordenadaPorNombreSitio(PresentacionSitio.nombre_sitio,
    this);
    float latitud = coordenada.obtenerLatitudPorIdCoordenada(id_coordenada,
    this);
    float longitud = coordenada.obtenerLongitudPorIdCoordenada(id_coordenada,
    this);
    mV_mapa.getController().setCenter(new GeoPoint(latitud, longitud));

    /* Itemized Overlay */
    {
        Sitio sitio = new Sitio();
        String direccion_sitio =
        sitio.obtenerDireccionSitioPorNombre(PresentacionSitio.nombre_sitio,
        this);

        //Create a static ItemizedOverlay showing a some Markers on some
        //cities.
        mMyLocationOverlayWithFocus = new ItemizedOverlayWithFocus<OverlayItem>(
        this, new GeoPoint(latitud, longitud));

        /* OnItemClickListener for the Markers, shows a simple Toast. */
        mMyLocationOverlay.setOnItemClickListener(new OnItemClickListener() {
            public boolean onItemClick(int index, OverlayItem item) {
                Toast.makeText(Localizacion.this, "Item " +
                item.getTitle() + " [" + index + "] got tapped",
                Toast.LENGTH_LONG).show();
                return true; // We "handled" this event.
            }
        });

        @Override
        public boolean onItemLongPress(int index, OverlayItem item) {
            // TODO Auto-generated method stub
            return false;
        }

        @Override
        public boolean onItemSingleTapUp(int index, OverlayItem
        item) {
            // TODO Auto-generated method stub
            return false;
        }
    }

    mMyLocationOverlay.setFocusItemsOnTap(true);
    mMyLocationOverlay.setFocusItem(0);
    mV_mapa.getOverlays().add(this.mMyLocationOverlay);
}

```

Figura 20.4 Código de la clase Localizacion.java



incomodaría y provocaría el abandono de la aplicación, para evitar esta deficiencia en el tiempo de respuesta del proceso de scrap o raspado de los datos de varios sitios, se ha logrado establecer otro proceso de carga de los datos.

Para ello se ha planteado realizar pruebas desde la aplicación web con dos sitios, el primero corresponde a <http://www.eluniverso.com/> y <http://www.suitesguayaquil.com/eventos-guayaquil.html>; y medir el tiempo en que demora la aplicación GeYE en cargar los datos haciendo scrap de los mismos.

- El otro punto importante de evaluación es verificar el mejor tiempo de respuesta del aplicativo móvil, presentando la información de dos formas, la primera accediendo a una conexión externa y la segunda, mediante la carga de datos desde una base local.

#### **4.4. Tiempo de consulta y optimización**

Así tenemos las siguientes mediciones para obtener:

- El tiempo promedio de carga de los datos haciendo scraping en dos sitio web.

En la tabla 1.4 se presentan los tiempos en que los datos fueron presentados en la aplicación GeYE.

		Tiempo de Carga (s)	
		El universo	Suites GYE
# prueba	1	1	0,5
	2	0,8	0,3
	3	0,9	0,5
	4	0,6	0,4
	5	0,7	0,5
Promedio		0,8	0,46

**Tabla 1.4 Tiempo promedio de carga de datos – El universo vs Suites GYE**

- El tiempo promedio de respuesta de la información que estamos analizando, es comparada cuando se realiza el scrap a las páginas web vs el proceso de parsear un rss esto nos muestra que parsear un rss es más eficiente que hacer web scraping.

En la tabla 2.4 se presentan los tiempos en que los datos fueron presentados en la aplicación desde el rss vs BD.

		Tiempo de Carga (s)	
		RSS	Base de Datos
# prueba	1	2	1
	2	1	0,8
	3	2	1,5
	4	0,8	0,5
	5	1,5	1
Promedio		1,46	0,96

**Tabla 2.4 Tiempo promedio de carga de datos – Sitio Web vs BD**

Siempre va hacer más eficiente cargar información desde una BD con respecto a realizar un proceso de extracción de información, la obtención de los datos desde su base es directa por lo que hace que la información se cargue rápidamente.

## 4.5. Evaluación de los objetivos

Con la iniciativa de sustentar los objetivos propuestos de las aplicaciones propuestas y desarrolladas, se detalla lo siguiente:

- En la aplicación móvil el usuario tiene acceso a la información detallada de los diferentes sitios y eventos que presenta la ciudad en la aplicación.
- Obtener la información en un menor tiempo.
- El usuario en su móvil observa la correcta ubicación de su evento en el mapa.
