



UNIVERSIDAD SIMÓN BOLÍVAR  
DECANATO DE ESTUDIOS PROFESIONALES  
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

**Desarrollo de una aplicación móvil basada en el sistema operativo AndroidOS  
para el portal Tuguia.de**

Por:  
Juan Vicente Rosas Ramirez

Realizado con la asesoría de:  
Tutor Académico: Carmen Rosseline Rodríguez  
Tutor Industrial: Gianpaolo Valero

PROYECTO DE PASANTÍA LARGA  
Presentado ante la Ilustre Universidad Simón Bolívar  
como requisito parcial para optar al título de  
Ingeniero en Computación

**Sartenejas, Septiembre de 2013**



UNIVERSIDAD SIMÓN BOLÍVAR  
DECANATO DE ESTUDIOS PROFESIONALES  
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

ACTA FINAL PASANTÍA LARGA

**Desarrollo de una aplicación móvil basada en el sistema operativo AndroidOS  
para el portal Tuguia.de**

Presentado por:

**Juan Vicente Rosas Ramirez**

Esta Pasantía Larga ha sido aprobado por el siguiente jurado examinador:

---

PROFESOR 1

---

PROFESOR 2

---

Ing. Gianpaolo Valero

**Sartenejas, FECHA (dd/mm/aa)**

## Resumen

Empresas TD es una empresa dedicada a ofrecer soluciones tecnológicas e innovadoras orientadas a mejorar la calidad del e-commerce en Venezuela y a brindar la oportunidad de darse a conocer a los diferentes comercios y locales de las principales ciudades del país en este medio.

Bajo este precepto Empresas TD crea el portal Tuguia.De con el objetivo de convertirse en la principal guía de locales y comercios del país. Dado el crecimiento de Tuguia.De como proyecto, la empresa decide llevar esta idea a las plataformas móviles comenzando con el desarrollo de una aplicación nativa para el sistema operativo AndroidOs.

Este informe describe las actividades realizadas durante el desarrollo del proyecto de pasantía larga, que consistió en la elaboración de una aplicación móvil para el portal Tuguia.de que permita llevar la iniciativa de este portal a un mayor número de usuarios de una forma sencilla, permitiendo así que éstos puedan acceder a los diferentes servicios prestados por esta guía desde cualquier lugar usando su teléfono celular Android.

El desarrollo de esta aplicación móvil, se realizó siguiendo el proceso iterativo e incremental de la metodología ágil Scrum, y los principios del desarrollo guiado por pruebas y comportamientos (*Test-driven development y behavior-driven development*); fue realizada con el lenguaje de programación java, haciendo uso del entorno de trabajo dispuesto para Android (*Android SDK*) y SQLite como manejador de base de datos de la aplicación en el dispositivo.

Como resultado de este proyecto de pasantía se realizó el análisis, diseño e implementación de la aplicación móvil que cubre los casos de usos y requerimientos planteados en el alcance inicial. Adicionalmente se realizó la configuración de un servidor de ApacheSolr para mejorar el resultado de las búsquedas y diversas mejoras funcionales al API (*Application Programming Interface*) que permite acceder a las funcionalidades del portal Tuguia.De siendo este la materia prima de la aplicación móvil.

A Dios, mi familia, profesores y amigos.

# Agradecimientos

El presente

En primer lugar

# Índice general

Índice general	VII
<b>1. Introducción</b>	<b>1</b>
<b>2. Entorno Empresarial</b>	<b>6</b>
2.1. Antecedentes de la Empresa . . . . .	6
2.2. Estructura Organizacional . . . . .	7
2.3. Ubicación del Pasante . . . . .	8
<b>3. Marco teórico</b>	<b>10</b>
3.1. Aspectos del Negocio . . . . .	10
3.1.1. Local . . . . .	10
3.1.2. Taxonomías . . . . .	11
3.1.3. Reseña . . . . .	11
3.2. Aspectos Tecnológicos . . . . .	11
3.2.1. Tecnologías Asociadas con la Aplicación Móvil . . . . .	12
3.2.1.1. Teléfono Inteligente . . . . .	12
3.2.1.2. Android . . . . .	12
3.2.1.2.1. Arquitectura del Sistema Android . . . . .	13
3.2.1.2.2. Estructura básica de una aplicación Android . . . . .	14
3.2.1.2.3. <i>SQLite</i> . . . . .	15
3.2.1.3. Eclipse . . . . .	16
3.2.2. Tecnologías Asociadas con Tuguia.de . . . . .	16
3.2.2.1. Drupal . . . . .	17
3.2.2.2. Apache Solr . . . . .	17

<b>4. Marco Metodológico</b>	<b>19</b>
4.1. Scrum . . . . .	19
4.1.1. Roles en Scrum . . . . .	20
4.1.2. El proceso de Scrum . . . . .	20
4.2. Desarrollo Basado en Pruebas y Comportamientos . . . . .	22
<b>5. Desarrollo del Proyecto</b>	<b>25</b>
5.1. Descripción del Sistema . . . . .	25
5.1.1. Contenido del Sitio . . . . .	25
5.2. Tareas de Realizadas en Cada Iteración . . . . .	27
5.2.1. Primera Iteración . . . . .	27
5.2.1.1. Objetivos Planteados . . . . .	28
5.2.1.2. Resultados Alcanzados . . . . .	28
5.2.2. Segunda Iteración . . . . .	30
5.2.2.1. Objetivos Planteados . . . . .	31
5.2.2.2. Resultados Alcanzados . . . . .	31
5.2.3. Tercera Iteración . . . . .	33
5.2.3.1. Objetivos Planteados . . . . .	33
5.2.3.2. Resultados Alcanzados . . . . .	33
5.2.4. Cuarta Iteración . . . . .	34
5.2.4.1. Objetivos Planteados . . . . .	34
5.2.4.2. Resultados Alcanzados . . . . .	34
5.2.5. Quinta Iteración . . . . .	36
5.2.5.1. Objetivos Planteados . . . . .	36
5.2.5.2. Resultados Alcanzados . . . . .	36
5.2.6. Sexta Iteración . . . . .	37
5.2.6.1. Objetivos Planteados . . . . .	37
5.2.6.2. Resultados Alcanzados . . . . .	37
5.2.7. Séptima Iteración . . . . .	38
5.2.7.1. Objetivos Planteados . . . . .	38
5.2.7.2. Resultados Alcanzados . . . . .	39
5.2.8. Octava Iteración . . . . .	40
5.2.8.1. Objetivos Planteados . . . . .	40
5.2.8.2. Resultados Alcanzados . . . . .	40

<b>6. Conclusiones y Recomendaciones</b>	<b>42</b>
<b>Bibliografía</b>	<b>43</b>



# Capítulo 1

## Introducción

En la sociedad actual se hace imprescindible que las empresas posean una pagina web; el mundo que hoy conocemos se basa en Internet y es en éste donde la gente busca las respuestas a sus inquietudes cotidianas. Actualmente en el mundo existen casi el doble de *smartphones* que computadores personales [DGT]. Según los resultados obtenidos en una encuesta llevada a cabo por Google en conjunto con IPSOS OTX mediaCT al final del 2010, aparte de hacer y recibir llamadas, las actividades realizadas con mayor frecuencia en los teléfonos inteligentes son: navegar en Internet, el uso de motores de búsquedas y en tercer lugar el uso de aplicaciones [TMM]. Para las empresas, ya no sólo es necesario estar en la web, deben estar al alcance de los teléfonos inteligentes.

Los *smartphones* han llegado a ser parte de la vida de las personas y son dispositivos mucho mas personales que una computadora portátil o de escritorio, estos teléfonos han penetrado hasta el punto que muchos usuarios raramente se separan de sus equipos móviles durante el día.

Tuguia.de es una empresa joven que pretende ser la principal guía de comercios y locales de Venezuela, su objetivo no solo es brindar información actualizada de los locales, sino que también exponer la experiencia que tuvieron los usuarios al visitar estos sitios. Tomando en

cuenta esto se hace importante para Tuguia.de convertirse en un servicio disponible a los usuarios 24 horas durante los 7 días de la semana, accesible desde cualquier momento y lugar por lo que se hace necesario el desarrollo de un mecanismo para que los usuarios puedan acceder a la información del portal desde sus celulares.

Conscientes de la influencia que pueda tener el éxito de la versión móvil de Tuguia.de, se decide emprender un desarrollo que permita llevar la aplicación a los celulares inteligentes. Sin embargo para lograr esto existen varias alternativas a considerar; aplicaciones nativas, aplicaciones híbridas o multiplataforma y paginas web diseñadas específicamente celulares.

Las aplicaciones nativas, llevan este nombre porque son desarrollos implementados en el lenguaje nativo del dispositivo. Gracias a esto, sacan el mejor rendimiento posible del *hardware* del móvil, así que tienen un alto desempeño y permiten acceder a las diferentes características presentes en los teléfonos inteligentes con mayor facilidad. Sin embargo, el hecho de necesitar un desarrollo en un lenguaje específico, distinto para cada plataforma eleva los costos y el tiempo de desarrollo.

