

Daniel Martín Yerga

Investigador del Departamento de Química Física y Analítica de la Universidad de Oviedo; Ganador del IV Concurso Universitario de Software Libre en la categoría "Mejor proyecto de movilidad"

<dyerga@gmail.com>

1. Introducción

CasualServices es una aplicación para realizar búsquedas de servicios, negocios y lugares de interés. El usuario puede obtener información sobre estos servicios, como, por ejemplo, la dirección, teléfono, localización en un mapa, etc. A su vez, el usuario puede compartir esta información con sus contactos de diferentes maneras.

Actualmente, CasualServices utiliza dos fuentes principales de información, 11870.com [1] y Google Local. 11870.com es una comunidad web donde los usuarios comparten información sobre negocios, comentarios sobre sus experiencias en esos lugares, fotografías, videos, etc. Google Local es la parte de Google Maps que tiene información sobre diferentes negocios.

CasualServices es una aplicación que funciona sobre la plataforma Maemo 5 [2], disponible para el Nokia N900.

2. Razón de ser

Este tipo de aplicaciones son realmente útiles cuando una persona se encuentra en una ciudad desconocida, y no sabe muy bien dónde acudir para temas de ocio, alojamiento, etc. También es muy interesante para conocer nuevos lugares recomendados por otros usuarios. Y qué mejor manera que tener todo esto disponible en la palma de la mano, en un dispositivo portable y que siempre se lleva encima, como es un teléfono móvil.

Aunque no es una idea nueva, y hay bastantes aplicaciones del mismo estilo en otras plataformas diferentes a Maemo, rara vez el código fuente de esas aplicaciones está disponible. La mayoría de los desarrolladores de aplicaciones para móviles no escriben software libre.

En el momento que se decidió empezar con el desarrollo, no había ninguna aplicación similar disponible para Maemo.

Así el mayor objetivo del proyecto era cubrir este hueco de funcionalidad que existía en Maemo, y a la vez realizar una aplicación que fuera útil para los usuarios, y que mejor forma que hacerlo con una aplicación de software libre, donde el código fuente estuviera disponible para otros desarrolladores y así poder ser mejorada fácilmente con ayuda de la comunidad.

CasualServices: Busca y comparte tus servicios favoritos

Resumen: en los últimos años, la integración de sistemas de posición global en los dispositivos móviles está siendo una gran revolución. CasualServices es una aplicación que hace uso de estos sistemas, permite buscar información sobre negocios o servicios en el entorno de los usuarios desde un dispositivo móvil. Ágil, simple y siempre disponible, CasualServices hace muy fácil el descubrimiento de nuevos lugares que quizás puedan cambiar la vida de sus usuarios.

Palabras clave: buscador, directorio, información geográfica, Maemo, MeeGo, negocios, servicios.

Autor

Daniel Martín Yerga es Licenciado en Química, estudiante de postgrado e investigador colaborador en el grupo de Inmuno electroanálisis del Departamento de Química Física y Analítica de la Universidad de Oviedo. En 2006 comenzó su interés por la programación en dispositivos móviles, especialmente aquellos basados en software libre, como el sistema operativo Maemo. Desde entonces ha desarrollado aplicaciones interesantes a la par que populares para esta plataforma, como, por ejemplo, MaStory, StockThis, PyRecipe o Copernicium. Pero sin duda, su aplicación más exitosa ha sido CasualServices, proyecto galardonado con el Primer Premio en la categoría "Mejor proyecto movilidad" en el IV Concurso Universitario de Software Libre.

3. Estado del proyecto

En la actualidad CasualServices se encuentra en un estado funcional, es decir, hay una versión estable publicada y disponible para el sistema operativo Maemo 5 (ver **figura 1**). Por lo tanto, existe una base de usuarios que utilizan la aplicación.

En la actualidad ya existen otras aplicaciones para Maemo que poseen la misma función que CasualServices.

Uno de los aspectos más importantes de una aplicación para una plataforma móvil es la interfaz de usuario y la experiencia que tiene el usuario al utilizarla. La interfaz de usuario es algo donde CasualServices se ha diferen-

ciado de otras aplicaciones. Se ha conseguido llegar a un nivel donde la interfaz de usuario es muy simple, intuitiva y fácil de utilizar, pero que tiene todas las características necesarias para ser útil a los usuarios. El resto de aplicaciones de este tipo existentes en Maemo no llegan a este nivel de compromiso entre funcionalidad y experiencia de usuario, y o bien no poseen toda la funcionalidad necesaria, o bien la interfaz de usuario está muy cargada, haciendo la aplicación más difícil de utilizar, pudiendo incluso en algunos casos llegar a confundir al usuario.

