UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

Título: SIU Guaraní Mobile
Autores: Luis Manuel Sala Espiell
Director: Lic. Javier Díaz
Codirector: Lic. Ana Paola Amadeo
Asesor profesional: Lic. María Alejandra Osorio
Carrera: Licenciatura en sistemas

Resumen

SIU Guaraní Mobile es una aplicación desarrollada para dispositivos móviles, la cual permitirá tanto a alumnos como a docentes operar sobre el sistema SIU Guaraní sin la necesidad de estar frente a una computadora, con la facilidad, disponibilidad y ubiquidad que un dispositivo móvil puede otorgar.

Desde sus inicios se pensó para brindar un mejor servicio a los estudiantes sobre las cuestiones administrativas y poder estar siempre presente en el día a día del estudiante, incorporando aspectos sociales como visualizar compañeros cercanos en un mapa cuando se está próximo a la Facultad, teniendo la posibilidad de dejar un mensaje para poder ser visualizado por los mismos.

Palabras Claves

Siu Guaraní, Android, API, Celular, REST, Angular, Aplicaciones Móviles, Nativa, Web, Híbrida, SiuBack, SiuFront, Javascript, Java, MVC, Maven, Spring, Bower.

Trabajos Realizados

Se generaron dos aplicaciones. Una API REST que expone servicios para poder operar con SIU Guaraní, estructurada de manera tal que pueda ser fácilmente conectada con el sistema en producción. Otra aplicación, en este caso para Android, que consume estos servicios y permite al usuario realizar las operaciones más comunes sobre el sistema. Por último, se realizó una encuesta de satisfacción para evaluar el impacto que tiene una aplicación de estas características en los usuarios finales.

Conclusiones

El futuro de las aplicaciones está centrado en el desarrollo para dispositivos móviles. Dada la necesidad de obtener resultados rápidos, los desarrollos se inclinan hacia aplicaciones web o híbridas. Las APIs REST abren nuevas posibilidades facilitando la comunicación entre aplicaciones, ampliando y mejorando servicios. SIU Guaraní para móviles es una necesidad, y los focus groups implementados permitieron identificar mejoras y nuevas posibilidades.

Trabajos Futuros

Desarrollar una versión nativa para Android.

Conectar la aplicación con la API que brinde SIU Guaraní.

Contemplar el perfil docente e incluir nuevas características orientadas a explotar la funcionalidad social que la aplicación ofrece.

Fecha de la presentación: **Marzo 2016**

Índice

Índice....1
Prólogo....3
Introducción....4
Historia....4
Los dispositivos móviles....5
Android....7
Evolución....8
Software Libre....11
SIU Guaraní....14
Motivación....16
Aplicaciones Móviles....18
Tipo de aplicaciones según su desarrollo....20
Elección del Tipo de Aplicación....23
Aplicación Nativa....24
Aplicación Híbrida....25
Aplicaciones Desarrolladas....26
Siuback....26
API....26
REST....27
Servicios Disponibles....28
SIU Back - Arquitectura de la Solución....36
Servidor....39
OpenShift de RedHat....39
SiuFront....40
Pantallas creadas....40
SIU Front - Arquitectura de la solución....45
Análisis de Uso....48
Formulario de Análisis de Uso....49
Respuestas....54
Conclusión sobre la encuesta....58

Conclusión.....59

Trabajo a futuro.....61

Referencias y Bibliografía.....62

Prólogo

El trabajo de grado SIU Guaraní Mobile responde a una necesidad evidenciada por los estudiantes durante su paso por la Facultad para agilizar la gestión administrativa y estar en contacto con sus compañeros de estudio en forma dinámica, acompañado por el crecimiento exponencial del uso de celulares, su potencialidad y la versatilidad de apps disponibles en el mercado actual.

El siguiente informe presenta una síntesis del trabajo de investigación y desarrollo realizado. El documento está organizado en 4 secciones. La primera sección incluye una introducción y marco teórico para el trabajo realizado: la tecnología móvil, Android, SIU Guaraní como sistema de gestión académica instalado en todas las Facultades de la UNLP y en la mayoría de las universidades nacionales en todo el país. La segunda sección explica la motivación que llevó a realizar el trabajo en sí, analizando las necesidades observadas y planteando alternativas de solución a través de la tecnología móvil, aplicaciones nativas e hibridas, necesidades de contemplar aspectos sociales. La tercera sección presenta el desarrollo mismo, abordando los primeros pasos hechos para la elección del tipo de aplicación a desarrollar, la aplicación de backend que ofrece los servicios a través de una API REST que consume la app desarrollada. Se describen los servicios necesarios para una aplicación de este tipo y pone a disposición un catálogo a implementar por parte del equipo desarrollador de la solución SIU Guaraní. Una vez ya seleccionado el tipo de aplicación comienza la cuarta parte donde se explica las aplicaciones desarrolladas de manera más detallada, incluyendo la arquitectura y capturas de pantallas. Luego, una quinta sección en la cual se plasma el resultado de un análisis de uso de la aplicación frente a un grupo de personas que usan el sistema actual en distintas universidades nacionales. Finalmente, el documento termina con una conclusión sobre el trabajo realizado, junto a todo lo que se tuvo que aprender para realizarlo, los puntos fuertes, puntos a mejorar y que serían incluidos en un trabajo a futuro.

SECCIÓN 1

Introducción

Historia

Al echar la vista atrás y observar el desarrollo tecnológico que ha experimentado la Humanidad desde mediados del siglo XX hasta hoy, no cabe duda de que más que un avance se ha producido una verdadera revolución. El descubrimiento de la informática, su aplicación paulatina en todo tipo de áreas de conocimiento y de producción, así como su introducción en el común de la población a través de todo tipo de componentes ha cambiado nuestra sociedad y nuestra economía más rápido que cualquier otro hecho o descubrimiento anterior. [01]

La computadora u ordenador es uno de los inventos que mejor resume la nueva situación tecnológica. Aparecieron primero como enormes y costosas máquinas que solamente estaban disponibles en importantes universidades o centros de investigación. Con la aparición de nuevas técnicas de fabricación, como los circuitos integrados, su tamaño, sus capacidades, y sobre todo precio, variaron de tal forma que se convirtieron en un producto de masas más, como lo podrían ser la televisión o la radio. La aparición de Internet, y sobre todo su apertura al público general, determinaron la importancia de las computadoras en la vida social, laboral o académica de cualquier persona hasta el día de hoy.

Simultáneamente a la aparición de Internet como servicio abierto, a principios de la década de 1990, surgió otro medio de comunicación que, si bien era más antiguo, se reinventó a sí mismo gracias a los cambios en la telefonía móvil. La implantación de Internet, junto al furor de la telefonía móvil, confirmaba que esta revolución tecnológica no sólo afectaba a la investigación o la actividad económica, sino que implicaba un fenómeno sociológico donde la comunicación y el acceso a la información en cualquier lugar y momento eran sus pilares básicos.

Como no podía ser de otra manera, la reducción del tamaño de los componentes y el aumento de sus prestaciones permitió acercar cada vez más ambos mundos, de forma que a través de un teléfono móvil no sólo se podían hacer llamadas o enviar SMS, sino que además se podía tener un acceso más o menos limitado a Internet, o incluso funciones añadidas como realizar fotografías o vídeos. Otros aparatos de similar tamaño, no directamente relacionados con la telefonía, surgieron y se hicieron tan populares como los primeros. Desde ese momento puede empezar a usarse el término genérico dispositivo móvil.

Así entonces, un dispositivo móvil es un término general que describe una amplísima familia de aparatos electrónicos surgidos en los últimos años, de reducido tamaño, que ofrecen alguna capacidad de procesamiento y almacenamiento de datos y que están orientados a una función concreta o varias de ellas. Desde los teléfonos móviles más evolucionados (los llamados Smartphone), a computadoras portátiles, cámaras digitales, reproductores de música o consolas de videojuegos.

La mayoría de estos aparatos cuentan con un sistema operativo de mayor o menor complejidad, que permita realizar las tareas de gestión de memoria y control de hardware que precisan. En el caso de

los ordenadores portátiles, con tanta o incluso mayor capacidad que los de sobremesa, los sistemas operativos habituales son perfectamente compatibles y funcionan sin diferencias. Sin embargo, en otros dispositivos móviles es preciso diseñar nuevos sistemas operativos adaptados específicamente a sus características: restricciones de memoria y procesamiento, consumo mínimo de energía o gran estabilidad en su funcionamiento, entre otros.

Actualmente los dispositivos móviles se están convirtiendo en algo indispensable en la vida de toda persona, cumpliendo no solo la función de realizar llamadas y enviar mensajes de texto, sino también consumir y producir información, haciendo que sean un gran punto de entrada para cualquier persona de la sociedad al momento de querer exponer un producto o servicio.

Los dispositivos móviles

Veinte años antes de la aparición del primer teléfono móvil, el ingeniero de IBM Theodore George Paraskevokos creó el concepto de aparatos inteligentes capaces de procesar datos, además de pantallas que representan estos de manera visual, así dio lugar el concepto “Smartphone”. Las primeras patentes, están asociadas con el concepto de banca, ya que con dichas características se quiso enfocar al cálculo de operaciones bancarias.

Es en 1992 cuando se creó el primer Smartphone, diseñado en ese mismo año y mostrado al público en la feria COMDEX celebrada en las Vegas, Nevada. No fue hasta 1993 cuando se empezó a comercializar por la distribuidora BellSouth.
Además de poder realizar llamadas, posee una serie de funciones revolucionarias para la época: calendario, libreta de direcciones, reloj mundial, calculadora, bloc de nota, cliente de correo electrónico, enviar y recibir faxes y juegos. No disponía de ningún botón para manejar el teléfono, por lo que su funcionamiento se centraba en el uso de un lápiz para este tipo de pantallas monocromáticas o el uso de los dedos como hoy en día.
Tras la aparición de este revolucionario concepto de teléfono móvil, la empresa Finlandesa Nokia lanzó su línea Communicator, uno de los primeros Smartphone de Nokia. Con esta línea comenzó con el Nokia 9000.

El Nokia 9000, comercializado en 1996, supuso una diferenciación en el sector por el estilo de ordenador personal y Smartphone. Fue fruto de la colaboración entre Nokia y Hewlett-Packard, donde se unió el teléfono más vendido de la época junto a uno de los primeros prototipos de HP suponiendo un avance en el concepto de asistente personal digital (PDA). Este además destacaba por tener un teléfono totalmente funcional y al abrirlo, podía verse la pantalla de alta resolución de 640×200 píxeles y un teclado qwerty físico. Funcionaba con el sistema operativo GEOS V3.0, el cual resaltaba por la utilización del correo electrónico y la posibilidad de navegación web mediante texto. Más tarde, en 1998, le sucedería el Nokia 9110 y en el 2000 el Nokia 9110i con una mejora en la navegación web .

Entre estas fechas (1997) es cuando apareció el primer terminal en el que se utiliza el concepto “Smartphone” este es el Ericsson GS88 el primer teléfono inteligente. Tras el éxito de este

nuevo dispositivo, Ericsson saco en el 2000 la R380, primer dispositivo en utilizar el sistema operativo libre Symbian. Dicho teléfono era la combinación de teléfono móvil y PDA. Tal fue la revolución que la revista Popular Science lo nombró como uno de los avances más importantes de la ciencia y tecnología, innovando en su diseño pequeño y ligero como un teléfono móvil normal. Dos años más tarde en el 2002 tomaría el relevo de este el P800 con pantalla a color y reproducción de videos, pero enfocado al ámbito empresarial.

En el mismo periodo que el R380, Nokia sacó el 9210 communicator, el primer teléfono inteligente con pantalla a color, pudiendo decir que se trata del verdadero Smartphone contemporáneo. A este le siguió el Nokia 9500 con cámara digital y Wi-FI, características que le dotaron de un status de gama alta, al mismo tiempo que sacaron el 9300 más económico. El último que presentó Nokia de la gama es el E90, con las características del 9500 y GPS. Ésta línea de teléfonos costaban una considerable cantidad de dinero en esa época, más incluso que lo que cuesta hoy en día un iPhone.