Las aplicaciones híbridas permiten escribir código en un solo lenguaje para después exportarlo a código nativo y de esta manera obtener una aplicación que funcione en las múltiples plataformas con un solo desarrollo. Es de notar que estas aplicaciones tienen un bajo desempeño si las comparamos con las nativas, además en estos entornos multiplataforma se dificulta la tarea de acceder a los diferentes componentes del equipo móvil con efectividad.

Las páginas web de diseño específico para equipos móviles son sitios web que pueden cambiar la forma en la que muestran su contenido de acuerdo al equipo en el que se esta visualizando y de esta manera aprovechar al máximo las dimensiones de las pantallas de los dispositivos. Así se consigue que con un solo desarrollo se obtiene un producto que puede operar en tanto en las computadoras como en equipos móviles. Al igual que las otras paginas web éstas se ejecutan en un navegador, por lo que se torna bastante complejo acceder a las

características propias de los teléfonos inteligentes de hoy en día.

Luego de evaluar estas alternativas, la junta directiva de Tuguia.de decide que desarrollar una aplicación nativa para su proyecto, a pesar de los costos elevados y los altos tiempos de desarrollo desean que la aplicación sea lo mas extensible posible, no obstante, es necesario determinar cuál sistema operativo móvil es el más adecuado para iniciar las versiones móviles de Tuguia.de

Según un estudio realizado para finales del 2012 en los Estados Unidos, [NTD] para el tercer trimestre de ese año la participación de Android en el mercado de teléfonos inteligentes era del 52 %. Se hace notable la superioridad de Android sobre los demás sistemas operativos, por lo que es el sistema escogido para iniciar el desarrollo móvil de Tuguia.de, pensado en un futuro a corto plazo expandirse al iOS de Apple.

Actualmente Tuguia.de cuenta con funcionalidades tales como búsqueda de locales dadas sus ubicación y/o una serie de taxonomías asociadas a éstos, además siempre que se muestra información de un local se debe mostrar su ubicación en el mapa, dirección, fotografías asociadas e información de contacto. Un aspecto importante para la visión de negocio de Tuguia.de es la capacidad de que los usuarios puedan realizar comentarios y establecer una puntuación de uno a cinco sobre los locales, esto con el objetivo de brindar no solo la información básica de un establecimiento, sino también priorizar los locales de acuerdo a la opinión de los usuarios. Adicionalmente, Tuguia.de prevé ofrecer la posibilidad a los dueños de cada local de gestionar la información de su comercio y campañas de publicidad en el sitio.

El Proyecto de pasantía consiste en llevar las funcionalidades presentes a una aplicación móvil para el sistema operativo Android. Gracias a los elementos propios de los teléfonos inteligentes como geolocalización, se pretende agregar nuevas funcionalidades y así realizar búsquedas usando la posición actual del equipo, de esta manera brindar información más precisa a los usuarios de Tuguia.de.

Para exportar la funcionalidad de Tuguia.de se desarrolló un API (*Application Programming Interface*), que permite gestionar el contenido de la página a través de él. La aplicación móvil debe usar este API como materia prima para realizar sus operaciones y de esta forma mantener una comunicación constante con la aplicación web de Tuguia.de logrando así ofrecer contenido siempre actualizado.

Se espera que la aplicación móvil, pueda operar efectivamente en las diferentes versiones activas del sistema operativo Android, al menos en las que concentran la mayor cantidad de usuarios, además debe tener un uso mínimo de la red de datos y poseer una interfaz sencilla y fácil de usar.

Por estas razones se plantea como objetivo general del proyecto de pasantía desarrollar una aplicación móvil basada en el sistema operativo AndroidOS para el portal Tuguia.de, que permita realizar búsquedas avanzadas sobre los locales, mostrar la información detallada de un local, realizar comentarios y que posea un manejo básico de usuarios.

Los objetivos específicos que se desean alcanzar con la ejecución de este proyecto, son los siguientes:

- Estudiar la arquitectura de la plataforma de desarrollo de Android, entornos de pruebas y demás tecnologías asociadas.
- Diseñar y desarrollar un mecanismo de comunicación con el API de Tuguia.de.
- Realizar los ajustes y extensiones necesarias sobre el API de Tuguia.de para garantizar el funcionamiento deseado.
- Diseñar e implementar un modelo de datos que permita gestionar los diferentes nodos de información presentes en Tuguia.de dentro la aplicación móvil.
- Desarrollar un módulo para la aplicación móvil que permita realizar búsquedas avanzadas, usando palabras claves y una ubicación, que puede ser las coordenadas geográficas

o el nombre de un determinado lugar.

- Diseñar y desarrollar un componente que permita visualizar los locales de Tuguia.de en la aplicación móvil.
- Diseñar e implementar la funcionalidad que permita subir comentarios asociados a un local de Tuguia.de.
- Implementar diferentes mecanismos que permitan que la aplicación realice el menor uso posible de la red de datos.

El presente proyecto pretende generar una versión beta de la aplicación móvil de Tuguia.de en la plataforma Android; de manera que esta sea estable y se encuentre en un estado de completitud suficiente para ser aprobada por la directiva de la empresa y pasar a una fase de prueba mas extensa pre-despliegue.

# Capítulo 2

## Entorno Empresarial

En este capítulo se realizará una descripción de Empresas TD y sus antecedentes, además se explicará en que consiste el portal web Tuguia.de que se tiene planteado se convierta en el núcleo de negocio para Empresas TD, así como el resto de su portafolio de productos y de esta forma dar a conocer el entorno laboral en el que se realizó el proyecto de pasantía.

### 2.1. Antecedentes de la Empresa

En el año 2010 la compañía Servicios 1488 C.A. realizó el lanzamiento del portal Tudescounton.com, con el objetivo de brindar a todos los venezolanos los mejores descuentos en los diferentes comercios asociados, convirtiéndose así en la primera empresa de venta de cupones de descuento del país.

El crecimiento de Tudescounton.com y las nuevas tendencias de comercio electrónico llevó a la compañía Servicios 1488 C.A. a consolidarse en lo que hoy se conoce como Empresas TD añadiendo a su portafolio de productos diferentes iniciativas que le permiten ofrecer soluciones tecnológicas orientadas a mejorar la calidad del comercio electrónico en Venezuela

y a brindar la oportunidad de darse a conocer a los diferentes comercios y locales de las principales ciudades del país en este medio.

De esta forma surge el portal Tuguia.de con el objetivo de convertirse en la principal guía de locales y comercios del país, y en un espacio donde los usuarios puedan compartir sus experiencias al visitar estos comercios. Se tiene previsto que Tuguia.de se convierta en el núcleo del negocio de Empresas TD, ya que este portal permitirá centralizar los diferentes servicios prestados por la compañía.

## 2.2. Estructura Organizacional

Empresas TD es una compañía con una gran proyección a futuro compuesta mayormente por gente joven. El perfil buscado en los miembros del equipo de trabajo es el de un profesional dinámico, creativo e innovador, comprometido por brindar a los asociados y usuarios experiencias de alta calidad.

Actualmente se gestionan cuatro proyectos en Empresas TD, como se puede ver en la figura 2.1y se describen a continuación:

- **Tudescuenton:** es una plataforma le brinda a los venezolanos, la oportunidad de conocer las mejores cosas que hacer, en las principales ciudades del país, a través de descuentos insuperables.” [TDC]
- **Lealtag:** ”LealTag es la plataforma que te permite recibir premios, descuentos y regalos en tus sitios favoritos por ser un cliente frecuente.” [LTG]
- **Sucucho:** .ofrece a los creativos venezolanos un espacio en donde pueden exponer su talento ante el país, en el cual brindamos la oportunidad de que cada uno tenga su tiendita propia 365 días al año.” [SCC].

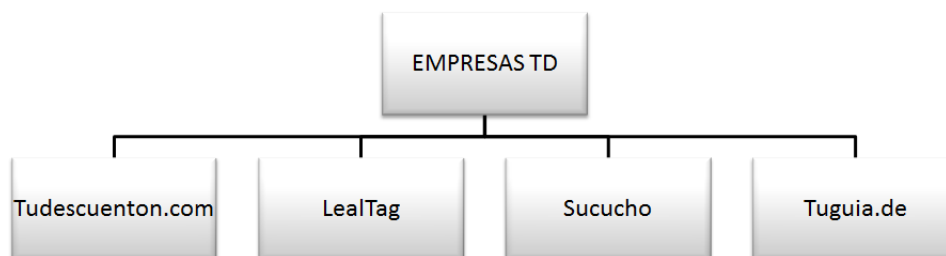


Figura 2.1: Organigrama de Empresas TD.

- **Tuguiade:** "Tuguía.de es una página que sirve para conectar los locales con sus clientes. Ofreciéndoles a los usuarios información verificada de los mismos y la posibilidad de escribir sobre sus experiencias, subir fotos e intercambiar historias con los otros clientes." [TGD].

