



# TESINA DE LICENCIATURA

**Título:** weegoo: una red social turística  
**Autores:** Patricio Mac Adden  
**Director:** Claudia Banchoff - Claudia Queiruga  
**Carrera:** Licenciatura en Informática

## Resumen

*La tecnología avanza velozmente y todos los aspectos de nuestra vida se ven afectados por los avances tecnológicos. En la actualidad es común que al realizar viajes utilicemos dispositivos GPS (**Global Positioning System**), que no sólo nos permite consultarlo a modo de mapa sino que además nos da indicaciones acerca de qué ruta utilizar para llegar a un sitio de la manera más rápida (en tiempo), más corta (en distancia) o incluso más económica.*

*La evolución del hardware se ve estrechamente ligada a la evolución del software (y viceversa). Con la aparición de las primeras computadoras se desarrollaron los primeros programas para aplicaciones específicas. Luego, con la evolución de las computadoras y la conectividad de las mismas, fueron surgiendo los sistemas operativos y las redes de computadoras, que junto con el uso masivo de las computadoras y el auge de internet, permitieron el surgimiento de los llamados medios sociales, como blogs, wikis y redes sociales (Facebook, Twitter, etc.).*

*El objetivo de esta tesina es el desarrollo de una red social orientada al turismo utilizando herramientas open source, que permita a un conjunto de personas interactuar compartiendo lugares y eventos, que pueda ser utilizada con tecnologías web y móviles.*

## Palabras Claves

*Red social, turismo, open source, GPS, dispositivos móviles*

## Conclusiones

*Como resultado del desarrollo de esta tesina ha surgido un prototipo de red social location aware orientada al turismo, weegoo, que tiene la particularidad de ser open source. Dicho prototipo provee dos interfaces totalmente funcionales: una interface web y una interface web adaptada para dispositivos móviles.*

*Una característica única en weegoo (con respecto a las redes sociales location aware evaluadas) es la incorporación de eventos.*

## Trabajos Realizados

*Evaluación de diferentes redes sociales location aware, tales como Gowalla, Foursquare, Google Latitude, Travellerspoint y WAYN.*

*Análisis de herramientas para el lenguaje Ruby y en especial el framework Ruby on Rails.*

*Análisis de herramientas de soporte y metodologías de desarrollo open source.*

*Análisis de las posibilidades de integración del prototipo desarrollado con otras aplicaciones, como Facebook, Twitter, Gravatar, etc.*

*tica open source: weegoo.*

## Trabajos Futuros

*Mantener a weegoo en evolución constante a partir de la comunidad, gracias a su aspecto open source.*

*Mejorar aún más la integración con otras redes sociales, con el objetivo de que más usuarios se sumen a weegoo.*

*Desarrollo de aplicaciones nativas para dispositivos móviles Android e iOS, basados en estudios de usabilidad.*

# Universidad Nacional de La Plata

## Facultad de Informática



### Licenciatura en Informática

#### Tesina de grado

### weegoo: Una red social turística

**Directores** Claudia Banchoff  
Claudia Queiruga

**Alumno** Patricio Mac Adden (7327/5)

# Índice

1 - Introducción.....	3
2 - Contexto.....	5
Un poco de historia.....	5
La evolución de los dispositivos móviles.....	6
Geolocalización.....	13
¿Qué es la geolocalización?.....	13
Principio básico de la geolocalización.....	16
Técnicas de geolocalización.....	16
Social media.....	20
3 - Redes sociales location-aware.....	22
Gowalla.....	22
Foursquare.....	27
Google Latitude.....	32
Travellerspoint.....	35
WAYN.....	37
Tabla comparativa de funcionalidad.....	39
4 - weegoo: una red social turística.....	42
¿Por qué otra red social location-aware?.....	42
5 - Implementación de weegoo.....	44
La especificación de weegoo.....	44
Decisiones de diseño.....	45
¿Aplicación web adaptada para dispositivos móviles o aplicación nativa?.....	45
El desarrollo de weegoo.....	47
Decisiones.....	47
Primera etapa del desarrollo de weegoo.....	56
Segunda etapa del desarrollo de weegoo.....	64
Prototipo de weegoo.....	66
Perspectiva de un usuario no registrado.....	66
Perspectiva de un usuario registrado.....	80
Interface web adaptada para dispositivos móviles.....	89
6 - Resultados.....	98
Uso de GitHub.....	98
¿Cómo contribuir con weegoo?.....	99
¿Es posible instanciar una copia de weegoo?.....	99
Validación del HTML generado por weegoo.....	99
Datos sobre uso de weegoo.....	101
7 - Conclusiones.....	102

# 1 - Introducción

La tecnología avanza muy velozmente y todos los aspectos de nuestra vida se ven afectados por los avances tecnológicos. En la actualidad es común que al realizar viajes utilicemos dispositivos GPS (**Global Positioning System**), que no sólo nos permite consultarlo a modo de mapa sino que además nos da indicaciones acerca de qué ruta utilizar para llegar a un sitio de la manera más rápida (en tiempo), más corta (en distancia) o incluso más económica.

Además de esto, los dispositivos GPS suelen tener una funcionalidad muy útil: búsqueda de puntos de interés a partir de la ubicación actual del dispositivo. Un punto de interés es más que una ubicación, un punto en el mapa. Puede representar una atracción turística, un servicio, alojamiento, etc.. Esto le da una importancia extra, por ejemplo, a la hora de realizar un viaje: siempre que tengamos el dispositivo al alcance de la mano, podremos decidir a dónde ir, o qué lugar nuevo conocer.

Un problema con los dispositivos GPS es que algunos fabricantes cobran por las actualizaciones de la cartografía, lo que puede implicar falta de información en la misma si es que el dispositivo no se actualiza a menudo.

Otro problema que tienen los dispositivos GPS es que no es sencillo agregar información (puntos de interés, rutas, etc.) a la cartografía, lo cual dificulta a los usuarios sin experiencia que puedan realizar esta tarea. Esto significa que si bien a grandes rasgos los dispositivos GPS son grandiosos para la navegación, no todos los aspectos de su uso son óptimos.

Luego, con el avance en el hardware (conectividad 3G, wifi y GPS) sumado al avance en el software de los dispositivos móviles, es posible utilizarlos tanto para la navegación como para la búsqueda de información.

La evolución del hardware se ve estrechamente ligada a la evolución del software (y viceversa). Con la aparición de las primeras computadoras se desarrollaron los primeros programas para aplicaciones específicas. Luego, con la evolución de las computadoras y la conectividad de las mismas, fueron surgiendo los sistemas operativos y las redes de computadoras, lo que, junto con el uso masivo de las computadoras y el auge de internet, permitieron el surgimiento de los llamados medios sociales, como blogs, wikis y redes sociales (tales como Facebook, Twitter, etc.).

El objetivo de esta tesina es el desarrollo de una red social orientada al turismo open source, que permita a un conjunto de personas interactuar compartiendo lugares y eventos, que pueda ser utilizada con tecnologías web y móviles.

En el capítulo 2 presentaré el contexto que motiva esta tesina, tanto en el aspecto tecnológico (nuevas tecnologías, geolocalización) como social (medios sociales).



Luego, en el capítulo 3 evaluaré las diferentes aplicaciones web que aprovechan estas nuevas tecnologías y la geolocalización para socializar su información y en el capítulo 4 describiré los motivos que llevaron al desarrollo de una nueva red social que reúna las características expuestas como objetivo de esta tesina.

En el capítulo 5 describiré el prototipo desarrollado que permitirá utilizar alguna de las tecnologías/APIs estudiadas en un caso concreto de uso y mostraré capturas de dicho prototipo.

En el capítulo 6 presentaré algunos resultados de **weegoo** (<http://weegoo.com.ar>), la red social turística desarrollada en el capítulo 5, y por último, en el capítulo 7, se realizarán unas conclusiones respecto al trabajo realizado.

## 2 - Contexto

En este capítulo se presentan los distintos contextos relacionados a esta tesina. Se abordarán los aspectos tecnológicos, comenzando por un breve repaso de la historia de la informática y los dispositivos móviles, siguiendo con una introducción a la geolocalización y a los medios sociales.

### Un poco de historia

Desde la aparición de las primeras computadoras, la informática evolucionó rápidamente. Las computadoras evolucionaron de tal manera que cada vez son más pequeñas y potentes, mientras que las redes son cada vez más rápidas y las aplicaciones se integran cada vez más entre sí y con nuestro entorno.

Podemos dividir la evolución de las computadoras en 4 distintas generaciones. Cada generación involucra una mejora con respecto a la anterior, ya sea en términos tecnológicos, económicos o de eficiencia.

La primera generación fue la comprendida entre los años 1940 y 1956. Esta generación se denomina la generación de los tubos de vacío debido a que éstos fueron utilizados para los circuitos computacionales. Estas computadoras eran de gran tamaño y su uso era para fines científicos, debido a que eran el medio computacional más rápido de su época. El software utilizado por estas computadoras eran tarjetas perforadas, y el lenguaje que estas computadoras entendían era el lenguaje de máquina. La salida que generaban era mediante impresoras.

La segunda generación, comprendida entre los años 1956 y 1963, introdujo el uso de transistores. Los transistores reemplazaron a los tubos de vacío, que consumían mucha energía y por lo tanto producían mucho calor. Esto trajo como consecuencia una reducción del tamaño, de consumo de energía, y de costo de las computadoras. Si bien las tarjetas perforadas seguían siendo el software de estas computadoras, ya no eran programadas en lenguaje de máquina, sino que en lenguaje ensamblador, lo que hizo que los programas sean más fáciles de escribir.

La tercera generación, comprendida entre los años 1964 y 1971 introdujo los chips de circuitos integrados. Un chip de circuitos integrados consiste de múltiples transistores colocados en un chip de silicio. El uso de estos chips incrementó la velocidad y eficiencia de las computadoras y decremento aún más el tamaño. En esta generación se introduce el teclado y el monitor como medios de interacción. Esto llevó a que las tarjetas perforadas dejen de ser el método para ejecutar programas y que las impresoras sean la única salida. A partir de ésta generación, comienzan a ejecutarse múltiples aplicaciones a la vez, lo que requiere el uso de un sistema operativo. Además, el lenguaje de máquina y el lenguaje ensamblador ya no eran los únicos lenguajes disponibles: comenzaron a usarse los lenguajes de alto nivel para programar las aplicaciones.

La tercera generación de computadoras empezó a ser comercializada masivamente, debido a que eran menos costosa que sus antecesoras.

La cuarta generación comenzó en 1971 y viene hasta la actualidad. En esta generación, los avances en materia de hardware fueron sustanciales:

- Los chips de circuitos integrados son construidos con integración a gran escala (LSI<sup>1</sup>) y con integración a muy gran escala (VLSI<sup>2</sup>), lo que llevó al desarrollo de los microprocesadores, que son chips que contienen millones de transistores.
- Las memorias semiconductoras reemplazan a las memorias magnéticas, mejorando notablemente el acceso a memoria RAM.
- El almacenamiento secundario devino en dispositivos más pequeños en tamaño pero con más capacidad.

Otro avance clave en esta generación fue el de la conectividad. Durante esta generación las computadoras fueron conectadas en redes, lo que llevó al surgimiento de Internet.

En materia de software, durante esta generación fueron desarrollados sistemas operativos como MS-DOS, MS-Windows, etc., y la interacción del usuario con dichos sistemas operativos se empiezan a realizar mediante interfaces gráficas.

A lo largo de la historia, las computadoras han tenido diferentes aplicaciones. En los inicios, las aplicaciones eran más que nada científicas. Luego, con la comercialización en masa de las computadoras (especialmente las PCs), comienzan las aplicaciones comerciales, como por ejemplo manejo de stock de un negocio, administración de clientes, pedidos, etc. y aplicaciones personales, como por ejemplo, manejo de contactos, documentos, etc..

Con la evolución de Internet, estas aplicaciones dejan de ser personales para ser sociales. No sólo una persona puede administrar sus imágenes, videos, documentos, contactos, etc. sino que también permite que otras personas interactúen con ellos y su información. Este tipo de aplicaciones son comúnmente denominadas “redes sociales”.

## **La evolución de los dispositivos móviles [HTML5GEO]**

El teléfono es uno de los más grandes inventos de la historia. Ha revolucionado las comunicaciones, permitiendo que dos personas separadas por grandes distancias puedan comunicarse, con lo cual es considerado indispensable con los negocios, los hogares y los gobiernos. De hecho, es considerado además como el artefacto tecnológico más utilizado en el mundo el día de hoy.

La historia de los dispositivos móviles puede dividirse en 5 grandes eras, en donde cada una describe el tipo de dispositivos móviles que predominaron.

La primera era fue la comprendida entre los años 1973 y 1988. Los dispositivos móviles de esta era no eran más que un teléfono conectado a un receptor de radio del tamaño de una batería de automóvil. Esto se debía a que necesitaban demasiada energía para alcanzar la red de

---

<sup>1</sup> LSI, Large Scale Integration. La cantidad de transistores en un chip de circuito integrado LSI ronda los 10.000.

<sup>2</sup> VLSI, Very Large Scale Integration. La cantidad de transistores de un chip de circuito integrado VLSI ronda los 100.000.

telefonía celular más cercana al lugar, que durante esta época, eran poco y alejados unos de otros. Además, eran muy costosos, con lo cual no todo el mundo podía adquirirlos.

A medida que los fabricantes de tecnología móvil comenzaron a competir en el mercado, se fueron incrementando las torres de telefonía celular para proveer cobertura constante a los usuarios. Mientras más torres se instalaban, menor era la demanda de energía de los dispositivos, con lo cual fue disminuyendo el tamaño de las baterías (y en consecuencia de los dispositivos móviles). A continuación, las ilustraciones 1 y 2 muestran teléfonos celulares de la primera era, el Ericsson Hotline, comercializado entre 1986 y 1992, y el Motorola dynaTAC 8000x, respectivamente.



*Ilustración 1: Ericsson Hotline, comercializado entre 1986 y 1992.*



*Ilustración 2: Motorola  
dynaTAC 8000x,  
comercializado entre 1983 y  
1994.*

La segunda era, comprendida entre los años 1988 y 1998, representó uno de los cambios más significativos en los dispositivos móviles. La mayoría de los teléfonos móviles de esta era eran largos, angostos y de forma rectangular. Al incrementarse la densidad de torres de telefonía celular y decrementarse los requerimientos de energía de los dispositivos móviles el tamaño de los mismos disminuyó considerablemente. Esto, sumado a la competencia de los fabricantes de los dispositivos, contribuyó a que más personas tuvieran acceso a los dispositivos móviles.

Sin embargo, esta era no sólo se caracteriza por los grandes avances en portabilidad a la telefonía móvil, sino que introdujo el servicio conocido como SMS<sup>3</sup>, que inicialmente eran para enviar a los usuarios notificaciones sobre correo de voz u otras notificaciones. El Nokia 2110, que puede verse en la ilustración 3, es un dispositivo típico de esta era.

---

<sup>3</sup> SMS, Short Message Service.



*Ilustración 3: Nokia 2110, en venta a partir de 1994.*

La tercera era, comprendida entre los años 1998 y 2008, no significó un cambio radical en la tecnología de dispositivos móviles, sino que fue considerada una evolución. Hasta este momento, los dispositivos móviles sólo eran capaces de realizar llamados, enviar mensajes de texto y, en algunos casos, jugar algún juego. En esta era, sin embargo, se agregaron aplicaciones y servicios a los dispositivos, como por ejemplo reproductores de audio, cámaras de foto y video, y se introdujo el uso de Internet.

Con la introducción de cámaras en los dispositivos móviles, la demanda por este tipo de dispositivos móviles se incrementó notablemente.

Como se ha dicho, en esta era se introdujo el uso de Internet en los dispositivos móviles. Sin embargo, debido a la falta de buenos navegadores móviles, el uso de Internet en este tipo de dispositivos no fue aprovechado al máximo, motivo por el cual las compañías se enfocaron en la explotación de la venta de ringtones, wallpapers, aplicaciones y juegos. A continuación, la ilustración 4 muestra el Motorola RAZR V3, el dispositivo móvil más popular de esta era.



*Ilustración 4: Motorola RAZR V3. Comercializado desde noviembre de 2004.*

La cuarta era, denominada la era de los smartphones, se solapa con la tercera y quinta era, y se extiende desde aproximadamente el año 2002. Si bien el término “smartphone” es ampliamente utilizado, no está claramente definido cuál dispositivo es un smartphone y cuál no lo es, lo que explica el solapamiento de esta cuarta era con la tercera y la quinta.

Aunque los smartphones tienen las mismas características que los dispositivos móviles de la tercera era, suelen distinguirse de éstos por contar con un teclado QWERTY, una pantalla más amplia, un sistema operativo y algún tipo de conectividad de alta velocidad, como por ejemplo, WiFi.

Lo que muchos fabricantes intentaron durante esta era fue mezclar las características de las computadoras con los teléfonos. La ilustración 5, que se muestra a continuación, muestra el smartphone Motorola Backflip, lanzado en 2010.



*Ilustración 5: Motorola Backflip, lanzado en 2010.*

La quinta y última era (hasta el momento) es la denominada era de los dispositivos móviles táctiles. El año 2007 es considerado el inicio de ésta era, cuando el 9 de Enero de ese año, Steve Jobs presentó el iPhone.

A pesar de que la tecnología que compone el diseño del iPhone ya estaba disponible incluso en otros dispositivos móviles anteriores, el iPhone cambió la forma de ver la tecnología móvil: no es un teléfono, y no es una computadora: es ambas.

Los dispositivos móviles de la era táctil son un medio completamente diferente mediante el cual la gente encuentra nuevas formas de interactuar y entender la información.

Este dispositivo tan revolucionario hizo también que la forma de ver la web cambiara, pronto comenzaron a desarrollarse aplicaciones web adaptadas para dispositivos móviles. A continuación, la ilustración 6 muestra el iPhone, un ícono de la era táctil.



*Ilustración 6: iPhone 4, lanzado en 2010.*

Una de las innovaciones más importantes en el software que pueden ejecutar los dispositivos móviles de esta era es la tienda de aplicaciones en línea, tales como el Appstore de Apple [APPSTORE], el Android Market [MARKET] o el BlackBerry App World [APPWORLD]. Estas tiendas de aplicaciones permiten instalar nuevas aplicaciones o actualizar las ya instaladas, lo que permite extender la funcionalidad de los dispositivos en muchos sentidos. Tanto el Appstore de Apple, el Android Market como el BlackBerry App World permiten la descarga de aplicaciones pagas como de aplicaciones gratuitas.

### **Otros dispositivos móviles**

Junto con los teléfonos móviles de la quinta era (la era de los teléfonos táctiles) surgieron nuevos dispositivos móviles: las tabletas. Las tabletas son dispositivos que tienen las mismas prestaciones que los teléfonos móviles de la quinta era excepto por el incremento en el tamaño de la pantalla y, en la mayoría de los casos, la imposibilidad de realizar llamados telefónicos. En el año 2010, Apple lanza una de las tabletas más populares, el iPad. A continuación, la ilustración 7 muestra el iPad.





*Ilustración 7: iPad de Apple. Lanzado en enero de 2010.*

Luego del lanzamiento del iPad en 2010, otras compañías lanzaron productos similares pero basados en el sistema operativo Android. Tal es el caso de las tabletas Samsung Galaxy Tab o Motorola XOOM, que se muestra a continuación en la ilustración 8.



*Ilustración 8: Motorola XOOM, lanzada en enero de 2011.*

Al igual que los teléfonos móviles de la quinta era, estas tabletas suelen incorporar las tiendas en línea. Para el caso del iPad, la tienda en línea es la Appstore de Apple, mientras que para el caso de las tabletas basadas en Android la tienda en línea es el Android Market.

## Geolocalización

### ¿Qué es la geolocalización?

El término geolocalización se refiere como el mecanismo para determinar la ubicación geográfica de una persona, lugar o cosa con las tecnologías disponibles. [HTML5GEO]

A lo largo de la historia, la geolocalización ha sido posible de diversas maneras. Hace miles de años, la gente confiaba en señales visuales para orientarse. Uno de los mecanismos más primitivos utilizados para ubicarse visualmente fueron las señales de humo. Las señales de humo fueron utilizadas para ayudar a la ubicación y comunicación a través de grandes distancias.