Ya por el 2007, Nokia desarrolló el N95, supuso lo mismo que hoy el Samsung Galaxy SII, una revolución en el sector. Este terminal de gama alta cuenta con GPS, cámara de 5 Megapixeles autofocus, flash led, 3G, Wi-Fi y salida TV. Junto a este móvil, aparecía también el primer iPhone. Fue de los primeros en integrar una gran pantalla táctil en la cual solo hacía falta el uso de los dedos, nada de lápiz, teclado numérico, como ocurría en la mayoría de los Smartphone del momento. Así el iPhone se describe como un terminal superior a la competencia, afirmación hecha por el uso del navegador web Safari, además de por el sistema operativo iOS postulándose por un sistema cerrado y optimizado para el iPhone, el cual no tardó en abrirse de par en par mediante el jailbreak, acción que permite instalar todo tipo de software, incluso aplicaciones de pago de forma gratuita. Además de estas características, el iPhone destaca por incluir GPS, temporizador, radio, mapas, calendario, bloc de notas y varias funciones más. Esto supuso para los usuarios la posibilidad de prescindir de muchos aparatos tecnológicos.

Por el 2008 Android fue presentado como un sistema operativo de código abierto, respaldado por Google, junto a grandes desarrolladores de Hardware (HTC, Motorola, Samsung) formando así una poderosa alianza. Fruto de esta unión dio el primero de los Smartphone con Android, el HTC Dream, también conocido como G1 cuando los distribuía T-Mobile. Las características más relevantes de este SO se centran en la integración de aplicaciones propietarias de Google, como un navegador con soporte Flash, mapas, calendario y Gmail. Además, este soporta la ejecución de aplicaciones nativas y en multitarea. Al igual que Apple, sacó su tienda de aplicaciones conocida ahora como Google Play, tienda que destaca por la gran cantidad de aplicaciones que ofrece al usuario, aunque también se pueden encontrar aplicaciones de pago con una buena calidad.

Android

Android más que un sistema operativo, representa toda una pila de software para dispositivos móviles que incluye gran cantidad de drivers, gestor de bases de datos, un completo framework de aplicaciones, y numerosas aplicaciones de usuario. Android está basado en el núcleo de Linux y todas sus aplicaciones se escriben en lenguaje Java, disponiendo además de una máquina virtual específica llamada Dalvik.

Con la aparición de este sistema, Google pretende aprovechar al máximo la cada vez mayor capacidad de los dispositivos móviles, que llegan a incluir componentes como GPS, pantallas táctiles, conexiones rápidas a Internet, y por supuesto, todos los servicios asociados hasta ahora a los teléfonos móviles, además de aplicaciones de usuario hasta ahora limitadas a los ordenadores, como clientes de correo, aplicaciones ofimáticas o videojuegos. En Android, cada aplicación corre en su propio proceso, donde el sistema decide en cada momento qué aplicación debe ser eliminada para liberar recursos en caso de carencia, y responsabilizándose igualmente de restaurarla de forma totalmente transparente al usuario. Navegar entre varias aplicaciones abiertas deja de ser una característica propia de ordenadores.

Hoy en día es uno de los sistemas operativos más utilizados en dispositivos móviles con casi el 50% del mercado.

Este nuevo sistema introduce también interesantes conceptos, como es la composición de sus aplicaciones a través de combinación de módulos o bloques básicos, según la naturaleza de la aplicación, o la delegación en el sistema de determinadas acciones para que sean otras aplicaciones instaladas las que se hagan cargo de ellas.

Evolución

Android 1.5 Cupcake

Fue la primera versión que fue lanzada al mercado, aunque no la primera desarrollada. Los primeros Smartphone en incorporar esta versión del sistema operativo fueron el HTC Dream y el Samsung Galaxy GT-i7500. Esta versión no fue importante solo por ser la primera versión del sistema, ya que además supuso un cambio de los móviles en los que predominaba el teclado QWERTY a la implementación de pantallas táctiles, al más puro estilo iPhone, algo revolucionario en el año 2009. Su interfaz era muy básica y muy similar a Symbian. [02]

Android 1.6 Donut

No fue la versión más popular del sistema ya que los cambios que proporcionó no afectaban directamente al usuario, pero sí fue una de las versiones más importantes hasta la fecha, ya que fue la que incorporó la capacidad de poder soportar diferentes resoluciones de pantalla sin importar la densidad de píxeles de esta.

Android 2.0 Eclair

Esta versión introdujo algún que otro cambio en la interfaz de usuario, nada destacable, pero sí que fue un gran cambio para el aumento de la popularidad de la fotografía nocturna con un teléfono móvil, ya que, hasta su lanzamiento, Android no soportaba la incorporación del Flash LED para la cámara, pero gracias a Android 2.0 Eclair esta nueva característica se hizo posible. Con esta versión apareció el primer terminal de Google, el Google Nexus One.

Android 2.2 Froyo

Presentado en mayo de 2010, con esta versión se mejoró la interfaz y experiencia de usuario, cambiando, por ejemplo, el dock de aplicaciones y el escritorio, siendo mucho más elaborado. La velocidad general del sistema se multiplicó por cuatro y se incorporó la capacidad de Tethering & Portable Hotspot, es decir, de compartir nuestra conexión a Internet con otros dispositivos. Con esta nueva versión el Android Market (Google Play en los primeros años de Android) se hizo accesible en los ordenadores, desde los cuales era posible descargar directamente las aplicaciones en nuestro dispositivo.

Android 2.3 Gingerbread

Tal vez la versión más popular del sistema operativo de Google, aunque su lanzamiento fue en febrero de 2011, todavía está presente en el 11% de terminales de la actualidad. No fue la versión con más novedades, pero sí que tuvo la incorporación de más sensores como el barómetro o el importantísimo giroscopio, gracias al cual se incorpora la rotación automática de la pantalla y se mejoró la experiencia que un teléfono podía dar, sobre todo en los juegos. Con esta versión también llegó el segundo terminal de la familia Google Nexus, el Google Nexus S y terminales tan emblemáticos como el Samsung Galaxy Ace.

Android 3.0 Honeycomb

Tal vez la actualización menos conocida por todos, pero no por eso significa que no fuese importante. Gracias a esta actualización Android se optimizó para tablets, integró los botones en pantalla y la barra de notificaciones pasó a estar en la parte inferior. La interfaz también cambió y vimos una interfaz que nos empezaba a recordar a la futura interfaz Holo como por ejemplo la multitarea, que empezó a verse en pequeñas ventanas y no en iconos.

Android 4.0 Ice Cream Sandwich

El primer gran cambio en la interfaz de Android, se introdujo la interfaz Holo, que le daba mucho más de seriedad y personalidad. Entre los cambios de interfaz encontramos la barra de notificaciones mejorada, un escritorio más minimalista y ligero, en cuanto a la seguridad introduce el desbloqueo facial y mejora la experiencia de usuario del navegador nativo. Junto con su llegada en octubre de 2011 también se presentó el Google Galaxy Nexus.

Android 4.1 Jelly Bean

Presentada en junio del 2012 esta actualización no supuso enormes cambios, pero sí que aumentó la velocidad del sistema y la interfaz se hizo un poco más plana, y las notificaciones se podían expandir con un simple gesto. Posiblemente lo más importante de la llegada de esta versión fue que llegó con dos terminales, el Google Nexus 4 y la Google Nexus 7. En sus versiones posteriores (4.2 y 4.3) se solucionan bugs y con la versión 4.3 se presenta la Google Nexus 7 segunda generación.

Android 4.4 KitKat

Optimizada para dispositivos con 512 MB de memoria RAM, cambió la interfaz de algunas aplicaciones como el dialer o la de contactos y la barra de notificaciones cambió de color azul al color blanco, dándole un toque mucho más minimalista. Se incorporó la capacidad de grabar la pantalla de forma nativa y de imprimir gracias a ciertos complementos disponibles en el Google Play. Con esta versión salió al mercado el Google Nexus 5 en octubre de 2013.

Android 5.0 Lollipop

Esta versión supone un lavado de cara respecto a las anteriores versiones, con la nueva interfaz estilo Material Design, cambiando completamente el sistema de notificaciones, la multitarea también ha sido renovada ahora con un sistema de ventanas flotantes en carrusel. Esta versión, optimizada para procesadores de 64 bits, será lanzada al mercado junto al Google Nexus 6 y la Google Nexus 9 y utilizará ART en vez de Dalvik como máquina virtual.

Android 6.0 Marshmallow

Es una versión del sistema operativo para dispositivos móviles Android. Fue dada a conocer el 28 de mayo de 2015 en el evento Google I/O como Android M. Android Marshmallow, oficialmente presentado ya con esa nomenclatura el 17 de agosto de 2015, se centrará principalmente en mejoras

incrementales y nuevas características. Android 6.0 Marshmallow introduce un modelo de permisos rediseñado. Ofrece soporte nativo para el reconocimiento de huellas digitales. Introduce un nuevo sistema de administración de energía conocida como "doze" que reduce las actividades en fondo del dispositivo cuando el sistema operativo detecta que el dispositivo no está siendo tocado físicamente.

Software Libre

Android se lanza bajo la licencia Apache, lo que implica que, como software libre, cualquier desarrollador tiene acceso completo al SDK [09] del sistema, incluidas todas sus API, documentación y emulador para pruebas, pudiendo distribuirlo y modificarlo. Además, esta licencia permite a los desarrolladores tanto publicar a su vez sus creaciones, como distribuirlas únicamente bajo pago ocultando el código fuente. [06]

El término software libre refiere el conjunto de software que, por elección de su autor, puede ser copiado, estudiado, modificado, utilizado libremente con cualquier fin y redistribuido con o sin cambios o mejoras. Su definición está asociada al nacimiento del movimiento de software libre, encabezado por Richard Stallman y la consecuente fundación en 1985 de la Free Software Foundation, que coloca la libertad del usuario informático como propósito ético fundamental. Proviene del término en inglés free software, que presenta ambigüedad entre los significados “libre” y “gratis” asociados a la palabra free. Por esto que suele ser considerado como software gratuito y no como software que puede ser modificado sin restricciones de licencia. Es necesario resaltar que la libertad tiene que ver con el uso y no con la gratuidad.

El software libre suele estar disponible gratuitamente, o al precio de coste de la distribución a través de otros medios; sin embargo, no es obligatorio que sea así, por lo tanto, no hay que asociar software libre a software gratuito, ya que, conservando su carácter de libre, puede ser distribuido comercialmente. Análogamente, el software gratis o gratuito incluye en ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con software de dominio público. Éste último es aquel que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, consignando su autoría original. Este software sería aquel cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es del dominio público.

De acuerdo con la definición establecida por Richard Stallman, un software es "libre" cuando garantiza las siguientes libertades:

0. la libertad de usar el programa, con cualquier propósito (Uso).
1. la libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las propias necesidades (Estudio).
2. la libertad de distribuir copias del programa, con lo cual se puede ayudar a otros usuarios (Distribución).
3. la libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie (Mejora).

Las libertades 1 y 3 requieren acceso al código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.

Los formatos abiertos permiten al software libre mantener sus cuatro libertades y la libre difusión de todo el código y formatos utilizados, su distribución y estudio, debido a esto, los creadores de software libre desarrollan a la vez de programas libres, formatos libres para estos programas o utilizan formatos libres ya creados anteriormente.

Dentro de los proyectos más importantes de software libre se destacan los siguientes:

- Firefox: Nacido a partir de Mozilla, que incluía también gestor de correo electrónico (ahora Thunderbird), editor HTML, chat IRC y lista de contactos.
- Pidgin: Sirve para hablar con prácticamente cualquier red de mensajería instantánea, desde las populares MSN y Yahoo! hasta otras como Jabber, ICQ, Google Talk o IRC. aMSN y emesene, por su parte, están especializados en la red de Microsoft.
- jDowloader: Es una herramienta de administración de descargas directas, desde cualquier link de descarga de archivos hasta videos de youtube.
- VLC: Reproductor multimedia con soporte para varios tipos de datos (videos, música) y un gran número de codecs.
- Gimp: es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías
- LibreOffice: es un paquete de oficina libre y de código abierto desarrollado por The Document Foundation. Se creó como bifurcación de OpenOffice.org en 2010. Cuenta con un procesador de texto (Writer), un editor de hojas de cálculo (Calc), un gestor de presentaciones (Impress), un gestor de bases de datos (Base), un editor de gráficos vectoriales (Draw) y un editor de fórmulas matemáticas (Math).
- VirtualBox: es un software de virtualización para arquitecturas x86/amd64.
- 7-Zip: Es un archivador de ficheros libre Por defecto utiliza el formato de archivo 7z, también libre, (con extensión .7z)

Tipos de licencia

El software no se vende, se licencia. Una licencia es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer "actos de explotación legales". Es decir, el software no se compra, sino que se adquieren una serie de derechos sobre el uso que se le puede dar. En las licencias de software libre esos derechos son muy abiertos y permisivos, apenas hay restricciones al uso de los programas. Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatario.