Cada proyecto tiene asignado un director general y un esquema organizacional específico que se ajusta a sus necesidades. En el caso de Tuguia.de; proyecto en el que desarrolló el proyecto de pasantía, el organigrama se muestra en la figura 2.2 donde se puede observar la estructura organizativa de la que el pasante fue parte.

## 2.3. Ubicación del Pasante

El proyecto de pasantía llevado a cabo en el periodo Abril-Septiembre de 2013 se realizó bajo la tutela del Ing. Gianpaolo Valero Gerente del Departamento de Tecnología de Empresas TD y bajo la supervisión directa del Lic. Alexander Mayora, coordinador de



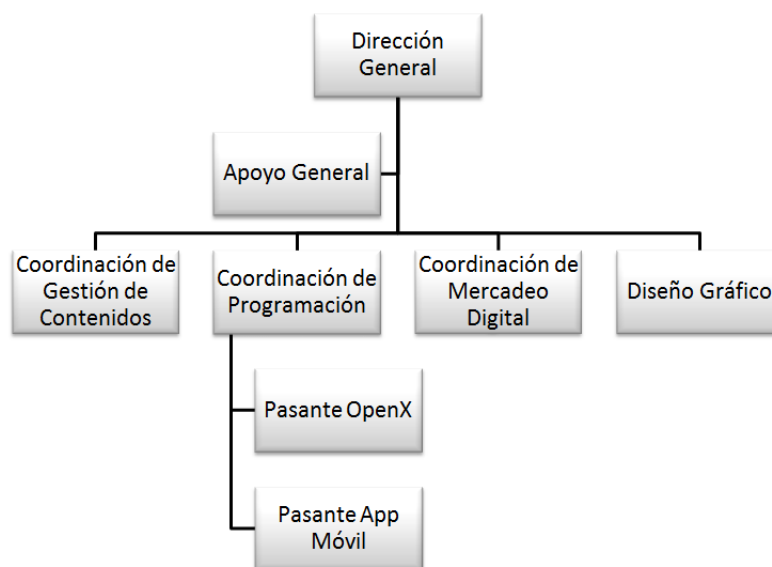


Figura 2.2: Organigrama de Tuguia.de

desarrollo del proyecto Tuguia.de.

Durante el desarrollo el proyecto el pasante fue miembro del equipo de Tecnología de Tuguia.de estando ubicado en la oficina correspondiente a este. Para el desarrollo le fue asignado un escritorio compartido y un computador personal con acceso a Internet.

# Capítulo 3

## Marco teórico

En este capítulo se presentan los fundamentos teóricos más relevantes involucrados con el proyecto de pasantía, así como también las herramientas y tecnologías aplicadas durante el desarrollo de la aplicación móvil.

### 3.1. Aspectos del Negocio

Al consistir el proyecto de pasantía en el desarrollo de la versión móvil para el portal Tugua.de, se hace necesario comprender algunos de los conceptos y términos asociados con el negocio y de esta manera facilitar el entendimiento de los posteriores capítulos.

#### 3.1.1. Local

Un local comercial es un establecimiento que tiene como objetivo principal el desarrollo de alguna actividad comercial o económica. Los locales pueden ofrecer productos o servicios. Algunos ejemplos para los primeros son supermercados, farmacias, tiendas, etc. En el caso de los locales que ofrecen servicios podemos mencionar restaurants, lugares de entretenimiento

como cines o discotecas, etc. [DLC]

### **3.1.2. Taxonomías**

Una taxonomía es un tipo de vocabulario o conjunto de palabras en que todos los términos están conectados, su objetivo es organizar la información y mejorar la búsqueda de contenidos en los sitios web [Cen05]. En el caso de Tuguia.de las taxonomías usadas son: ciudad, urbanización, categoría, subcategoría y atributos.

### **3.1.3. Reseña**

”Para Tuguía.de, una reseña es un pequeño texto en donde el usuario cuenta su experiencia. Puede ser también llamada comentario. Éstas serán escritas en los cuadros de texto dentro de cada local, y servirán para que cada usuario cuente su historia” [TGD]. Para que una reseña sea válida el usuario debe asignarle una calificación de uno a cinco al local que esta comentando.

## **3.2. Aspectos Tecnológicos**

En esta sección se presentan las tecnologías y herramientas utilizadas en el desarrollo de la pasantía, así como también algunos términos asociados que son de obligatoria referencia para facilitar al lector la comprensión del contenido subsiguiente. Es necesario hacer una separación entre las tecnologías y herramientas utilizadas para el desarrollo de la aplicación móvil y las asociadas a Tuguia.de con las que se interactuó en el transcurso de la pasantía.

### 3.2.1. Tecnologías Asociadas con la Aplicación Móvil

A continuación se presentan las diferentes tecnologías y herramientas involucradas en la construcción de la aplicación Android para Tuguia.de

#### 3.2.1.1. Teléfono Inteligente

Los teléfonos inteligentes o *smartphones* (nombre en ingles), son teléfonos celulares que además de poseer las características habituales de un celular común, permiten acceder a una serie de aplicaciones integradas, navegar en Internet, tomar fotografías, grabar vídeos, entre otras funcionalidades que lo asemejan a una computadora portátil móvil [PCM].

El objetivo del desarrollo es llevar a Tuguia.de a estos dispositivos.

#### 3.2.1.2. Android

Android es un sistema operativo basado en *Linux*, diseñado en principio para dispositivos móviles con pantalla táctil, es de código abierto y es liberado bajo la licencia Apache; una licencia bastante flexible, que permite que cualquier desarrollador realice aplicaciones y las integre al sistema sin dificultad.

Las aplicaciones nativas se desarrollan, usando el SDK (*Software Development Kit*) y escribiendo la aplicación en lenguaje *Java*. El SDK es un paquete proporcionado directamente por Google que encapsula todas las bibliotecas y herramientas de desarrollo necesarias para crear, probar, y depurar las aplicaciones [ASD]. Por otro lado existe la alternativa de usar el NDK (*Native Development Kit*) también provisto por Google que permite mediante el uso de lenguajes como *C* y *C++* interactuar directamente con el sistema operativo y crear bibliotecas. Esta última alternativa sólo es recomendada para situaciones específicas donde la aplicación en cuestión realiza cálculos intensivos sobre el *CPU* y debe emplearse sólo cuando el SDK no provee de la funcionalidad deseada [AND].

### 3.2.1.2.1. Arquitectura del Sistema Android

Gran parte del éxito de este sistema operativo se debe a su arquitectura. En un inicio el sistema fue pensado solo para ser usado en dispositivos móviles, sin embargo, dada su gran capacidad de adaptación, actualmente además de ser usado en celulares, también en televisores, reproductores de automóvil, *netbooks* y diversos equipos electrónicos de todo tipo.

Como se observa en la figura 3.1 el sistema operativo se desglosa de la siguiente manera:

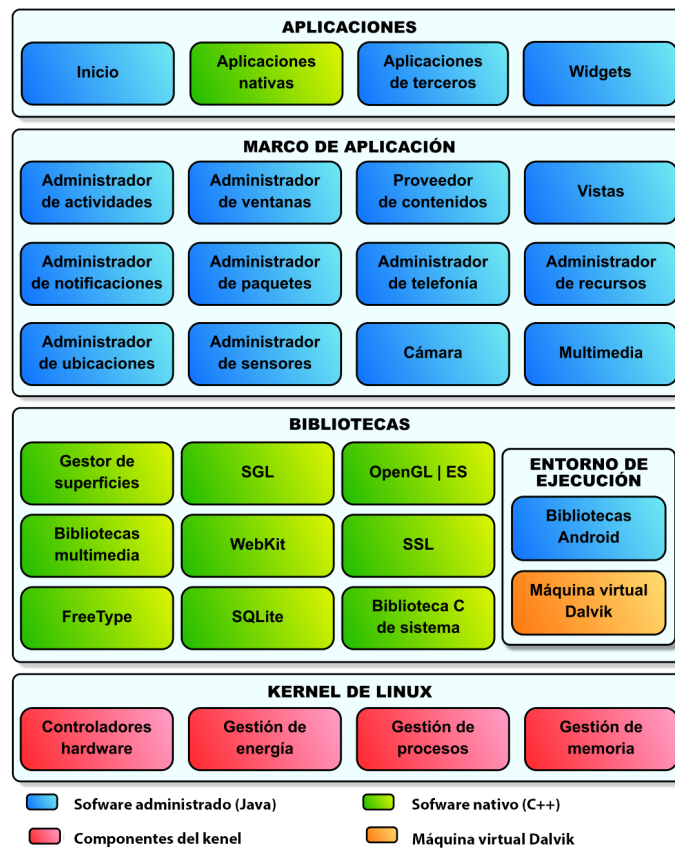


Figura 3.1: Arquitectura del Sistema Operativo Android [AAT]

- Aplicaciones: En este nivel se encuentran todas las aplicaciones presentes en el equipo, tanto las instaladas por el usuario como las provistas por el fabricante. Es importante

destacar que todas cuentan con la misma capacidad para acceder los servicios que proporciona el sistema operativo. Gracias a esto, se pueden hacerse desarrollos que usen los mismos recursos que usan las aplicaciones nativas. [AAT].