- El usuario móvil publica los eventos que desee postear en las redes sociales.
- En la aplicación web el usuario tiene acceso a la creación de eventos, publicidades y a la revisión de los mismos.
- La aplicación web genera un rss con la información de los eventos y otro con la información de las publicidades.

## 4.6. Estadísticas producidas por el sistema

En el gráfico 22.4 se representa el tiempo de carga q se demora al hacer web scraping a los dos sitios web, en el cual observamos

que el tiempo que corresponde al sitio web del universo es mayor al Suites GYE.

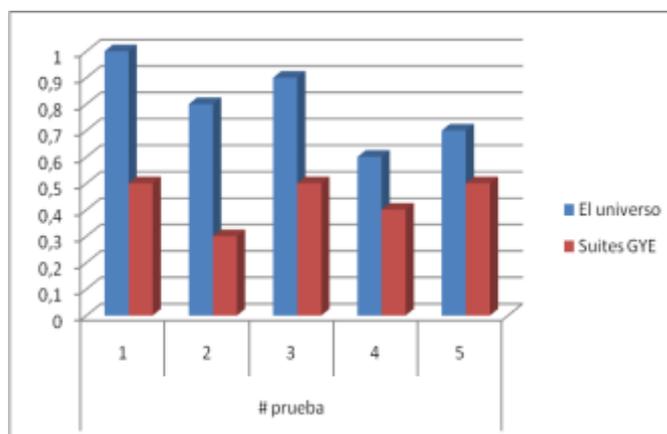


Figura 22.4 Diagrama de barras de los tiempos en que se carga la información de los sitios web, los datos son tomados de la Tabla 1

En el gráfico 23.4 se representa el tiempo de carga promedio de estos dos sitios web, en el cual visualizamos que un 0,8% corresponde al sitio web universo mientras que el 0,46% corresponde al suites GYE.

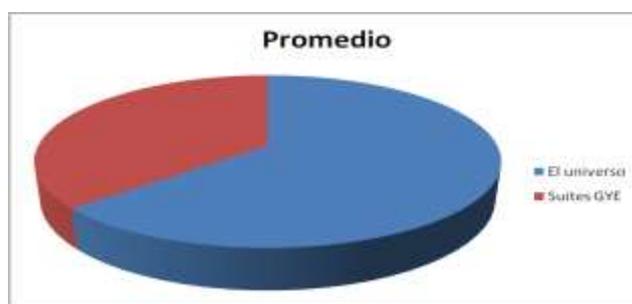
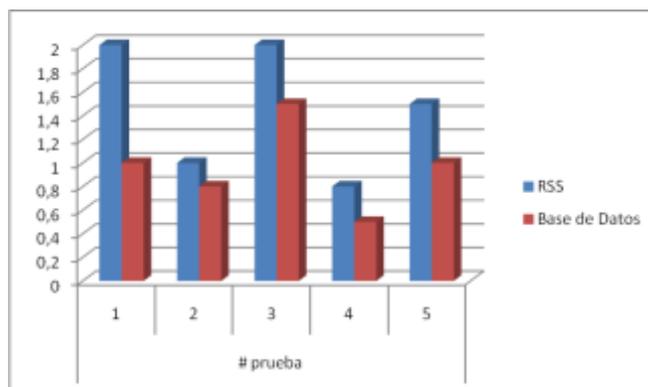


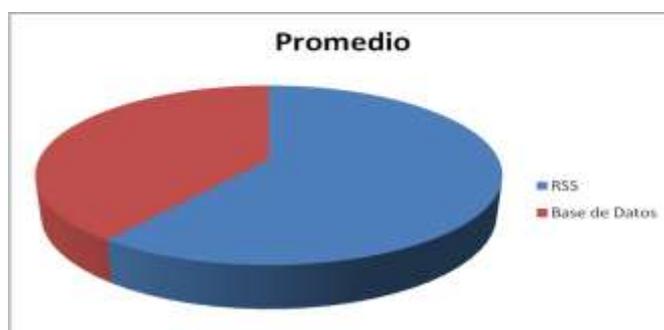
Figura 23.4 Gráfico circular en el que se representa el tiempo de carga promedio, datos tomados de la Tabla 1

En el gráfico 24.4 se representa el tiempo de respuesta que le toma a la aplicación móvil parsear un rss, con respecto al tiempo que le toma extraer los datos de la base.