Licencias GPL

Una de las más utilizadas es la Licencia Pública General de GNU (GNU GPL). El autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL.

Licencias AGPL

La Licencia Pública General de Affero (en inglés Affero General Public License, también Affero GPL o AGPL) es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red. La Affero GPL es íntegramente una GNU GPL con una cláusula nueva que añade la obligación de distribuir el software si éste se ejecuta para ofrecer servicios a través de una red de ordenadores.

Licencias estilo BSD

Llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles. Puede argumentarse que esta licencia asegura “verdadero” software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre.

Licencias estilo MPL y derivadas

Esta licencia es de Software Libre y tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y empezar ese proyecto tan importante para el mundo del Software Libre: Mozilla. Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos. La MPL es Software Libre y promueve eficazmente la colaboración evitando el efecto "viral" de la GPL (si usas código licenciado GPL, tu desarrollo final tiene que estar licenciado GPL). Desde un punto de vista del desarrollador la GPL presenta un inconveniente en este punto, y mucha gente se cierra en banda ante el uso de dicho código. No obstante, la MPL no es tan excesivamente permisiva como las licencias tipo BSD. Estas licencias son denominadas de copyleft débil. La NPL (luego la MPL) fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tomados en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

Copyleft

Hay que hacer constar que el titular de los derechos de autor (copyright) de un software bajo licencia copyleft puede también realizar una versión modificada bajo su copyright original, y venderla bajo cualquier licencia que desee, además de distribuir la versión original como software libre. Esta técnica ha sido usada como un modelo de negocio por una serie de empresas que realizan software libre (por ejemplo, MySQL); esta práctica no restringe ninguno de los derechos otorgados a los usuarios de la versión copyleft.

SIU Guaraní

El SIU Guaraní es un sistema de gestión de alumnos que registra y administra todas las actividades académicas de la universidad, desde que los alumnos ingresan como aspirantes hasta que obtienen su diploma. Fue concebido para administrar la gestión de alumnos en forma segura, con la finalidad de obtener información consistente para los niveles operativos y directivos. [03]

El sistema brinda servicios tanto para alumnos como docentes, usuarios administrativos y autoridades. Permite al estudiante consultar inscripción a exámenes y cursadas, reinscripción a carreras, consulta de inscripciones, consulta de plan de estudios e historia académica, consulta de cronograma de evaluaciones parciales, consulta de créditos, notas de evaluaciones parciales, materias regulares, agenda de clases, solicitud de certificados, actualización de datos censales y recepción de mensajes.

Actualmente SIU Guaraní está desplegado en las siguientes universidades de la Argentina:

IUSM - Instituto Universitario de Seguridad Marítima
UNCOMA - Universidad Nacional del Comahue
UNSJ - Universidad Nacional de San Juan
UNSA - Universidad Nacional de Salta
UNLAR - Universidad Nacional de La Rioja
UNAM - Universidad Nacional de Misiones
UNER - Universidad Nacional de Entre Ríos
UNPSJB - Universidad Nacional de la Patagonia San Juan Bosco
UNSE - Universidad Nacional de Santiago del Estero
UDC - Universidad del Chubut
UNSAM - Universidad Nacional de San Martín
UNR - Universidad Nacional de Rosario
UNSL - Universidad Nacional de San Luis
IUNA - Instituto Universitario Nacional del Arte
UNC - Universidad Nacional de Córdoba
UNNE - Universidad Nacional del Nordeste
UBA - Universidad de Buenos Aires
UNICEN - Universidad Nacional del Centro de la Provincia de Buenos Aires
UNL - Universidad Nacional del Litoral
UCEL - Universidad del Centro Educativo Latinoamericano
UNMdP - Universidad Nacional de Mar del Plata
UNPA - Universidad Nacional de la Patagonia Austral
UTN - Universidad Tecnológica Nacional
UNF - Universidad Nacional de Formosa
IFDCVM - Instituto de Formación Docente Continua de Villa Mercedes
UNO - Universidad Nacional del Oeste
UNCU - Universidad Nacional de Cuyo
UNLP - Universidad Nacional de La Plata
UNNOBA - Universidad Nacional del Noroeste de la Provincia de Buenos Aires
UNDEC - Universidad Nacional de Chilecito
UNLZ - Universidad Nacional de Lomas de Zamora
UNRN - Universidad Nacional de Río Negro
UNJU - Universidad Nacional de Jujuy
UADER - Universidad Autónoma de Entre Ríos
UNDAV - Universidad Nacional de Avellaneda
UNLA - Universidad Nacional de Lanús
UNCAus - Universidad Nacional del Chaco Austral
UNLaM - Universidad Nacional de La Matanza
UNM - Universidad Nacional de Moreno
UNAJ - Universidad Nacional Arturo Jauretche
ISALUD - Universidad ISALUD
IUE - Instituto Universitario del Ejército
MSFe - Municipalidad de Santa Fe
IUSAM - Instituto Universitario de Salud Mental (Asociación Psicoanalítica de BsAs)
UNQ - Universidad Nacional de Quilmes
IFTS6 - Instituto de Formación Técnica Superior Nro 6 GCBA
IUGR - Instituto Universitario del Gran Rosario
CEMIC - CEMIC
UNLPAM - Universidad Nacional de La Pampa
LMStaFe - Liceo Municipal Santa Fe \"Antonio Fuentes del Arco\"
UNGS - Universidad Nacional de General Sarmiento
UNCA - Universidad Nacional de Catamarca
HIBA - Instituto Universitario del Hospital Italiano
UNPAZ - Universidad Nacional de José C. Paz
UNT - Universidad Nacional de Tucumán
IUPFA - Instituto Universitario de la Policía Federal Argentina
MSFe - Municipalidad de Santa Fe
UNS - Universidad Nacional del Sur
UPSO - Universidad Provincial del Sudoeste de la Provincia de Buenos Aires
IFES - Instituto de Formación y Educación Superior
UNIPE - Universidad Pedagógica de la Provincia de Buenos Aires
UNPSJB - Universidad Nacional de la Patagonia San Juan Bosco

SIU Guaraní está implementado en la UNLP desde el año 2002 y se extendió a todas las Facultades a partir del año 2013.

SECCIÓN 2

Motivación

SIU Guaraní es un sistema ampliamente utilizado por todos los alumnos de la Universidad, por lo que es necesario que esté integrado a su vida cotidiana durante su época de estudio. Desde ver una nota hasta inscribirse a una cursada, a lo largo de la carrera un alumno accede innumerables veces para hacer consultas o interactuar con el sistema.

Actualmente si uno quisiera inscribirse a una materia, existen tres formas. La primera es acercarse a la oficina de alumnos, siendo una opción que los alumnos poco a poco están dejando de lado. Hoy en día la gran mayoría de las operaciones que una persona quiera hacer en su vida se pueden hacer a través de una computadora. Hacer compras, sacar un pasaje, o hasta pagar un servicio son algunas de las cosas que se pueden hacer desde internet, por lo que los alumnos cada vez están más acostumbrados a este tipo de transacciones no presenciales. El presentarse a la oficina de alumnos para hacer una tarea no es algo cómodo, y cada vez será algo más difícil de adoptar a medida que pasen los años y más cosas se puedan hacer a través de una computadora. Otro factor es el tema localización de los alumnos de la facultad. Dado que la institución es reconocida nacionalmente es un hecho que no sólo gente de la ciudad quiera cursar a sus carreras. Esto complica aún más el estar presente para originar inscripciones.

La aplicación SIU Guaraní surgió para suplir estas faltas. Ya no sería necesario estar físicamente en la facultad para poder realizar las operaciones personales. Cualquiera que tuviera acceso a una computadora podría acceder y ver su historia académica, o realizar una inscripción a un examen. Pero aun así no siempre se tiene una computadora cerca para poder acceder a internet.

Desde el 2007 hasta el día de hoy el boom de los teléfonos móviles es algo que no para de crecer. Estos dispositivos están cada vez más integrados en nuestras vidas, lo cual presenta un gran punto de entrada para cualquier aplicación que quiera ser accedida desde cualquier lugar del mundo. Hoy en día es difícil encontrar una persona que no disponga de un teléfono inteligente. Dado el tamaño de las pantallas de hoy en día, el hecho de tener que estar frente a una computadora se puede llegar a suplir accediendo a la web de SIU Guaraní desde cualquier dispositivo.

Al ser SIU Guaraní una aplicación web pensada para computadoras de escritorio presenta varias dificultades al momento de ser accedida desde un dispositivo con pantalla más chica y con otros métodos de entrada. No todas las páginas webs están adaptadas para ser accedidas desde varios dispositivos y ser usadas sin ningún problema. Crear una página adaptable implica capacitar a los desarrolladores en nuevas tecnologías y metodologías de trabajo, lo cual requiere mucho tiempo y esfuerzo. Agregado al tiempo de capacitación se encuentra el tiempo de implementación, actividad que aun sabiendo lo que se está haciendo es algo bastante tedioso también por el hecho que existen muchos dispositivos y no todos con características similares. Diferentes pantallas, diferentes resoluciones y otros aspectos agregan dificultad al desarrollo.

Aún con una web adaptada a los móviles de hoy en día, existe otro factor limitante en el acceso al sistema. La conectividad es un problema que afecta a un gran porcentaje de los que poseen un teléfono, haciendo que acceder a contenido online sea algo difícil si no se está conectado a una red wifi estable, dejando las opciones de internet móvil (como puede ser 2g, 3g y 4g) como alternativa. Dado que

en estos tiempos las compañías de celulares no ofrecen un buen servicio de conexión a internet, ingresar a una página web puede ser algo que cueste y tarde mucho desde un dispositivo. Cada carga y operación sobre la aplicación web adaptada al celular implicaría una carga inicial de datos grande, ya que es necesario descargar tanto la página HTML, como las imágenes, archivos JavaScript y CSS. Ésta demora obliga a los usuarios a usar cada vez menos las páginas webs para migrar al uso de aplicaciones móviles convencionales, lo que deja a los servicios que aún no tienen una versión móvil fuera de una gran parte de la población.

De estas limitaciones surge entonces la idea de tener una aplicación de gran disponibilidad geográfica, siendo necesario que opere con un consumo de datos mínimo y adaptada a la gran mayoría de los dispositivos móviles actuales.

Como adicional, y pensando en la interacción que cada alumno tiene con sus compañeros, se pensó en cómo tener un agregado que ayude a mejorar la comunicación. Aprovechando las características que hoy en día muchos dispositivos móviles tienen, se investigó cómo ayudar a los alumnos a encontrarse. Uno de los casos de uso posibles sería querer encontrarse para preparar un trabajo, o un examen final, o intercambiar notas de cátedra. Dado este requerimiento se optó finalmente por incorporar funcionalidades de localización. Esta característica permitiría a los alumnos tener información sobre la localización de sus compañeros siempre y cuando estén dentro del radio de la facultad y sean efectivamente compañeros de cursada.

Ésta última funcionalidad dejaría las puertas abiertas a muchas más funcionalidades relacionadas con la interacción de los alumnos y sus compañeros, pudiendo plantearse en un futuro compartir apuntes, prácticas, exámenes o incluso armar foros de discusión dentro de la misma aplicación, donde el alumno no necesitaría tener que acceder a una computadora para poder realizar todas estas tareas.