- Marco de aplicación: Esta capa esta formada por las diferentes clases que componen el *API* que permiten a las aplicaciones realizar sus funciones, constituyen en el entorno de trabajo que posibilita acceder a los diferentes recursos del dispositivo [AAT].
- Bibliotecas: Esta capa se ubica justo sobre el Kernel del sistema operativo y está constituida por un conjunto de bibliotecas escritas en *C* o *C++* que son compiladas para la arquitectura específica del equipo, su función es interactuar con el sistema operativo para acceder a los diferentes recursos gestionados por éste y poner al alcance de las aplicaciones la funcionalidad respectiva. [AAT].
- Entorno de ejecución: Este no se considera una capa como tal puesto que se compone principalmente de un conjunto de bibliotecas que proporcionan la funcionalidad base. El elemento principal de esta sección es la maquina virtual Dalvik que ejecuta cada de las aplicaciones instaladas en el dispositivo [AAT].
- Kernel de Linux: El núcleo del sistema operativo Android es un kernel Linux bastante similar al que puede incluir cualquier distribución de GNU/Linux conocida. En el caso de Android se realizan adaptaciones que permita ajustarse a las características del hardware en el que se ejecutará. Su función principal es proveer una capa de abstracción para el hardware y gestionar los diferentes recursos del teléfono y del sistema operativo [AAT].

### 3.2.1.2.2. Estructura básica de una aplicación Android

Una actividad es una clase java que contiene todo el código necesario para realizar una

determinada tarea. Las actividades son el componente principal de las aplicaciones Android. Una aplicación puede estar compuestas por una o varias actividades que en conjunto desarrollan la funcionalidad de la aplicación.

Por lo general, las actividades requieren la interacción del usuario. Para lograr esto, la actividad genera en la pantalla una serie de elementos, como campos de textos, botones, formularios, imágenes, entre otros. Estos elementos reciben el nombre de vistas, y son organizados y parametrizados mediante archivos *XML* (*Extensible Markup Language*) llamados *layouts*. A pesar de que la disposición de las vistas es constantes en los archivos XML, una actividad puede modificar totalmente las vistas y sus atributos a tiempo de ejecución.

Cada vez que una aplicación ejecuta una actividad específica, se genera una pila con las actividades relacionadas. Es decir, solo una actividad puede estar en primer plano a la vez, las actividades predecesoras son empiladas en la memoria del teléfono. Cuando una actividad termina o el usuario presiona el botón atrás se pasa la siguiente en la pila. La figura 3.2 ilustra el comportamiento.

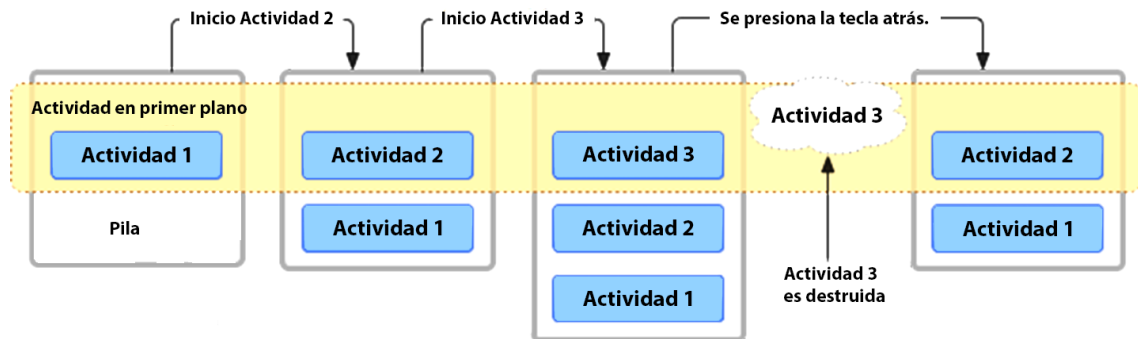


Figura 3.2: Interacción y flujo de Actividades en Android [TBS]

### 3.2.1.2.3. *SQLite*

*SQLite* es un manejador de base de datos relacional de código abierto, usa sintaxis *SQL* (*Structured Query Language*), implementa transacciones y soporta *triggers*. En *SQLite* el

conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo archivo y el requerimiento de memoria durante la ejecución es bastante bajo aprox 250KByte, lo que hace que este manejador sea fácilmente embebido en otros sistemas. Gracias a estas características el sistema Android cuenta con una base de datos *SQLite* integrada y de fácil acceso para las aplicaciones.

### 3.2.1.3. Eclipse

Eclipse es un entorno integrado de desarrollo (*Integrated development environment*, IDE por sus siglas en inglés) gratuito y de código abierto. Está escrito en *Java* con el objetivo de realizar aplicaciones en este lenguaje. sin embargo permite la integración con otros como *C*, *C++*, *Ada*, *Haskell*, *PHP*, *Python*, *Ruby*.

Este ambiente de trabajo es recomendado por Google para la construcción de aplicaciones Android, ya que a través de la instalación de un *plugin* distribuido en el sitio oficial, permite integrar en este entorno el soporte necesario para emplear diversas herramientas específicas para Android que facilitan el desarrollo, cuenta con plantillas para las diversas clases básicas, instrumentos para el análisis del desempeño de la aplicación y facilidades para la gestión de emuladores que permiten simular *smartphones* de diversas características y funciones.

Por ser el entorno oficial de desarrollo cuenta con un amplio soporte y el respaldo de la comunidad, por lo que es el ambiente escogido para el desarrollo de Tuguia.de móvil.

## 3.2.2. Tecnologías Asociadas con Tuguia.de

Durante el desarrollo del proyecto, fue necesario incidir directamente sobre el API de Tuguia.de para agregar funcionalidad, hacer mejoras y ajustes necesarios para garantizar un adecuado desarrollo de la aplicación móvil, las tecnologías implicadas en este proceso serán descritas a continuación.



### 3.2.2.1. Drupal

Drupal es un marco de gestión de contenidos de código abierto escrito en lenguaje *PHP*. En otras palabras, es una interfaz de programación de aplicaciones para crear sistema de gestión de contenidos (*Content Managemement System*, CMS por su siglas en ingles). Una instalación base del sistema o *el núcleo de Drupal* provee los elementos necesarios para el funcionamiento de un CMS básico, sin embargo, la funcionalidad se puede extender mediante la integración y desarrollo de módulos o paquetes [TV10].

El sitio Tuguia.de fue desarrollado con esta herramienta y mediante la instalación, configuración y elaboración de varios módulos fue implementado un API que permite mediante peticiones HTTP (*Hypertext Transfer Protocol*) gestionar el contenido del portal.

### 3.2.2.2. Apache Solr

Apache Solr es una plataforma de búsqueda de alto desempeño para sitios web, es de código abierto, esta escrita en lenguaje *Java* y se basa en la biblioteca *Apache Lucene* [APS]. Apache Solr opera como un servicio separado del servidor web y la base de datos, puede incluso llegar a requerir un servidor dedicado para su operación.

De forma sencilla el funcionamiento de Apache Solr puede ser descrito de la siguiente manera: Mediante archivos *XML* llamados Documentos, se especifican los contenidos que serán indexados para su posterior búsqueda, estos índices pueden ser vistos como una lista de palabras en la que cada palabra en la lista posee una referencia hacia los elementos que la contienen, cuando se realiza una consulta a Apache Solr se ejecuta una búsqueda sobre las listas y se retornan los elementos que cumplan con el criterio establecido en la consulta.

Apache Solr permite realizar búsquedas de texto (*Full-Text Search*) por palabra clave de forma similar al buscador de Google y tiene la capacidad para manejar búsquedas geoespaciales, por lo que en Tuguia.de, es ésta la herramienta elegida para realizar las búsquedas y

es integrada a Drupal mediante un módulo de código abierto ampliamente configurable.

# Capítulo 4

## Marco Metodológico

En empresas TD es costumbre usar la metodología ágil Scrum, y los principios del desarrollo guiado por pruebas y comportamientos (*Test-driven development* y *Behavior-driven development*), por lo tanto, el proceso de creación de la aplicación móvil para el portal Tuguia.de estuvo enmarcado de ésta manera. En este capítulo se presenta y describe esta metodología.

### 4.1. Scrum

Scrum es un marco de trabajo iterativo e incremental para el desarrollo de proyectos y aplicaciones. El desarrollo esta organizado en ciclos de trabajo llamados *Sprints* o iteraciones que son de duración fija y van sucediendo uno detrás de otro. Es importante destacar que sin importar que no se haya terminado el trabajo un *Sprint* nunca se alarga [DPB09]. En el caso de Empresas TD las iteraciones tienen una duración de dos semanas.

### 4.1.1. Roles en Scrum

Dentro de la metodología Scrum existen tres roles principales que debe ser cumplidos para la aplicación efectiva del proceso, estos son el el Dueño de Producto (DP), el Equipo y el ScrumMaster (SM). A continuación se describe cada rol, estas definiciones fueron extraídas de [DPB09].