**Figura 24.4** Diagrama de barras de los tiempos en que se carga la información desde el rss, los datos son tomados de la Tabla 1

El gráfico 25.4 representa el tiempo de respuesta promedio entre el parseador rss y la base de datos local, en el cual visualizamos que un 1,46% corresponde al rss mientras que el 0,96% corresponde a la base.



**Figura 25.4** Gráfico circular en el que se representa el tiempo de carga promedio, datos tomados de la Tabla 1

# CONCLUSIONES Y RECOMENDACIONES

## CONCLUSIONES

1. Se determinó que no todos los sitios en la web permite el “raspado” de sus datos; hay que tener muy en cuenta los derechos de propietarios de los sitios web.
2. Dependiendo de lo que se esté desarrollando en algunos casos el proceso de web scraping podría ser un poco tedioso por que la estructura puede variar y eso generaría problemas en el proceso de obtención de los datos
3. Existen varias formas de presentación de información en mapas, hemos concluido que Openstreetmap supera a las demás en precisión, por que presenta en detalle mucho mejor la información sobre carreteras puesto que lo editan los contribuyentes.
4. La carga de los datos puede ser extensa cuando la información sea obtenida por primera vez.
5. Aplicar un análisis de los datos genera información de calidad y la tolerancia a fallos puede ser controlada.

## RECOMENDACIONES

1. Se recomienda que las páginas mantengan el mismo markup determinado al inicio del desarrollo, ya que la forma de extracción de la información tienen referenciados tags debidamente ordenados.
2. Los administradores de las paginas analizadas que suministran los eventos deben mantener un proceso de actualización continuo, y de ser posible una comunicación con el desarrollador de la aplicación para mantener a tanto de las posibles modificaciones realizadas a dichas páginas web.
3. Los usuarios que adquieran la aplicación deben asegurarse que su dispositivo sea compatible con la aplicación que desean instalar, es decir que el nivel de api expresada por la aplicación debe ser mayor o igual al nivel de api de su dispositivo android de lo contrario Android evitara que el usuario instale la aplicación.

# APÉNDICE

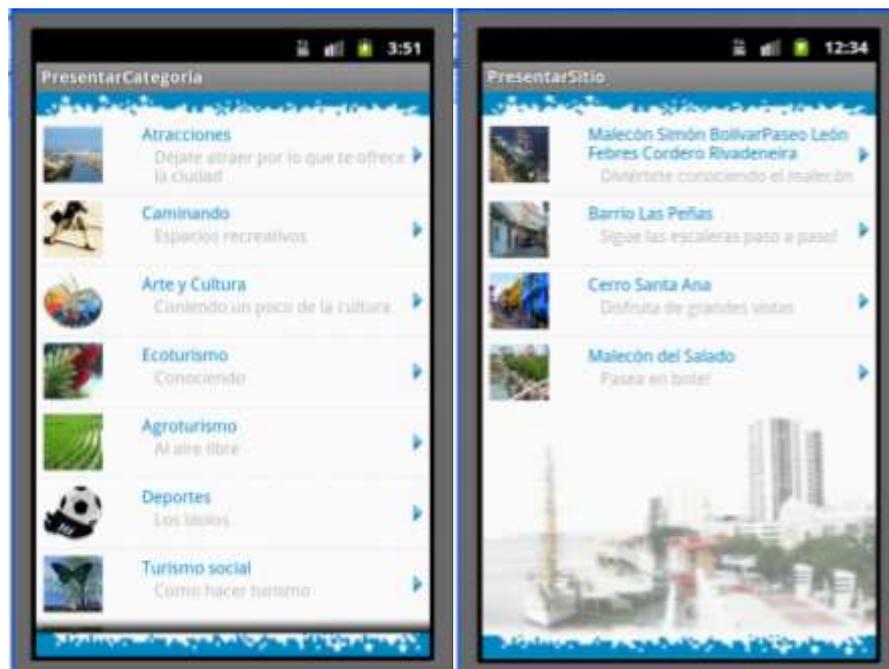
## DISEÑO DE LAYOUT DE LA APLICACIÓN MÓVIL