Aplicaciones Móviles

Una aplicación móvil es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, BlackBerry OS, Windows Phone, entre otros. Existen tanto aplicaciones móviles gratuitas como pagas. [05]

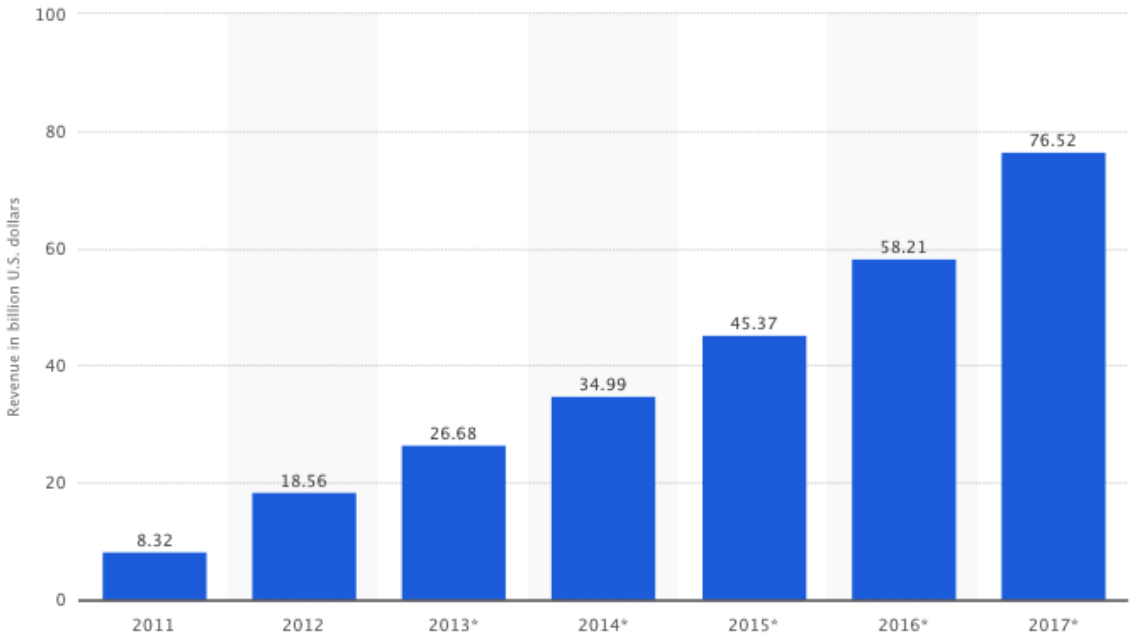
Las primeras aplicaciones móviles que se desarrollaron datan de finales de los noventa. Estas eran lo que conocemos como la agenda, arcade games, editores de ringtones, etc. Dichas aplicaciones cumplían con funciones muy elementales y su diseño era bastante simple y poco atractivo. La evolución de las apps se dio rápidamente gracias a las innovaciones en tecnología WAP y la transmisión de data (EDGE) esto vino acompañado de un desarrollo muy fuerte de los celulares.

Finalmente, la evolución de dichas aplicaciones nace con el lanzamiento del iPhone de Apple y el desarrollo del sistema operativo para móviles Android. Junto a estos desarrollos llegan muchas más propuestas de Smartphone, y de esta forma empieza el boom de las apps, juegos, noticias, diseño, arte, educación, fotografía, medicina; todo inmerso en lo que antes eran un simple equipo de comunicación celular, la incorporación de Internet en los celulares y la creación de las Tablets revolucionó el mundo de las aplicaciones móviles.

El surgimiento de las Apps Stores terminó de impulsar el éxito de las aplicaciones móviles y un significativo cambio en la manera en que se distribuye y comercializa el software. Android Market entró al negocio a los pocos meses, con un repositorio de 50 apps, el cual luego se nombraría Play Store [27]. La tercera fue BlackBerry App World [28] y Ovi Store de Nokia en el 2009. Microsoft llegó mucho más tarde abriendo en el 2010 con Windows Phone Marketplace [29] dejando obsoleta la tienda Ovi de Nokia.

Hoy en día las aplicaciones móviles están en su auge. Según un informe de KPCB del año pasado en 2015 habrá 1.600 millones de Smartphone en el mundo. En cuanto a dispositivos móviles se calcula una suma de 5.600 millones. Según un reciente estudio de la GSMA para 2020 la cifra de Smartphone podría ser de 6.000 millones.

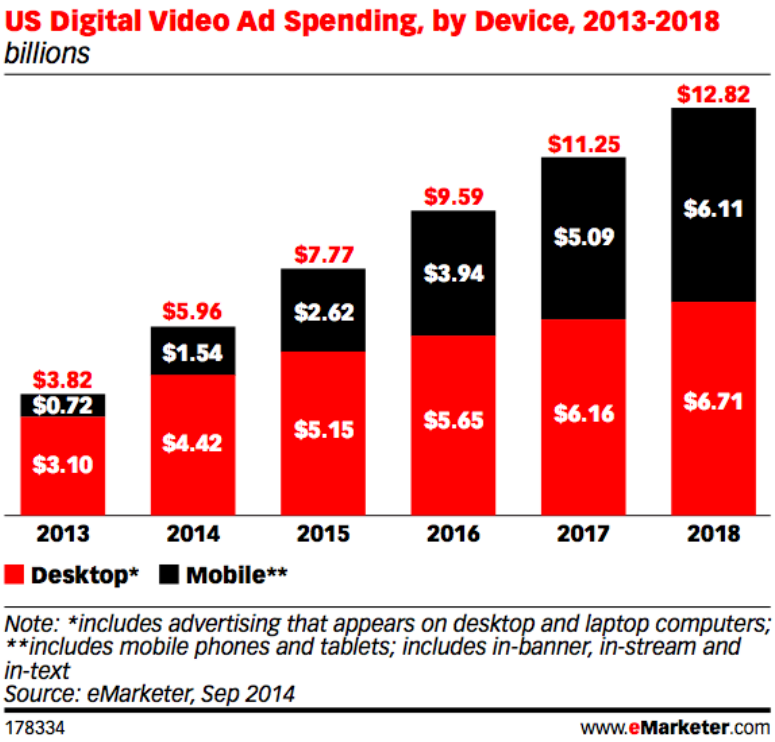
Los ingresos generados en las diferentes apps stores están previstos en 45.400 millones de dólares para 2015, subiendo a 76.520 millones para 2017, según previsiones de Statista. Esto va unido a las previsiones de descargas de apps, estimadas en 167.050 millones para 2015 y en 211.310 millones para 2016, también según Statista.

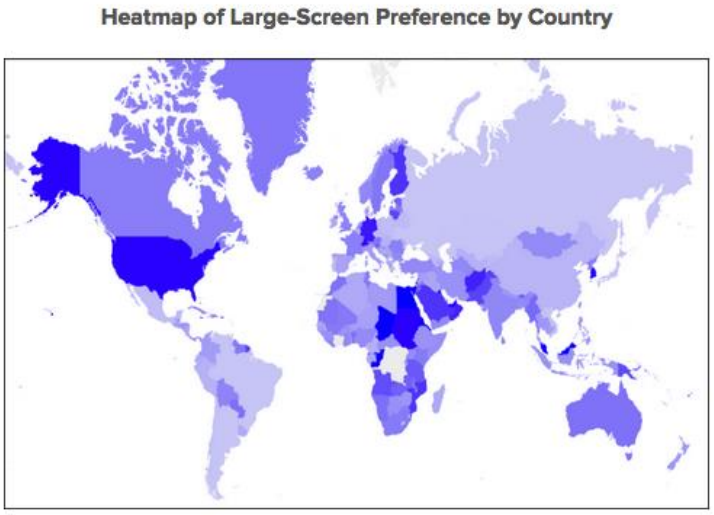


También aumentará la inversión en publicidad móvil, y seguirá aumentado hasta alcanzar los 42.000 millones en 2018, según un estudio de BI Intelligence. Según Medialets los anuncios mostrados dentro de las apps son más eficaces que los mostrados en las webs móviles. Destaca aquí una tendencia en mobile marketing como es el uso cada vez más intensivo del video como canal de publicidad en móviles. eMarketer estima que se pasará de una inversión de 1.540 millones de dólares en 2014 a 12.820 millones en 2018.

Todos estos números del punto anterior se reflejan también en el uso que se hace de las apps. Según un estudio de Octubre de 2014 de Comscore, el 52% el tiempo que los consumidores estuvieron online lo hicieron desde un dispositivo móvil, bien un smartphone o una tablet.

Por último, una razón para centrar los esfuerzos en el desarrollo de aplicaciones móviles para tablets y phablets. Las pantallas más grandes parecen ser más usadas que las pequeñas. Según un informe de Loclytics los usuarios de tabletas se pasan un 13% más de tiempo dentro de las apps y las abren un 11% más frecuentemente. Así mismo los dueños de phablets usan un 38% más las apps y permanecen usándolas un 10% más que los usuarios de móviles con pantallas pequeñas.





Mapa de calor de países con preferencia de pantallas grandes

Tipo de aplicaciones según su desarrollo

A nivel programación existen varias formas de desarrollar una aplicación. Cada una de ellas tiene diferentes características y limitaciones, especialmente desde el punto de vista técnico.

Aplicaciones nativas

Una aplicación nativa es la que se desarrolla de forma específica para un determinado sistema operativo, llamado Software Development Kit o SDK [09]. Cada una de las plataformas (Android, iOS o

Windows Phone) tienen un sistema diferente, por lo que, si es necesario que la aplicación esté disponible en todas las plataformas, se deberán crear varias aplicaciones, también llamadas apps, con el lenguaje del sistema operativo seleccionado. [04] [07] [08]

Cuando hablamos de desarrollo móvil casi siempre nos referimos a aplicaciones nativas. La principal ventaja con respecto a los otros dos tipos es la posibilidad de acceder a todas las características del hardware del móvil: cámara, GPS, agenda, dispositivos de almacenamiento y varias más. Esto hace que la experiencia del usuario sea mucho más positiva que con otro tipo de aplicaciones. Además, las aplicaciones nativas no necesitan conexión a internet para que funcionen. Una característica generalmente menospreciada de las aplicaciones nativas, es que pueden hacer uso de las notificaciones del sistema operativo para mostrar avisos importantes al usuario, aun cuando no se esté usando la aplicación, como los mensajes de WhatsApp, por ejemplo.

La descarga e instalación de estas aplicaciones se realiza siempre a través de las tiendas de aplicaciones (*app store* de los fabricantes). Esto facilita el proceso de marketing y promoción, proceso que es vital para dar visibilidad a una aplicación.

Está claro que, si el coste no es un obstáculo, o se tiene la certeza de que la aplicación será rentable, la mejor opción será siempre el desarrollo de una aplicación nativa para cada plataforma (iOS, Android y Windows Phone). Si el presupuesto o tiempo es limitado, las aplicaciones web tienen también grandes ventajas para el negocio.
Las aplicaciones nativas se actualizan frecuentemente y en esos casos, el usuario debe volver a descargarlas para obtener la última versión, que a veces corrige errores o añade mejoras.

A nivel de diseño, esta clase de aplicaciones tiene una interfaz basada en las guías de cada sistema operativo, logrando mayor coherencia y consistencia con el resto de aplicaciones y con el propio SO. Esto favorece la usabilidad y beneficia directamente al usuario que encuentra interfaces familiares.

El mejor ejemplo de aplicaciones nativas son las que directamente están preinstaladas dentro del mismo sistema operativo, como puede ser la aplicación de llamadas, mensajes, correo electrónico. Muchas empresas también optan por esta modalidad de desarrollo ya que mejora mucho la experiencia del usuario y ayuda a dar renombre a la misma. Un ejemplo puede ser WhatsApp con su aplicación de mensajería, que logró ser una de las más utilizadas en estos últimos años.

Aplicaciones Web

La base de programación de las aplicaciones web (también llamadas webapps) es el HTML, conjuntamente con JavaScript y CSS, herramientas ya conocidas para los programadores web. La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones.

Las aplicaciones web no necesitan instalarse ya que se visualizan usando el navegador del teléfono como un sitio web normal. Por esta misma razón no se distribuyen en una tienda de aplicaciones, sino que se comercializan y promocionan de forma independiente. Al tratarse de aplicaciones que funcionan sobre la web, no es necesario que el usuario reciba actualizaciones, ya que siempre va a estar viendo la última versión. Pero, a diferencia de las apps nativas, requieren de una conexión a Internet para funcionar correctamente.