- El **Dueño del Producto** es el encargado de identificar las funcionalidades del producto y asignarles prioridad de acuerdo a las necesidades del negocio y debe revisar el resultado de cada iteración. Esta persona debe interactuar continuamente con el Equipo y tiene la autoridad final en el proyecto. En el caso de las pasantías el DP era el director general de Tuguia.de.
- El **Equipo** es el encargado de construir el producto, en este caso el pasante. Debe poder autogestionarse y contar con un alto grado de autonomía y responsabilidad ya que al final de cada *Sprint* debe poder tener un producto *entregable*.
- El **ScrumMaster** es el encargado de guiar al equipo y al DP en la aplicación y uso fructífero de Scrum, lo que hace es facilitar el proceso y ser un arbitro garante de la metodología y de que los demás actores ejecuten efectivamente su rol. Esta responsabilidad era de la gerencia e tecnología de Empresas TD.

### 4.1.2. El proceso de Scrum

Como ya fue mencionado la metodología Scrum esta organizada en ciclos de trabajo que van ocurriendo uno tras otro. Estos ciclos están compuestos de actividades tal que se describe a continuación y como se muestra en la figura 4.1. Las definiciones siguientes fueron obtenidas de [DPB09].

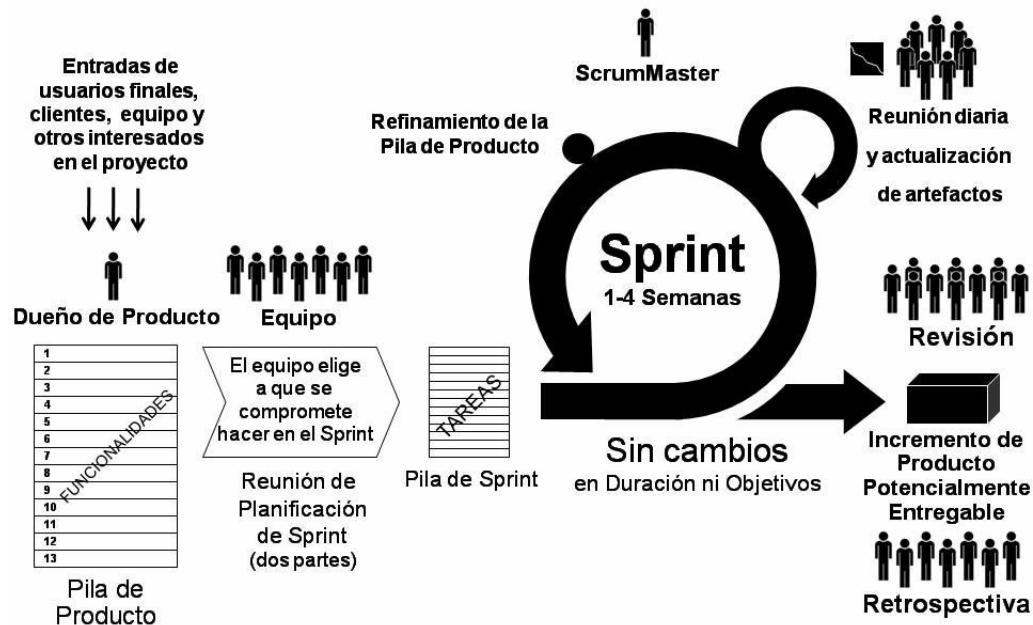


Figura 4.1: Proceso Scrum [DPB09]

- **Planificación de la iteración:** La primera fase consiste en una reunión realizada al comienzo de la iteración donde al comienzo el DP presenta al equipo la lista de requisitos priorizada, se resuelven las dudas y se seleccionan los requisitos que se compromete a completar en la iteración, cada requisito es separado en tareas y se estima la cantidad de esfuerzo en tiempo necesaria para completar cada una función del grado de dificultad de la misma.
- **Ejecución de la iteración:** En esta fase más larga, en ésta se realiza el desarrollo del entregable. Durante esta fase, el equipo se reúne brevemente cada día para informar del progreso y dificultades. Cada miembro del equipo debe contestar únicamente las siguientes preguntas.

- ¿Qué ha hecho desde la última reunión?
  - ¿Qué tiene planificado hacer antes de la siguiente reunión?
  - ¿Qué impedimentos tiene o considera que va a tener?
- **Inspección y adaptación:** Al finalizar el *Sprint* se realiza una reunión, donde el equipo presenta el Dueño de la Aplicación los resultados obtenidos. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el DP realiza las adaptaciones necesarias de manera objetiva sobre la lista de requisitos. Por último el equipo debe analizar cómo ha sido su desempeño y cuales son los problemas que podrían impedirle progresar adecuadamente, tratando de mejorar de manera continua su productividad.

## 4.2. Desarrollo Basado en Pruebas y Comportamientos

El proceso de descrito por la metodología *Scrum* no establece practicas especificas para la construcción del producto, es por esto que en Empresas TD se integran los principios del desarrollo Basado en Pruebas o *Test Driven Development* (TDD) y el Desarrollo Basado en Comportamientos, en inglés *Behavior Driven Development* (BDD), como practicas a seguir durante el fase de programación.

El procedimiento de desarrollo usando TDD se muestra en la figura 4.2.

En primer lugar, se escribe una prueba y se verifica que ésta falla. Luego se implementa el código que hace que la prueba pase satisfactoriamente y seguidamente se refactoriza el código escrito. Por último, se agrega una nueva prueba y se inicia el proceso nuevamente. Esta técnica es utilizada hasta que funcionalidad deseada este completa y funciona adecuadamente.

La forma más efectiva de aplicar esta técnica es segmentar la funcionalidad en pequeñas unidades que puedan ser probadas de manera automatizada e inequívoca, de esta manera

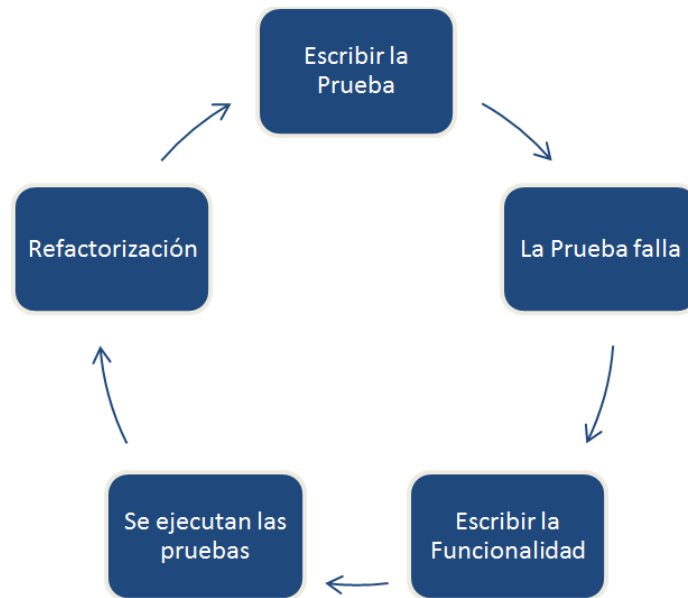


Figura 4.2: Ciclo de *Test Driven Development*

generar código simple y fácil de mantener gracias al proceso de refactorización asociado. Sin embargo aplicar esta técnica por sí sola es complejo, puesto que en ocasiones es difícil determinar qué probar y en qué nivel de detalle.

BDD surge como una manera de subsanar esta dificultad. En lugar basar las pruebas en pequeñas unidades de funcionalidad, BDD establece que las pruebas deben reflejar los comportamientos deseados por los diferentes actores involucrados en el proyecto.

Para que se pueda aplicar BDD de manera adecuada es de vital importancia que todos los involucrados manejen un mismo léxico con respecto al proyecto, que les permita describir los comportamientos deseados en términos comunes sin caer en detalles de implementación. BDD sugiere la utilización de un formato similar al siguiente para especificar los comportamientos [IBD].

Dado: Un contexto inicial.

Cuando: El evento a probar ocurre.

Entonces: Se aseguran que se alcancen los resultados deseados.

Este enfoque permite que las especificaciones no deban ser reescritas ya que el comportamiento pocas veces cambia durante el desarrollo, solo cambia la implementación. Además facilita el proceso de determinar los requerimientos mientras que acelera la documentación.



# Capítulo 5

## Desarrollo del Proyecto

Este capítulo describe, el proceso de investigación, diseño, desarrollo e implementación del proyecto de pasantías. Se realiza una descripción del sistema actual y exponen las distintas tareas que conformaron las iteraciones de trabajo y se presentan los resultados obtenidos en cada una de ellas.

### 5.1. Descripción del Sistema

Antes de iniciar el con el desarrollo de la aplicación móvil se realizó un estudio de Tugua.de con el objetivo familiarizarse con el contenido del sitio y el funcionamiento del API que alimenta la aplicación móvil.

#### 5.1.1. Contenido del Sitio

El contenido de Tugua.de es bastante variado, si bien el sitio esta planteado como una guía de locales, el portal no solo ofrece información sobre estos, también cuenta con contenido que corresponde a otro tipo de establecimientos y servicios, que van desde entidades financieras,

instituciones gubernamentales, rutas de transporte, hasta instituciones educativas.