De este modo, CasualServices se ha convertido en la aplicación de este estilo más descargada por los usuarios de Maemo (más de



Figura 1. Versión estable de Casual Services funcionando sobre Maemo 5.

35.000 descargas en octubre de 2010 desde los repositorios).

En cuanto al desarrollo del proyecto, para llegar a este punto ha pasado por un camino de altos y bajos, ilusiones y desilusiones, como cualquier otro proyecto de software hecho como un hobby, pero el resultado conseguido es bastante satisfactorio y el trabajo ha dado sus frutos, tanto en el Concurso Universitario de Software Libre, como con los comentarios recibidos por parte de los usuarios de la aplicación.

Para la programación de CasualServices se utilizó el lenguaje Python [3]. Mientras que este lenguaje no está oficialmente soportado por Maemo, la comunidad está manteniendo las bibliotecas, y se encuentra en un gran estado de calidad y de características, con una fácil instalación por parte del usuario.

La interfaz utiliza las bibliotecas gráficas Hildon [4], que son una extensión de GTK+ [5] adaptada para dispositivos móviles como el Nokia N900. Estas bibliotecas son las utilizadas para la creación de la interfaz de usuario. La otra biblioteca importante en el desarrollo de CasualServices es *osm-gps-map* [6]. Es un añadido a GTK+ para la visualización de mapas como OpenStreetMap o Google Maps. Esto es necesario para la localización de los servicios en un mapa, de manera que los usuarios tengan una fácil visualización de donde se encuentran los servicios.

Básicamente, el proyecto se enfocó a tener algo minimalista pero funcional en poco tiempo y a partir de ese momento, añadir nuevas características a la base atómica de la aplicación.

La arquitectura a seguir en el desarrollo de la aplicación estuvo clara desde el primer momento, y no era más compleja que tener un módulo definiendo la interfaz de usuario y distintos módulos separados para las demás funciones de la aplicación. El objetivo a seguir era que el módulo de la interfaz de usuario fuera todo lo abstracto posible, es decir, separar la lógica de la aplicación de la interfaz de usuario, y conseguir así que cambios en una de las partes no implicasen cambios en la otra. El objetivo se cumplió en gran medida, y es posible añadir nuevos servicios a CasualServices sin tener que hacer prácticamente cambios en el resto del código.

Los diferentes módulos que componen la aplicación son los siguientes:

- *cservices*: define todos los elementos gráficos de la aplicación que son los expuestos al usuario (la interfaz de usuario).
- *settings*, *agenda*, *portrait*: manejan funciones básicas de la aplicación que interactúan con características del sistema opera-

tivo. Son funciones como la administración de las preferencias de la aplicación, la interacción de CasualServices con la agenda de contactos del usuario, e incluso el soporte de la aplicación de los diferentes modos de visualización (vertical, horizontal) utilizando el acelerómetro del dispositivo móvil.

- *services.glocal*: este módulo interactúa con Google Local para la obtención de datos sobre los servicios que busca el usuario a través de la aplicación. Básicamente, consiste en preguntar a la base de datos de Google Local sobre los términos que el usuario ha buscado con diferentes opciones condicionales (en qué ciudad, utilizando la localización del GPS o no, etc.). Google Local devuelve una respuesta en formato JSON, que es traducida de manera que la interfaz de usuario muestre los datos más importantes al usuario.

- *services.once*: se trata del módulo que interactúa con la API de 11870.com. El funcionamiento es muy similar al módulo de Google Local, la única diferencia es que algunas funciones de 11870.com necesitan autenticación y la respuesta en lugar de ser en formato JSON es en formato Atom.

Durante el desarrollo se intentó utilizar otras funciones más avanzadas de la API de 11870.com, como por ejemplo la adición de nuevos servicios por parte del usuario desde la aplicación, o comentar en los servicios ya existentes. Esto no se consiguió realizar ya que la aplicación no conseguía autenticarse con la API para usar estos métodos específicos. No se pudo saber de dónde provenía el problema, si era de parte del cliente o del servidor. Se decidió dejar la solución de estos problemas para el futuro y seguir mejorando CasualServices en otras áreas diferentes.

Pero el punto más importante a destacar en el estado del proyecto se encuentra en la versión en desarrollo. Esta versión seguirá básicamente la misma arquitectura que la versión estable pero se está reescribiendo completamente, según explicaremos en la sección siguiente. Por lo tanto, la versión estable se encuentra en modo mantenimiento, un estado en el que se publican nuevas versiones mínimas donde sólo se arreglan errores.