Adicionalmente, tienen algunas restricciones e inconvenientes en factores importantes como gestión de memoria y no permiten aprovechar al máximo la potencia de los diferentes componentes de hardware del teléfono.

Muchas de los sitios web de hoy en día están pensados para poder ser llevados a la pantalla chica, de manera de ser navegables desde dispositivos móviles sin dificultades. Esto asegura una ventaja contra los competidores que aún no están preparados para mostrar su contenido de forma adaptable.

Aplicaciones Híbridas

Este tipo de aplicaciones es una especie de combinación entre las dos anteriores. La forma de desarrollarlas es parecida a la de una aplicación web (usando HTML, CSS y JavaScript), y una vez terminada, se compila o empaqueta de forma tal que el resultado final es como si se tratara de una aplicación nativa. Esto permite casi con un mismo código obtener diferentes aplicaciones (por ejemplo, para Android o iOS) y distribuirlas en cada una de sus tiendas.

A diferencia de las aplicaciones web, estas permiten acceder a las capacidades del teléfono usando librerías, así como lo hace una app nativa.

Las aplicaciones híbridas también tienen un diseño visual que no se identifica en gran medida con el sistema operativo. Sin embargo, hay formas de usar controles y botones nativos de cada plataforma para apegarse más a la estética propia de cada una.

Un caso de gran éxito de este tipo de aplicaciones es Netflix, siendo una de las plataformas más utilizada para la reproducción de videos. Siendo que es una aplicación híbrida, cuenta con el mismo código fuente para todos los dispositivos móviles, independientemente de que tipo de sistema operativo corran, permitiendo hacer arreglos y modificaciones para su aplicación y plasmarlo en cada versión sin necesidad de realizar un gran esfuerzo, siendo el contenedor de la aplicación lo único diferente entre las versiones.

SECCIÓN 3

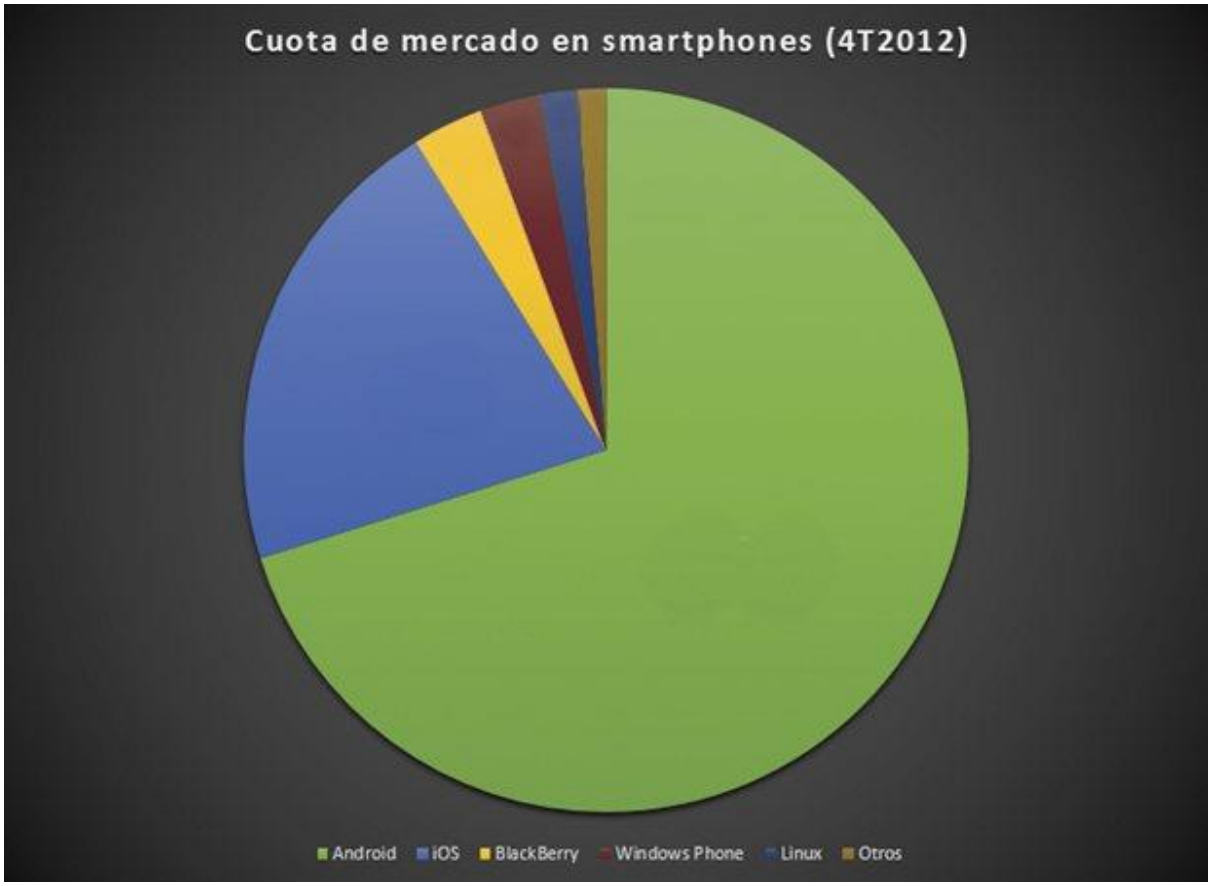
Elección del Tipo de Aplicación

Como ya se nombró previamente, existen tres tipos de aplicaciones. SIU Guaraní mobile tendría que ser una aplicación que no requiera una carga inicial grande de información, por lo que una aplicación web adaptada a un dispositivo no era la solución ideal para suplir esta necesidad. Se optó por investigar las dos variantes restantes.

Hoy en día, Apple y Google se reparten prácticamente todo el mercado con sus respectivas plataformas móviles, iOS y Android, que son la base de millones de productos con los que hablamos, navegamos, trabajamos y jugamos. Apple fue el primero en ofrecer un sistema operativo móvil, el cual empezaría a marcar el camino del resto, aunque en un principio no se vio como una gran innovación. Google tuvo que lanzar la competencia, aunque las primeras versiones de Android no parecían una amenaza directa a Apple. El crecimiento de Android fue inmensurable y, versión tras versión, se fueron recortando las distancias con respecto a un iOS que trataba sin éxito de mantener la ventaja.

Esa particular esta competencia entre Android e iOS se terminó igualando en el terreno técnico: cada plataforma tiene su filosofía, pero, incluso en detalles como la interfaz o el funcionamiento de la distribución del software, sus posturas han sido muy similares. Sin embargo, Android ha contado con un apoyo crítico: su base Open Source ha permitido que operadoras y sobre todo fabricantes puedan basar sus dispositivos en una solución que se ha adaptado a todas las necesidades.

Por su parte, Apple seguía sin licenciar su plataforma a terceros (como ocurría con Mac OS X en las computadoras), mientras que Google simplemente quería que Android lo usase tanta gente como fuese posible –como hizo Microsoft con un Windows que funcionaba en todo tipo de PC–. Esa versatilidad y adaptabilidad de Android ha hecho que hoy en día esta plataforma domine el mercado de forma indiscutible. Según las cifras de un estudio publicado reciente (IDC Worldwide Mobile Phone Tracker [30]), en el cuarto trimestre de 2012, Android tenía una cuota de mercado del 70,1%, por un 21,0% de iOS, un 3,2% de BlackBerry, un 2,6% de Windows Phone, y un 1,7% de Linux.



Aplicación Nativa

Al comenzar con la investigación para el desarrollo, se optó por la modalidad de realizar una aplicación nativa. La puesta a puesta a punto del entorno de desarrollo es simple y rápida, estando respaldada por una gran comunidad de desarrolladores y una amplia documentación provista por Google.

Tanto el IDE (integrated development environment) y el SDK [09] (software development kit) son provistos por Google y su instalación tanto para entornos Windows como Linux son simples y rápidas.

Las aplicaciones nativas Android se desarrollan en lenguaje Java, pero tienen varios conceptos nuevos diferentes a las aplicaciones visuales como puede ser una aplicación web, por lo que la curva de aprendizaje puede ser más larga.

Las primeras pruebas que se hicieron fueron principalmente para entender el enfoque que se le da a la parte visual. Para ello se hicieron los primeros tutoriales ofrecidos en la web de desarrolladores de Google. Los resultados fueron varias aplicaciones, cada una de ellas incluía sólo una idea de todas las cosas que se querían hacer, teniendo así una aplicación que mostraba un menú desplegable, otra un formulario de inicio de sesión y otra una lista de ítems presentados en la pantalla.

Dado que durante 3 meses de estudio se pudieron obtener funcionalidades aisladas y pruebas de concepto, por lo que se llegó a la conclusión que para poder completar el desarrollo había muchos

conceptos que investigar y poco tiempo para hacerlo, razón por la cual se descartó este enfoque en pos del desarrollo de una aplicación híbrida.

Aplicación Híbrida

Luego de haber descartado la modalidad nativa, se empezó a investigar la modalidad híbrida. Ésta presentaba varias ventajas sobre la anterior. El desarrollo se hacía sobre una tecnología ya trabajada (JavaScript, CSS y HTML) y altamente portable a cualquier tipo de dispositivo de acceso, ya sea un dispositivo móvil o un ordenador.

Este tipo de aplicaciones son básicamente aplicaciones web, pero como se explica anteriormente, tienen la posibilidad de acceder a funcionalidades propias del dispositivo, como el GPS, la cámara y otras funciones más. Al ser una aplicación web, la curva de aprendizaje es un poco más corta; y la documentación, tutoriales y ayuda disponibles es más extensa que la disponible para el desarrollo de una aplicación móvil nativa, por el sólo hecho de ser una tecnología que está vigente hace mucho más tiempo.

Como fue dicho anteriormente, este enfoque permite que la aplicación sea altamente portable, porque la aplicación es siempre la misma, lo único que cambia es el contenedor, puede ser compilada sin dificultades para Android, iOS o Windows Phone sin mayores problemas.

Aplicaciones Desarrolladas

Inicialmente la aplicación iba a ser orientada a consumir servicios expuestos a través de la API de SIU Guaraní pero, al no estar completo el desarrollo de ésta última, se tuvieron que generar dos aplicaciones.

La primera iba a ser un clon de SIU Guaraní, con menos lógica, pero centralizando su funcionalidad en la disponibilidad de servicios REST para ejecutar las tareas necesarias sobre el sistema. Tener una aplicación REST implica que se exponga una interfaz para poder, a través de http, enviar y recibir información que impacte sobre el sistema.

La segunda aplicación seguiría siendo entonces la aplicación móvil. Ésta se comunicaría con la primera para realizar todas las operaciones soportadas para el manejo de la carrera del alumno.

A continuación, se describirán ambas aplicaciones con más detalle en cuanto a su proceso de desarrollo, arquitectura y servicios o funcionalidades que ofrecen.

Siuback

Como se dijo anteriormente, ésta aplicación expone varios servicios para operar con los datos del alumno. Los servicios disponibles abarcan la consulta de la historia académica del alumno, generar o borrar inscripciones y servicios de localización de compañeros de cursada. Está pensada para ser solo una aplicación de prueba, pero está lo suficientemente organizada como para poder ser un intermediario entre la aplicación móvil y SIU Guaraní en su momento.

SiuBack implementa una api REST para poder realizar todas las operaciones públicas permitidas para un usuario incluyendo los correspondientes estados http [11]. Esto permite que no solo se pueda acceder desde una aplicación móvil, sino desde cualquier tipo de aplicación desarrollada para entender una API REST y sepa interpretar JSON como respuesta.

API

Una API (application programming interface) como su nombre lo indica, son interfaces de acceso a una aplicación. Son un conjunto de subrutinas, funciones y procedimientos que permiten adicionar a una aplicación una capa de abstracción, evitando que un consumidor de la misma tenga que estar pendiente de la implementación al momento de utilizarla para un nuevo desarrollo.

Representa la capacidad de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio. Las API asimismo son abstractas: el software que proporciona una cierta API generalmente es llamado la implementación de esa API.