Los atributos relacionados con cada uno de estos *nodos* de información es la siguiente: nombre, ciudad, urbanización, dirección, coordenadas geográficas, telefono de contacto, página web, correo electronico, cuenta en *Facebook*, cuenta en *Twitter* una o varias imagenes relacionadas, una puntuación que va del uno al cinco; esta puntuación es el resultado de promediar las calificaciones otorgadas por los usuarios en sus reseñas, por último, los nodos poseen una serie de taxonomías que permiten clasificarlos en diferentes categorías.

Mediante la realización de peticiones de tipo HTTP en los servios provistos por el API es posible gestionar los diferentes contenidos del sitio y sus atributos.

A continuación se presenta una imagen de un Local en Tuguia.de.



Figura 5.1: Local visto en Tuguia.de [CTG]

## 5.2. Tareas de Realizadas en Cada Iteración

Antes de iniciar cada iteración, se sostenía una reunión entre el Dueño del Producto, el *ScrumMaster* y el Pasante, donde se seleccionaban las actividades a realizar durante *sprint*. Cada actividad se descomponía en tareas y se realizaba un guión sencillo. A continuación se presenta un guión basado en una actividad realizada en el proceso de desarrollo.

Dado que soy un usuario con la aplicación de TGD en Android.

Debo ver la página home de la aplicación (Login por ahora).

Al ingresar usuario y contraseña.

Debo ver una vista con mi nombre completo.

Tareas:

- Desarrollo de la funcionalidad de "Login".
- Desarrollo y ejecución de las pruebas asociadas.
- Refactoring del código desarrollado.

En las secciones siguientes se realizará una descripción de las tareas realizadas, los objetivos planteados en función de las actividades y los resultados obtenidos en cada una de las ocho iteraciones en las que se desarrolló la aplicación móvil. Aunque no está planteado de forma implícita en los objetivos de cada iteración, en todos los desarrollos se incluye la construcción y ejecución de los casos de prueba.

### 5.2.1. Primera Iteración

Por ser este *spring* el comienzo del proyecto, consistió en una investigación con el fin de adquirir conocimientos acerca del funcionamiento de las aplicaciones móviles Android, así como de las metodologías y herramientas necesarias para el desarrollo del proyecto. Además fue necesario profundizar en el funcionamiento del API y sus características.

#### 5.2.1.1. Objetivos Planteados

A continuación se enumeran los objetivos planteados:

- Adquirir conocimientos del funcionamiento y arquitectura del Sistema Operativo Android y de la estructura de las aplicaciones desarrolladas para esta plataforma.
- Evaluar las herramientas necesarias para el desarrollo del proyecto.
- Investigar los mecanismos para la aplicación de las metodologías TDD y BDD, en el entorno Android.
- Analizar las características prestadas por cada uno de los servicios prestados por el API de Tuguia.de.

#### 5.2.1.2. Resultados Alcanzados

En la sección 3.2.2 referente a las tecnologías asociadas con la aplicación móvil se expone la información recaudada en esta iteración. Luego de analizarla, se preparó en el entorno de trabajo instalando el IDE en la computadora asignada y se construyeron varias aplicaciones simples que reafirmaran los conceptos.

Para la aplicación de las metodologías se evaluaron varias herramientas entre las que destacan el entorno de pruebas provisto en el SDK y *Robolectric* que es un proyecto de código abierto realizado por terceros.

En el SDK están incluidos todos los elementos necesarios para probar cada aspecto que compone a una aplicación Android, gracias a un entorno de pruebas basado en *JUnit*<sup>1</sup>. La ventaja principal de esta herramienta es que al estar embebida no requiere instalar ni aprender ningún complemento externo. Sin embargo, esto trae como desventaja la dependencia de un emulador para su ejecución, lo que retrasa las pruebas en gran medida. *Robolectric* es un

---

<sup>1</sup>*JUnit*: Es un marco de trabajo para el desarrollo de pruebas sobre aplicaciones Java [JUN]

marco de trabajo que subsana esta situación, ya que permite simular el entorno Android y ejecutar los casos de prueba directamente en la maquina virtual de Java, no obstante al ser una biblioteca externa implica un aprendizaje extra.

Dadas estas características, se decidió implementar la metodología usando *Robolectric*, ya que se consideró que el tiempo a invertir en aprender la herramienta es inferior al tiempo que se emplearía ejecutando los casos de pruebas en un entorno de trabajo emulado a lo largo de todo el proyecto.

La última fase de esta iteración consistió en el análisis del API de Tuguia.de. Este servicio se encuentra bajo dos capas de seguridad, la primera es una credencial compuesta por un nombre de usuario y una contraseña que deben ser suministrada en la cabecera de cada petición , esto con el objetivo de que el API no pueda ser accedido por entes externos a la organización. El segundo nivel es la jerarquía de permisos de usuario del sistema de Tuguia.de, de tal forma que un usuario del sistema puede realizar las mismas operaciones tanto en el sitio web como a través del API.

Al momento de iniciar la pasantía los recursos disponibles eran los siguiente:

- buscar local: Este servicio se emplea para hacer consultas sobre locales utilizando varios criterios de búsqueda, de no ser utilizado ningún parámetro de búsqueda, se devolverán todos los locales activos de Tuguia.de.
- buscar comentario: mediante este recurso se pueden obtener los comentarios para uno o mas locales, de no utilizar ningún parámetro de búsqueda, se devolverán todos los comentarios aprobados realizados sobre locales activos de Tuguia.de
- buscar taxonomia: A través de esta funcionalidad se obtienen todos los términos de un vocabulario dado dentro de Tuguia.de, por ejemplo, ciudades, categorías, subcategorías, atributos, etc;

- user: Mediante este elemento se accede a las funcionalidades que permiten el manejo de usuarios, específicamente a *login*, *logout* y registro.
- comentario: Este recurso permite registrar un comentario en Tuguia.de, para su uso es necesario estar conectado como un usuario registrado.
- local: Este servicio permite agregar locales al sitio. Al igual que en recurso anterior para su uso es necesario estar conectado como un usuario registrado.

Cada uno de estos servicios fue probado y a pesar de funcionar adecuadamente en el caso de éxito, no disponían de un manejo de errores para el resto de los casos. Esta situación debía ser solventada puesto que representaba un gran riesgo para la aplicación móvil, por lo que se decidió investigar un poco más a fondo el funcionamiento del API. Afortunadamente, esta interfaz contaba con una documentación lo suficientemente extensa que permitió familiarizarse con los módulos que lo componían sin dificultad. Gracias a esto se agregó un manejo de errores básico sin alterar la planificación.

Las respuesta del API eran en formato *JSON*<sup>2</sup> lo que representó una ventaja gracias a la facilidad existente para manejar este lenguaje. sin embargo, estas respuestas debían ser ajustadas a la funcionalidad deseada en la aplicación móvil. Este proceso de ajuste se pospuso y se realizó a medida que fue necesario y de esta forma invertir el menor tiempo posible fuera de la aplicación móvil.

### 5.2.2. Segunda Iteración

Luego del trabajo realizado en la primera iteración, se poseían los elementos necesarios para iniciar el desarrollo de la aplicación móvil, así que se inició con login, logout, registro y una vista sencilla para ver locales.

---

<sup>2</sup>*JSON: JavaScript Object Notation*, es un estándar para el intercambio de datos ligero y legible [JSO]

#### 5.2.2.1. Objetivos Planteados

En la segunda iteración se plantearon los siguientes objetivos:

- Desarrollar la funcionalidad de *login* y *logout* para los usuarios registrados.
- Implementar un mecanismo de registro para los usuarios nuevos.
- Construir la funcionalidad que permita visualizar una lista de locales. Este mecanismo será usado posteriormente en las búsquedas.

#### 5.2.2.2. Resultados Alcanzados

Por ser esta la primera iteración de desarrollo fue necesario construir un conjunto de funcionalidades secundarias antes de trabajar directamente en los objetivos planteados.

En primer lugar se elaboró un mecanismo *cliente* para realizar peticiones al API que se ajustasen a las políticas de seguridad de este y que pueda recibir las respuestas en el formato JSON. Este mecanismo debe poder ser configurado para generar cualquier tipo de petición.

En segundo lugar se diseñó y construyó un procedimiento para transformar las respuestas provenientes del API a objetos que pudieran ser manejados con mayor facilidad dentro del entorno de la aplicación. Esto fue posible gracias a la biblioteca GSON, que permite transformar un objeto Java en su representación en JSON y viceversa.

Con estas dos herramientas en funcionamiento se desarrollaron los elementos necesarios para alcanzar los objetivos planteados para la iteración siguiendo el esquema planteado en la figura ??