1. Pantalla principal y menú de eventos.

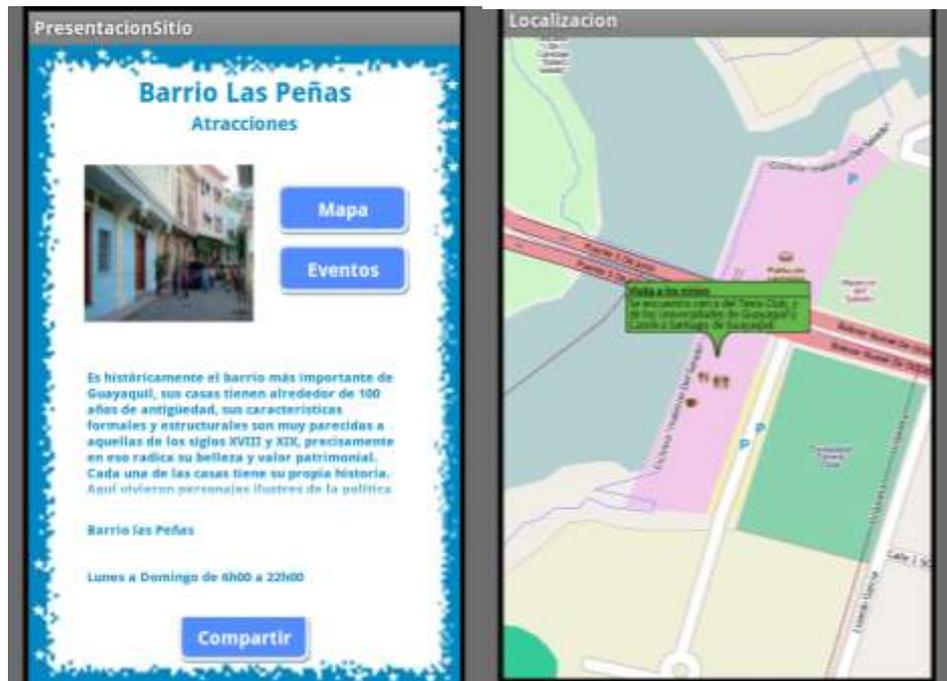


- 2.