Hoy en día la gran mayoría de las empresas y sitios web implementan una API de comunicación al exterior, permitiendo a los desarrolladores utilizar sus funcionalidades e incluirlas en los desarrollos propios. Algunas de las empresas o sitios que implementan API para desarrolladores son:

- Facebook [23]
- Twitter [24]
- Google Maps [25]
- MercadoLibre [26]

REST

La Transferencia de Estado Representacional (Representational State Transfer) o REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo.

Si bien el término REST se refería originalmente a un conjunto de principios de arquitectura, en la actualidad se usa en el sentido más amplio para describir cualquier interfaz entre sistemas que utilice directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML, JSON, etc) sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes, como por ejemplo SOAP. Es posible diseñar sistemas de servicios Web de acuerdo con el estilo arquitectural REST y también es posible diseñar interfaces XMLHTTP de acuerdo con el estilo de llamada a procedimiento remoto (RPC), pero sin usar SOAP. Estos dos usos diferentes del término REST causan cierta confusión en las discusiones técnicas, aunque RPC no es un ejemplo de REST.

Un concepto importante en REST es la existencia de recursos (elementos de información), que pueden ser accedidos utilizando un identificador global (un Identificador Uniforme de Recurso). Para manipular estos recursos, los componentes de la red (clientes y servidores) se comunican a través de una interfaz estándar (HTTP) e intercambian representaciones de estos recursos (los ficheros que se descargan y se envían).

Servicios Disponibles

El sistema soporte de la aplicación móvil tiene implementados todos los servicios básicos para operar como alumno.

Login

Este servicio se encarga a partir de un usuario y contraseña, generar una entrada de login en el sistema con los datos del usuario logueado más una clave generada única para mantener sesión a lo largo de los requests. Ésta clave generada tiene una validez de una hora, por lo que, si el usuario no realiza ninguna acción dentro de ese tiempo, tendrá que volver a autenticarse para seguir operando. Cada operación que el usuario realiza usando la clave se reestablece el tiempo de espera nuevamente a una hora, extendiendo el tiempo en que el usuario puede seguir operando.

Para el envío de usuario y contraseña, se debe agregar al header “Authorization” el valor “Basic” que es estático, y separado por espacio el usuario y contraseña separados por dos puntos y encodeado en base 64. Una vez recuperado el loginKey, éste debe pasarse a todos los request que requieran autenticación en un nuevo header llamado “siuback-loginkey”

Url:
`{server}/user/login`
Method Type: GET
Headers:

- Authorization. Ej: Basic c2FsYWx1aXM6MTIzNA==

Respuesta:

```
{
  "loginKey": "c2FsYWx1aXMxMjM0MTQ0NDA0NDkwMTczNQ==",
  "status": "SUCCESSFUL"
}
```

Datos propios

Este servicio fue inicialmente pensado para devolver todos los datos referentes al usuario, como por ejemplo nombre, apellido, teléfono, carreras y todos los datos que estén guardados en SIU. Actualmente y para esta solución solo fue necesario recuperar las carreras a las que el usuario está inscripto.

Url:
`{server}/user/me`
Method Type: GET
Headers

- siuback-loginkey. Ej: c2FsYWx1aXMxMjM0MTQ0NDA0NDkwMTczNQ==

Respuesta:

```
{
  "careers": [
    {
      "academicUnit": {
        "id": "INF",
        "name": "Informática"
      },
      "career": {
        "id": "LS",
        "name": "Licenciatura en Sistemas"
      },
```

```
      "careerPlan": {
        "id": "LS2003",
        "name": "Licenciatura en Sistemas Plan 2003"
      },
      "date": "Oct 5, 2015 7:34:30 AM",
      "id": "21454126",
      "name": "21454126"
    }
  ]
}
```


Historia Académica

Este servicio, como lo dice el título, recupera la historia académica del usuario que está logueado. Como requiere información sensible del usuario, es necesario enviar el loginKey junto con el servicio. Aquí se recuperarán tanto los exámenes como las cursadas aprobadas del usuario, así como el porcentaje completo de la carrera seleccionada. Por cada materia en el listado, se agrega la cursada, y si existe, la información del examen final. Como se nombró anteriormente, estos servicios responden al estándar REST, por lo que las urls se componen de recursos.

Url:
`{server}/academicunit/{idUnidadAcademica}/career/{idCarrera}/careerplan/{idPlan}/.`
Ej: `{server}/academicunit/INF/career/LS/careerplan/LS2003/`
Method Type: GET
Headers

- siuback-loginkey. Ej: c2FsYWx1aXMxMjM0MTQ0NDA0NDkwMTczNQ==

Respuesta:


```
{
  "subjects": [
    {
      "exam": {
        "subject": {
          "id": "SI101",
          "name": "Matemática 1"
        },
        "grade": 6,
        "date": "Oct 5, 2015 7:34:30 AM",
        "id": "7199108",
        "name": "7199108"
      },
      "subjectRegularity": {
        "subject": {
          "id": "SI101",
          "name": "Matemática 1"
        },
        "date": "Oct 5, 2015 7:34:30 AM",
        "id": "5640064",
        "name": "5640064"
      }
    }
  ],
  "examPercentage": 14.285714285714286,
  "subjectsPercentage": 46.42857142857143,
  "fullPercentage": 30.357142857142858
}
```


Servicio de Materias

El servicio de recuperación de materias está hecho para obtener las materias disponibles para inscripción del alumno, tanto como para una nueva cursada como para rendir el examen final. Cada materia tiene junto a su código y nombre, información de las variantes o posibles mesas a las que se puede inscribir el alumno. Adicionalmente, se puede pedir al servicio que se recupere también información de las regularidades e inscripciones actualmente vigentes.

Url:
`{server}/academicunit/{idUnidadAcademica}/career/{idCarrera}/careerplan/{idPlan}/subject`
Ej: `{server}/academicunit/INF/career/LS/careerplan/LS2003/subject`
Method Type: GET
Headers
siuback-loginkey. Ej: c2FsYWx1aXMxMjM0MTQ0NDA0NDkwMTczNQ==
Parámetros

- regularities: boolean. Incluye regularidades en la respuesta.
- inscriptions: boolean. Incluye inscripciones en la respuesta.

Respuesta:


```
{
  "subjectRegularities": [
    {
      "subject": {
        "examVariants": [
          {
            "id": "1",
            "date": "Oct 5, 2015 7:34:29 AM"
          },
          {
            "id": "2",
            "date": "Oct 5, 2015 7:34:29 AM"
          }
        ],
        "subjectInscriptionVariants": [
          {
            "id": "1",
            "date": "Oct 5, 2015 7:34:29 AM"
          },
          {
            "id": "2",
            "date": "Oct 5, 2015 7:34:29 AM"
          }
        ],
        "id": "SI208",
        "name": "Taller de lecto-comprensión y traducción en Ingles"
      },
      "date": "Oct 5, 2015 7:34:30 AM",
      "id": "33083095",
      "name": "33083095"
    }
  ],
"subjectInscriptions": [
    {
      "variant": {
        "id": "1",
        "date": "Oct 9, 2015 7:17:51 AM"
      },
      "subject": {
        "examVariants": [
          {
            "id": "1",
            "date": "Oct 9, 2015 7:17:51 AM"
          },
          {
            "id": "2",
            "date": "Oct 9, 2015 7:17:51 AM"
          }
        ],
```

```
        "subjectInscriptionVariants": [
          {
            "id": "1",
            "date": "Oct 9, 2015 7:17:51 AM"
          },
          {
            "id": "2",
            "date": "Oct 9, 2015 7:17:51 AM"
          }
        ],
        "id": "S0303",
        "name": "Bases de Datos 1 (*)"
      },
      "date": "Oct 9, 2015 7:34:43 AM",
      "id": "33073840",
      "name": "inscription1625071223"
    }
  ],
  "availableSubjects": [
    {
      "examVariants": [
        {
          "id": "1",
          "date": "Oct 5, 2015 7:34:29 AM"
        },
        {
          "id": "2",
          "date": "Oct 5, 2015 7:34:29 AM"
        }
      ],
      "subjectInscriptionVariants": [
        {
          "id": "1",
          "date": "Oct 5, 2015 7:34:29 AM"
        },
        {
          "id": "2",
          "date": "Oct 5, 2015 7:34:29 AM"
        }
      ],
      "id": "SI306",
      "name": "Conceptos y Paradigmas de Lenguajes de Programación"
    }
  ],
  "availableExams": [
    {
      "examVariants": [
        {
          "id": "1",
          "date": "Oct 5, 2015 7:34:29 AM"
        },
        {
          "id": "2",
          "date": "Oct 5, 2015 7:34:29 AM"
        }
      ],
      "subjectInscriptionVariants": [
        {
          "id": "1",
          "date": "Oct 5, 2015 7:34:29 AM"
        },
        {
          "id": "2",
          "date": "Oct 5, 2015 7:34:29 AM"
        }
      ],
      "id": "SI105",
      "name": "Arquitectura de Computadoras"
    }
  ]
}
```

Servicio de Exámenes

El servicio de recuperación de exámenes recupera toda la información de los exámenes del alumno, es decir, la nota, fecha en que se rindió el examen y la materia asociada. Adicionalmente se le puede pedir al servicio que devuelva también las inscripciones actuales a exámenes a través de un parámetro llamado “inscriptions”.

Url:
`{server}/academicunit/{idUnidadAcademica}/career/{idCarrera}/careerplan/{idPlan}/exams`
Ej: `{server}/academicunit/INF/career/LS/careerplan/LS2003/exams`
Method Type: GET
Headers

- siuback-loginkey. Ej: c2FsYWx1aXMxMjM0MTQ0NDA0NDkwMTczNQ==

Parámetros

- inscriptions: boolean. Incluye inscripciones en la respuesta.

Respuesta:


```
{
  "exams": [
    {
      "subject": {
        "examVariants": [
          {
            "id": "1",
            "date": "Oct 9, 2015 7:17:51 AM"
          },
          {
            "id": "2",
            "date": "Oct 9, 2015 7:17:51 AM"
          }
        ],
        "subjectInscriptionVariants": [
          {
            "id": "1",
            "date": "Oct 9, 2015 7:17:51 AM"
          },
          {
            "id": "2",
            "date": "Oct 9, 2015 7:17:51 AM"
          }
        ],
        "id": "SI103",
        "name": "Algoritmos, Datos y Programas"
      },
      "grade": 8,
      "date": "Oct 9, 2015 7:17:51 AM",
      "id": "20479133",
      "name": "20479133"
    }
  ],
  "examInscriptions": []
}
```


Servicio de Inscripciones

El servicio de inscripciones permite al alumno inscribirse tanto a materias como exámenes. Internamente se hacen las comprobaciones correspondientes para que no se realice una inscripción a

una materia que no corresponda. Si es un examen, se comprueba que la cursada esté aprobada y que estén rendidos los exámenes finales de sus correlativas. Si es una inscripción a una materia, también se realiza el análisis de materias correlativas necesarias.

En el cuerpo del mensaje enviado al servidor, es necesario especificar si se trata de una inscripción de examen o materia. Para ello se agrega un campo “inscriptionAction”, y los posibles valores son “SUBJECT_INSCRIPTION” en el caso de ser una inscripción a una cursada, o “EXAM_INSCRIPTION” en el caso de ser una inscripción a un examen. Junto a estos datos, también es necesario indicar la variante (ya sea mesa u horario de cursada) a la que se quiere inscribir, enviando el valor en la clave “variant”.

La respuesta al servicio retorna la información de la inscripción recién creada, junto a la información de la materia asociada.