#### Diagrama de desarrollo

En primer lugar se realizó el *layout*, una interfaz sencilla para la funcionalidad en cuestión, no se hace énfasis en la apariencia de éste ya que al finalizar la aplicación, las vistas serán realizadas por el diseñador de Tuguia.de. Luego se revisó a fondo el funcionamiento del servicio

del API con el que se iba a interactuar, con el fin de garantizar su correcto funcionamiento. En tercer lugar se realizó el diseño de la actividad que representaría el comportamiento a implementar. Seguidamente se actualizó el mecanismo cliente para poder realizar las peticiones necesarias para este comportamiento y se extendió el procedimiento para traducir las respuestas en formato JSON a objetos Java de acuerdo a las necesidades. Por último se realizó la construcción de la actividad diseñada.

Dadas las características incrementales de la metodología empleada, este flujo de trabajo fue usado repetidamente cada vez que se agregaba una nueva funcionalidad, adaptándolo a las necesidades según el caso y realizando los respectivos casos de prueba.

En el caso del *login*, *registro* la funcionalidad se desarrolló sin un manejo de flujos alternos adecuado, es decir, no se realizó un manejo para los casos de fallo en los que se requiere la participación del usuario. Por otro lado para esta iteración el mecanismo de visualizar la lista de locales construido operaba con una serie de locales simulados sin conectarse a Tuguia.de. En la lista para cada local se muestra la siguiente información: nombre, puntuación, número de comentarios, ciudad, urbanización, subcategoría y en caso de que no posea categoría y la imagen principal de este local.

Un aspecto importante a resaltar es que la información ofrecida de la imagen por el API es el url (*uniform resource locator*); que no es más que la dirección donde se ubica este recurso en Internet. Para acceder a la imagen como tal es necesario realizar una petición a esta dirección para obtenerla. Gracias a esta situación y a el gran tamaño de las imágenes manejadas en Tuguia.de el consumo de memoria y de la red de datos es bastante alto. Quedó pendiente para futuras iteraciones mejorar esta condición.

## IMAGEN DE LOGIN Y REGISTRO



### 5.2.3. Tercera Iteración

En esta iteración se planteó la realización de búsquedas dada una palabra clave y una ubicación y se tenía pendiente elaborar el manejo de casos de fallo de la iteración anterior.

En este punto del proyecto el módulo del API donde se implementó el manejo de errores, recibió una actualización, por lo que había que instalarla y migrar el manejo de errores como una tarea de mantenimiento.

#### 5.2.3.1. Objetivos Planteados

Se plantearon los siguientes objetivos.

- Actualizar el módulo del API.
- Migrar el manejo de errores.
- Desarrollar la funcionalidad que permita realizar búsquedas dada una palabra clave y una ubicación.
- Completar los casos de fallo pendientes de la iteración dos.

#### 5.2.3.2. Resultados Alcanzados

Luego de investigar los mecanismos para la actualización de módulos en Drupal, se hizo un respaldo del sistema y se procedió a actualizar el módulo en cuestión, este proceso resultó ser bastante automatizado y no presentó mayores inconvenientes. No obstante al inspeccionar el módulo, este había sufrido una refactorización completa, por lo que el proceso de migrar la funcionalidad correspondiente al manejo de errores fue extenso pero se pudo lograr efectivamente.

Para implementar la funcionalidad de búsqueda nuevamente se siguió el esquema de presentando en la fig ??, sin embargo en el API no existía un mecanismo para realizar la búsqueda

dado una ubicación o unas coordenadas geográficas, por lo que sólo se implementó esta funcionalidad usando palabras clave y se integró con el mecanismo de visualizar locales desarrollado en la iteración anterior. El desarrollo necesario para poder realizar las búsquedas dadas una palabra clave y una ubicación o un par de coordenadas se pasan a la siguiente iteración.

La funcionalidades pendientes fueron desarrolladas con éxito. **IMAGEN DE MANEJO DE FALLOS**

#### 5.2.4. Cuarta Iteración

En esta iteración se dejó hizo una pausa sobre el desarrollo de la aplicación móvil y se realizó un análisis de la tecnología ApacheSolr y su integración con Drupal dadas las deficiencias encontradas en la iteración anterior, con el objetivo de poder extender la funcionalidad del API.

##### 5.2.4.1. Objetivos Planteados

- Adquirir conocimientos acerca del funcionamiento Apache Solr.
- Investigar el mecanismo de integración de Drupal y Apache Solr.
- Implementar la funcionalidad necesaria para realizar búsquedas desde el API, dadas una palabra clave y una ubicación.
- Desarrollar los mecanismos necesarios para realizar búsquedas desde el API, dadas una palabra clave, unas coordenada geográfica y un radio.

##### 5.2.4.2. Resultados Alcanzados

En la sección 3.2.2.2 referente a las tecnologías asociadas con Tuguia.de se expone la información recaudada en esta iteración acerca de la herramienta Apache Solr. Luego de

analizarla, se determinó la manera de incluir la funcionalidad deseada en su comportamiento.

Para implementar la búsqueda por palabra clave y una ubicación se accedió a la configuración del módulo de Drupal que permite integrar Apache Solr al sistema. Mediante la interfaz administrativa de este modulo se agregó un campo de texto llamado ubicación en el documento de configuración de Apache Solr.

Luego se desarrolló una extensión para este módulo donde se insertó en el campo ubicación, la información contenida, en los atributos ciudad, urbanización y dirección de un local. Cuando se corre el proceso de generación de índices en Apache Solr mediante la interfaz de configuración observado en la figura ??, el contenido de este campo se segmenta en palabras para que cada una sea tomada como un elemento para la búsqueda.

### **Foto de la configuración de drupal de indexación**

Por último fue necesario agregar un nuevo servicio al API, para lograr esto se desarrolló y configuró un complemento, mediante éste, se hizo posible acceder a la nueva funcionalidad de búsqueda dada una palabra clave y una ubicación. Por ejemplo al pasar las palabras *pizza* y *chacao*, se obtienen todos los locales que en algún atributo tenga la palabra *pizza* y la palabra *chacao* en su dirección, ciudad o urbanización.

Como se determinó en la investigación, Apache Solr posee la capacidad para realizar búsquedas dadas un par coordenadas geográficas. Sin embargo esta funcionalidad no se había configurado en Tuguia.de. Para realizar búsquedas de este tipo fue necesario editar manualmente los archivos de configuración y los documentos de Apache Solr y de esta manera agregar un campo del tipo *location*. Esta configuración se hizo de forma manual y no mediante el módulo como en el caso anterior, ya que éste actualmente no presenta soporte para este tipo de campos.

El proceso seguido a partir de este punto es igual al caso anterior, se agregó la extensión para el módulo de integración de Apache Solr y se agregó un nuevo servicio al API que

permite acceder a la funcionalidad.

### 5.2.5. Quinta Iteración

Con la nueva funcionalidad disponible en el API de *Tuguia.de*, se retoma el desarrollo de la aplicación móvil con el fin de incluir la búsqueda de locales, elemento fundamental de Tuguia.de.

#### 5.2.5.1. Objetivos Planteados

- Modificar a la funcionalidad de búsqueda implementada en la tercera iteración para que adicional a la palabra clave, también reciba ubicación.
- Implementar la funcionalidad necesaria para realizar búsquedas desde el API, dadas una palabra clave, unas coordenada geográfica y un radio.
- Desarrollar un mecanismo de paginación para el resultado de las búsquedas.

#### 5.2.5.2. Resultados Alcanzados

Como la funcionalidad de búsqueda dada una palabra clave fue implementada en la tercera iteración, en este caso sólo fue necesario modificar el cliente para que usase el nuevo servicio elaborado en la iteración anterior.

En cuanto a la búsqueda por coordenadas, se siguió el flujo de desarrollo planteado en la segunda iteración mostrado en la figura ??, gracias a este esquema se concluyó con éxito la funcionalidad.

La paginación se implementó modificando el mecanismo para listar locales implementado en el segundo *sprint*, de tal forma de que se muestra un máximo de veinte locales por página. Si se realizó una búsqueda con más de veinte locales en el resultado, se listan los veinte

primeros y aparece un botón de siguientes al final de la vista, al ser pulsado, realiza la petición al API solicitando los veinte siguientes y así sucesivamente hasta mostrar todos los resultados de la búsqueda. Siempre es posible retornar a la página de procedencia, pulsando el botón anteriores que realiza una petición por los locales que corresponden. Es botón es mostrado sólo si se ha avanzado de la primera página.

## **FOTO DE BUSQUEDA CON PAGINADO**

### **5.2.6. Sexta Iteración**

Continuando con la lista de comportamientos, luego de buscar el local de su preferencia usando los diferentes criterios, la siguiente actividad que haría un usuario es ver la información detallada de un local, sus fotografías y los comentarios asociadas a este. Esas son las tareas que se desarrollaron en esta iteración.

#### **5.2.6.1. Objetivos Planteados**

- Implementar una funcionalidad que permite ver información detallada de un local
- Desarrollar un mecanismo que permita mostrar una lista de comentarios.
- Construir la funcionalidad que muestre un comentario completo.

#### **5.2.6.2. Resultados Alcanzados**

Para la implementación de estas tres funcionalidades se siguió el flujo de desarrollo empleado en las iteraciones anteriores (figura ??) y en los tres casos se lograron desarrollar con éxito. A continuación se detallan los aspectos mas relevantes del desarrollo.