A medida que las culturas avanzaban, también lo hizo el entendimiento sobre las matemáticas. Los primeros navegantes descubrieron la posición del sol y las estrellas con relación a la tierra

y calcularon cómo usar los ángulos de ciertas estrellas fijas, como por ejemplo la Estrella Polar, para realizar cálculos de navegación.

Durante la edad media la navegación cumplió un rol fundamental, principalmente para el comercio entre diferentes partes del mundo. Era necesario recorrer largas distancias sobre el océano, y como consecuencia de ello fue necesario construir instrumentos que permitan obtener la ubicación de forma precisa. Los primeros instrumentos utilizados fueron la brújula y el kamal<sup>4</sup>, siglos más adelante surgieron instrumentos como la vara de Jacob<sup>5</sup>, la cual funciona respetando los mismos principios que el kamal. Eventualmente, la vara de Jacob fue reemplazada por instrumentos más precisos, como el astrolabio<sup>6</sup> o el cuadrante<sup>7</sup>.

A esta altura de la historia, todos los instrumentos de navegación servían para determinar la latitud, pero no había métodos precisos para determinar la longitud. La longitud puede calcularse en virtud de las diferencias de tiempo entre dos desplazamientos. La solución más lógica consiste entonces en portar un cronómetro ajustado a la hora local del meridiano de salida para poder comparar la diferencia horaria entre el origen y el destino. Luego, esa diferencia puede traducirse en grados, a razón de 1° cada 4 minutos (que es lo que resulta de dividir los 360° de la circunferencia entre las 24 horas del día).

Sin una forma precisa de calcular el tiempo, es casi imposible calcular la longitud. A esta altura, los navegantes comenzaron experimentar con relojes de agua<sup>8</sup> y de arena. Hasta la precisión de los relojes actuales y de los cronómetros (a partir del siglo 18), el instrumento utilizado fue el reloj de arena.

A partir del siglo 20, los navegantes comenzaron a usar radios para comunicarse, aunque también para determinar la exactitud de los cronómetros y para determinar la dirección. Esto se logra estableciendo la dirección desde la cual fue transmitida la señal recibida. Cuando la información de la dirección de 2 o más receptores es combinada, la ubicación de la transmisión puede determinarse. Este cálculo es llamado triangulación. La triangulación es el proceso de medir la distancia de una señal recibida usando 2 o más transmisores.

A continuación se muestran dos métodos de triangulación, la triangulación radial (ilustración 9) y la triangulación direccional (ilustración 10).

---

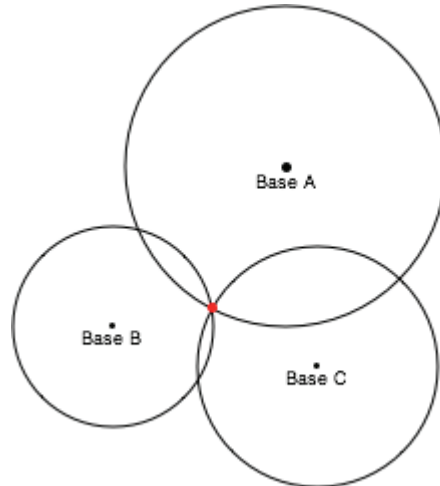
<sup>4</sup> El kamal es un instrumento de navegación que sirve para determinar la latitud.

<sup>5</sup> La vara de Jacob es un instrumento de navegación de apariencia similar a una cruz que sirve para determinar la latitud.

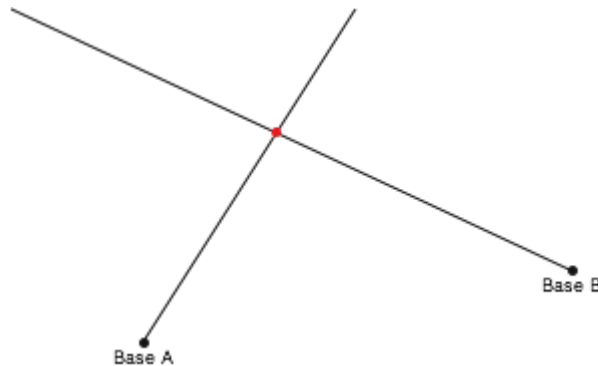
<sup>6</sup> El astrolabio es un instrumento de navegación que sirve para determinar la latitud a partir de la hora local, basándose en la posición de las estrellas.

<sup>7</sup> El cuadrante es un instrumento de navegación que sirve para determinar la latitud, midiendo la altura de la estrella polar (o el sol del mediodía) y para determinar la hora.

<sup>8</sup> Los relojes de agua consisten de una vasija que contenía agua hasta cierto nivel con un orificio en la base, de modo que se vacíe a una velocidad determinada y, por lo tanto, en un tiempo prefijado.



*Ilustración 9: Triangulación radial usando 3 bases. El punto rojo representa al receptor.*



*Ilustración 10: Triangulación direccional usando 2 bases. El punto rojo representa al receptor.*

En 1957, más precisamente el 4 de Octubre, la Unión Soviética lanzó el Sputnik, el primer satélite hecho por el hombre, sobre la órbita de la tierra. Dicho satélite contaba con 2 transmisores de radio, utilizados para medir la concentración de electrones en la ionósfera. Al mismo tiempo, los científicos estadounidenses descubrieron que podían monitorear las transmisiones de radio del Sputnik y, gracias al efecto Doppler<sup>9</sup>, se dieron cuenta que podían saber dónde se ubicaba el satélite en un momento dado.

En las siguientes décadas, los Estados Unidos lanzaron una serie de satélites con el fin de utilizarlos para la navegación satelital, creándose así el Sistema de Navegación Satelital de Defensa<sup>10</sup>, que a partir de 1973 fue renombrado a Navstar. El sistema Navstar fue creado para propósitos militares, hasta que en 1983, el presidente de los Estados Unidos, Ronald Reagan, dispusiera la creación de un sistema de navegación disponible para el uso civil, el Sistema de

<sup>9</sup> El efecto Doppler, llamado así por su descubridor, Christian Andreas Doppler, describe el cambio en la frecuencia de una onda producido por el movimiento relativo de la fuente respecto a su observador.

<sup>10</sup> DNSS, Defense Navigation Satellite System.

Posicionamiento Global, más conocido por su acrónimo en inglés: GPS<sup>11</sup>. Entre 1989 y 1994 se lanzaron 24 satélites para ser utilizados por el proyecto GPS. Desde sus inicios, las señales de mejor calidad fueron reservadas para uso militar, mientras que las de peor calidad eran destinadas para el uso civil, en lo que fue conocido como Disponibilidad Selectiva. En el año 2000, más precisamente el 1ro de mayo, el presidente de los Estados Unidos, Bill Clinton, ordenó que se elimine la Disponibilidad Selectiva, haciendo que el GPS sea mucho más preciso para el uso civil.

Desde 1978 fueron lanzados 59 satélites destinados al uso del GPS, aunque en la actualidad sólo 30 son considerados utilizables.

Como podemos observar, la habilidad de determinar precisamente la ubicación ha ido evolucionando desde hace miles de años. El GPS ha hecho posible la precisión y es considerado como el presente y el futuro de la Geolocalización.

## **Principio básico de la geolocalización**

La posición de un lugar, persona u objeto en el mapa es un punto en el mundo. Dicho punto está compuesto de 2 partes: la longitud y la latitud. Ese punto puede ser utilizado para obtener información acerca del contexto, como por ejemplo, negocios cercanos, otras personas o incluso el nivel de tránsito.

El proceso de obtener las coordenadas geográficas asociadas a la información textual de la ubicación, como puede ser el domicilio o el código postal, es llamada geocodificación. La geocodificación reversa es esencialmente el proceso inverso, obtener a partir de las coordenadas geográficas a partir de la información textual de la ubicación. [HTML5GEO]

## **Técnicas de geolocalización**

En la actualidad existen diversos métodos para obtener información acerca de la ubicación:

- GPS
- Dirección IP
- Id de celda GSM/CDMA

### **GPS**

Los satélites GPS transmiten información que los dispositivos capaces de recibir esas señales (de ahora en más dispositivos GPS) pueden analizar, como por ejemplo la disponibilidad general de la matriz de satélites GPS, donde se encuentran en la órbita de la tierra, el tiempo de transmisión, etc.. El receptor calcula su posición sincronizando las señales enviadas por los satélites, determina el tiempo que tarda en recibir cada mensaje y luego calcula la distancia a cada satélite basada en esta información. Luego, el receptor realiza un cálculo llamado trilateración satelital para obtener su ubicación actual.

---

<sup>11</sup> GPS, Global Positioning System.

La trilateración satelital es el proceso de calcular la distancia de un punto a un conjunto de satélites. Para calcular la posición de un receptor GPS usando 4 satélites, el cálculo se realiza de la siguiente manera:

- El receptor GPS capta la señal de un primer satélite, determinando la distancia entre ambos creando una esfera. De esta forma, el receptor GPS puede estar ubicado en un punto cualquiera dentro de la esfera. La ilustración 11 muestra este primer paso de la trilateración.

*Ilustración 11: Primer paso de la trilateración.*

- El receptor GPS capta la señal de un segundo satélite, determinando de la misma forma la distancia entre ambos creando una segunda esfera. De esta forma, el receptor GPS puede estar ubicado en un punto cualquiera en la intersección de estas dos esferas. La ilustración 12 muestra el segundo paso de la trilateración.

*Ilustración 12: Segundo paso de la trilateración.*

- El receptor GPS capta la señal de un tercer satélite, determinando de la misma forma la distancia entre ambos creando una tercera esfera. De esta forma, el receptor GPS puede estar ubicado en uno de los dos puntos que forma la intersección entre las tres esferas. La ilustración 13 muestra el tercer paso de la trilateración.

*Ilustración 13: Tercer paso de la trilateración.*

- El receptor GPS capta la señal de un cuarto satélite, determinando de la misma forma la distancia entre ambos creando una cuarta esfera. De esta forma, la posición es

determinada por la intersección de ésta cuarta esfera con uno de los dos puntos. La ilustración 14 muestra el último paso de la trilateración.

*Ilustración 14: Cuarto paso de la trilateración.*

Luego, habiéndose creado las 4 esferas (tal cual lo muestra la ilustración 14), puede determinarse que la ubicación del receptor GPS es el punto resultante de la intersección de esas 4 esferas.

### **Dirección IP**

Cada dispositivo conectado a Internet tiene una única dirección IP, lo que permite la comunicación con otros dispositivos también conectados a Internet.

En la mayoría de los casos, una dirección IP es asignada por un Proveedor de Servicios de Internet (ISP<sup>12</sup>), el cual posee bloques de direcciones basados en regiones. Esto quiere decir que es posible identificar, dada una dirección IP, el país, región, e incluso la ciudad de esa dirección IP. En la actualidad, incluso es posible determinar la ubicación geográfica del dispositivo, aunque tal vez con algunos kilómetros de diferencia. Esta diferencia puede deberse a que los datos viajan por diversos dispositivos, como por ejemplo servidores proxy, firewalls, etc..

### **Id de celda GSM/CDMA**

En las redes de telefonía móvil, las regiones geográficas se dividen en celdas. Un id de celda es el número que identifica unívocamente a un dispositivo móvil en una celda. Las 2 redes de telefonía más populares son GSM<sup>13</sup> y CDMA<sup>14</sup>.

---

<sup>12</sup> ISP, Internet Service Provider.

<sup>13</sup> GSM, Global System for Mobile Communications.

<sup>14</sup> CDMA, Code Division Multiple Access.



Utilizando triangulación entre el dispositivo móvil y algunas torres de la red de telefonía puede identificarse la latitud y longitud en donde se encuentra dicho dispositivo móvil, es decir, es posible geolocalizarlo.

## Social media

El término "social media"<sup>15</sup> representa los medios de comunicación que los usuarios pueden participar y contribuir. [APISM]

Los medios sociales suelen basarse en los siguientes principios:

- participación: alentando las contribuciones y reacciones de todos los que estén interesados.
- conversación: mientras los medios tradicionales tienden a comunicar un mensaje (el contenido se transmite o distribuye a una audiencia), los medios sociales son vistos como una comunicación de dos vías.
- comunidad: los medios sociales permiten a las comunidades la comunicación rápida y efectiva. Las comunidades son grupos de usuarios que comparten intereses, como por ejemplo gustos artísticos, deportivos o políticos.
- conectividad: la mayoría de los medios sociales prosperan gracias a su conectividad, esto es, la conexión con diferentes medios sociales, personas y recursos.

La mayor diferencia entre los medios tradicionales, como por ejemplo los diarios, la radio, la televisión, libros, etc. y los medios sociales es la naturaleza dinámica y flexible de estos últimos.

Existen diferentes formas de medios sociales, entre los que se incluyen los blogs, foros, mundos virtuales, wikis, comunidades de contenido y redes sociales.

Un blog es un sitio web actualizado con una cierta frecuencia que contiene artículos de uno o más autores, ordenados cronológicamente.

Los foros son aplicaciones web que permiten discusiones sobre temas específicos.

Las wikis son sitios que permiten que la gente agregue o edite contenido, que es visto como un documento o base de datos público. El ejemplo más conocido de este tipo de medios sociales es Wikipedia<sup>16</sup>.

Las comunidades de contenido son comunidades que organizan y comparten un tipo particular de contenido. Los más populares son Flickr<sup>17</sup>, cuyo contenido principal son las fotografías y Youtube<sup>18</sup>, cuyo contenido principal son los videos, entre otros.

---

<sup>15</sup> social media, medios sociales en español.

<sup>16</sup> <http://www.wikipedia.org>

<sup>17</sup> <http://www.flickr.com>

<sup>18</sup> <http://www.youtube.com>

Las redes sociales son sitios que permiten a la gente construir un perfil personal y conectarse con amigos para compartir contenido y comunicarse. Los ejemplos más populares de este tipo de medio social son Facebook<sup>19</sup>, Twitter<sup>20</sup> y MySpace<sup>21</sup>.

Muchas de las aplicaciones mencionadas de las últimas dos categorías de medios sociales (comunidades de contenido y redes sociales), podrían enmarcarse en más de una categoría. Esto se debe a que las redes sociales como Facebook y MySpace (y hasta cierto punto Twitter) permiten compartir contenido tal como imágenes y videos. Y desde otro punto de vista, las comunidades de contenido tales como Youtube y Flickr permiten la comunicación entre sus usuarios.

---

<sup>19</sup> <http://www.facebook.com>

<sup>20</sup> <https://twitter.com>

<sup>21</sup> <http://www.myspace.com>

## 3 - Redes sociales location-aware

En la actualidad las redes sociales ocupan un lugar importante en la vida de los usuarios de Internet. Esto se debe a que les proveen las siguientes características:

- permanecer en contacto constante con sus amigos
- información sobre actualidad e intereses
- entretenimiento, mediante juegos, videos, imágenes, etc.

Con el auge de redes sociales como Facebook y Twitter, muchas redes sociales enfocaron su funcionamiento en la ubicación de sus usuarios, haciendo este aspecto la principal característica de la red social.

En este capítulo se evaluarán diferentes redes sociales location-aware. De cada aplicación se analizarán los siguientes aspectos:

- Sus características principales: con qué objetivo fueron creadas, cómo funcionan, etc.
- Su integración con otras redes sociales
- La disponibilidad o no de una aplicación móvil para acceder a la misma.
- La cantidad de usuarios que la utilizan
- Por último se evaluará si se trata de una aplicación es open source. Esta es una característica importante, debido a que de ser open source, la aplicación podrá ser extendida (o podrán solicitarse nuevas características a quienes la mantengan) o instalada para ser usada de manera privada.

### Gowalla

Gowalla [GOWALLA] es una red social creada con la finalidad de que los usuarios registrados indiquen dónde se encuentran actualmente. Al proceso de indicar dónde un usuario se encuentra, en Gowalla se lo denomina *check in*.

Gowalla cuenta con una interface de usuario web (mediante la cual los usuarios pueden registrarse, buscar lugares, conectarse con otras redes sociales para encontrar amigos, etc.), una interface web adaptada para dispositivos móviles (mediante la cual puede iniciar sesión, check in en lugares, etc.) y aplicaciones nativas para dispositivos móviles (las cuales tienen básicamente la misma funcionalidad que la interface web). Más adelante, la ilustración 15 muestra una captura de la página de inicio de Gowalla.

Una particularidad con respecto a los check in es que no pueden realizarse desde la versión normal de Gowalla, sino que desde la versión adaptada para dispositivos móviles o las aplicaciones nativas para dispositivos móviles (Android, iPhone, BlackBerry y PALM).

*Ilustración 15: Pantalla inicial de Gowalla.*

En Gowalla, cada usuario registrado posee un Passport, que no es más que una colección de Stamps. Estos Stamps representan a los lugares que el usuario ha visitado. El Passport y los Stamps pueden relacionarse al pasaporte usado para hacer viajes internacionales y a los sellos de cada aeropuerto, respectivamente (sólo que en vez de aeropuertos se trata de lugares). A continuación, la ilustración 16 muestra una captura del Passport de Gowalla.

*Ilustración 16: Passport de Gowalla.*

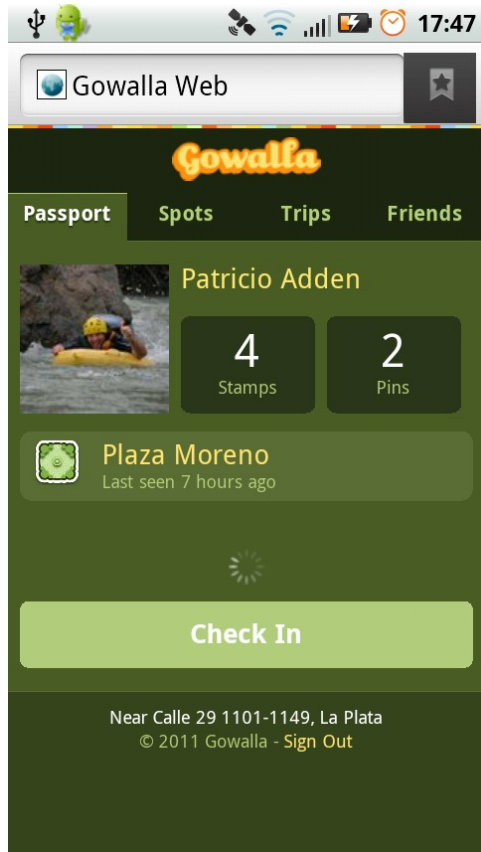
Un usuario de Gowalla puede conectarse con otros usuarios de Gowalla, ya sea mediante el buscador de usuarios de Gowalla o a través de la búsqueda en otra aplicación externa, como por ejemplo Facebook, Twitter, Gmail, Yahoo o Hotmail. Luego, puede compartir los lugares donde hace check in y las fotos tomadas en esos lugares, además de comentar los check ins de los otros usuarios. Para hacer esto posible, el usuario debe conectar su perfil de Gowalla con su perfil en Facebook, Twitter, Gmail, Yahoo o Hotmail. Una vez conectados los perfiles, el usuario podrá conectarse con otro usuario que esté usando esa aplicación externa y Gowalla o invitar usuarios de esa aplicación externa. A continuación se muestra, en la ilustración 17, la pantalla de búsqueda de amigos en Gowalla.

*Ilustración 17: Pantalla de búsqueda de amigos en Gowalla.*

Una característica interesante de Gowalla son los **Trips** (viajes, salidas), que son simplemente conjuntos de lugares. Una vez que un usuario hace check in en todos los lugares de un Trip, se le otorga un Pin, que también es agregado al Passport para recordarle los logros conseguidos. Además, algunos lugares contienen souvenirs digitales, que en algunos casos pueden ser intercambiados por premios reales, como por ejemplo entradas de cine, ropa, etc.. En la siguiente ilustración (ilustración 18) se muestra la pantalla de Trips de Gowalla.

*Ilustración 18: Pantalla de trips de Gowalla.*

En cuanto al aspecto móvil, Gowalla ofrece una versión web diseñada específicamente adaptada para ser usada por dispositivos móviles y aplicaciones nativas para los dispositivos Android[GOWALLAANDROID], iPhone[GOWALLAITUNES], BlackBerry y PALM. A continuación se muestran las ilustraciones 19 y 20, que muestran la versión web adaptada para dispositivos móviles y la aplicación nativa para android, respectivamente, de Gowalla.