4. Planes futuros

Como ya hemos dicho, en la actualidad existen otras aplicaciones con características similares en la plataforma Maemo, así que al ser la competencia mayor, el trabajo deberá ser mayor para que CasualServices destaque sobre las demás aplicaciones.

El primer punto a destacar (y el más importante) entre los planes futuros es la reescritura de CasualServices con tecnologías modernas e innovadoras. Este trabajo ya se ha comenzado pero está en una etapa inicial todavía. Las tecnologías elegidas para la reescritura

son Qt [7] (para la lógica de la aplicación) y Qt Quick [8] (para la interfaz de usuario).

Una reescritura total, usando diferentes tecnologías de un proyecto de un año de antigüedad, es algo serio y se necesitan motivos muy poderosos para que sea una decisión razonable. En el mundo móvil siempre hay que estar en la cresta de la ola, innovando, es necesario estar un paso por delante de tus competidores, por lo que se necesita usar tecnologías con una gran versatilidad que ayuden a cumplir con estos objetivos.

Aparte de esto, las razones más importantes para la reescritura son las siguientes:

- El sistema operativo de Nokia, Maemo, basado en las bibliotecas Gtk y su extensión Hildon, ha sido fusionado con Moblin (un sistema de Intel), en una nueva plataforma móvil llamada MeeGo [9]. La interfaz de usuario de MeeGo está escrita en Qt. Por lo tanto, el siguiente paso de CasualServices tendría que ser su correcto funcionamiento en este nuevo sistema y el cambio de tecnologías usadas en CasualServices pareció una buena idea en este sentido. Como complemento, Maemo también soporta Qt, en su versión 4.7, por lo que la nueva versión de la aplicación escrita en Qt y Qt Quick funciona perfectamente, también, en Maemo.

- El lenguaje utilizado por Qt Quick es un lenguaje muy similar a Javascript, llamado QML [10]. Con Qt Quick se está escribiendo la interfaz de usuario de CasualServices. Es una tecnología moderna, que soporta animaciones, efectos varios, siendo posible modificar hasta el más mínimo detalle de la interfaz de usuario, y con ello conseguir un aspecto visual muy llamativo para el usuario. Y, todo esto es posible hacerlo con un lenguaje de programación muy simple, fácil de aprender, y que hace muy sencilla la interacción de los desarrolladores con los diseñadores. Con todas estas ventajas de Qt Quick, siendo una tecnología moderna, muy portable entre plataformas, y con un gran futuro en el horizonte, el paso más lógico era que CasualServices hiciera uso de esta nueva tecnología.

- Qt está empezando a ser una gran plataforma para desarrollar aplicaciones móviles. Tanto MeeGo como Symbian soportan Qt. Y aunque Symbian está siendo muy criticado últimamente, no hay que olvidar que sigue siendo la plataforma móvil que más vende y más número de usuarios tiene. Esto significa que una aplicación escrita en Qt (como será CasualServices), fácilmente portable entre estas plataformas, va a estar disponible para una cantidad inmensa de posibles usuarios.

- Si todo esto fuera poco, por parte de la comunidad se está trabajando para hacer funcionar Qt en otras plataformas como Android e iOS, con resultados bastante satisfactorios hasta ahora. Quizás en un futuro, CasualServices pueda estar disponible para la mayoría de sistemas operativos móviles.



Figura 2. Versión en desarrollo de Casual Services funcionando sobre MeeGo.

Esta versión en desarrollo todavía está en un estado inestable, y no posee las mismas funcionalidades que la versión estable. Necesita diferentes mejoras hasta que pueda ser considerada la publicación de una nueva versión para el público general (ver figura 2).

De todas maneras, se deberán implementar otra serie de características tanto en la aplicación en sí misma, como en la administración del proyecto, entre las que destaca añadir más fuentes donde obtener resultados, por ejemplo Yelp [11], Yahoo Local o Qype [12]. Esto es importante, ya que no todas las bases de datos tienen los mismos servicios, por lo que una manera de diferenciarse positivamente de otras aplicaciones parecidas es dar la mayor cantidad de resultados posibles al usuario.

Por otro lado, dado que la API de 11870.com se ha actualizado a una nueva versión, será necesario actualizar la aplicación para utilizar esta API, y será una buena oportunidad para implementar esas características de 11870.com que no se consiguieron implementar durante el desarrollo inicial del proyecto.