En la vista detallada de un local se muestran la siguiente información: nombre, puntuación, número de comentarios, las categorías a las que pertenece, un botón con el número de

teléfono que funciona como un acceso directo, al presionarlo lleva directamente a la aplicación usada para hacer llamadas marcando este número y la dirección. Además en este *layout* se muestra una lista de cinco comentarios realizados sobre el local y un botón de ver todos los comentarios, que lleva a otra pantalla donde se muestra la lista completa.

En Tuguia.de no existe un límite para la longitud de la reseña de un usuario, por lo que en las listas de comentarios, se acotó el texto de la descripción a dos líneas. si se presiona sobre un comentario, se va a una vista que muestra el contenido completo.

### **IMAGEN DE UN LOCAL, LISTA DE COMENTARIO, COMENTARIO COMPLETO**

#### **5.2.7. Séptima Iteración**

En esta iteración se realizó un desarrollo orientado al manejo de las bases de datos internas de la aplicación, con el fin de registrar los usuarios y los locales vistos. De esta manera poder ofrecer un registro de los locales vistos recientemente por el usuario y la posibilidad de almacenar los locales favoritos, además de un manejo de sesión para el usuario en la aplicación.

##### **5.2.7.1. Objetivos Planteados**

- Crear el modelo de datos necesario para almacenar locales y usuarios en una base de datos de la aplicación
- Construir la funcionalidad que permita almacenar automáticamente los locales recientemente vistos por el usuario.
- Implementar la funcionalidad para que los usuarios pueden almacenar sus locales favoritos.

- Desarrollar un manejo de sesión para que una instancia de la aplicación pueda ser usada por múltiples usuarios.

#### 5.2.7.2. Resultados Alcanzados

Gracias a la base de datos SQLite embebida dentro del sistema operativo Android, se disponía de todos los elementos necesarios para almacenar los locales. Se tomó la decisión de crear dos tablas independientes, una para los locales recientes y la segunda para los favoritos, a pesar de q ambas manejan los mismos atributos.

Esta decisión fue tomada para facilitar el manejo de las operaciones sobre estas ya que a pesar de que manejan la misma información el comportamiento es distinto, tambien se eligió esta opción de almacenar los locales en tablas diferentes, para minimizar el tamaño de la base de datos. En una sola tabla se hubiese requerido una columna extra que identificase los locales recientes y los favoritos, esta columna adicional a la larga podría representar mas espacio, que el costo por mantener dos tablas separadas.

Cuando se visualiza la información detallada de un local este es agregado en la tabla de recientes automaticamente. Ademas en este layout se dispuso de un botón que permite que el usuario identifique un local como favoritos. Para que tener acceso a esta funcionalidad los usuarios deben haber sido autenticados pasando exitosamente por el proceso de *login*. De esta forma todos los locales almacenados en la aplicación son marcados con un identificador único correspondiente al usuario autenticado que este usando la aplicación en ese momento.

Otro aspecto importante de la gestión de sesión que se implementó fue el manejo de *cookies*<sup>3</sup> ya que este mecanismo permite que no sólo se esté autenticado en la aplicación móvil, sino que también a través del envío de un identificador único de usuario en un *cookie*, se puede interactuar con el sitio web como si este nos hubiera autenticado.

---

<sup>3</sup>*Cookies*: pequeños fragmentos de información enviados al servidor en cada petición realizada

### 5.2.8. Octava Iteración

Al ser este la última iteración, se enfocó en mejorar la eficiencia del manejo de recursos en la aplicación y de agregar la funcionalidad que permite realizar comentarios en Tuguia.de.

#### 5.2.8.1. Objetivos Planteados

- Diseñar e implementar la funcionalidad que permita subir comentarios asociados a un local de Tuguia.de.
- Implementar diferentes mecanismos que permitan que la aplicación realice el menor uso posible de la red de datos

#### 5.2.8.2. Resultados Alcanzados

La funcionalidad correspondiente a los comentarios siguió el esquema planteado y fue desarrollada sin problema, este comportamiento solo esta disponible una vez que el usuario esta autenticado y se puede acceder a él, a través de la vista detallada de un local.

A lo largo del desarrollo de la aplicación se idearon diversos mecanismos que permitieron optimizar no sólo el uso de la red, si no que también la memoria que consume la aplicación en ejecución.

La primera mejora es la integración de la biblioteca de manejo de imágenes *Picasso*, ésta ofrece, la posibilidad de realizar diversas transformaciones imágenes, gracias a esto es posible reducir el tamaño de las imágenes recibidas, de tal forma que el espacio ocupado en memoria sea el menor. Además provee un mecanismo de soporte a fallas, cada petición es intentada tres veces antes de descartar el recurso, si el recurso es descartado, se coloca una imagen por defecto y de esta forma no se dejan espacios vacíos en la interfaz. Por último, permite implementar mecanismos de caché en la memoria del teléfono, de esta forma cuando se descarga una imagen, una copia es guardada en la memoria, sí por cualquier razón la



aplicación intenta descargar nuevamente la misma imagen en lugar de esto, es retornada la copia que se encuentra ya en memoria. Como se mencionó anteriormente, las imágenes manejadas en Tuguia.de son de gran tamaño, la suma de los elementos brindados por ésta librería (ajuste de tamaño, caché y soporte de fallos) representó una cambio considerable en el funcionamiento de la aplicación.

Por otro lado se modificó también el comportamiento del paginado, buscando reducir el uso de la red de datos. Se implementó un mecanismo para guardar en archivos de texto las paginas de resultado. De esta forma al moverse entre las páginas, se verifica si existe un archivo que la contenga, antes de hacer una petición al API para obtenerla. Este enfoque no es solo mas eficiente sino que permite un desplazamiento más fluido dentro de la aplicación.

Como ultima mejora se editaron las respuestas del API para cada recurso eliminando todos los campos que no eran necesarios en la aplicación móvil. De esta forma en cada petición sólo se obtiene la información que es necesaria y mantener la cantidad de información transferida via la red al minimo.

## Capítulo 6

### Conclusiones y Recomendaciones

# Bibliografía

- [AAT] Arquitectura de android. <http://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>, consultado el 20 de septiembre de 2013.
- [AND] Android ndk. <http://developer.android.com/tools/sdk/ndk/index.html>, consultado el 20 de septiembre de 2013.
- [APS] Apache solr. <http://lucene.apache.org/solr/>, consultado el 20 de septiembre de 2013.
- [ASD] Get the android sdk. <http://developer.android.com/sdk/index.html>, consultado el 20 de septiembre de 2013.
- [Cen05] Miquel Centelles. Taxonomías para la categorización y la organización de la información en sitios web. *Universitat de Barcelona*, 2005.
- [CTG] El cine restaurant. <http://tuguia.de/local/el-cine-restaurant#.UjpGYVFDtKY>, consultado el 20 de septiembre de 2013.
- [DGT] By 2017 there will almost be as many smartphones as literate adults. <http://www.digitaltrends.com/mobile/2017-literate-adults-smartphones/>, consultado el 20 de septiembre de 2013.

- [DLC] Definición de local. <http://www.definicionabc.com/economia/local.php>, consultado el 20 de septiembre de 2013.
- [DPB09] Larman C Deemer P, Benefield G and Vodde B. *The Scrum Primer*. Agile-Spain, 1 edition, 2009.
- [IBD] Introducing bdd. <http://dannorth.net/introducing-bdd/>, consultado el 20 de septiembre de 2013.
- [JSO] Json. <http://www.json.org/>, consultado el 20 de septiembre de 2013.
- [JUN] Junit. <http://junit.org/>, consultado el 20 de septiembre de 2013.
- [LTG] Lealtag. <http://www.lealtag.com/>, consultado el 20 de septiembre de 2013.
- [NTD] Nielsen tops of 2012: Digital. <http://www.nielsen.com/us/en/newswire/2012/nielsen-tops-of-2012-digital.html>, consultado el 20 de septiembre de 2013.
- [PCM] Definition of:smartphone. <http://www.pcmag.com/encyclopedia/term/51537/smartphone>, consultado el 20 de septiembre de 2013.
- [SCC] Sucucho. <http://sucucho.com/ve/sucucho/nosotros>, consultado el 20 de septiembre de 2013.
- [TBS] Tasks and back stack. <http://developer.android.com/guide/components/tasks-and-back-stack.html>, consultado el 20 de septiembre de 2013.
- [TDC] Tudescuenton.com. <http://www.tudescuenton.com/about/us.php>, consultado el 20 de septiembre de 2013.
- [TGD] Tugua.de. <http://tugua.de/preguntas-frecuentes>, consultado el 20 de septiembre de 2013.

- [TMM] The mobile movement. <http://www.google.com/think/research-studies/the-mobile-movement.html>, consultado el 20 de septiembre de 2013.
- [TV10] T Tomlinson and J VanDyk. *Pro Drupal 7 Development*. Apress, 3 edition, 2010.