*Ilustración 19: Versión adaptada para dispositivos móviles.*

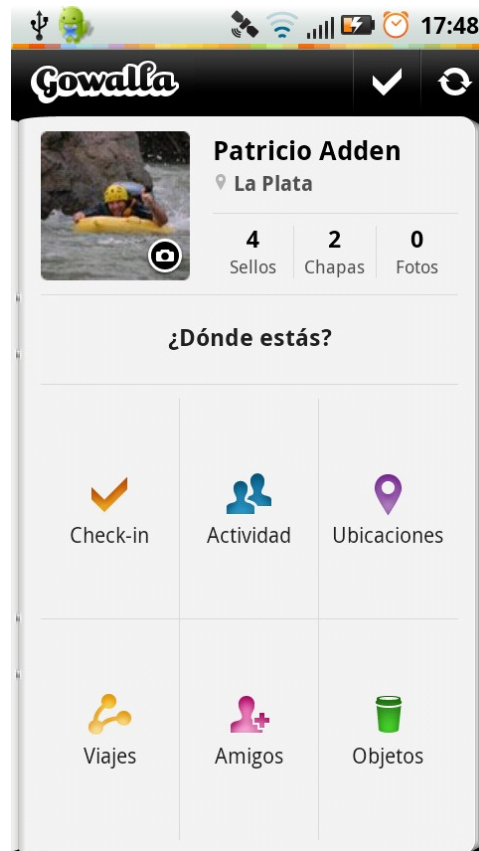


Ilustración 20: Gowalla para Android.

Actualmente, cerca de 3 millones de personas usan Gowalla.

Gowalla no es open source. Esto significa que no hay forma de modificar el código fuente de la aplicación (ni la web ni la móvil adaptada para dispositivos móviles ni la nativa para dispositivos móviles) para corregir errores o agregar funcionalidad.

## Foursquare

Foursquare [FOURSQUARE] también es una red social que sigue los mismos principios básicos que Gowalla, es decir, basa su funcionamiento en el check in de sus usuarios. En la siguiente ilustración, la ilustración 21, se muestra una captura de la pantalla principal de Foursquare.



*Ilustración 21: Pantalla principal de Foursquare.*

A diferencia de Gowalla, Foursquare no tiene lo que se llama Passport, pero si tiene Badges, que son lo que en Gowalla se denominan Stamps. Los Badges son “premios” por hacer check in en ciertos lugares, o hacer una cantidad determinada de check ins en diferentes lugares, o incluso hacer check in tantos días consecutivos. A continuación, la ilustración 22 muestra la pantalla de Badges de Foursquare.

*Ilustración 22: Pantalla que muestra los Badges conseguidos.*

En Foursquare, cualquier usuario puede dejar sus Tips (consejos para otros usuarios) y subir fotos en los lugares. Además, al usuario que más veces ha hecho check in en un lugar se lo

denomina Mayor del lugar, que es una mención especial. A continuación, la ilustración 23 muestra un lugar en Foursquare, donde se puede ver comentarios, el Mayor y fotos del mismo.

*Ilustración 23: Un lugar en Foursquare. Muestra los tips, el Mayor y las fotos del mismo.*

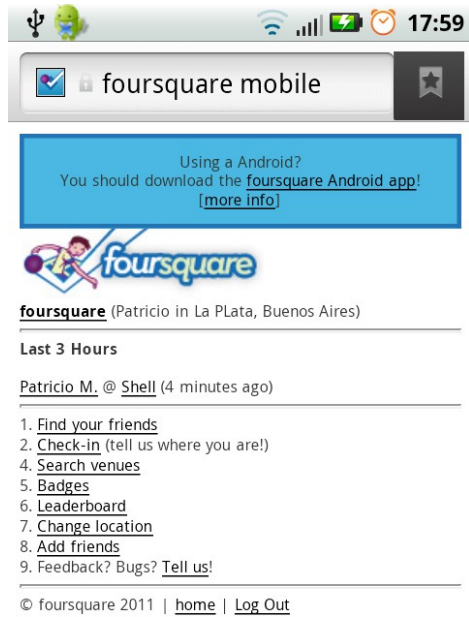
Foursquare no posee Trips como Gowalla, sin embargo hay un concepto similar que son los **TO-DO** lists, es decir, una lista de lugares por visitar. A continuación se muestra, en la ilustración 24, la pantalla de Foursquare que muestra las TO-DO lists de un usuario.

*Ilustración 24: Pantalla que muestra las listas de un usuario.*

De manera similar a Gowalla, Foursquare permite conectar al perfil de Foursquare con el perfil de otra aplicación externa, como Facebook, Google, Twitter y Yahoo! con el fin de encontrar usuarios de estas aplicaciones que ya estén usando Foursquare. Además de la búsqueda de usuarios, la conexión con otra aplicación externa permite que los check ins, obtención de Badges, etc.. La siguiente ilustración (ilustración 25) muestra la pantalla de Foursquare que permite conectarse con otras aplicaciones.

*Ilustración 25: Conexión con otras aplicaciones.*

Al igual que Gowalla, Foursquare no permite hacer check in en su aplicación web normal, sino que mediante su aplicación web adaptada para dispositivos móviles o mediante sus aplicaciones nativas para Android[FOURSQUAREANDROID], iPhone[FOURSQUAREITUNES], BlackBerry, PALM, etc.. En las siguientes 2 capturas, ilustraciones 26 y 27, se muestran la versión web adaptada para dispositivos móviles y la aplicación nativa para android de Foursquare, respectivamente.



*Ilustración 26: Versión adaptada para dispositivos móviles.*



Ilustración 27: Aplicación para Android.

Para Junio de 2011 Foursquare alcanzó los 10 millones de usuarios. [FOURSQUARE10M]

Foursquare tampoco es open source. Esto significa que no hay forma de modificar el código fuente de la aplicación (ni la web ni la móvil adaptada para dispositivos móviles ni la nativa para dispositivos móviles) para corregir errores o agregar funcionalidad.

## Google Latitude

Google Latitude [LATITUDE] nació como una aplicación totalmente diferente a Gowalla y a Foursquare. En sus inicios, el propósito de Google Latitude era simplemente averiguar la ubicación de los contactos. Cada usuario podía compartir su ubicación, permitiendo controlar la privacidad. En la actualidad, con la velocidad con la que Google mejora sus productos, se agregó el check in automático en distintos lugares (aunque sólo para el cliente nativo de Android). La siguiente captura, ilustración 28, muestra la pantalla principal de Google Latitude, donde se puede observar la ubicación actual del usuario y sus contactos.

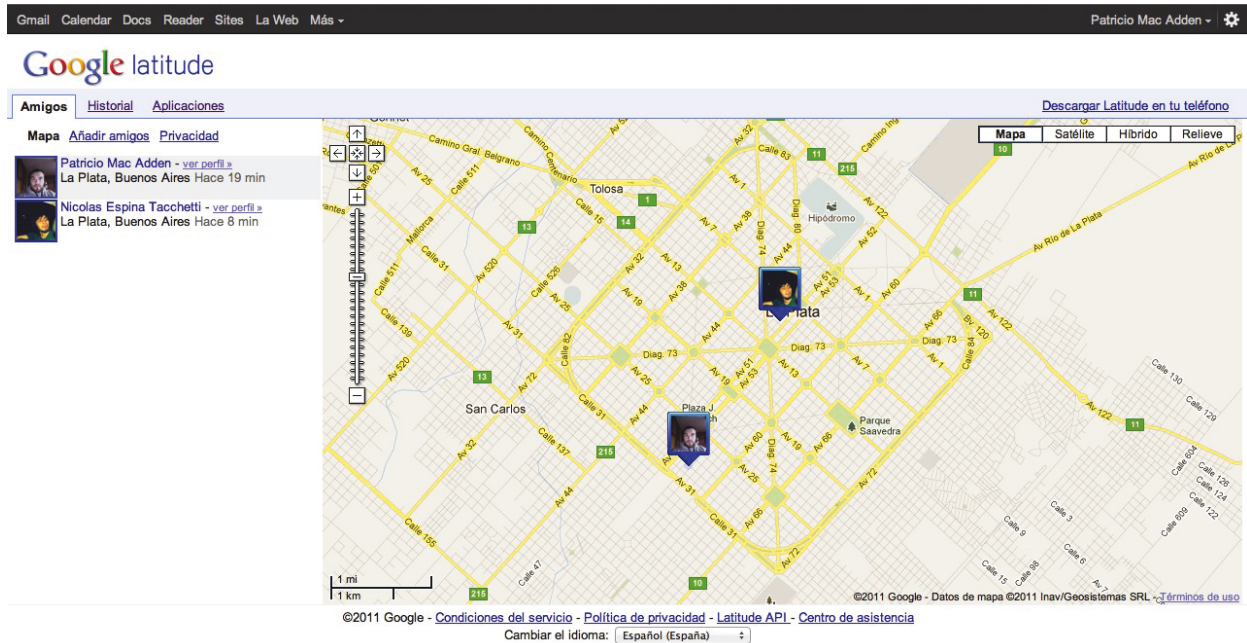


Ilustración 28: Pantalla principal de Google Latitude. Muestra la ubicación actual del usuario y de sus contactos.

Google Latitude no sólo consta en indicar dónde se encuentra el usuario en un momento determinado, sino que también le da al usuario la posibilidad de llevar un historial de las ubicaciones donde ha estado y llevar un control de la cantidad de kilómetros recorridos. A continuación se muestra, en la ilustración 29, el historial de ubicaciones del usuario en Google Latitude.

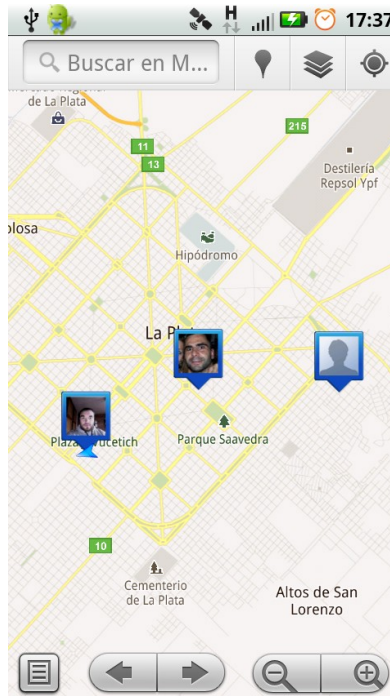
Ilustración 29: Historial de movimientos en Google Latitude.

A diferencia de Gowalla y Foursquare, Google Latitude no permite la conexión con otras aplicaciones externas. Esto significa que las personas con las cuales se puede compartir la información son aquellos que forman parte de los contactos de Google.

Google Latitude no posee una versión web adaptada para dispositivos móviles, pero, al igual que Gowalla y Foursquare, provee clientes nativos Android [LATITUDEANDROID], iPhone [LATITUDEITUNES], BlackBerry, Windows mobile, etc.. A continuación se muestran dos capturas, la primera (ilustración 30) muestra la versión adaptada para dispositivos móviles de Google Latitude, donde se indica que se descargue la aplicación nativa, mientras que la segunda (ilustración 31), muestra la aplicación nativa para android.



*Ilustración 30: Versión adaptada para dispositivos móviles.*



*Ilustración 31: Aplicación para Android de Google Latitude.*

Para febrero de 2011, Google Latitude alcanzó los 10 millones de usuarios. [LATITUDE10M]

Google Latitude es una aplicación de Google de código privativo. Esto significa que no hay forma de modificar el código fuente de la aplicación (ni la web ni la móvil adaptada para dispositivos móviles ni la nativa para dispositivos móviles) para corregir errores o agregar funcionalidad.

## **Travellerspoint**

Travellerspoint [TRAVELLERS] es una red social que funciona de una forma totalmente diferente a las redes sociales evaluadas anteriormente. No se basa en el check in de sus usuarios en la ubicación donde se encuentran, sino que es enriquecer la experiencia turística ofreciendo numerosos servicios, contenido y productos con un valor significativo para los turistas de todo el mundo. [TRAVELLERSABOUT]

A continuación, se muestra la pantalla principal de Travellerspoint, en la ilustración 32.



*Ilustración 32: Pantalla principal de Travellerspoint.*

Al igual que las redes sociales evaluadas anteriormente, Travellerspoint permite a sus usuarios conectarse con otros usuarios. Estos usuarios pueden compartir fotos, guías de viaje, comunicarse a través de foros, escribir en blogs personales, planear viajes, etc..

Una funcionalidad interesante que posee Travellerspoint es “Plan your journey”, que permite a un usuario crear mapas de los viajes realizados (o por realizar), indicando cada uno de los lugares donde se efectuaron (o efectuarán) las paradas. Desafortunadamente, a estos viajes no se pueden agregar puntos de interés como atracciones, restaurantes, alojamiento, etc. visitados (o por visitar). Además, las rutas que dibuja entre los lugares visitados (o por visitar) no son rutas reales. A continuación se muestra una captura con la pantalla “Plan your journey”, en la ilustración 33.

*Ilustración 33: "Plan your journey"*

Travellerspoint no tiene una versión adaptada para dispositivos móviles ni una aplicación nativa para clientes Android, iPhone ni BlackBerry.

Travellerspoint permite que sus usuarios inicien sesión de la forma tradicional (usuario y contraseña) o mediante Facebook. Además, permite a sus usuarios que compartan el contenido de la red social con sus contactos de otras redes sociales como Facebook, Twitter, entre otras.

Para Agosto de 2011, Travellerspoint tiene 413.032 usuarios.

Travellerspoint no es open source. Esto significa que no hay forma de modificar el código fuente de la aplicación para corregir errores o agregar funcionalidad.

## **WAYN**

WAYN<sup>22</sup> [WAYN] es la red social sobre turismo y estilo de vida de mayor crecimiento en el mundo. [WAYNABOUT]

La siguiente captura (ilustración 34) muestra la página de inicio de WAYN para un usuario registrado.

---

<sup>22</sup> WAYN, Where Are You Now?

*Ilustración 34: Página de inicio de usuario de WAYN.*

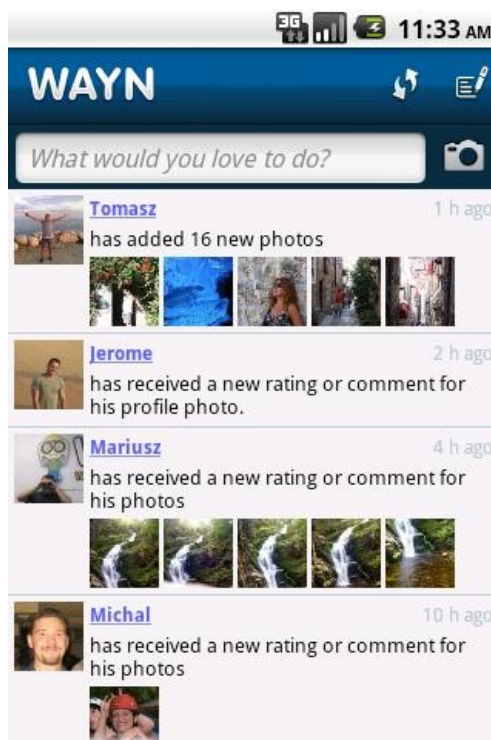
El objetivo principal de WAYN es que sus usuarios se conecten entre sí y compartan experiencias y actividades. Un usuario puede indicar a los lugares que ha ido y a los que quiere ir y en base a esta información se le sugiere conectarse con gente del lugar o que se encuentre allí, ver fotos del lugar y ver que actividades se desarrollan allí. A continuación, la ilustración 35 muestra una captura WAYN, donde dado un lugar, se sugieren cosas para hacer.

*Ilustración 35: Cosas para hacer en Ciudades.*

Sin embargo, WAYN no hace uso de la tecnología disponible para aprovechar al máximo la información que maneja.

WAYN permite el inicio de sesión con Facebook y Twitter, y a partir de estas conexiones permite invitar a contactos de estas redes sociales para que se unan a WAYN y en algunas ocasiones publicar información en estas redes sociales.

WAYN no provee una versión adaptada para dispositivos móviles, pero si una aplicación Android [WAYNANDROID] y iPhone [WAYNITUNES]. A continuación se muestra, en la ilustración 36, la aplicación nativa para Android de WAYN.



*Ilustración 36: Aplicación para Android.*

Para Agosto de 2011, WAYN tiene más de 17 millones de usuarios.

WAYN no es open source. Esto significa que no hay forma de modificar el código fuente de la aplicación (ni la web ni la nativa para dispositivos móviles) para corregir errores o agregar funcionalidad.

## **Tabla comparativa de funcionalidad**

Según el análisis realizado en las secciones anteriores, existen varias redes sociales location-aware de características similares que brindan distintas funcionalidades a sus usuarios. La

siguiente tabla muestra una comparación entre todas las redes sociales de manera tal de poder analizar ventajas y desventajas de cada una para justificar el desarrollo de weegoo.

Analicé los aspectos que consideré más destacados que conciernen a la motivación de esta tesina:

	Gowalla	Foursquare	Google Latitude	Travellerspoint	WAYN
Check in en lugares	Sí	Sí	Sí	No	No
Conexión con otras redes sociales	Sí	Sí	No	Sí	Sí
Ver/agregar lugares visitados con anterioridad al uso de la red social	No	No	No	Sí	Sí
Planificación de viajes	No	No	No	Sí	Sí
Uso de geolocalización	Sí	Sí	Sí	No	No
Manejo de puntos de interés	Sí	Sí	Sí	No (maneja ciudades)	No (maneja ciudades)
Aplicación web adaptada para dispositivos móviles	Sí	Sí	No	No	No
Aplicaciones nativas para Android, iPhone, BlackBerry, etc.	Sí	Sí	Sí	No	Sí
Cantidad de usuarios	3 millones (aproximada)	10 millones (aproximada)	10 millones (aproximada)	413.032	17 millones

	mente)	mente)	mente)		
¿Es open source?	<b>No</b>	<b>No</b>	<b>No</b>	<b>No</b>	<b>No</b>

Como se puede observar, ninguna de las aplicaciones analizadas es open source, por lo tanto ninguna cumple con uno de los aspectos deseados más importantes.

## 4 - weegoo: una red social turística

Como mencioné anteriormente, el objetivo de esta tesina es el desarrollo de una red social orientada al turismo que permita a un conjunto de personas interactuar compartiendo lugares y eventos de una cierta ciudad o región turística.

### ¿Por qué otra red social location-aware?

Habiendo evaluado las diferentes redes sociales location-aware existentes, llegué a la conclusión de que ninguna reúne todas estas características deseadas:

- **orientada al turismo**
- uso de geolocalización
- manejo de puntos de interés
- **manejo de eventos**
- versión móvil o aplicación nativa para dispositivos móviles
- **y fundamentalmente, ser open source, lo que la hace extensible**

Tanto Gowalla, Foursquare como Google Latitude se basan en el concepto de **check in** en ubicaciones (Spots en el caso de Gowalla, Venues en el de Foursquare y en el caso de Google Latitude simplemente un punto en el mapa), con lo cual dejan de lado el aspecto turístico, ya que básicamente constan en saber en qué ubicación se encuentra actualmente el usuario que hace check in.

Un aspecto fundamental en el turismo es poder saber qué lugares han sido visitados y qué lugares quedan pendientes de visitar. Al ser basados en check in, ni Gowalla, ni Foursquare, ni Google Latitude pueden indicar qué lugares han sido visitados con anterioridad al comienzo del uso de la red social. Si bien Gowalla y Foursquare manejan Trips y TO-DO lists de Spots y Venues respectivamente, no siempre permiten que se sepa qué lugares quiere visitar un usuario. En el caso puntual de los Trips de Gowalla, no sirven para indicar que un usuario quiere visitar ese conjunto de Spots, sino que se obtiene una recompensa al hacer check in en la totalidad de los Spots del Trip. En cambio, Travellerspoint y WAYN, permiten crear Trips, que son similares a los Trips de Gowalla, con la salvedad de que estos últimos no manejan puntos de interés, sino que manejan ciudades. Tanto Travellerspoint como WAYN permiten que los usuarios creen Trips ya realizados o planear nuevos.