Url:
`{server}/academicunit/{idUnidadAcademica}/career/{idCarrera}/careerplan/{idPlan}/subject/{subjectId}`
Ej: `{server}/academicunit/INF/career/LS/careerplan/LS2003/subject/SI104`
Method Type: POST
Headers

- siuback-loginkey. Ej: c2FsYWx1aXMxMjM0MTQ0NDA0NDkwMTczNQ==

Body:

```
{
"inscriptionAction":"SUBJECT_INSCRIPTION" / "EXAM_INSCRIPTION" ,
        "variant":"1"
}
```

Respuesta:

```
{
  "variant": {
    "id": "1",
    "date": "Oct 10, 2015 2:19:50 PM"
  },
  "subject": {
    "examVariants": [
      {
        "id": "1",
        "date": "Oct 10, 2015 2:19:50 PM"
      },
      {
        "id": "2",
        "date": "Oct 10, 2015 2:19:50 PM"
      }
    ],
    "subjectInscriptionVariants": [
      {
        "id": "1",
        "date": "Oct 10, 2015 2:19:50 PM"
      },
      {
        "id": "2",
        "date": "Oct 10, 2015 2:19:50 PM"
      }
    ],
    "id": "S0303",
    "name": "Bases de Datos 1 (*)"
  },
  "date": "Oct 10, 2015 2:43:43 PM",
  "id": "4285139",
  "name": "inscription1625071223"
}
```

Servicio de borrado de inscripciones

Una vez creadas las inscripciones, las mismas pueden ser borradas. Dado que sigue el estándar REST, el borrado es simplemente usar el método DELETE al recurso seleccionado. La respuesta a este servicio son los datos de la inscripción borrada.

Url:
`{server}/academicunit/{idUnidadAcademica}/career/{idCarrera}/careerplan/{idPlan}/subject/{subjectId}/inscription/{inscriptionId}`
Ej: `{server}/academicunit/INF/career/LS/careerplan/LS2003/subject/S0303/inscription/4285139`
Method Type: DELETE
Headers

- siuback-loginkey. Ej: c2FsYWx1aXMxMjM0MTQ0NDA0NDkwMTczNQ==

Respuesta:


```
{
  "variant": {
    "id": "1",
    "date": "Oct 10, 2015 2:19:50 PM"
  },
  "subject": {
    "examVariants": [
      {
        "id": "1",
        "date": "Oct 10, 2015 2:19:50 PM"
      },
      {
        "id": "2",
        "date": "Oct 10, 2015 2:19:50 PM"
      }
    ],
    "subjectInscriptionVariants": [
      {
        "id": "1",
        "date": "Oct 10, 2015 2:19:50 PM"
      },
      {
        "id": "2",
        "date": "Oct 10, 2015 2:19:50 PM"
      }
    ],
    "id": "S0303",
    "name": "Bases de Datos 1 (*)"
  },
  "date": "Oct 10, 2015 2:43:43 PM",
  "id": "4285139",
  "name": "inscription1625071223"
}
```


Servicio de compañeros cerca

Este servicio se encarga de recuperar los compañeros que se encuentran cerca de su ubicación. Solo busca entre los alumnos que están efectivamente cursando las mismas materias que el alumno que lo esté consultando, y está limitado a un radio específico cerca de la facultad, por lo que solo se podrá ver los compañeros cerca cuando se esté cerca de la facultad y los compañeros lo estén también.

Este servicio soporta al mismo momento, actualizar la ubicación actual del usuario logueado, utilizando en vez del método GET, el método POST. La respuesta al servicio es un listado de compañeros cerca, con datos de coordenadas y nombre del usuario en sí. En caso de que algún usuario haya agregado un estado a su localización, también se retorna el mismo.

Url:
`{server}/user/me/nearby`
Method Type: GET / POST
Headers

- siuback-loginkey. Ej: c2FsYWx1aXMxMjM0MTQ0NDA0NDkwMTczNQ==

Body:
```
{
    "coordinates":"-34.903808,-57.938117"
}
```

Respuesta:
```
[
  {
    "student": {
      "id": "2",
      "name": "Don Rodrigo de la Vega"
    },
    "status": "un estado",
    "coordinate": {
      "longitude": -57.938117,
      "latitude": -34.903808
    }
  }
]
```

Servicio de actualización de localización

Este servicio se encarga de actualizar la información de localización del usuario logueado. Al enviar la localización al servidor, esta comprueba que el usuario esté dentro del área cerca de la facultad, y en caso de no estarlo, no se guarda la localización. Este servicio también permite asignar un estado a la actualización, permitiendo que luego al consultarse, los compañeros puedan ver ese estado. La respuesta a este servicio es simplemente un resumen de los datos del usuario junto a la información de la localización modificada.

Url:
`{server}/user/me`
Method Type: PUT
Headers

- siuback-loginkey. Ej: c2FsYWx1aXMxMjM0MTQ0NDA0NDkwMTczNQ==

Body:
```
{
    "coordinates":"-34.903808,-57.938117",
    "coordinates":"un estado"
}
```

Respuesta:
```
        {
  "careers": [
    {
      "academicUnit": {
        "id": "INF",
        "name": "Informática"
      },
      "career": {
        "id": "LS",
```

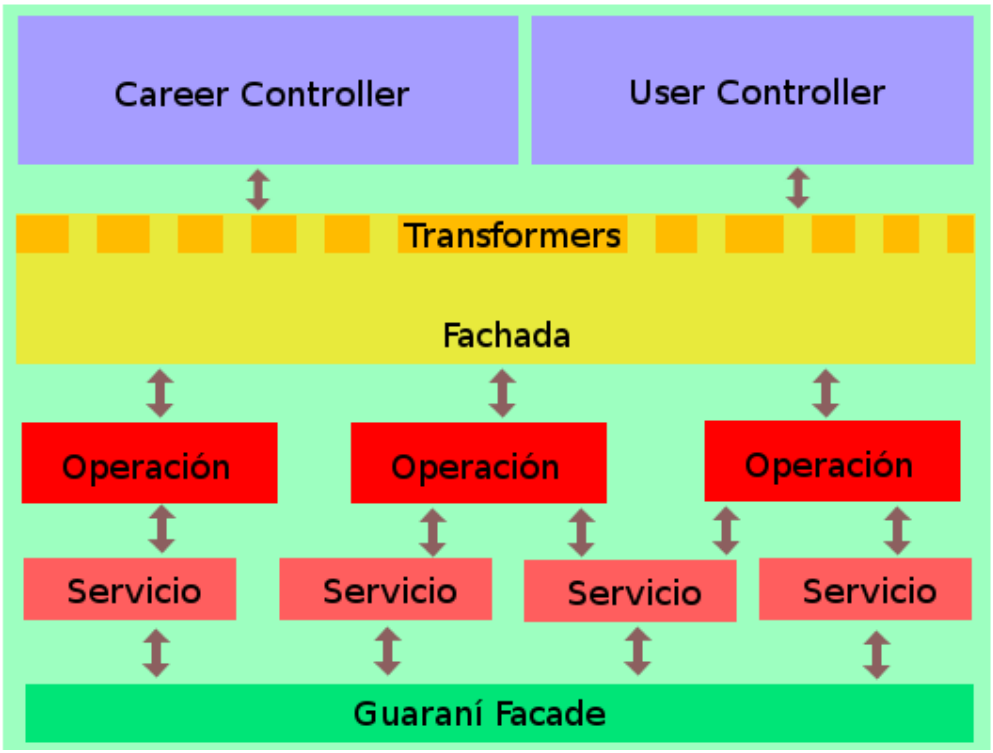
```
        "name": "Licenciatura en Sistemas"
      },
      "careerPlan": {
        "id": "LS2003",
        "name": "Licenciatura en Sistemas Plan 2003"
      },
      "date": "Oct 10, 2015 2:19:50 PM",
      "id": "31193592",
      "name": "31193592"
    }
  ],
  "lastSeen": {
    "coordinates": "-34.903808, -57.938117",
    "lastChecked": "Oct 10, 2015 3:38:54 PM"
  }
}
```


SIU Back - Arquitectura de la Solución

La aplicación está desarrollada enteramente en JAVA, con el uso de varios frameworks y herramientas comunes para este tipo de desarrollo web, como, por ejemplo, Spring para el manejo de inyección de dependencias, Spring MVC para la creación de los servicios, y maven para la compilación y empaquetación.

Está organizada en capas. Cada capa tiene su función de lógica y abstracción de datos, lo que permite que sea de fácil mantenimiento y, como se menciona anteriormente, en caso de tener disponibles los servicios de SIU Guaraní, permite que la conexión sea simple y transparente para las capas superiores.

Actualmente los datos de la aplicación existen en archivos JSON. Estos archivos se leen en una importación inicial al momento de iniciar el sistema. Todos los datos de la aplicación están guardados en memoria, ya que no sería necesario incluir una base de datos al momento de completar la integración.



Controllers

Existen principalmente dos controllers. Un controller es aquel que atiende request HTTP y, en este caso, responde a ellos con un documento JSON. Las URL están organizadas de manera tal que estén lo más acorde posible a la arquitectura REST, donde cada URL corresponde a un recurso. Por ejemplo:

- siuback.com/carrera: Respondería a todos los request que tengan que ver con una carrera
- siuback.com/carrera/10: Responde a todas las operaciones soportadas para la carrera número 10.

Fachada

La primera fachada va a ser la encargada de ocultar información interna de la aplicación a los servicios expuestos al exterior. Básicamente ésta nuclea las llamadas a todos los servicios internos. La lógica de esta capa es prácticamente nula, su única responsabilidad es hacer de conector entre las operaciones y los controllers.

Dentro de la fachada existen varios objetos encargados de transformar todos los objetos de respuesta que se reciben desde las operaciones en objetos de respuesta que esperan los controllers. Cada uno de estos objetos que transforman información se encarga de una sola parte del objeto y solo para los tipos básicos (String, Integer, Double, Boolean) y para los tipos complejos creados para contener información del sistema, tienen inyectados el transformer específico para ese objeto. Esto permite que cada uno sea muy específico, y el cambio de código en caso de agregar algo sea muy puntual y prolijo.

Operaciones

Cada operación es una porción de código que nuclea una acción única, atómica y completa. Está encargada de orquestar las llamadas a los servicios necesarios para completar esta acción, permitiendo al que lo consume únicamente hacer el llamado, sabiendo que es lo que vendrá como respuesta, pero sin preocuparse por saber cómo está llevado a cabo. Por ejemplo, cuando un usuario inicia sesión es necesario obtener los datos del usuario para comprobar la sesión y generar una nueva entrada en la lista de códigos de acceso generado (funcionalidad que será explicada más adelante). Estas dos partes de la acción de iniciar sesión están en distintos servicios, pero no es necesario que el que consume esto tenga que tener la lógica de cómo llamarlos y qué hacer con sus porciones de datos. Para ello existe la operación de iniciar sesión, que se encarga de, con los datos de usuario y contraseña, hacer las dos llamadas a los servicios y devolver únicamente el token generado para el usuario que inició la operación.

Servicios

Los servicios están organizados de tal manera que contengan el código referido a cada vertical del sistema. Es decir, existe un servicio para exámenes, que tiene las funcionalidades de inscribirse, eliminar la inscripción, recuperar inscripciones y recuperar exámenes; existe un servicio de cursadas que, donde están los servicios análogos al servicio de exámenes; y así para el servicio de usuarios, de carreras, de geolocalización, etc.

Guaraní Facade

Esta fachada es la encargada de ocultar la implementación de la comunicación con SIU Guaraní. Para ello se expone una interfaz que nuclea todas las operaciones necesitadas del sistema SIU. Actualmente tiene una única implementación que simula ser el SIU Guaraní y guarda todos los datos del alumno y las operaciones hechas sobre el sistema en memoria.

Dado que está organizado de esta manera, una vez desarrollados los servicios del SIU Guaraní, ésta sería la única capa a reemplazar, dejando las otras capas sin modificaciones, evitando introducir errores que pueden aparecer al modificar gran parte del sistema.

Manejo de Errores

La aplicación cuenta con un manejo de errores basado en estados http. Junto a estos, también se obtiene una respuesta detallada sobre el error ocurrido en formato json. El json de respuesta tendrá el siguiente formato:

```
{
  "message":"{mensaje}",
  "cause": "{causa}"
}
```

Los posibles códigos de error que se pueden obtener son los siguientes:

- 404: El recurso no existe. Este error no devuelve ningún json ya que no es algo que esté manejado por la aplicación.