Se hará un esfuerzo para conseguir que haya una mayor comunidad alrededor de Casual Services, especialmente de desarrolladores para aumentar la rapidez del desarrollo, pero también de otro tipo de contribuyentes. En este aspecto, se espera que las nuevas tecnologías utilizadas por CasualServices puedan ayudar a crear una mayor comunidad, ya que será más fácil contribuir y se espera que el número de usuarios de la aplicación se multiplique al estar disponible en un mayor número de plataformas.

En referencia a esto último, se hará todo lo posible para que CasualServices esté disponible en la mayor cantidad de lenguajes, con la ayuda de los usuarios.

Visto el auge de las redes sociales en estos momentos, se espera implementar algunas características más sociales en la aplicación, como, por ejemplo:

- Nuevas formas de compartir la información de un servicio con los contactos, entre ellas: correo electrónico, Facebook, SMS, incluso podría ser interesante alguna forma de interacción con Foursquare y Gowalla.
- Visualización de los comentarios que los usuarios escriben sobre los servicios.
- Comentarios por parte de los usuarios sobre los servicios.

5. Colabora con Casual Services

CasualServices es una aplicación de software libre, donde el código fuente está disponible, y se desarrolla de una forma abierta y pública, por lo tanto, cualquier aportación es bienvenida.

Hay muchos aspectos donde es posible colaborar con el proyecto, siendo los principales:

Código: lo más usual para colaborar en un proyecto de software, es escribir código. CasualServices está siendo escrito en Qt y Qt Quick. Tecnologías atractivas y en un gran momento en la actualidad, fáciles de aprender, y fáciles de escribir. Estoy seguro de que hay personas en estos momentos que están pensando por dónde empezar a utilizar estas tecnologías. Pues bien, lo mejor sería ayudar a CasualServices. ¡Te esperamos!

Interfaz de Usuario y Gráficos: un elemento clave en el desarrollo de aplicaciones móviles es la interfaz de usuario y la interacción del usuario con la aplicación. A CasualServices le ayudarían mucho colaboradores expertos en este tema, para la creación de una interfaz de usuario que sea útil, pero fácil de utilizar y lo más divertida posible para los usuarios. En este aspecto, también serían de gran ayuda diseñadores gráficos, para la creación de las diferentes imágenes y elementos gráficos usados en la aplicación. Está comprobado que una aplicación cuanto más atractiva a la vista más va a llamar la atención del usuario y más posibilidades tiene de ser utilizada y de destacar sobre el resto.

Otros: cualquier otra aportación al proyecto es más que bienvenida, por ejemplo, apor-

tes en la documentación, pruebas de la aplicación en las diferentes plataformas, búsqueda y reporte de errores, o también, la traducción a cualquier idioma.

Desde CasualServices se ofrece un interesante desarrollo, utilizando tecnologías punteras y actuales (Qt Quick). Esto hace que vaya a estar disponible en varios sistemas operativos en el futuro, como por ejemplo MeeGo o Symbian, e incluso en sistemas operativos de escritorio como GNU/Linux y Microsoft Windows. Esto significa un gran número de posibles usuarios para un proyecto excitante. Entonces, ¿a qué esperas?

Enlaces útiles

Código: <<http://gitorious.org/casual-services/>>.

Web: <<http://cusl4-cservices.forja.rediris.es/>>.

Traducciones: <<http://www.transifex.net/projects/p/cservices/>>.

Referencias

[1] 11870.com. <<http://11870.com/>>.

[2] Maemo. <<http://maemo.nokia.com>>

[3] Python Programming Language. <<http://python.org>>.

[4] Hildon application framework. <<http://en.wikipedia.org/wiki/Hildon>>.

[5] The GTK+ Project. <<http://www.gtk.org>>.

[6] osm-gps-map. GTK+ Widget for displaying maps, <<http://nzjrs.github.com/osm-gps-map>>.

[7] Qt. Cross platform application and UI framework, <<http://qt.nokia.com>>.

[8] Qt Quick. Design and code together, <<http://qt.nokia.com/products/qt-quick>>.

[9] MeeGo. MeeGo, a Linux Foundation project, <<http://meego.com>>.

[10] Qt. Introduction to the QML language, <<http://doc.qt.nokia.com/4.7/qdeclarativeintroduction.html>>.

[11] Yelp. To connect people with great local businesses, <<http://www.yelp.com>>.

[12] Qype. Find it. Share it, <<http://www.qype.co.uk>>.