Yendo un poco más lejos, uno podría deducir que tanto Gowalla como Foursquare se basan en la competencia entre sus usuarios para ver quién ha visitado más lugares, quién ha visitado más veces un lugar y quién ha obtenido más premios. En el caso de Google Latitude, uno podría deducir se basa simplemente en tener conocimiento de dónde se encuentran los contactos de un usuario a todo momento.

Es un hecho que tanto Gowalla como Foursquare permiten agregar comentarios e imágenes en los Spots y Venues respectivamente, pero en la práctica esta información no es relevante, ya que el común de los usuarios no utilizan estas redes sociales para planear viajes, sino para contar a sus contactos dónde se encuentran.

Con esto se quiere decir que los usuarios de estas redes sociales no se utilizan con fines turísticos, sino que con fines competitivos o con el fin de indicar a sus amigos donde se encuentran actualmente.

Con respecto al aprovechamiento de la geolocalización, sólo Gowalla, Foursquare y Google Latitude hacen uso de tecnologías que permiten conocer la ubicación del usuario.

La única red social que no posee ni versión web adaptada para dispositivos móviles ni aplicación nativa para dispositivos móviles es Travellerspoint, mientras que únicamente Gowalla y Foursquare posee ambas versiones. Google Latitude y WAYN tienen únicamente aplicaciones nativas para dispositivos móviles.

A la hora de hacer turismo, no sólo los lugares son importantes. Cuando uno realiza un viaje, probablemente también quiera aprovechar los eventos sociales de la ciudad o región turística donde se encuentre, como pueden ser eventos deportivos, funciones de teatro, recitales, etc.. Ninguna de las redes sociales evaluadas provee la funcionalidad de poder ver qué eventos ocurrirán dependiendo del lugar donde se encuentre el usuario.

Por último, ninguna de estas redes sociales es open source, por lo tanto es imposible agregar la funcionalidad suficiente que haga que cumplan con todos los puntos enumerados anteriormente. Esto quiere decir que si quisiera extender alguna de estas aplicaciones para incluir, por ejemplo, la publicación de eventos, no podría debido, justamente, a no contar con el código fuente ni la licencia de uso que permita hacerlo.



## 5 - Implementación de weegoo

weegoo es una red social orientada al turismo, donde los usuarios registrados puedan relacionarse de manera similar a las redes sociales más populares hoy en día (Facebook, Twitter e incluso las analizadas en el capítulo 3), pero haciendo énfasis en compartir tanto lugares turísticos como eventos, no sólo marcar check in en los lugares que se encuentren (como en el caso de Gowalla, Foursquare y Google Latitude).

Al momento de finalizada esta tesina, la aplicación está funcionando en <http://weegoo.com.ar>. Y el repositorio donde se encuentra el código fuente de la misma se encuentra en: <https://github.com/patriciomacadden/weegoo>.

### La especificación de weegoo

weegoo es una red social open source que permite que los usuarios interactúen compartiendo información turística. Esta información turística se ve representada por lugares y los eventos que allí ocurren. Los usuarios pueden indicar qué lugares o eventos han visitado y cuáles planean visitar. A su vez, los lugares y eventos son enriquecidos con comentarios e imágenes de los usuarios, con el fin de que otros usuarios decidan a qué lugar o evento visitar.

Los usuarios de weegoo pueden registrarse completando un formulario de registración con información personal o bien mediante Facebook o Twitter. La única diferencia entre registrarse completando el formulario de registración o mediante Facebook o Twitter es que el registro mediante Facebook o Twitter no requiere contraseña (aunque si solicita algunos de los datos personales del formulario de registración), ya que el inicio de sesión se realizará mediante Facebook o Twitter.

Luego, los usuarios de weegoo pueden buscar amigos en la red social, con el fin de conectarse entre sí. La búsqueda puede realizarse mediante un buscador o mediante la conexión con Facebook o Twitter. En este caso, se buscarán aquellos usuarios que estén en weegoo y Facebook o Twitter. Una vez que se encuentra un amigo usando cualquiera de las dos formas, el usuario puede solicitar su amistad. Entonces, el usuario que solicita la amistad espera a que el otro usuario acepte o rechace dicha solicitud.

Un aspecto importante en weegoo es su aspecto móvil. Debido a que weegoo es una red social turística, debe poder ser accedido desde dispositivos móviles. Esto se pensó principalmente para las situaciones donde el usuario se encuentra recorriendo ciertos lugares y desea saber qué puntos de interés se encuentran cerca.

Por último, weegoo debe ser open source. De esta forma, los usuarios de weegoo podrán sugerir nuevas características, reportar errores (si es que hubiere) e inclusive contribuir con parches que agreguen características o corrección de errores. Además, al ser open source permitiría la instalación para cualquier tipo de uso, ya se para uso regional (es decir, para

agregar lugares/eventos de una región en particular) o para uso privado (por ejemplo, para una organización, grupo de amigos, etc)..

## **Decisiones de diseño**

### **¿Aplicación web adaptada para dispositivos móviles o aplicación nativa?**

Básicamente, una aplicación web adaptada para dispositivos móviles es una aplicación web especialmente optimizada para ser usada en dispositivos móviles. Obviamente, esta aplicación web no se instala en los dispositivos móviles, sino que es accedida a través de un navegador web mediante el protocolo HTTP y puede estar desarrollada usando cualquier lenguaje.

Por el contrario, las aplicaciones nativas son instaladas en el dispositivo móvil y tienen acceso al hardware del dispositivo (acelerómetro, cámara, GPS, etc.) y deben ser desarrolladas en algún lenguaje en particular, generalmente definido o limitado por la plataforma. En el caso de las aplicaciones nativas para dispositivos Android, las aplicaciones nativas, por lo general se escriben en Java [ANDROIDSDK], mientras que en el caso de las aplicaciones nativas para dispositivos iPhone deben ser escritas en ObjectiveC [IPHONESDK].

Cada enfoque tiene sus puntos a favor y sus puntos en contra. Es fundamental una buena elección para el desarrollo de un prototipo óptimo. A continuación destacaré los puntos a favor y en contra de cada uno de los enfoques.

#### **Puntos a favor de las aplicaciones web optimizadas para dispositivos móviles**

Uno de los puntos a favor más importantes de las aplicaciones web optimizadas para dispositivos móviles es que los desarrolladores pueden usar cualquier tecnología para desarrollarlas, incluyendo lenguajes, frameworks y librerías. Como consecuencia de esto, no es necesario ningún conocimiento sobre una plataforma determinada para desarrollar una aplicación, ya que la tecnología empleadas para desarrollar la aplicación web es conocida.

Otro punto a favor de las aplicaciones web optimizadas para dispositivos móviles es la disponibilidad. Siempre que el dispositivo cuente con un navegador web, y obviamente, una conexión a internet, la aplicación va a poder ser utilizada.

#### **Puntos en contra de las aplicaciones web optimizadas para dispositivos móviles**

El primer punto en contra que podemos encontrar es que las aplicaciones web optimizadas para dispositivos móviles no pueden acceder al hardware del dispositivo, como el acelerómetro, el GPS o la cámara.

Otro punto en contra de las aplicaciones web optimizadas para dispositivos móviles es que, a diferencia de las aplicaciones nativas, es difícil lograr algunos efectos visuales, como por ejemplo el desplazamiento entre páginas o los diálogos que se aparecen en la pantalla deslizando desde la parte superior de la pantalla.

## **Puntos a favor de las aplicaciones nativas**

Las aplicaciones nativas pueden acceder al hardware del dispositivo móvil, lo cual en el caso de una aplicación que necesita obtener la ubicación en la que se encuentra el dispositivo es de suma importancia, ya que el acceso al hardware de GPS del dispositivo se realiza a través de una API.

## **Puntos en contra de las aplicaciones nativas**

Para que una aplicación Android pueda ser descargada desde el market de Android, el desarrollador debe registrarse. El proceso de registración como desarrollador de Android no es gratuito. [ANDROIDDEVREG]

Otro punto en contra es que las aplicaciones nativas sólo podrán ser ejecutadas en la plataforma para la cual fue desarrollada, lo que significa que para ejecutarse en diferentes dispositivos móviles de diferentes plataformas, la misma aplicación debe desarrollarse varias veces. Por ejemplo, para que una aplicación pueda ser ejecutada en Android e iOS (plataforma de los iPhones) deberá ser desarrollada 2 veces, una usando el SDK de Android, y otra usando el SDK de iOS.

Además, desarrollar una aplicación nativa implica que el ciclo de desarrollo sea más lento que el ciclo de desarrollo de una aplicación web adaptada para dispositivos móviles, debido a que hay que desarrollar, compilar y desplegar la aplicación para realizar cualquier cambio.

## **Entonces, ¿aplicación web optimizada para dispositivos móviles o aplicación nativa?**

weegoo debe hacer uso de la información geográfica donde se encuentra el usuario ya sea para crear nuevos puntos de interés como para indicar qué puntos de interés se encuentran cerca. Es por eso que es necesario acceder al hardware GPS del dispositivo móvil. Según este requerimiento y habiendo clarificado los puntos a favor y en contra de los dos enfoques, parecería que la aplicación nativa será la opción a utilizar. Sin embargo, HTML5 implementa la API de geolocalización especificada por el W3C<sup>23</sup> (explicada más adelante), la cual permite obtener de diferentes maneras la ubicación actual del dispositivo. En la actualidad, la mayoría de los navegadores web de los dispositivos móviles soportan HTML5, con lo cual es posible obtener la información geográfica del dispositivo móvil. [MOBILEHTML5]

Debido a que se trata de un prototipo, desarrollar una aplicación nativa es demasiado costosa con respecto a desarrollar una aplicación web y luego optimizarla para que se vea correctamente en dispositivos móviles. En primer lugar, porque para que sea utilizada por más personas es necesario desarrollar más de una aplicación nativa (por ejemplo una para Android y otra para iPhone) y en segundo lugar por el registro de la aplicación, que no es gratuito.

Por estos motivos decido desarrollar el cliente móvil como una aplicación web optimizada para este tipo de dispositivos.

---

<sup>23</sup> W3C, World Wide Web Consortium. <http://www.w3.org/>

# El desarrollo de weegoo

El desarrollo de weegoo fue dividido en dos etapas. Esto no es un hecho arbitrario, sino que se debe a la búsqueda de las mejores herramientas y a la incorporación de conocimientos sobre las mismas.

La primera etapa consistió en el desarrollo completo de la especificación. Una vez finalizada esta etapa, llegué a la conclusión de que había muchas cosas por mejorar, con lo cual decidí reimplementar la aplicación en una segunda etapa que incluye tecnologías más nuevas, mejoras en el modelo de datos y en las metodologías.

## Decisiones

### Sistema de control de versiones

El primer punto a decidir fue qué sistema de control de versiones utilizar. Las opciones más fuertes fueron:

- git [GIT]
- subversion [SUBVERSION]
- cvs [CVS]

**git** es un sistema de control de versiones distribuido open source, diseñado para manejar desde proyectos pequeños a proyectos grandes con velocidad y eficiencia. [GIT]

Muchos proyectos importantes utilizan git como sistema de control de versiones, entre ellos:

- git [GITGIT]
- El kernel de linux [GITLINUX]
- perl [GITPERL]
- Eclipse [GITECLIPSE]
- Ruby on Rails [GITRAILS]
- Android [GITANDROID]
- PostgreSQL [GITPOSTGRESQL]
- Debian [GITDEBIAN]

Al igual que git, **Subversion** es un sistema de control de versiones open source, pero a diferencia de git, no es distribuido sino que centralizado. [SUBVERSION]

**CVS** es un sistema de control de versiones diseñado para proyectos de software. CVS mantiene una trazabilidad de los cambios sobre un archivo o un conjunto de archivos.

La decisión sobre qué sistema de control de versiones fue sencilla. Decidí utilizar **git** en parte porque muchos de los proyectos open source más importantes (algunos de los mencionados arriba) utilizan este sistema de control de versiones y en parte por la posibilidad de trabajar con GitHub [GITHUB]. GitHub es una red social de repositorios de software, es la mejor manera de colaborar con otros. Se pueden bifurcar repositorios (fork), enviar solicitudes de pull (para

mejorar los proyectos) y manejar repositorios públicos y privados. Además, cada repositorio tiene una wiki y un issue tracker, para solicitar funcionalidad y reportar errores.

GitHub tiene más de 1 millón de usuarios y de 2 millones y medio de repositorios git. Entre los usuarios de github más destacados se encuentran: Twitter, Facebook, Yahoo!, EMI, etc., y entre los repositorios destacados se encuentran: jQuery[GITJQUERY], Ruby on Rails[GITRAILS], CakePHP[GITCAKE], Symfony[GITSYMFONY].

## Lenguaje y ecosistema

A la hora de elegir el lenguaje de implementación, las opciones manejadas fueron PHP y Ruby. Mi decisión fue implementar weegoo usando Ruby por sobre PHP debido a que Ruby es un lenguaje no muy popular en nuestro país, pero que en el exterior es usado en diversos proyectos muy importantes, como Twitter, GitHub, Groupon[GROUPON], entre otros. Además de este motivo, es de destacar la amplia comunidad de usuarios a lo largo del mundo, y con más de 30 mil librerías [RUBYGEMS]. Por otro lado, Ruby y su ecosistema no fue abordado en ninguna materia a lo largo de la carrera, lo cual despierta un interés extra.

A esto se suma mi experiencia de 4 años como desarrollador Symfony<sup>24</sup> [SYMFONY], ya que conozco las ventajas y desventajas del framework y el lenguaje en el cual está escrito (PHP). Además es sabido que Symfony basa muchas de sus características en muchas características de Ruby on Rails.

Como consecuencia de esto, el lenguaje elegido fue Ruby. “Ruby es un lenguaje de programación dinámico y open source enfocado en la simplicidad y productividad. Su elegante sintaxis se siente natural al leerla y fácil al escribirla.” [RUBY]

Ruby es un lenguaje de programación interpretado, no tipado y orientado a objetos. Entre sus características principales se destacan:

- *todo es un objeto*: a cada dato y fragmento de código pueden asignársele propiedades y acciones. Este aspecto es una influencia de Smalltalk<sup>25</sup>.
- *flexibilidad*: sus usuarios pueden alterarlo libremente para agregar, quitar y redefinir funcionalidad.
- *bloques*: inspirado en los lenguajes funcionales, los bloques sirven para anexar cualquier funcionalidad a cualquier método, describiendo como debe actuar.
- *mixin*: Ruby posee herencia simple. Sin embargo, incorpora el concepto de módulos, que son conjuntos de métodos. Las clases pueden mezclar (mixin) un módulo y utilizar sus métodos como si fueran propios.
- *portabilidad*: Ruby corre en GNU/Linux, Windows, Mac OS X, etc.

---

<sup>24</sup> Symfony es un framework para el desarrollo de aplicaciones web, escrito en PHP.

<sup>25</sup> Smalltalk es un lenguaje de programación orientado a objetos.

Existen dos versiones estables de Ruby, la rama 1.8 y la rama 1.9. La versión utilizada para el desarrollo de weegoo es la rama 1.8.7 en la primera etapa, sin embargo, para la segunda etapa opté por utilizar la rama 1.9, en particular la versión 1.9.2.

Además, utilizo RubyGems [RUBYGEMS], que es gestor de paquetes que proporciona una forma de distribuir programas o librerías en Ruby. A través de RubyGems se instala, por ejemplo, Ruby on Rails.

A partir de la segunda etapa en el desarrollo de weegoo utilizo RVM [RVM] para crear un entorno que permita la ejecución de weegoo, aislando el intérprete de Ruby y los gems (paquete de software instalado con RubyGems) necesarios para que weegoo funcione de otras aplicaciones.

RVM es una utilidad que permite instalar, administrar y trabajar con múltiples entornos Ruby, desde intérpretes hasta conjuntos de gems (gemsets).

Por último, decidí utilizar un framework MVC<sup>26</sup> que facilite la generación del prototipo. En este caso, el framework elegido para el desarrollo de weegoo es Ruby on Rails [RAILS].

Ruby on Rails es un framework de aplicaciones web escrito en Ruby. Está diseñado para desarrollar aplicaciones de una manera simple, mediante la asunción de cosas que el desarrollador tiene que saber. De esta forma permite escribir menos líneas de código para obtener el mismo resultado que con otros lenguajes y frameworks.

La filosofía de Rails se basa en los siguientes principios:

- DRY<sup>27</sup>: repetir código es un mal hábito.
- Convention over configuration<sup>28</sup>: Rails asume qué cosas se quieren hacer y cómo hacerlas en vez de tener que especificar cómo hacer cada cosa en particular usando, por ejemplo, archivos de configuración.
- REST<sup>29</sup> es el mejor patrón para aplicaciones web. Básicamente, REST consiste en asociar un verbo HTTP a una acción sobre los recursos de la aplicación. Un identificador de recurso es, por ejemplo, una URL.

Rails sigue la arquitectura MVC. Esta arquitectura trae muchos beneficios, entre los cuales se destacan el aislamiento de la lógica de la aplicación con su interfaz gráfica y una organización simple que permite un buen mantenimiento de la aplicación.

## Base de datos

---

<sup>26</sup> MVC (Model-View-Controller) es una arquitectura de software en donde la lógica de la aplicación (controlador) está aislada de los otros dos componentes (modelo y vista).

<sup>27</sup> DRY: Don't Repeat Yourself.

<sup>28</sup> Convention over configuration: Convención sobre configuración.

<sup>29</sup> REST: Representational State Transfer. [REST]

A la hora de elegir la base de datos, analicé las bases de datos open source más populares, tales como MySQL [MYSQL], PostgreSQL [POSTGRESQL] y SQLite [SQLITE]. Debido a mi experiencia en el uso de MySQL y dadas las características de weegoo, la base de datos elegida es MySQL.

De todas formas, por la arquitectura del framework Ruby on Rails, la base de datos puede cambiarse sin afectar a la aplicación. Por ejemplo, se podría pasar de MySQL a SQLite o a PostgreSQL sólo cambiando un archivo con la configuración de la base de datos.

## Interface móvil

Si bien la interface móvil será la misma aplicación web adaptada para dispositivos móviles, es necesario facilitar el desarrollo de la misma. Hoy en día existen diversos frameworks javascript open source que, usando ciertas convenciones construyen una interface unificada adaptada para diversas plataformas de dispositivos móviles.

Los frameworks evaluados fueron:

- jQuery Mobile [JQUERYMOBILE]
- jQTouch [JQTOUCH]
- Sencha Touch [SENCHATOUCH]

El objetivo de **jQuery Mobile** es proveer una interface de usuario que funcione del mismo modo en todas las plataformas de dispositivos móviles. Está construido sobre jQuery<sup>30</sup> [JQUERY] y jQuery UI<sup>31</sup> [JQUERYUI] y tiene un diseño customizable.

En jQuery Mobile todas las páginas son construidas sobre las bases de HTML semántico<sup>32</sup> para asegurar la compatibilidad con todos los dispositivos. jQuery Mobile transforma no obstrusivamente<sup>33</sup> ese HTML semántico en una página rica e interactiva. Además integra características de accesibilidad como WAI-ARIA<sup>34</sup> [W3CWAIARIA] para proveer soporte a lectores de pantalla y otras tecnologías.

Las características más importantes de jQuery Mobile son:

- compatible con las plataformas móviles y de escritorio más importantes: iOS, Android, BlackBerry, Plam WebOS, etc..
- las páginas y el comportamiento necesitan una programación mínima y se basan en configuración usando HTML5.

---

<sup>30</sup> jQuery es una librería de javascript que simplifica el acceso al DOM HTML, el manejo de eventos, animación y llamadas ajax.

<sup>31</sup> jQuery UI es una librería escrita por sobre jQuery que provee abstracciones para la interacción y animaciones, además widgets customizables.

<sup>32</sup> HTML semántico se refiere al uso de las etiquetas HTML para darle sentido a la información que se muestra en las páginas web, no solo para definir como se verá.

<sup>33</sup> Transformar no obstrusivamente significa transformar mediante la separación del comportamiento y la estructura/contenido y presentación.

<sup>34</sup> WAI-ARIA: Accessible Rich Internet Applications

- Inicialización automática usando atributos data-role de HTML5.
- Soporte para eventos del mouse y táctiles.
- widgets de UI optimizados para táctiles, como sliders.
- framework para modificación de temas visuales.