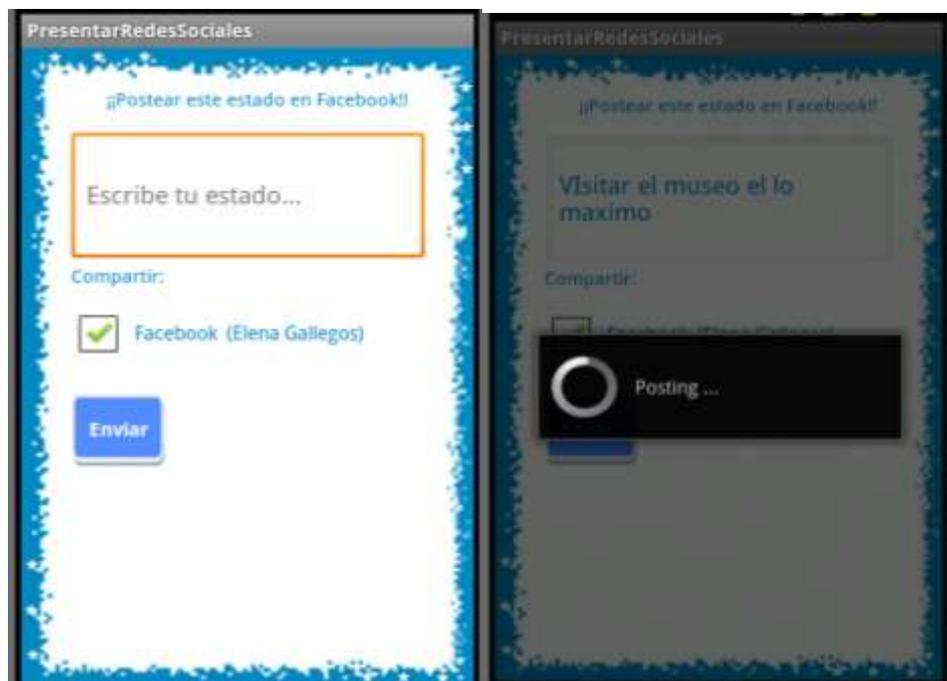
3. Que hacer? Se presentan las categorías y las subcategorías de los sitios más relevantes de la ciudad.



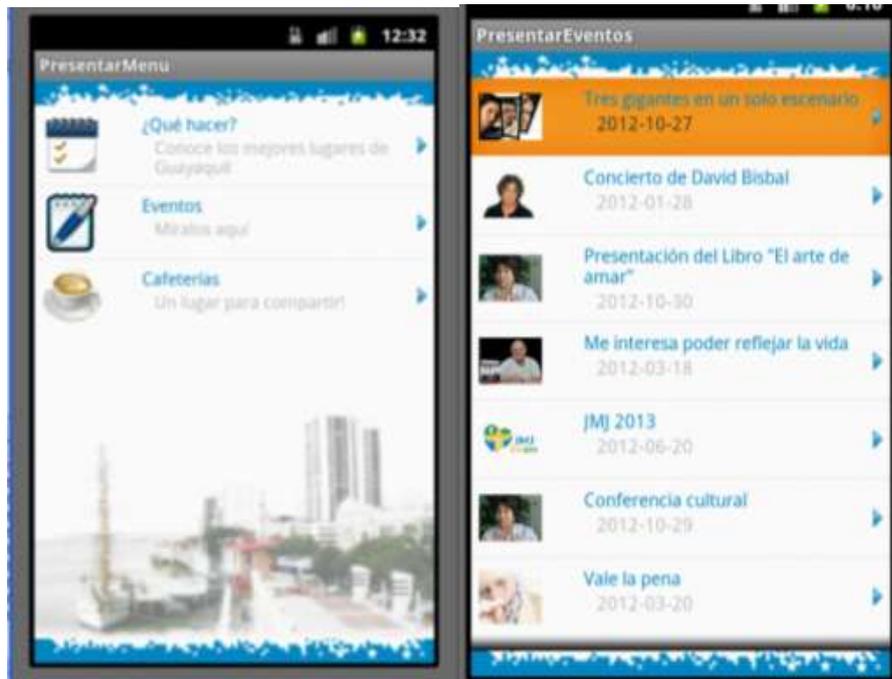
4. Presentación del sitio y la localización del sitio en el mapa.



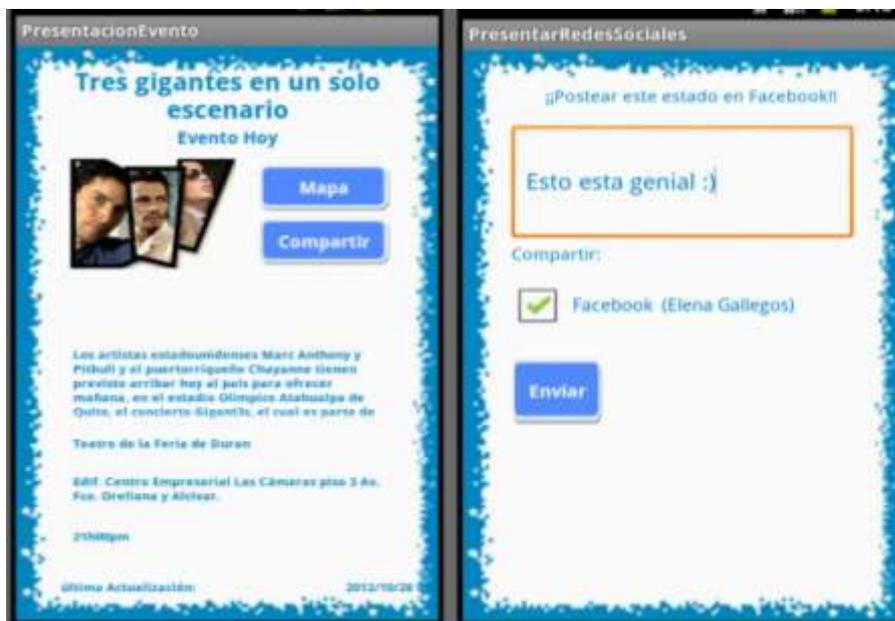
5. Layout de publicación en facebook