La siguiente tabla (ilustración 37) muestra el soporte que provee jQuery Mobile para los distintos navegadores en las distintas plataformas.

*Ilustración 37: Tabla de soporte de plataformas de dispositivos móviles. A = Alta calidad - B = Calidad media - C = Baja calidad. [JQUERYMOBILETABLE]*

**jQTouch** es un plugin para jQuery para desarrollo móvil en dispositivos iPhone, Android, Palm, etc.. Sus características principales son:

- totalmente customizable.



- soporte para temas visuales, incluyendo el tema default de Apple.
- funciones callback, incluyendo cambio de orientación y swipe.
- manejo del historial de páginas y transiciones de páginas usando CSS3, incluyendo flip 3D.
- usa jQuery para construir aplicaciones AJAX<sup>35</sup>.

**Sencha Touch** es un framework que permite desarrollar aplicaciones web con “look and feel” nativo en dispositivos táctiles como iPhone, Android y BlackBerry. Sus principales características son:

- uso de estándares web: HTML5, CSS3, javascript.
- eventos táctiles: además de los eventos estándar soportados por el browser (touchstart y touchend) agrega otros eventos customizados como tap, double tap, swipe, etc..
- multi plataforma: apunta a los dispositivos móviles táctiles. Actualmente soporta iOS 3+, Android 2.1+ y BlackBerry 6+.

Debido a que jQuery Mobile soporta muchas más plataformas y navegadores y a que su desarrollo tiene más actividad, ha sido elegido como el framework a utilizar.

## API para manejo de mapas

**Google Maps JavaScript API** permite insertar Google Maps en cualquier aplicación web. La versión 3 de esta API está especialmente diseñada para proporcionar una mayor velocidad y que se pueda aplicar más fácilmente tanto a móviles como a las aplicaciones de navegador de escritorio tradicionales. [GMAPSAPIV3]

Junto con la API para el manejo de mapas es necesario utilizar alguna API para obtener la ubicación actual de los usuarios. Como se mencionó anteriormente, HTML5 provee la API para geolocalización, estandarizada por el W3C.

## W3C Geolocation API

La API de geolocalización del W3C define una interface de alto nivel para acceder a la información de ubicación asociada al dispositivo. La API es independiente de las fuentes de información subyacentes. Las fuentes de información más comunes incluyen al GPS, la geolocalización usando direcciones IP, Id de celda GSM/CDMA, etc.. La API no ofrece garantías sobre la ubicación real del dispositivo. [W3CGEOAPI]

La especificación se limita a proveer una API para obtener la información de la posición geográfica asociada al dispositivo. Esta información de la posición geográfica es provista en términos de coordenadas del Sistema Geodésico Mundial [WGS84].

## Interface Geolocation

Los objetos que implementan la interface **Geolocation** son usados para determinar programáticamente la información de localización asociada al dispositivo. La información de

---

<sup>35</sup> AJAX: Asynchronous Javascript And XML.

localización se obtiene aplicando un algoritmo específico del agente de usuario<sup>36</sup>, creando un objeto **Position** con la información acorde a la ubicación obtenida.

*Ilustración 38: interface NavigatorGeolocation*

*Ilustración 39: interface Geolocation*

El método **getCurrentPosition()** recibe uno, dos o tres parámetros. Cuando se lo invoca debe retornar inmediatamente y luego intentar obtener la localización del dispositivo asincrónicamente. Si el intento es exitoso debe invocarse **successCallback**, que debe recibir un objeto **Position** como parámetro. Si el intento es fallido, y si se ha especificado, debe invocarse **errorCallback**, que debe recibir un objeto **PositionError** como parámetro, el cual debe reflejar el motivo del error.

El método **watchPosition()** recibe uno, dos o tres parámetros. Cuando se lo invoca debe retornar inmediatamente un valor de tipo **long** que identifique unívocamente la operación y luego comience la operación asincrónicamente. Si el intento es exitoso debe invocarse **successCallback**, que debe recibir un objeto **Position** como parámetro. Si el intento es fallido, y si se ha especificado, debe invocarse **errorCallback**, que debe recibir un objeto

---

<sup>36</sup> Un agente de usuario es una cadena que identifica a un cliente web. Este cliente web puede ser un navegador, un lector de pantalla, un web crawler, etc..

**PositionError** como parámetro, el cual debe reflejar el motivo del error. Luego, la operación debe continuar monitoreando la posición del dispositivo e invocar al callback correspondiente cada vez que cambie la posición del dispositivo hasta que se invoque el método **clearWatch()**, que debe ser invocado con el identificador de la operación.

Tanto para **getCurrentPosition()** como **watchPosition()**, la implementación no debe nunca invocar a **successCallback** sin la autorización del usuario a compartir su ubicación. Si el usuario autoriza a compartir su ubicación se invocará el callback correspondiente. Si en cambio el usuario no autoriza a compartir su ubicación y si está presente, **errorCallback** deberá ser invocado con el código **PERMISSION\_DENIED**.

### Interface PositionOptions

Los métodos **getCurrentPosition()** y **watchPosition()** de la interface **Geolocation** reciben un objeto que implementa la interface **PositionOptions** como tercer parámetro. Esta interface representa opciones que estos métodos utilizarán.

#### *Ilustración 40: interface PositionOptions*

El atributo **enableHighAccuracy** provee a las aplicaciones una forma de especificar que quiere recibir los mejores resultados posibles, lo que puede llevar a respuestas más lentas o que consuman más energía. Esto es especialmente útil para aplicaciones que corran en dispositivos a batería (como teléfonos móviles), debido a que permite evitar el consumo excesivo de batería. Este atributo es seteado como **false** por defecto.

El atributo **timeout** indica la duración máxima permitida (en milisegundos) para invocar a **successCallback**. Si la implementación no es capaz de obtener un objeto **Position** antes de que finalice el **timeout** y no ocurrieron errores en este intervalo de tiempo, entonces se invoca a **errorCallback** con un **PositionError** como parámetro indicando que el código de error es **TIMEOUT**. El tiempo en que se solicita al usuario la autorización para compartir su ubicación no es incluido en el intervalo de tiempo de **timeout**. Por defecto, el valor del atributo es infinito.

El atributo **maximumAge** indica que la aplicación aceptará posiciones cacheadas que no hayan sido cacheadas por más de el tiempo especificado en milisegundos. Si **maximumAge** es 0 (valor por defecto), la implementación debe intentar obtener un nuevo objeto **Position**.

### Interface Position

La interface **Position** es el contenedor para la información de geolocalización.

*Ilustración 41: interface Position*

El atributo **coords** contiene las coordenadas geográficas asociadas con su precisión, además de atributos opcionales como altitud y velocidad.

El atributo **timestamp** representa la hora en que el objeto **Position** fue obtenido.

### **Interface Coordinates**

Los atributos de esta interface son los definidos por el Sistema Geodésico.

*Ilustración 42: interface Coordinates*

Los atributos **latitude** (latitud) y **longitude** (longitud) son las coordenadas geográficas expresadas en decimales.

El atributo **altitude** (altitud) representa la altura de la posición, expresada en metros. Si la implementación no puede proveer esta información, el valor debe ser nulo.

El atributo **accuracy** (precisión) denota el nivel de precisión de las coordenadas. Es especificada en metros y debe ser soportada por todas las implementaciones. El valor de este atributo debe ser un número real positivo.

El atributo **altitudeAccuracy** es especificado en metros y tiene el mismo significado que el atributo **accuracy**. Si la implementación no puede proveer esta información el valor de este atributo debe ser null, de lo contrario debe ser un número real positivo.

El atributo **heading** denota el sentido en que el dispositivo va marchando, especificado en grados, donde  $0^\circ \leq \text{heading} \leq 360^\circ$ . Si la implementación no puede proveer esta información, el valor de éste atributo debe ser null. Si el dispositivo es sedentario (el valor del atributo **speed** es 0), entonces el valor de éste atributo debe ser NaN<sup>37</sup>.

---

<sup>37</sup> NaN: Not a number.

El atributo **speed** representa la velocidad con la que se desplaza el dispositivo expresado en metros por segundo. Si la implementación no puede proveer esta información, el valor de este atributo debe ser null. De lo contrario, el atributo debe ser un valor real positivo.

## Interface PositionError

*Ilustración 43: interface PositionError*

El atributo **code** debe retornar el código de error. Debe ser alguno de estos valores:

- **PERMISSION\_DENIED**: El proceso de obtención de localización falló por falta de autorización.
- **POSITION\_UNAVAILABLE**: La posición del dispositivo no puede ser determinada.
- **TIMEOUT**: El tiempo especificado en timeout ha finalizado antes de que se haya podido obtener la posición del dispositivo.

El atributo **message** debe retornar un mensaje de error que describa los detalles del error encontrado. Este atributo tiene la intención de ser usado con propósitos de debugging.

## Primera etapa del desarrollo de weegoo

Habiendo seleccionado las herramientas principales para el desarrollo, procedí al desarrollo de la especificación de la aplicación web y, una vez finalizada, la aplicación móvil.

### Primeros bocetos de la interface web

Antes de comenzar a desarrollar, realicé unos bocetos usando una herramienta llamada Balsamiq Mockups [BALSAMIQMOCKUPS], los cuales me permitieron seguir una línea en cuanto al diseño visual de weegoo. Luego, como todo boceto, fue mutando en cómo se ve weegoo actualmente (se verá más adelante).

A continuación se muestran 3 bocetos: la ilustración 44 muestra como se vería la página de inicio de un usuario registrado, la ilustración 45 muestra la página de búsqueda de personas, y por último, la ilustración 46 muestra la página de búsqueda de puntos de interés (lugares).

*Ilustración 44: Pagina de inicio de un usuario.*

*Ilustración 45: Búsqueda de personas (para usuarios autenticados).*

*Ilustración 46: Búsqueda de lugares (para usuarios autenticados).*

## Herramientas utilizadas

En la sección anterior han sido especificadas las herramientas principales para el desarrollo de weegoo. Sin embargo, en esta sección se especifican otras herramientas utilizadas que han facilitado mucho el desarrollo.

### Ruby on Rails

Se utilizó la versión 3.0.7 del framework.

### bundler

bundler [BUNDLER] es un gem que permite manejar las dependencias de las aplicaciones Ruby de una manera simple. A partir de un archivo de configuración (Gemfile) es capaz de descargar e instalar las dependencias de la aplicación recursivamente (es decir, si una dependencia tiene a su vez dependencias, éstas son descargadas e instaladas también). Además es capaz de actualizar las dependencias en cualquier momento (siempre y cuando existan esas actualizaciones).

### devise

devise [DEVISE] es un gem que provee una solución de autenticación para Ruby on Rails. Es una solución modular: sólo se activan los módulos que se necesitan. Entre los diferentes módulos que provee se destacan:

- *Database Authenticatable*: encripta y guarda las contraseñas en la base de datos. La autenticación puede hacerse mediante solicitudes POST o autenticación HTTP básica.
- *Token Authenticatable*: el usuario inicia sesión usando un token.
- *Confirmable*: Agrega mecanismos de confirmación de identidad. Esto se logra enviando un correo electrónico con instrucciones de activación.

- *Trackable*: mantiene un registro con la actividad de las cuentas.
- *Lockable*: bloquea aquellas cuentas que realicen una cantidad determinada de intentos de inicio de sesión fallidos. Luego provee diferentes mecanismos para desbloquear esas cuentas.

### **simple\_form**

`simple_form` [SIMPLEFORM] permite redefinir la forma en que se visualizan los formularios en Ruby on Rails. Por defecto, el `scaffolding`<sup>38</sup> de Ruby on Rails genera en las vistas de los formularios la forma en que éstos se renderizan. El problema es que si la aplicación tiene muchos formularios, las vistas pueden diferir si no se lleva un buen control de ellas. `simple_form` provee una forma de especificar los campos de los formularios, y mediante un archivo de configuración, la forma en que se renderiza. De esta forma, todos los formularios de la aplicación se verán de la misma manera.

### **jquery-rails**

`jquery-rails` [JQUERYRAILS] es un gem que provee el framework jQuery a las aplicaciones Ruby on Rails.

Si bien jQuery puede descargarse y utilizarse directamente, es una buena práctica utilizar gems que provean de una manera transparente el uso de frameworks javascript. Esto simplifica la actualización del framework javascript, ya que simplemente actualizando el gem se actualiza el framework.

### **kaminari**

`kaminari` [KAMINARI] es un paginador para Ruby on Rails. Agrega unos métodos a ActiveRecord de manera tal que podemos especificar la página y la cantidad de elementos a mostrar en esa página.

### **permalink\_fu**

`permalink_fu` [PERMALINKFU] es un plugin que permite crear permalinks<sup>39</sup> a partir de un atributo del modelo. Se especifica a partir de que atributo del modelo se quiere generar el permalink y en que otro atributo del mismo modelo se almacenará. El permalink se genera cuando se guarda por primera vez el objeto y se actualiza cada vez que se actualiza el modelo.

### **carrierwave**

`carrierwave` [CARRIERWAVE] provee una forma simple y flexible para subir archivos en aplicaciones Ruby on Rails. Permite que los archivos sean guardados en el file system del servidor o en una ubicación remota, como por ejemplo usando amazon web services<sup>40</sup>. Además

---

<sup>38</sup> `scaffolding` es una forma de generar partes de código de una aplicación. Genera el modelo, controlador y vistas para un recurso nuevo ejecutando un simple comando: `rails g scaffold Model`.

<sup>39</sup> Un permalink es un enlace permanente. Permite identificar un recurso de manera unívoca.

<sup>40</sup> Amazon Web Services es un servicio de Amazon que permite a sus usuarios una plataforma de servicios web. Se puede solicitar potencia de cómputo como capacidad de almacenamiento, entre otros servicios.



permite mantener distintas versiones de las imágenes, por ejemplo, la versión en tamaño original y otra versión en tamaño de ícono.

### **rmagick**

rmagick [RMAGICK] es una interface entre el Ruby y las librerías ImageMagick<sup>41</sup> y GraphicsMagick<sup>42</sup>, utilizadas por carrierwave para crear las diferentes versiones de las imágenes.

### **omniauth**

omniauth [OMNIAUTH] es un sistema de autenticación para proveedores externos, por ejemplo Facebook y Twitter. Además de Facebook y Twitter soporta muchos otros proveedores, como Flickr, Google, Foursquare, etc..

### **twitter**

twitter [TWITTERGEM] es un gem que permite el acceso a la API REST y de búsqueda de Twitter. Mediante el uso de este gem es posible ver el timeline de los usuarios, sus followers e incluso actualizar su estado.

### **css3buttons**

css3buttons [CSS3BUTTONS] es un gem que agrega métodos a los helpers<sup>43</sup> de Ruby on Rails que permiten crear botones más vistosos.

### **fb\_graph**

fb\_graph [FBGRAPH] es un gem que permite el acceso al grafo de información de Facebook. Al igual que el gem twitter, permite ver los amigos de un usuario, su estado e incluso realizar actualizaciones.

### **mobile-fu**

mobile-fu [MOBILEFU] permite detectar automáticamente si el dispositivo que accedió a la aplicación fue un dispositivo estándar (por ejemplo una PC de escritorio usando un navegador web) o un dispositivo móvil. Teniendo esa información, se puede presentar una vista diferente a los dispositivos móviles.

## **Setup inicial**

---

<sup>41</sup> ImageMagick es una utilidad open source que permite manipular imágenes, tales como transformaciones, cambios de formato, generar animaciones etc.. <http://www.imagemagick.org>

<sup>42</sup> GraphicsMagick es una utilidad open source que permite manipular imágenes. <http://www.graphicsmagick.org>

<sup>43</sup> Un helper de Ruby on Rails es un módulo de que provee funcionalidad extra a las vistas.

El primer paso a seguir luego de haber elegido las herramientas fue crear el repositorio git en GitHub.

*Ilustración 47: Repositorio de weegoo en GitHub.*

El repositorio resultante puede verse en <https://github.com/patriciomacadden/weegoo>.

### **Registro de weegoo en las aplicaciones externas**

A partir de la especificación de weegoo se inició el desarrollo incremental de la aplicación, comenzando por la autenticación de usuarios. Una vez definido esto fue necesario registrar la aplicación en Twitter y en Facebook, de modo que los usuarios puedan elegir entre registrarse en weegoo o utilizar la identidad de Facebook o de Twitter.

*Ilustración 48: Página de registro de weegoo en Facebook.*

*Ilustración 49: Página de registro de weegoo en Twitter.*

Una vez registradas las aplicaciones, utilicé el gem omniauth para que weegoo sea capaz de autenticar sus usuarios con Facebook o con Twitter y los gems fb\_graph y twitter para que aquellos usuarios autenticados con facebook o con twitter puedan publicar su actividad en esas redes sociales (siempre y cuando hayan configurado su cuenta para que así lo hagan).

Además de Facebook y Twitter, weegoo hace uso de Gravatar [GRAVATAR] para mostrar los avatares de los usuarios registrados. El uso de gravatar es muy sencillo: un usuario se registra en Gravatar y carga su imagen, que puede ser usada por cualquier sitio web que haga uso de la API de gravatar. En weegoo, los usuarios no necesitan cargar su avatar, sino que se obtiene mediante el uso de la API de Gravatar, y en caso de no tener usuario de gravatar, se mostrará una imagen por defecto (provista por la misma API de Gravatar).

## **Manejo de lugares y eventos**

Una vez que los usuarios podían iniciar sesión en weegoo debí pensar en cómo manejar los lugares y eventos. De cada lugar y evento se necesita saber su nombre, descripción, categoría y subcategoría (de modo que puedan ser navegables desde el dispositivo móvil), y para el caso de los eventos fecha y hora. Además de esta información es necesario guardar la ubicación geográfica donde se encuentran los lugares y eventos. Esto se logra guardando dos campos extra: latitud y longitud. Luego, esta información se calcula utilizando la APIv3 de GMaps, a partir de la entrada del usuario (el usuario selecciona un punto en el mapa).

## **Actividad**

Las actividades que los usuarios pueden realizar en weegoo son:

- conectarse con otros usuarios
- crear un lugar o evento
- comentar lugares y eventos
- agregar imágenes a los lugares y eventos
- indicar qué lugares y eventos visitaron, y sus respectivas acciones inversas (qué lugares y eventos no visitaron)
- indicar qué lugares y eventos quieren visitar, y sus respectivas acciones inversas (qué lugares y eventos no quieren visitar)

Esta actividad se ve reflejada en la línea de tiempo del usuario (Actividad reciente), y, siempre que se den las condiciones, puede ser publicada en las redes sociales a las que esté conectado. Esta publicación es posible a través de los gems `fb_graph` y `twitter`.

## **Interface móvil**

En esta etapa decidí que para la interface móvil era necesario crear una nueva aplicación que contenga la funcionalidad común (modelos y controladores) y luego por intermedio del gem `mobile-fu` se haga una redirección a la aplicación indicada dependiendo del agente de usuario de donde provenga la solicitud HTTP. Esto significa que una aplicación correría en el dominio [weegoo.com.ar](http://weegoo.com.ar) (la aplicación web) mientras que la otra correría en el subdominio [m.weegoo.com.ar](http://m.weegoo.com.ar) (la aplicación web adaptada para dispositivos móviles). De esta forma, si un dispositivo móvil ingresara a [weegoo.com.ar](http://weegoo.com.ar) sería automáticamente redirigido a [m.weegoo.com.ar](http://m.weegoo.com.ar).

Debido a las prestaciones de los dispositivos móviles (sobre todo velocidad de la transferencia de datos y el tamaño de la pantalla), no todas las características de weegoo estarán disponibles en la versión web adaptada para dispositivos móviles. Esto quiere decir que la aplicación web adaptada para dispositivos móviles será limitada con respecto a la aplicación web, por ejemplo, con las imágenes. Esto se debe a que las imágenes pueden enlentecer la carga de la aplicación, debido a la mayor transferencia de datos, lo cual no es deseable en una aplicación móvil.

Además, debido a que se supone que la interface móvil se usará en momentos donde el usuario se encuentra en movimiento (o al menos no en un lugar fijo), decidí que la interface móvil no provea la funcionalidad para hacer amistades y buscar personas.

Haciendo un resumen, la interface web incluirá la siguiente funcionalidad:

- buscar lugares cercanos a la ubicación actual del dispositivo
- buscar eventos próximos cercanos a la ubicación actual del dispositivos
- navegar lugares por categoría y subcategoría
- ver la información de los lugares y eventos, incluyendo la información básica (nombre, descripción, categoría, subcategoría), información acerca de los usuarios que estuvieron o quieren ir al mismo y comentarios.

A pesar de que la funcionalidad planteada estaba completa, al revisar el diseño y la implementación de weegoo me di cuenta tanto el modelo de datos como la implementación de alguna de las características de weegoo podían ser mejoradas/optimizadas.

En primer lugar el modelo de datos suponía que la única diferencia entre un lugar y un evento era que este último tenía información acerca de fecha y hora de realización. Esto significa que para crear un nuevo evento en un lugar determinado había que ingresar la información de la ubicación del evento, es decir, el punto en el mapa. Esto tiene varias contras: la primera y más evidente es que por cada evento que se realice en un mismo lugar, la información sobre la ubicación del mismo se duplicará, mientras que la segunda es algo más sutil y es que no permite obtener una estadística cierta acerca de en que lugares se realizaron más eventos.

En segundo lugar, de acuerdo a los conocimientos obtenidos acerca del lenguaje Ruby, el framework Ruby on Rails y el ecosistema, me di cuenta que podía mejorar/optimar la implementación de mucha de las funcionalidades de weegoo.

## **Segunda etapa del desarrollo de weegoo**

Como se mencionó anteriormente, la segunda etapa agrega o mejora algunas funcionalidades con respecto a la primera etapa. Esta sección se organiza de la misma forma que la anterior, y sólo se especificarán los cambios y/o mejoras con respecto a la primera etapa (las herramientas y funcionalidad en común se omite).

### **Herramientas utilizadas**

Con respecto a las herramientas utilizadas, las diferencias con respecto a la primera etapa son las siguientes.

## **Ruby on Rails**

Utilicé la versión 3.1.rc4, que agrega mejoras al framework, entre ella la herencia de templates, que es especialmente útil para mostrar las vistas adaptadas para dispositivos móviles. Gracias a esta funcionalidad se elimina el gem mobile-fu y se redefine la forma en que se desarrolla la aplicación móvil (la cual será explicada más adelante).

## **fancybox-rails**

fancybox-rails [FANCYBOXRAILS] es un gem que provee la extensión FancyBox [FANCYBOX] a las aplicaciones Ruby on Rails.

FancyBox es una herramienta que se utiliza para mostrar imágenes en una capa por encima de la página que se está viendo. Fue construida sobre jQuery y es open source.

## **acts\_as\_ferret**

acts\_as\_ferret [ACTSASFERRET] es un gem que provee una extensión para ActiveRecord que permite la búsqueda full-text sobre los modelos de la aplicación usando Ferret [FERRET].

Ferret es una librería de motor de búsqueda full-text escrito en C con una api nativa para Ruby. Originalmente fue una implementación del motor de búsqueda Lucene [LUCENE] pero luego usó sus propios formatos de archivos y algoritmos de indexación.

En particular, los modelos indexados fueron Venue (que representa los lugares) y Event (que representa los eventos).

## **Manejo de lugares y eventos**

Durante las pruebas de la aplicación (resultado del desarrollo de la primera etapa), se evidenció la duplicación de información en la carga de eventos. Esto se debe a que los lugares y los eventos contenían los mismos datos a diferencia de fecha y hora, es decir, eran subclase de un clase llamada punto de interés. De esta forma, cuando se quería crear un lugar se ingresaba:

- nombre
- descripción
- categoría
- sub categoría
- ubicación (se selecciona un lugar en el mapa)

y cuando se quería crear un lugar se ingresaba la misma información y además:

- fecha
- hora

Como se puede apreciar, los eventos siempre ocurren en lugares que ya existen, por lo tanto la información sobre un lugar debía ingresarse cada vez que se quería crear un evento. Gracias a esto, los eventos fueron desacoplados de los lugares y luego de esta modificación pertenecen a ellos. De esta forma, un lugar tiene una lista de eventos que ocurrieron (u ocurrirán) allí y no es necesario duplicar la información del lugar donde se realizará.

## **Interface móvil**

Como se explicó anteriormente, una de las mejoras en el framework en la versión 3.1.rc4 (hasta el momento del desarrollo de weegoo no ha sido publicada la versión estable del framework) es la herencia de vistas. Esto permite que por algún criterio podamos mostrar una vista u otra. En particular, cuando se acceda a [weegoo.com.ar](http://weegoo.com.ar) mediante un dispositivo móvil, la aplicación se encarga automáticamente de mostrar las vistas adaptadas para dispositivos móviles. Esto se logra detectando el user agent del navegador que hace el requerimiento HTTP. En caso de ser un user agent de un navegador móvil, la aplicación se ejecutará usando un subdominio (en particular, el subdominio m), y luego mostrará las vistas para la interface móvil.

De esta forma no es necesario desarrollar dos aplicaciones que dupliquen los modelos y los controladores, sino que en la misma aplicación se definen dos tipos de vistas, la vista estándar y la vista adaptada para dispositivos móviles. Esto influye mucho también en el mantenimiento de la aplicación.

## **Prototipo de weegoo**

Habiendo desarrollado la aplicación fue necesario hacerla pública. Decidí, como mencioné anteriormente, solicitar el dominio [weegoo.com.ar](http://weegoo.com.ar) con el fin de publicarla allí.

Esta sección se dividirá en 3 partes. La primera parte describirá mediante capturas las pantallas de la aplicación la funcionalidad de weegoo para usuarios no registrados, la segunda parte la funcionalidad para un usuario registrado y la tercera parte la funcionalidad de la interface web adaptada para dispositivos móviles.

## **Perspectiva de un usuario no registrado**



¡Empezá a encontrar amigos y compartir lugares y eventos!

¡Regístrate ahora!

o iniciá sesión con



o



#### Lugares más visitados

-  [Antares](#)
-  [Teatro Argentino de La Plata](#)
-  [Teatro Coliseo Podestá](#)
-  [Cervecería Falkner](#)
-  [Catedral de La Plata](#)
-  [Cinema 8](#)
-  [Facultad de Informática - UNLP](#)
-  [Cervecería Alemana](#)
-  [Cinema City](#)
-  [Jardín Zoológico de Buenos Aires](#)

#### Usuarios más activos

-  [patricio](#) (Patricio Mac Adden)
-  [BaRbY](#) (Bárbara Nounou)
-  [nicolaide](#) (Nicolas Espina Tacchetti)
-  [torresmatias](#) (Matías Torres)
-  [barbyteper](#) (barbara teper)
-  [macpiojo](#) (Marcelo Mac Adden)
-  [guiman](#) (Alvaro Fernando Lara)
-  [mil](#) (María Inés Lascano)
-  [ClauB](#) (Claudia Banchoff)
-  [adrianf](#) (adrian feliciani)

#### Eventos próximos

[Almirante Brown vs. Gimnasia](#)



La captura anterior muestra la página de inicio de weegoo. Dicha página de inicio permite que usuarios no registrados se registren o que los usuarios registrados inicien sesión, tanto mediante el manejo de usuarios propio de weegoo como mediante Facebook o Twitter. Además muestra un listado de los lugares más visitados, los usuarios con más actividad y los eventos próximos a ocurrir.

La siguiente captura muestra como un usuario no registrado puede realizar una búsqueda en weegoo utilizando el buscador de la barra superior. La búsqueda realizada es utilizando el término “unlp”.

*Ilustración 51: Búsqueda utilizando el buscador de la barra superior.*

Como se puede observar, la búsqueda utilizando el buscador de la barra superior realiza dicha búsqueda sobre los usuarios, lugares y eventos de weegoo. Para el caso puntual de ésta búsqueda, nos indica que no hay ni usuarios ni eventos relacionados a “unlp” pero si una serie de lugares. Otra característica de esta búsqueda es que sólo retorna 20 resultados como

máximo por tipo, es decir que como máximo obtendremos 20 usuarios, 20 lugares y 20 eventos relacionados con el término buscado. Sin embargo, si quisiéramos refinar la búsqueda deberíamos usar la búsqueda avanzada de usuarios o lugares.

La siguiente captura muestra cómo un usuario no registrado puede realizar una búsqueda utilizando la búsqueda avanzada de lugares.

*Ilustración 52: Búsqueda avanzada de lugares.*

Como se puede apreciar, la búsqueda avanzada de lugares tiene un parecido con la búsqueda mediante el buscador de la barra superior, pero a diferencia de esta última, si el resultado de búsqueda arroja más de 20 resultados, estos resultados son paginados utilizando el link “Ver más”, que agrega la siguiente página de resultados de búsqueda al final de la lista, permitiéndonos ver la lista completa.

Otra característica importante de este buscador es que permite visualizar los resultados de búsqueda como lista (tal cual se ve en la imagen anterior) o como mapa. Esto es de suma importancia, debido a que se trata de lugares turísticos, por lo tanto es imprescindible poder ver la ubicación de los mismos. La siguiente captura muestra los mismos resultados de búsqueda pero visualizados como mapa.

*Ilustración 53: Búsqueda avanzada de lugares, vista de mapa.*

Como se puede apreciar, al igual que en la búsqueda avanzada visualizada como lista, si la búsqueda tiene más de 20 resultados se paginarán, pero a diferencia de la visualización como lista, en vez de agregarse debajo de la lista, se agregan los marcadores en el mapa, indicando la ubicación del lugar. Otra particularidad sobre la visualización como mapa es que los lugares tienen un marcador diferente dependiendo de su categoría.

Hasta ahora se vio cómo realizar búsquedas pero no cómo ver los lugares y su información. Tanto para la búsqueda realizada mediante la barra superior como la búsqueda avanzada de lugares, los resultados de búsqueda son cliqueables, lo que significa que al ser cliqueados iremos a ver la información del lugar. Para el caso de la visualización de los resultados de búsqueda como mapa, los marcadores son cliqueables, en cuyo caso se muestra información

del lugar (y haciendo click en el nombre del lugar podemos ver la información completa del lugar). La siguiente captura ilustra esta característica.

*Ilustración 54: Ventana que muestra información básica del lugar seleccionado.*

La siguiente captura muestra la información de un lugar seleccionado. En este caso, muestra el lugar "Facultad de Informática - UNLP".

*Ilustración 55: Página del lugar "Facultad de Informática - UNLP".*

En la página de un lugar, se puede ver la información básica del mismo:

- nombre
- descripción (si tiene)
- categoría
- sub categoría
- creador
- cantidad de visitantes, es decir, personas que estuvieron ahí (haciendo click aquí se puede ver el listado de visitantes)
- cantidad de posibles visitantes, es decir, personas que quieren ir ahí (haciendo click aquí se puede ver el listado de posibles visitantes)
- ubicación (mediante un mapa)

Y además información más detallada:

- últimos 5 visitantes del lugar (si tiene), con la posibilidad de ver todos

- últimos 5 posibles visitantes del lugar (si tiene), con la posibilidad de ver todos
- últimos 5 eventos que se realizaron allí (o se están por realizar), con la posibilidad de ver todos
- últimos 5 comentarios del lugar, con la posibilidad de ver todos
- últimas 5 imágenes del lugar, con la posibilidad de ver todas (haciendo click en las imágenes se puede ver la misma imagen ampliada y en forma de galería, es decir, pasar a las imágenes anteriores y siguientes)

*Ilustración 56: Página de un usuario de weegoo.*

La captura anterior muestra la página de un usuario weegoo desde la perspectiva de un usuario no registrado. Como se puede observar, muestra datos básicos como:

- nombre y apellido
- cantidad de amigos (haciendo click aquí vemos un listado de sus amigos)
- cantidad de lugares visitados (haciendo click aquí vemos un listado de los lugares visitados, tanto lista como mapa)
- cantidad de lugares por visitar (haciendo click aquí vemos un listado de los lugares por visitar, tanto lista como mapa)
- cantidad de eventos concurridos (haciendo click aquí vemos un listado de los eventos a los que concurrió el usuario)
- cantidad de eventos por concurrir (haciendo click aquí vemos un listado de los eventos a los que el usuario marcó como para concurrir)
- actividad reciente (se muestra en caso de configurar el perfil para hacerlo, lo cual se verá más adelante)

La siguiente captura muestra los lugares visitados por un usuario como lista. Al igual que en la búsqueda, permite visualizar los resultados como lista y como mapa. Además realiza la paginación en caso de tener más de 20 resultados.

*Ilustración 57: Lugares visitados por un usuario, como lista.*

La siguiente captura muestra el mismo listado pero como mapa.

*Ilustración 58: Lugares visitados por un usuario, como mapa.*

La siguiente captura muestra los eventos visitados por un usuario. Como se puede observar, el listado de eventos visitados por un usuario sólo se puede visualizar como lista, para ver donde ocurrió ese evento es necesario hacer click sobre el mismo.



*Ilustración 59: Eventos visitados por un usuario.*

La siguiente captura muestra la página de un evento.

*Ilustración 60: Página de un evento.*

Como se puede observar, la página de un evento es muy similar a la página de un lugar. En la página de un evento, se muestra información básica de éste:

- nombre
- lugar donde ocurrirá/ocurrió
- fecha de inicio
- hora de inicio
- fecha de fin (si tiene)
- hora de fin (si tiene)
- creador
- cantidad de visitantes, es decir, usuarios que concurrieron (haciendo click aquí se puede ver la lista de visitantes)
- cantidad de posibles visitantes, es decir, usuarios que quieren o quisieron concurrir (haciendo click aquí se puede ver la lista de visitantes)

Además, de manera similar a los lugares muestra:

- últimos 5 visitantes del evento (si tiene), con la posibilidad de ver el listado completo
- últimos 5 posibles visitantes del evento (si tiene), con la posibilidad de ver el listado completo
- últimos comentarios del evento, con la posibilidad de ver el listado completo
- últimas 5 imágenes del evento, con la posibilidad de ver todas (haciendo click en las imágenes se puede ver la misma imagen ampliada y en forma de galería, es decir, pasar a las imágenes anteriores y siguientes)

Por último mostraré las páginas de registración e inicio de sesión en weegoo.

La siguiente captura representa la pantalla de registración para usuarios no registrados. Como mencioné anteriormente, esto puede realizarse mediante el manejo de usuarios de weegoo o mediante Facebook o Twitter. Para el caso de escoger la registración nativa de weegoo, es decir, con el manejo de usuarios en el propio weegoo, el usuario debe ingresar la siguiente información básica:

- Nombre
- Apellido
- Email
- Ubicación, que debe ser algo aproximado, por ejemplo: "La Plata"
- Nombre de usuario
- Contraseña (y confirmación de la misma)

*Ilustración 61: Página de registraci3n de weegoo.*

En caso de elegir la registraci3n mediante Facebook o Twitter, el usuario es redirigido a Facebook o Twitter (seg3n haya escogido), donde deber3 autorizar a weegoo para el inicio de sesi3n y actualizaci3n del estado. La siguiente captura muestra esto.

*Ilustración 62: Solicitud de autorización para la registración en weegoo.*

Una vez que se realiza la autorización, al usuario se le solicita que complete ciertos datos:

- Nombre
- Apellido
- Email
- Ubicación
- Nombre de usuario

Mientras que la contraseña no es obligatoria debido a que el usuario iniciará sesión usando Facebook o Twitter (según haya escogido).

La última captura de esta sección muestra la página de inicio de sesión de weegoo. Como se puede observar, hay dos formas de iniciar sesión, mediante la autenticación con weegoo o mediante la autenticación con Facebook o Twitter.

*Ilustración 63: Página de inicio de sesión de weegoo.*

## **Perspectiva de un usuario registrado**

La primera captura de esta sección corresponde a la página de inicio de un usuario registrado. Esta es la primera página que ve un usuario al iniciar sesión y es la misma que ven los demás usuarios cuando acceden al perfil de éste.

Como se puede observar, la información que muestra esta página es muy similar a la que ve un usuario no registrado, a diferencia que ésta muestra los datos completos del usuario y toda la actividad reciente (con la posibilidad de paginarla).

En esta página de inicio, se permite al usuario buscar amigos en weegoo que son amigos en otras redes sociales. Esto se realiza a partir de las conexiones con otras redes sociales. Si un usuario se registró con weegoo mediante Facebook o Twitter tiene la funcionalidad lista para usar, mientras que si se registró mediante weegoo únicamente, necesita conectarse con Facebook o Twitter para contar con la funcionalidad.

Además, al igual que la página de inicio para los usuario no registrados, muestra el listado de lugares más visitados, de usuarios más activos y de eventos próximos.



patricio (Patricio Mac Adden)

Nombre Patricio Mac Adden  
 Email patriciomacadden@gmail.com  
 Ubicación La Plata

7  
Amigos

110  
Lugares  
visitados

1  
Lugares por  
visitar

5  
Eventos  
concurridos

0  
Eventos por ir

+ Nuevo lugar

### Actividad reciente

- Creaste el evento **Almirante Brown vs. Gimnasia.**  
 04 de Noviembre a las 09:26
- Comentaste el evento **Sin ensayo 28/10/2011.**  
 30 de Octubre a las 19:43
- Marcaste el evento **Sin ensayo 28/10/2011** como visitado.  
 30 de Octubre a las 19:43
- Creaste el evento **Sin ensayo 28/10/2011.**  
 30 de Octubre a las 19:43
- Marcaste el lugar **Teatro Ópera (La Plata)** como visitado.  
 30 de Octubre a las 19:42
- Creaste el lugar **Teatro Ópera (La Plata).**  
 30 de Octubre a las 19:42
- Marcaste el evento **Gimnasia vs. Boca Unidos** como visitado.  
 30 de Octubre a las 19:41
- Creaste el evento **Gimnasia vs. Boca Unidos.**  
 30 de Octubre a las 19:41
- barbara teper** y **Matías Torres** son ahora amigos.  
 23 de Octubre a las 16:33
- Patricio Mac Adden** y **Matías Torres** son ahora amigos.  
 23 de Octubre a las 14:09
- Matías Torres** comentó el lugar **Viva Martital.**  
 23 de Octubre a las 12:36
- Matías Torres** marcó el lugar **Viva Martital** como visitado.  
 23 de Octubre a las 12:35
- Matías Torres** comentó el lugar **Antares.**  
 23 de Octubre a las 12:34
- Matías Torres** marcó el lugar **Antares** como visitado.  
 23 de Octubre a las 12:33
- Matías Torres** comentó el lugar **Cervecería Falkner.**  
 23 de Octubre a las 12:33
- Matías Torres** comentó el lugar **Cervecería Falkner.**  
 23 de Octubre a las 12:32
- Matías Torres** comentó el lugar **Cervecería Modelo.**  
 23 de Octubre a las 12:30
- Matías Torres** marcó el lugar **Cervecería Modelo** como visitado.  
 23 de Octubre a las 12:30
- Matías Torres** marcó el lugar **Cervecería Alemana** como para ir.  
 23 de Octubre a las 12:28
- Matías Torres** marcó el lugar **Facultad de Informática - UNLP** como visitado.  
 23 de Octubre a las 12:27

Ver más ↓

### Buscar amigos en otras redes sociales que ya estén usando weegoo

- facebook Eliminar Buscar amigos
- twitter Eliminar Buscar amigos

### Lugares más visitados

- Antares** ✕ Desmarcar como visitado
- Teatro Argentino de La Plata** ✕ Desmarcar como visitado
- Teatro Coliseo Podestá** ✕ Desmarcar como visitado
- Cervecería Falkner** ✕ Desmarcar como visitado
- Catedral de La Plata** ✕ Desmarcar como visitado
- Cinema 8** ✕ Desmarcar como visitado
- Facultad de Informática - UNLP** ✕ Desmarcar como visitado
- Cervecería Alemana** ✕ Desmarcar como visitado
- Cinema City** ✕ Desmarcar como visitado
- Jardín Zoológico de Buenos Aires** ✕ Desmarcar como visitado

### Usuarios más activos

- patricio** (Patricio Mac Adden) ¡Este sos vos!
- BarBY** (Bárbara Nounou) ✕ Eliminar amistad
- nicolaide** (Nicolas Espina Tacchetti) ✕ Eliminar amistad
- torresmatias** (Matías Torres) ✕ Eliminar amistad
- barbyteper** (barbara teper) ✕ Eliminar amistad
- macpiojo** (Marcelo Mac Adden) ✕ Eliminar amistad
- guiman** (Alvaro Fernando Lara) ✕ Eliminar amistad
- mil** (María Inés Lascano) ✕ Eliminar amistad
- CiauB** (Claudia Banchoff) ✕ Cancelar solicitud de amistad
- adrianf** (adrian feliciani) ✕ Cancelar solicitud de amistad

### Eventos próximos

- Almirante Brown vs. Gimn**  Marcar como visitado  Marcar como para ir

Ilustración 64: Página de inicio de un usuario registrado.