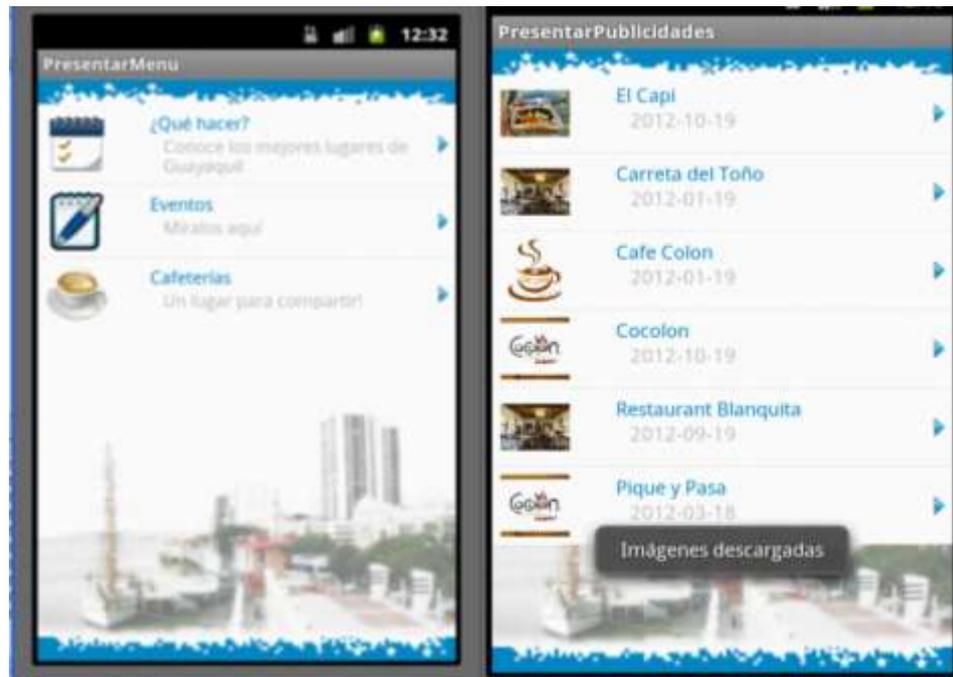
6. Layout partiendo del menú de eventos.



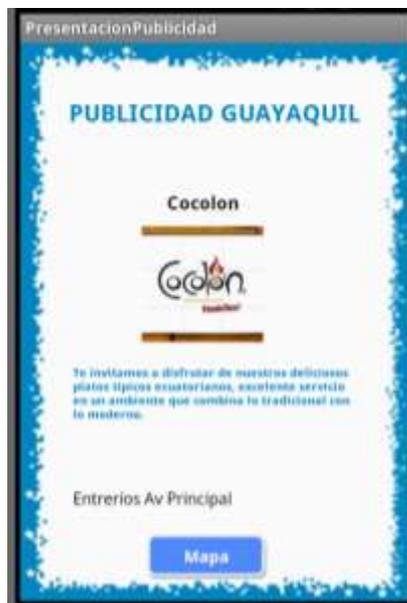
## 7. Presentación del evento y publicación



## 8. Menú publicitario y presentación de los lugares publicitarios que en este proyecto hemos propuesto locales alimenticios-cafeterías.



9. Presentación del layout publicidad.



10. Menus: sitios- eventos - publicidad



11. Errores cuando no tenemos acceso al internet la aplicación descarga los últimos eventos y publicidades que tiene registrados en su base de datos.

PresentarEventos	
Tres gigantes en un solo escenario	2012-10-27
Concierto de David Bisbal	2012-01-28
Presentación del Libro "El arte de amar"	2012-10-30
Me interesa poder reflejar la vida	2012-03-18
JMJ 2013	2012-06-20
Conferencia cultural	2012-10-20
¡Error en la descarga!	
Conferencia cultural	2012-03-20

PresentarPublicidades	
Carreta del Toño	2012-01-19
Restaurant Blanquita	2012-01-19
Pique y Pasa	2012-03-18
El chalet	2012-01-20
La Morenita	2012-04-19
Alibaba	2012-04-19
¡Error en la descarga!	
	2012-04-19

## BIBLIOGRAFÍA

[1] Manfred Moser. "Open Source Android Development Tools". [Online]  
[Cited: Julio, 2011.]

<http://assets.en.oreilly.com/1/event/61/Open%20Source%20Android%20Development%20Tools%20-%20SDK,%20ADT%20and%20Beyond%20Presentation.pdf>

[2] Mateo Carles. "Desarrollo de Aplicaciones web". FUOC  
XP06/M2108/01497. Universidad Oberta de Catalunya, Marzo 2004. pag.  
232-234.

[3] Scripting Intelligence: "Web 3.0 Information Gathering and  
Processing". [Online]

[http://books.google.com.ec/books?id=ElSxSLsc3M0C&pg=PA205&dq=Web+scraping+in+Java&hl=es&sa=X&ei=54\\_eT--CGZK08AS3\\_ryECw&sqi=2&ved=0CDkQ6AEwAA#v=onepage&q=Web%20scraping%20in%20Java&f=false](http://books.google.com.ec/books?id=ElSxSLsc3M0C&pg=PA205&dq=Web+scraping+in+Java&hl=es&sa=X&ei=54_eT--CGZK08AS3_ryECw&sqi=2&ved=0CDkQ6AEwAA#v=onepage&q=Web%20scraping%20in%20Java&f=false)

[4] "Use selector-syntax to find elements". [Online]

<http://jsoup.org/cookbook/extracting-data/selector-syntax>