La siguiente captura muestra la búsqueda realizada por un usuario registrado para el término “unlp”. Si bien la búsqueda funciona de la misma forma para los usuarios registrados que para los usuarios no registrados, presenta diferentes acciones para el usuario.

*Ilustración 65: Búsqueda usando el buscador de la barra superior.*

Como se puede observar, a diferencia de los usuarios no registrados, para los resultados de búsqueda presenta acciones. En el caso de los lugares, las opciones son:

- Marcar como visitado (en caso de estar marcado como visitado, la acción inversa: Desmarcar como visitado)
- Marcar como para ir (en caso de estar marcado como para ir, la acción inversa: Desmarcar como para ir)

Para el caso de los usuarios (no es el caso de esta búsqueda), se muestra la opción de agregar como amigo (en el caso de que aún no lo sea) o eliminar amigo (en caso de que si lo sea).

Para el caso de los eventos, en cambio, las opciones son las mismas que para los lugares.

La siguiente captura muestra la página de un lugar desde la perspectiva de un usuario registrado.

*Ilustración 66: Página de un lugar desde la perspectiva de un usuario registrado.*

Como se puede apreciar, la vista de un lugar desde la perspectiva de un usuario registrado es la misma que para un usuario no registrado. La única diferencia es que presenta acciones para cada una de las secciones. En el caso de los visitantes y posibles visitantes, permite agregarlos como amigos, eliminar amistad y cancelar la solicitud de amistad según corresponda. Esto



significa que si un usuario no es amigo, puede ser agregado como tal, si un usuario es amigo se puede eliminar amistad mientras que si se solicitó la amistad a un usuario, puede cancelarse la solicitud.

Para el caso de los últimos eventos, comentarios e imágenes, el comportamiento es el mismo que para la perspectiva de un usuario no registrado, excepto que es posible agregar eventos, comentarios o imágenes.

La siguiente pantalla muestra el listado de comentarios de un lugar. En esta lista de comentarios puede verse que es posible crear un nuevo comentario y que los comentarios propios pueden eliminarse.

*Ilustración 67: Comentario de un lugar.*

En esta sección se omiten las capturas de la búsqueda avanzada de lugares. Esto se debe a que la búsqueda avanzada de lugares para los usuarios registrados es la misma que para los usuarios no registrados, a diferencia que se presentan las opciones para los usuarios registrados.

La siguiente captura muestra la página de otro usuario registrado. La página es idéntica a la de un usuario registrado, a excepción de que no muestra los lugares más visitados, ni los usuarios más activos ni los eventos próximos. Tampoco permite que el usuario que inició sesión busque amigos en Facebook o Twitter.



BaRbY (Bárbara Nounou)

Nombre Bárbara Nounou  
Email barbaranounou@gmail.com  
Ubicación La Plata

5

Amigos

58

Lugares  
visitados

1

Lugares por  
visitar

0

Eventos  
concurridos

0

Eventos por ir

4 amigos en común con Bárbara Nounou [Ver todos](#) →

## Actividad reciente

- Patricio Mac Adden creó el evento [Almirante Brown vs. Gimnasia](#).  
04 de Noviembre a las 09:26
- Patricio Mac Adden comentó el evento [Sin ensayo 28/10/2011](#).  
30 de Octubre a las 19:43
- Patricio Mac Adden marcó el evento [Sin ensayo 28/10/2011](#) como visitado.  
30 de Octubre a las 19:43
- Patricio Mac Adden creó el evento [Sin ensayo 28/10/2011](#).  
30 de Octubre a las 19:43
- Patricio Mac Adden marcó el lugar [Teatro Ópera \(La Plata\)](#) como visitado.  
30 de Octubre a las 19:42
- Patricio Mac Adden creó el lugar [Teatro Ópera \(La Plata\)](#).  
30 de Octubre a las 19:42
- Patricio Mac Adden marcó el evento [Gimnasia vs. Boca Unidos](#) como visitado.  
30 de Octubre a las 19:41
- Patricio Mac Adden creó el evento [Gimnasia vs. Boca Unidos](#).  
30 de Octubre a las 19:41
- barbara teper y [Matías Torres](#) son ahora amigos.  
23 de Octubre a las 16:33
- Patricio Mac Adden y [Matías Torres](#) son ahora amigos.  
23 de Octubre a las 14:09
- [Nicolas Espina Tacchetti](#) marcó el lugar [Hospital Italiano de La Plata](#) como visitado.  
22 de Octubre a las 14:59
- [Nicolas Espina Tacchetti](#) marcó el lugar [Hospital de niños de La Plata](#) como visitado.  
22 de Octubre a las 14:59
- [Bárbara Nounou](#) marcó el lugar [Hospital Español de La Plata](#) como visitado.  
17 de Octubre a las 20:40
- [Bárbara Nounou](#) marcó el lugar [Hospital de niños de La Plata](#) como visitado.  
17 de Octubre a las 20:40
- [Bárbara Nounou](#) marcó el lugar [Hospital San Roque de Gonnet](#) como visitado.  
17 de Octubre a las 20:40
- [Bárbara Nounou](#) marcó el lugar [Hospital Rossi de La Plata](#) como visitado.  
17 de Octubre a las 20:40
- [Bárbara Nounou](#) marcó el lugar [Óptica Bermudez](#) como visitado.  
17 de Octubre a las 20:39
- [Bárbara Nounou](#) creó el lugar [Óptica Bermudez](#).  
17 de Octubre a las 20:39
- [Bárbara Nounou](#) marcó el lugar [Hospital Italiano de La Plata](#) como visitado.  
17 de Octubre a las 20:37
- [Bárbara Nounou](#) marcó el lugar [Hospital San Martin \(H.I.G.A\)](#) como visitado.  
17 de Octubre a las 20:37

[Ver más ↓](#)

Ilustración 68: Página de un usuario.

Si la página del usuario pertenece a un amigo del usuario registrado, muestra los amigos en común de ambos. En el caso de la captura anterior, el usuario que inició sesión tiene 4 amigos en común con el usuario de la página que se está viendo.

La siguiente captura muestra los lugares visitados de un usuario. Como puede observarse, tiene las mismas características que para la perspectiva de un usuario no registrado, a diferencia de las acciones que puede realizar un usuario registrado (marcar como visitado, demarcar como visitado, marcar como para ir, desmarcar como para ir).

*Ilustración 69: Lugares visitados por un usuario.*

En esta sección también se omitirán las acciones para visualizar como mapa, ver el listado de eventos y el listado de amigos, ya que son idénticos a la versión para usuario no registrados,

siempre con la diferencia de que esta perspectiva (para usuarios registrados) incluye las acciones que éstos pueden realizar.

La siguiente captura muestra algo que mencioné anteriormente: la búsqueda de amigos en otras redes sociales. Esta búsqueda no incluye invitación a usuarios, sino que muestra aquellos amigos de Facebook o Twitter que ya están usando weegoo.

*Ilustración 70: Amigos que están usando facebook y weegoo.*

La siguiente captura corresponde a la página que permite al usuario registrado modificar su perfil. La edición del perfil permite a un usuario no sólo modificar los datos con los cuales se registró, sino también permite configurar cuestiones de seguridad, como:

- ¿Compartir email? Si se tilda, se compartirá el email con los usuarios no registrados y con aquellos usuarios que no sean amigos
- ¿Compartir ubicación? Si se tilda, se compartirá la ubicación con los usuarios no registrados y con aquellos usuarios que no sean amigos
- ¿Compartir actividad? Si se tilda, se compartirá la actividad con los usuarios no registrados y con aquellos usuarios que no sean amigos
- ¿Postear actividad en facebook? Si se tilda, se compartirá la actividad mediante la actualización del estado en Facebook
- ¿Tweeter actividad? Si se tilda, se compartirá la actividad mediante la actualización del estado en Twitter

*Ilustración 71: Editar perfil.*

Por último en esta sección, mostraré el formulario de creación de lugares. Para crear un nuevo lugar es necesario que el usuario ingrese los siguientes campos:

- Nombre
- Descripción
- Categoría
- Subcategoría
- Ubicación, mediante la interacción con un mapa. Esto significa que el usuario debe mover el marcador hasta la ubicación correcta. El marcador en la ubicación del usuario (la que especifica en el perfil).

*Ilustración 72: Formulario de creación de lugares.*

## **Interface web adaptada para dispositivos móviles**

Por último, en esta sección mostraré capturas de las distintas pantallas de la interface web adaptada para dispositivos móviles.

La primera captura muestra la página de inicio de weegoo en su versión adaptada para dispositivos móviles. Esta página consta únicamente de un formulario de inicio de sesión, debido a que está disponible únicamente para usuarios registrados. Como en la versión web estándar, la versión web adaptada para dispositivos móviles permite que un usuario inicie sesión con usuario y contraseña así como también mediante Facebook o Twitter.



*Ilustración 73: Página de inicio de sesión.*

La segunda captura muestra la página de inicio del usuario. Como se puede observar, el usuario tiene un conjunto de acciones a realizar y se muestra su ubicación en el mapa. Este punto se obtiene utilizando la API de geolocalización de la W3C.

Las acciones que se le presentan al usuario son:

- Buscar lugares
- Navegar por las diferentes categorías
- Crear un nuevo lugar (en la ubicación actual)
- Ver los eventos próximos

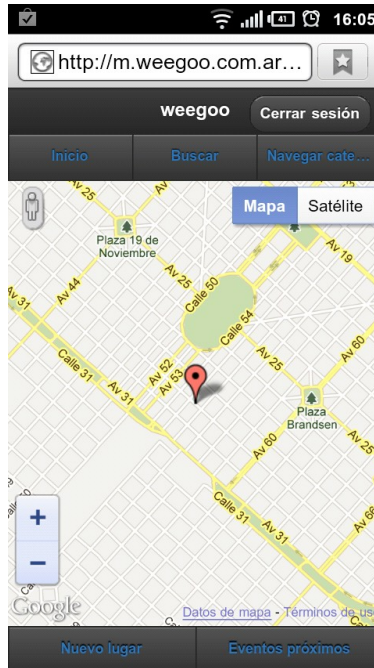


Ilustración 74: Página de inicio.

La próximas capturas muestran una característica disponible sólo para la interface web adaptada para dispositivos móviles de weegoo: la navegación de lugares por categorías.

Como se puede observar, la navegación de lugares por categorías consta de un listado de categorías que indica la cantidad de lugares bajo esa categoría. Si se selecciona una categoría, se pasa a un listado de las subcategorías de esa categoría, donde cada item muestra la cantidad de lugares bajo esa subcategoría.





Ilustración 75: Navegación por categorías.



Ilustración 76: Navegación por subcategorías.

Una vez que se selecciona una subcategoría, se muestran los lugares bajo esa subcategoría. La siguiente captura muestra esto.

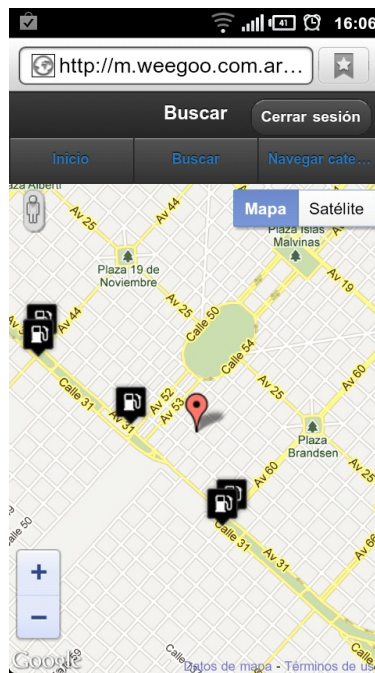
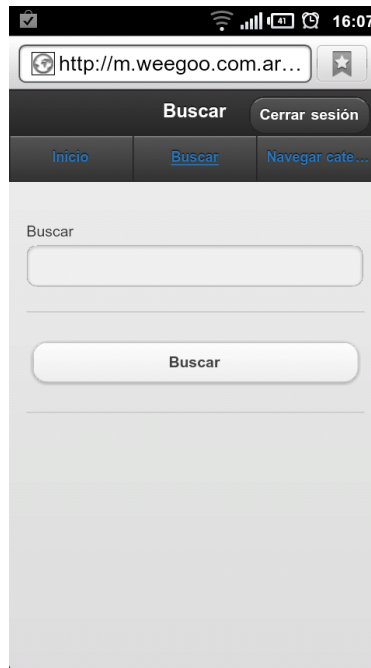


Ilustración 77: Selección de una subcategoría, en particular: Estación de servicio.

La próxima captura muestra la página de búsqueda de la interface web adaptada para dispositivos móviles. Como se puede observar se trata de un formulario de búsqueda sencillo.



*Ilustración 78: Página de búsqueda.*

La siguiente captura muestra los resultados de búsqueda para el término “unlp”. Una particularidad de la búsqueda es que sólo muestra los resultados como mapa, al igual que los resultados de la navegación por categorías y subcategorías.



*Ilustración 79: Página que muestra los resultados de búsqueda.*

La siguiente captura muestra la página de un lugar. A diferencia de la versión web estándar, en la versión web adaptada para dispositivos móviles la página de un lugar no muestra la ubicación en el mapa del lugar, ya que siempre esa ubicación puede verse antes de ver los detalles.

Como se puede ver, la página del lugar muestra la misma información que en la versión web estándar, a diferencia de la disposición de la misma.

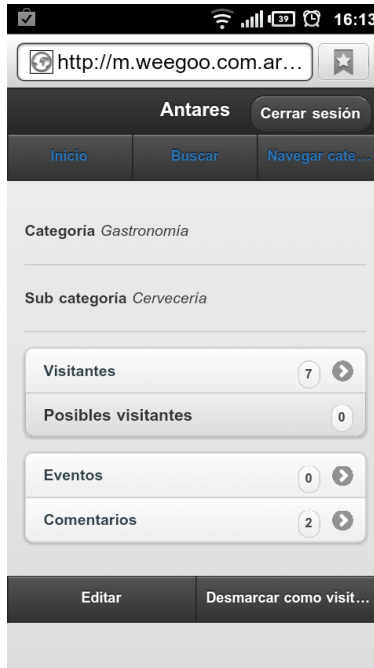


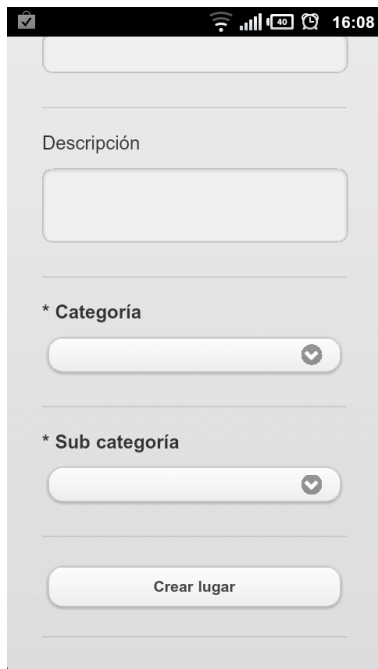
Ilustración 80: Página de un lugar.

La siguiente captura muestra los comentarios de un lugar. Del mismo modo que en la versión web estándar de weegoo, se muestra una lista con los comentarios del lugar más un formulario donde el usuario registrado puede agregar comentarios.



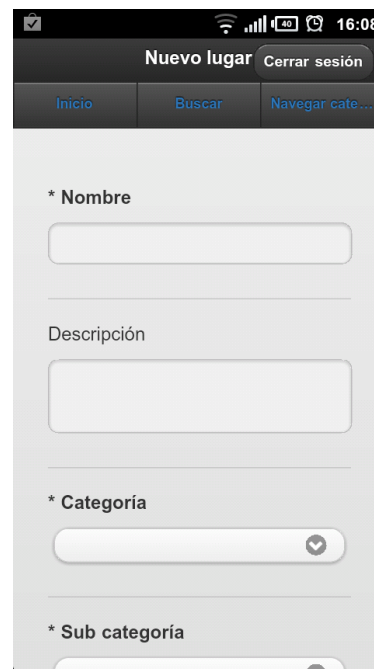
Ilustración 81: Comentarios de un lugar.

Las próximas dos capturas ilustran el formulario para creación de un lugar en la versión web adaptada para dispositivos móviles en weegoo. En la versión adaptada para dispositivos móviles para crear un lugar hay que ingresar nombre, descripción, categoría y subcategoría. La ubicación del lugar se obtiene a partir de la ubicación actual del usuario mediante la API de geolocalización del W3C.



This screenshot shows a mobile web form for creating a location. The form is displayed on a mobile device with a status bar at the top showing the time as 16:08. The form consists of several input fields: a text field for 'Descripción', a dropdown menu for '\* Categoría', and another dropdown menu for '\* Sub categoría'. At the bottom of the form is a button labeled 'Crear lugar'.

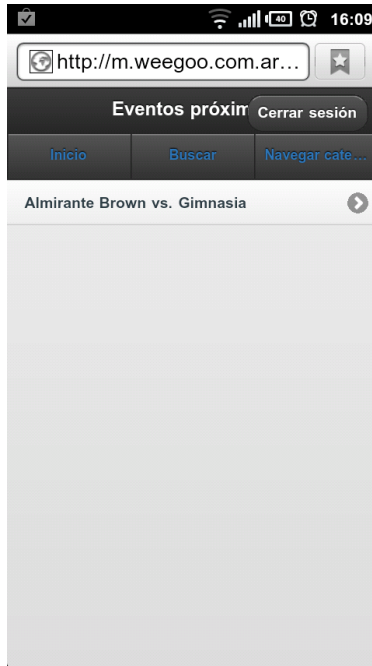
*Ilustración 82: Formulario de creación de lugares. (2)*



This screenshot shows the same mobile web form for creating a location, but from a different perspective. The top of the page features a header with the text 'Nuevo lugar' and 'Cerrar sesión'. Below the header is a navigation bar with three items: 'Inicio', 'Buscar', and 'Navegar cate...'. The main form area contains input fields for '\* Nombre', 'Descripción', '\* Categoría', and '\* Sub categoría'. A button is partially visible at the bottom of the form.

*Ilustración 83: Formulario de creación de lugares. (1)*

La última captura de esta sección muestra el listado de eventos próximos.



*Ilustración 84: Eventos próximos.*

## 6 - Resultados

En este capítulo expondré algunos resultados que surgieron a partir del desarrollo de weegoo. Entre estos resultados se encuentran la experiencia en el uso de GitHub como herramienta para organizar una aplicación open source, validaciones del código HTML generado por weegoo y algunos datos cuantificables sobre el uso de weegoo.

### Uso de GitHub

GitHub cuenta (como mencioné en el capítulo anterior) de un **issue tracker**, lo que significa que cualquier usuario registrado en GitHub puede solicitar nuevas características, reportar errores e incluso aportar nueva funcionalidad o corrección de errores en un proyecto.

Para el caso puntual de weegoo, en GitHub se han solicitado nuevas características y se han reportado errores.

*Ilustración 85: Issues abiertos en Github reportados por distintos usuarios (aunque algunos subidos por mi).*

*Ilustración 86: Issues cerrados.*

GitHub es una excelente herramienta para el desarrollo de una aplicación open source, en primer lugar por su issue tracker, y en segundo lugar porque favorece la colaboración en el desarrollo del proyecto mediante **pull requests**. El **pull request** es un mecanismo donde un usuario registrado en GitHub puede tener su copia del proyecto original, modificarlo (ya sea para agregar/modificar funcionalidad como para corregir errores) y solicitar que el poseedor de la copia principal acepte su modificación, incorporándola a la copia original.

## ¿Cómo contribuir con weegoo?

Es fundamental que los usuarios de weegoo puedan contribuir (de las maneras mencionadas anteriormente) de una forma sencilla.

Como en todo proyecto open source, las formas de contribuir son simples: reportando errores, solicitando nueva funcionalidad y corrigiendo o agregando funcionalidad existente. En weegoo, gracias a la funcionalidad provista de GitHub, un usuario puede reportar errores y/o solicitar nueva funcionalidad mediante el issue tracker mientras que si quiere corregir o agregar funcionalidad puede hacerlo realizando un fork del proyecto y luego enviando un pull request con los cambios realizados.

## ¿Es posible instanciar una copia de weegoo?

weegoo es open source, por lo tanto es posible descargar el código de la red social open source e instalarla de forma privada (o no) para el uso.

Esto puede hacerse descargando el código de la red social comprimida directamente desde GitHub o directamente clonando el repositorio git allí alojado.

## Validación del HTML generado por weegoo



La validación del código HTML de una página web es un paso importante que puede ayudar a mejorar o asegurar su calidad. [W3CVALABOUT]

A continuación se muestran las capturas de las validaciones realizadas sobre una muestra del conjunto de páginas de weegoo, tanto de la interface web como de la interface web adaptada para dispositivos móviles.

*Ilustración 87: Validación de la página de inicio.*

*Ilustración 88: Validación de la página de un lugar.*

*Ilustración 89: Validación de la página de un usuario.*

*Ilustración 90: Validación de la página de inicio de la versión web adaptada para dispositivos móviles (inicio de sesión).*

*Ilustración 91: Validación de la página de inicio de un usuario de la versión web adaptada para dispositivos móviles.*

Como se puede observar en la última captura, la validación no fue realizada mediante la URL de la página de inicio de sesión de un usuario, sino mediante la entrada directa del código HTML de la misma. Esto se debe a que el validador de HTML del W3C no tiene permisos para acceder a dicha URL.

## **Datos sobre uso de weegoo**

Como resultado de la publicación de weegoo, puedo decir que hasta el día de escribir este capítulo, la red social turística cuenta con:

- 15 usuarios
- 212 lugares
- 14 eventos
- 667 actividades realizadas (esto es amistades, visitas a lugares, creación de comentarios, creación de imágenes, etc.)

Teniendo en cuenta que es una red social nueva, y teniendo en cuenta que se necesita un envión con usuarios que lo usen activamente, se puede decir que es un conjunto de datos interesante para llevar poco tiempo de vida, pero suficiente para poder evaluar tanto el uso como su funcionamiento.

## 7 - Conclusiones

En la actualidad, existen numerosas herramientas para desarrollar aplicaciones location aware. Dichas herramientas, que incluyen tanto al hardware (sobre todo GPS y dispositivos móviles con GPS) como al software (APIs para acceder a la información proporcionada por el hardware), fueron analizadas durante el desarrollo de los capítulos 2 a 5. Personalmente, considero que las aplicaciones evaluadas en el capítulo 3, es decir, Gowalla, Foursquare, Google Latitude, Travellerspoint y WAYN, no aprovechan todas estas herramientas para proporcionar información turística relevante a sus usuarios teniendo en cuenta la ubicación en la que se encuentran.

Sin embargo, aunque según mi consideración ninguna de las aplicaciones evaluadas en el capítulo 3 hacen un uso provechoso de las herramientas disponibles para desarrollar aplicaciones location aware orientadas al turismo, de ser open source podrían ser modificadas para agregar funcionalidad relacionada con el turismo. Desafortunadamente, ninguna de las aplicaciones evaluadas son open source.

Estas dos razones motivaron el objetivo de esta tesina de grado. Dicho objetivo fue el desarrollo de una red social location aware orientada al turismo: weegoo, cuya url es <http://weegoo.com.ar>

Durante el desarrollo de weegoo he tenido 2 grandes desafíos. El primero fue el uso de tecnologías no abordadas en ninguna materia a lo largo de la carrera, tales como Ruby y Ruby on Rails (entre otras). Debido a mi experiencia de más de 4 años en el uso de tecnologías web como PHP, Symfony (framework web escrito en PHP), jQuery (framework javascript), y sumado a mi entusiasmo para aprender nuevas tecnologías, la curva de aprendizaje de Ruby y su framework web Ruby on Rails no fue demasiado larga. Además, habiendo investigado sobre Ruby y su ecosistema, llegué a la conclusión de que es un lenguaje muy prometedor, usado por muchas grandes empresas a lo largo del mundo, entre las que se destacan Twitter, Groupon, entre otras. El segundo fue el desafío de llevar a cabo un proyecto open source, donde otras personas pueden colaborar, ya sea mediante reporte de errores como con parches para corrección de errores o inclusión de funcionalidad. Investigué como se llevan a cabo grandes proyectos open source, desde frameworks hasta aplicaciones y librerías, y en la gran mayoría de los casos, GitHub fue la herramienta más utilizada, con lo cual decidí utilizarla para llevar a cabo este proyecto.

Como resultado del desarrollo de esta tesina ha surgido un prototipo de red social location aware orientada al turismo, weegoo, que tiene la particularidad de ser open source. Dicho prototipo provee dos interfaces totalmente funcionales: una interface web y una interface web adaptada para dispositivos móviles, aunque esta última, la versión web adaptada para dispositivos móviles cuenta con una funcionalidad reducida con respecto a la versión web estándar.

## Referencias

[ACTSASFERRET] Repositorio Git de acts\_as\_ferret, [https://github.com/jkraemer/acts\\_as\\_ferret](https://github.com/jkraemer/acts_as_ferret)

[ANDROIDDEVREG] Publicar software en el android market, <https://market.android.com/publish/signup>

[ANDROIDSDK] Android SDK, <http://developer.android.com/sdk/index.html>

[APISM] Eric Karjaluo, A Primer in Social Media,

[APPSTORE] Apple App store, <http://www.apple.com/iphone/built-in-apps/app-store.html>

[APPWORLD] BlackBerry AppWorld, <http://appworld.blackberry.com/webstore/>

[BALSAMIQMOCKUPS] Balsamiq Mockups, <http://balsamiq.com/products/mockups>

[BUNDLER] Bundler, <http://gembundler.com>

[CARRIERWAVE] Repositorio Git de Carrierwave, <https://github.com/jnicklas/carrierwave>

[CSS3BUTTONS] Repositorio Git de CSS3 buttons, [https://github.com/thetron/css3buttons\\_rails\\_helpers](https://github.com/thetron/css3buttons_rails_helpers)

[CVS] CVS, <http://www.cvshome.org/>

[DEVISE] Repositorio Git de devise, <https://github.com/plataformatec/devise>

[FANCYBOX] Fancybox, <http://fancybox.net/>

[FANCYBOXRAILS] Repositorio Git de fancybox-rails, <https://github.com/hecticjeff/fancybox-rails>

[FBGRAPH] Repositorio Git de fb\_graph, [https://github.com/nov/fb\\_graph](https://github.com/nov/fb_graph)

[FERRET] Ferret, <https://github.com/dbalmain/ferret>

[FOURSQUARE] Foursquare, <https://foursquare.com>

[FOURSQUARE10M] 10 millones de usuarios de Foursquare, <https://es.foursquare.com/10million>

[FOURSQUAREANDROID] Aplicacion nativa para Android de Foursquare, [https://market.android.com/details?id=com.joelapenna.foursquared&feature=search\\_result](https://market.android.com/details?id=com.joelapenna.foursquared&feature=search_result)

[FOURSQUAREITUNES] Aplicación nativa para iPhone de Foursquare, <http://itunes.apple.com/us/app/foursquare/id306934924?mt=8>

[GIT] Git, <http://git-scm.com/>

[GITANDROID] Repositorio Git de Android, <http://android.git.kernel.org/>

[GITCAKE] Repositorio Git de CakePHP, <https://github.com/cakephp/cakephp>

[GITDEBIAN] Repositorio Git de Debian, <http://anonscm.debian.org/gitweb/>

[GITECLIPSE] Repositorio GIT de eclipse, <http://git.eclipse.org/c/>

[GITGIT] Repositorio Git de Git, <http://git.kernel.org/?p=git/git.git;a=summary>

[GITHUB] GitHub, social coding, <https://github.com/>

[GITJQUERY] Repositorio Git de jQuery, <https://github.com/jquery/jquery>

[GITLINUX] Repositorio GIT del kernel de Linux, <http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=summary>

[GITPERL] Repositorio Git de Perl, <http://perl5.git.perl.org/perl.git>

[GITPOSTGRESQL] Repositorio Git de PostgreSQL, <http://git.postgresql.org/gitweb/>

[GITRAILS] Repositorio GIT de Ruby on Rails, <https://github.com/rails/rails>

[GITSYMFONY] Repositorio Git de Symfony, <https://github.com/symfony/symfony>

[GMAPSAPIV3] APIv3 de Google Maps, <http://code.google.com/intl/es-ES/apis/maps/documentation/javascript/>

[GOWALLA] Gowalla, <http://gowalla.com>

[GOWALLAANDROID] Aplicación nativa de Gowalla para Android, [https://market.android.com/details?id=com.gowalla&feature=search\\_result](https://market.android.com/details?id=com.gowalla&feature=search_result)

[GOWALLAITUNES] Aplicación nativa de Gowalla para iPhone, <http://itunes.apple.com/us/app/gowalla/id304510106?mt=8>

[GRAVATAR] Gravatar, Globally Recognized Avatar, <http://es.gravatar.com/>

[GROUPON] Groupon, <http://www.groupon.com.ar/>

[HTML5GEO] Anthony T. Holdener III, HTML5 Geolocation, 2011

[IPHONESDK] iPhone SDK, <http://developer.apple.com/devcenter/ios/index.action>

[JQTOUCH] jQTouch, <http://jqtouch.com/>

[JQUERY] jQuery, <http://jquery.com/>

[JQUERYMOBILE] jQuery Mobile, <http://jquerymobile.com/>

[JQUERYMOBILETABLE] Original Graded Browser Matrix, <http://jquerymobile.com/original-graded-browser-matrix/>

[JQUERYRAILS] Repositorio Git de jquery-rails, <https://github.com/rails/jquery-rails>

[JQUERYUI] jQuery UI, <http://jqueryui.com/>

[KAMINARI] Repositorio Git de kaminari, <https://github.com/amatsuda/kaminari>

[LATITUDE] Google Latitude, <https://www.google.com/latitude>

[LATITUDE10M] 10 millones de usuarios de Google Latitude, <http://techcrunch.com/2011/02/01/google-latitude-check-in/>

[LATITUDEANDROID] Aplicación nativa para Android de Google Latitude, [https://market.android.com/details?id=com.google.android.apps.maps&feature=search\\_result](https://market.android.com/details?id=com.google.android.apps.maps&feature=search_result)

[LATITUDEITUNES] Aplicación nativa para iPhone de Google Latitude, <http://itunes.apple.com/us/app/google-latitude/id306586497?mt=8>

[LUCENE] Apache Lucene, <http://lucene.apache.org/>

[MARKET] Android Market, <https://market.android.com/>

[MOBILEFU] Repositorio Git de mobile-fu, <https://github.com/brendanlim/mobile-fu>

[MOBILEHTML5] Mobile HTML5, <http://mobilehtml5.org/>

[MYSQL] MySQL, <http://mysql.com/>

[OMNIAUTH] Repositorio Git de omniauth, <https://github.com/intridea/omniauth>

[PERMALINKFU] Repositorio Git de permalink\_fu, [https://github.com/technoweenie/permalink\\_fu](https://github.com/technoweenie/permalink_fu)

[POSTGRESQL] PostgreSQL, <http://www.postgresql.org/>

[RAILS] Ruby on Rails, <http://rubyonrails.org/>

REST: Roy Thomas Fielding, Architectural Styles and the Design of Network-based Software Architectures,

[RMAGICK] Repositorio Git de RMagick, <https://github.com/rmagick/rmagick>

[RUBY] Ruby Programming Language, <http://www.ruby-lang.org/es/>

[RUBYGEMS] Rubygems, <http://rubygems.org/>

[RVM] Ruby Version Manager, <https://rvm.beginrescueend.com/>

[SENCHATOUCH] Sencha Touch, <http://www.sencha.com/products/touch/>

[SIMPLEFORM] Repositorio Git de simple\_form, [https://github.com/plataformatec/simple\\_form](https://github.com/plataformatec/simple_form)

[SQLITE] SQLite, <http://www.sqlite.org/>  
[SUBVERSION] Subversion, <http://subversion.apache.org/>  
[SYMFONY] Symfony, <http://symfony.com/>  
[TRAVELLERS] Travellerspoint, <http://www.travellerspoint.com/>  
[TRAVELLERSABOUT] Acerca de Travellerspoint, <http://www.travellerspoint.com/about.cfm>  
[TWITTERGEM] Repositorio git del gem twitter, <https://github.com/jnunemaker/twitter>  
[W3CGEOAPI] API de Geolocalización del W3C, <http://dev.w3.org/geo/api/spec-source.html>  
[W3CVLABOUT] Acerca del Validador HTML del W3C, <http://validator.w3.org/about.html>  
[W3CWAIARIA] Accessible Rich Internet Applications (WAI-ARIA) 1.0,  
<http://www.w3.org/TR/wai-aria/>  
[WAYN] WAYN, <http://www.wayn.com/>  
[WAYNABOUT] Acerca de WAYN, <http://www.wayn.com/aboutus>  
[WAYNANDROID] Aplicación nativa para Android de WAYN,  
[https://market.android.com/details?id=com.wayn.android.wayndroid&feature=search\\_result](https://market.android.com/details?id=com.wayn.android.wayndroid&feature=search_result)  
[WAYNITUNES] Aplicación nativa para iPhone de WAYN,  
<http://itunes.apple.com/es/app/wayn/id373563898?mt=8>  
[WGS84] National Imagery and Mapping Agency, <http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>

## Índice de ilustraciones

Ilustración 1: Ericsson Hotline, comercializado entre 1986 y 1992.....	6
Ilustración 2: Motorola dynaTAC 8000x, comercializado entre 1983 y 1994.....	7
Ilustración 3: Nokia 2110, en venta a partir de 1994.....	7
Ilustración 4: Motorola RAZR V3. Comercializado desde noviembre de 2004.....	8
Ilustración 5: Motorola Backflip, lanzado en 2010.....	9
Ilustración 6: iPhone 4, lanzado en 2010.....	10
Ilustración 7: iPad de Apple. Lanzado en enero de 2010.....	11
Ilustración 8: Motorola XOOM, lanzada en enero de 2011.....	12
Ilustración 9: Triangulación radial usando 3 bases. El punto rojo representa al receptor.....	14
Ilustración 10: Triangulación direccional usando 2 bases. El punto rojo representa al receptor.	14
Ilustración 11: Primer paso de la trilateración.....	16
Ilustración 12: Segundo paso de la trilateración.....	16
Ilustración 13: Tercer paso de la trilateración.....	17
Ilustración 14: Cuarto paso de la trilateración.....	17
Ilustración 15: Pantalla inicial de Gowalla.....	21
Ilustración 16: Passport de Gowalla.....	21
Ilustración 17: Pantalla de búsqueda de amigos en Gowalla. ....	22
Ilustración 18: Pantalla de trips de Gowalla.....	23
Ilustración 19: Versión adaptada para dispositivos móviles.....	24
Ilustración 20: Gowalla para Android.....	25
Ilustración 21: Pantalla principal de Foursquare.....	26
Ilustración 22: Pantalla que muestra los Badges conseguidos.....	26
Ilustración 23: Un lugar en Foursquare. Muestra los tips, el Mayor y las fotos del mismo.....	27
Ilustración 24: Pantalla que muestra las listas de un usuario.....	27
Ilustración 25: Conexión con otras aplicaciones.....	28
Ilustración 26: Versión adaptada para dispositivos móviles.....	29
Ilustración 27: Aplicación para Android.....	30
Ilustración 28: Pantalla principal de Google Latitude. Muestra la ubicación actual del usuario y de sus contactos.....	31
Ilustración 29: Historial de movimientos en Google Latitude.....	31
Ilustración 30: Versión adaptada para dispositivos móviles.....	32
Ilustración 31: Aplicación para Android de Google Latitude.....	33
Ilustración 32: Pantalla principal de Travellerspoint.....	34
Ilustración 33: “Plan your journey”.....	35
Ilustración 34: Página de inicio de usuario de WAYN.....	36
Ilustración 35: Cosas para hacer en Ciudades.....	36
Ilustración 36: Aplicación para Android.....	37
Ilustración 37: Tabla de soporte de plataformas de dispositivos móviles. A = Alta calidad - B = Calidad media - C = Baja calidad.....	49
Ilustración 38: interface NavigatorGeolocation.....	51
Ilustración 39: interface Geolocation.....	51
Ilustración 40: interface PositionOptions.....	52

Ilustración 41: interface Position.....	53
Ilustración 42: interface Coordinates.....	53
Ilustración 43: interface PositionError.....	54
Ilustración 44: Pagina de inicio de un usuario.....	55
Ilustración 45: Búsqueda de personas (para usuarios autenticados).....	55
Ilustración 46: Búsqueda de lugares (para usuarios autenticados).....	56
Ilustración 47: Repositorio de weegoo en GitHub.....	59
Ilustración 48: Página de registro de weegoo en Facebook.....	59
Ilustración 49: Página de registro de weegoo en Twitter.....	60
Ilustración 50: Página de inicio de weegoo.....	65
Ilustración 51: Búsqueda utilizando el buscador de la barra superior.....	66
Ilustración 52: Búsqueda avanzada de lugares.....	67
Ilustración 53: Búsqueda avanzada de lugares, vista de mapa.....	68
Ilustración 54: Ventana que muestra información básica del lugar seleccionado.....	69
Ilustración 55: Página del lugar “Facultad de Informática - UNLP”.....	70
Ilustración 56: Página de un usuario de weegoo.....	71
Ilustración 57: Lugares visitados por un usuario, como lista.....	72
Ilustración 58: Lugares visitados por un usuario, como mapa.....	73
Ilustración 59: Eventos visitados por un usuario.....	74
Ilustración 60: Página de un evento.....	74
Ilustración 61: Página de registración de weegoo.....	76
Ilustración 62: Solicitud de autorización para la registración en weegoo.....	77
Ilustración 63: Página de inicio de sesión de weegoo.....	78
Ilustración 64: Página de inicio de un usuario registrado.....	79
Ilustración 65: Búsqueda usando el buscador de la barra superior.....	80
Ilustración 66: Página de un lugar desde la perspectiva de un usuario registrado.....	81
Ilustración 67: Comentario de un lugar.....	82
Ilustración 68: Página de un usuario.....	83
Ilustración 69: Lugares visitados por un usuario.....	84
Ilustración 70: Amigos que están usando facebook y weegoo.....	85
Ilustración 71: Editar perfil.....	86
Ilustración 72: Formulario de creación de lugares.....	87
Ilustración 73: Página de inicio de sesión.....	88
Ilustración 74: Página de inicio.....	89
Ilustración 75: Navegación por categorías.....	90
Ilustración 76: Navegación por subcategorías.....	90
Ilustración 77: Selección de una subcategoría, en particular: Estación de servicio.....	90
Ilustración 78: Página de búsqueda.....	91
Ilustración 79: Página que muestra los resultados de búsqueda.....	92
Ilustración 80: Página de un lugar.....	93
Ilustración 81: Comentarios de un lugar.....	93
Ilustración 82: Formulario de creación de lugares. (2).....	94
Ilustración 83: Formulario de creación de lugares. (1).....	94
Ilustración 84: Eventos próximos.....	95



Ilustración 85: Issues abiertos en Github reportados por distintos usuarios (aunque algunos subidos por mi).....	96
Ilustración 86: Issues cerrados.....	97
Ilustración 87: Validación de la página de inicio.....	98
Ilustración 88: Validación de la página de un lugar.....	98
Ilustración 89: Validación de la página de un usuario.....	98
Ilustración 90: Validación de la página de inicio de la versión web adaptada para dispositivos móviles (inicio de sesión).....	98
Ilustración 91: Validación de la página de inicio de un usuario de la versión web adaptada para dispositivos móviles.....	99