- 500: Error interno de la aplicación. Idealmente este error no debería aparecer nunca ya que indica que hubo un error interno en la aplicación, ya sea un NullPointer o algún otro error similar que no fue manejado previamente.
- 401: No autorizado. Este error se obtiene al realizar alguna acción que requiera login evitando enviar el token o enviando un token que ya no es válido.
- 405: Método no permitido. Error que ocurre al momento de intentar hacer una inscripción y algún requisito no está cumplido, como, por ejemplo, faltan correlativas.
- 409: Conflicto. Éste código es utilizado al momento de notificar la localización a la aplicación estando fuera del radio permitido.
- 400: Request incorrecto. Cuando un usuario intenta autenticarse con usuario o contraseña inválidos, recibe este error indicando que las credenciales no son correctas.

Servidor

La aplicación está deployada en un servidor Java llamado OpenShift [10], de Redhat. Este servidor permite tener varias aplicaciones de manera gratuita, aunque puede que suspenda las aplicaciones por falta de uso, provocando que el primer acceso a la aplicación luego de un tiempo de inactividad sea algo lento. La URL de este servidor es “siuback-salaesp.rhcloud.com”.

OpenShift de RedHat

OpenShift es un producto de computación en la nube de plataforma como servicio de Red Hat. Es un servicio que es de código abierto bajo el nombre de "OpenShift Origin", y está disponible en GitHub [10]. Los desarrolladores pueden usar Git segurizado bajo ssh para desplegar sus aplicaciones Web en los diferentes lenguajes de la plataforma. Los lenguajes soportados son:

- Haskell
- Java
- JavaScript
- .NET[3]
- Perl
- PHP
- Python
- Ruby

OpenShift también soporta programas binarios que sean aplicaciones Web, con tal de que se puedan ejecutar en RHEL Linux. Esto permite el uso de lenguajes arbitrarios y frameworks. OpenShift se encarga de mantener los servicios subyacentes a la aplicación y la escalabilidad de la aplicación como se necesite.

Permite deployar la aplicación al momento de subir el código al repositorio de manera automática. La herramienta ofrece varios planes de inscripción. El plan que se eligió fue el gratis, el cual permite tener hasta 3 aplicaciones con equipos de poca potencia, apagando el equipo luego de 24hs de estar en desuso.

SiuFront

La aplicación SiuFront es la aplicación que será instalada en los móviles. Como se discutió previamente, está pensada para ser un híbrido entre una aplicación web y una de teléfonos celulares, permitiendo así poder llevar a cabo el desarrollo sin ser necesario probar el resultado en ningún dispositivo hasta las etapas finales del proyecto. El fin de ésta es hacer accesibles todos los servicios de SiuBack a través de un celular y de la forma más amigable posible.

Una vez instalada, la aplicación podría ser accedida desde el menú de aplicaciones del celular a través del siguiente ícono:

Ícono de la aplicación

Pantallas creadas

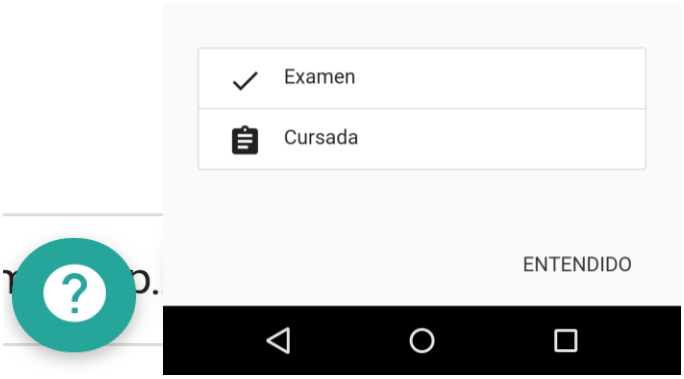
Se crearon entonces varias pantallas dedicadas a poder acceder a todos los servicios ofrecidos. Todas ellas pueden ser accedidas a través de un menú desplegable desde la izquierda de la pantalla. En la aplicación móvil, el menú se visualiza de la siguiente manera:

Contenido del menú lateral expansible

Cada una de las acciones hechas por el usuario están acompañadas por un feedback de la aplicación indicando el resultado de la misma, ya sea que se produjo un error o que la operación se completó exitosamente. La manera de verse en la aplicación es la siguiente:

Notificación Toast de carrera seleccionada

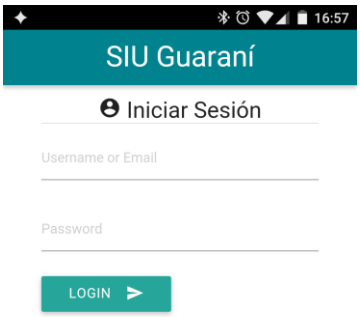
Para facilitar el uso y mejorar el entendimiento de la aplicación, se agregó un botón de ayuda que estará presente en todo momento durante la navegación de la aplicación, el cual muestra el significado de cada ícono:



A la izquierda el ícono de ayuda, a la derecha el pop up de ayuda

Inicio de Sesión

El inicio de sesión es lo primero que aparece al momento de abrir la aplicación. Se le pide al usuario que ingrese su nombre de usuario y contraseña para poder luego entrar al sistema.



Formulario de inicio de sesión

Selección de carrera

Al entrar a la aplicación, lo primero que se observa es la pantalla de selección de carrera. Esta carrera luego se podrá modificar mientras se está logueado, pero es necesario seleccionarla al inicio para comenzar a operar.

Tarjeta de carrera disponible

Historia Académica

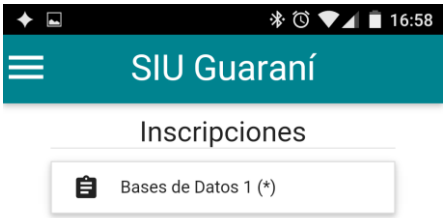
Una vez autenticado en el sistema y seleccionada la carrera, el usuario verá la historia académica. Es un listado de materias con la información de cada una. Las materias que están marcadas con una libreta, solo tienen la cursada aprobada, por lo que la información que se mostrará al hacer clic sobre ella será solo la fecha de fin de cursada. En caso de que la materia esté marcada con una tilde, ésta tendrá también el examen final aprobado, y a la información se le adicionará la fecha y nota del examen.

Arriba se mostrará también el porcentaje actual de la carrera, tomando en cuenta tanto las cursadas como exámenes finales restantes.

A la izquierda se muestra la historia académica, a la derecha los datos de una materia en particular.

Inscripciones

En esta pantalla se muestran las inscripciones actuales del usuario, ya sea para exámen o para cursada. Al igual que en la historia académica, existen diferentes íconos para identificarlas. Las materias que tienen una libreta a su lado, son inscripciones a cursadas, en cambio las que tienen una tilde son inscripciones a exámenes.



Listado de inscripciones

Al hacerle clic a cualquier inscripción, se pueden ver los detalles de la misma. También se muestra la opción de eliminarla, acción que una vez ejecutada requerirá una confirmación por parte del usuario para completarla.

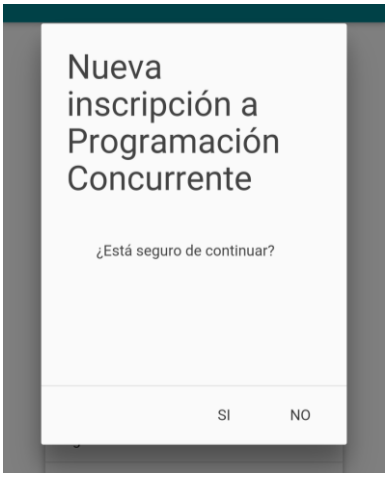
Pop-up de borrado de inscripción

Crear Inscripción

Para crear una inscripción, el usuario deberá hacer clic en el botón con un símbolo “+”. Al hacerlo se desplegará un menú con los dos tipos de inscripciones. Nuevamente se utilizan los mismos íconos que antes para identificar las materias. Una libreta para las cursadas y una tilde para los exámenes.

Botones para agregar inscripciones

Una vez hecho clic sobre cualquiera de las dos opciones, se mostrará la página de crear una nueva inscripción. En esta página no se utilizarán los íconos identificadores de materias ya que se seleccionó previamente el tipo de inscripción a realizar. Las materias que se son todas las disponibles, aunque puede que por falta de correlativas no se pueda llevar a cabo la inscripción. Al momento de hacer clic en una materia, se desplegarán las variantes o fechas disponibles. Cuando se selecciona una variante, una ventana de confirmación de la inscripción aparecerá.

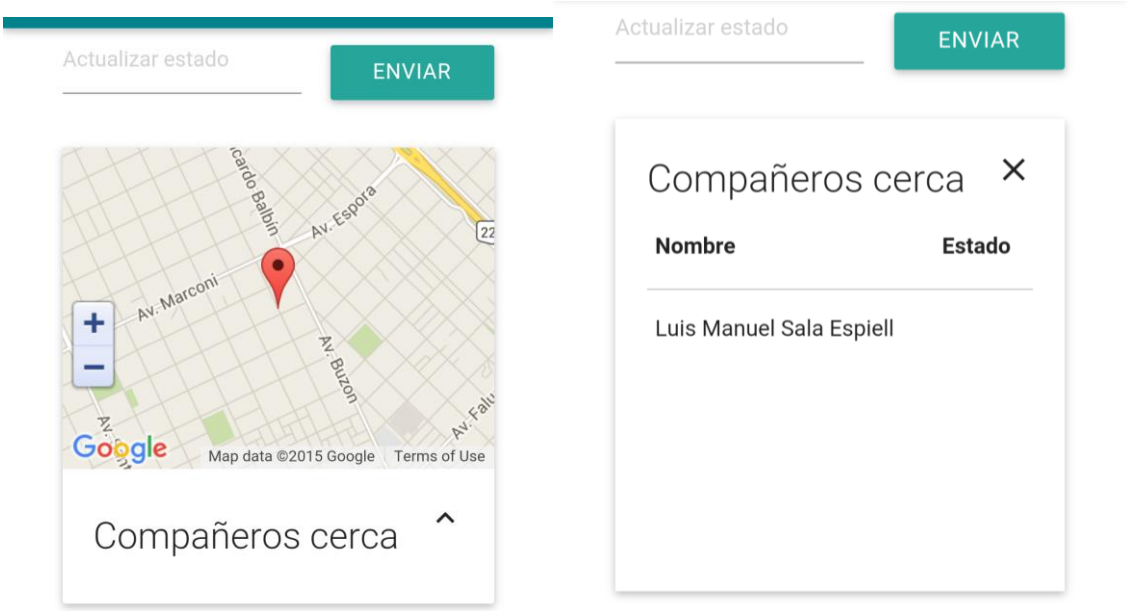


Pop-up de nueva inscripción

Pantalla de compañeros cerca

En esta última pantalla se puede ver un mapa con los compañeros cerca del usuario logueado. Cabe recordar que para poder consultar los compañeros cerca se debe estar dentro de un radio de 100 metros de la facultad. Junto con el mapa hay una lista de los compañeros que actualizaron su ubicación, y en caso de haber actualizado su estado, también se puede consultar.

Desde esta pantalla también se puede actualizar el estado personal. Al enviar el estado, éste se actualiza en el servidor y, junto con él, se actualiza también la ubicación del usuario conectado.



A la izquierda, mapa con compañeros cerca. A la derecha la lista de compañeros cerca.

SIU Front - Arquitectura de la solución

Como se comentó previamente, la aplicación móvil sería desarrollada como una página web, y luego empaquetada como una aplicación de celular, por lo que la gran mayoría de las tecnologías usadas son orientadas a la programación web. Nuevamente la aplicación está organizada en capas, facilitando así el análisis y mantenimiento de todo el sistema.

Una de los primeros frameworks involucrados en este desarrollo fue NodeJs. Es un framework que hace posible ejecutar código JavaScript del lado del servidor, permitiendo generar o incluir módulos (también en JavaScript) que solucionan varias funcionalidades de un desarrollo.

Para el startup de la aplicación se utilizó una herramienta de scaffolding. Este tipo de herramientas permite generar las estructuras de proyecto de una manera rápida, sencilla y robusta, ayudando a la organización del código. Para este desarrollo se utilizó Yeoman, un módulo generado para NodeJs. Como se comentó previamente, Yeoman incluye las herramientas y frameworks necesarios para poder realizar aplicaciones (en esta caso web) de manera rápida y simple.

En esta estructura de proyecto creada, se incluyó Grunt [15], una librería JavaScript que permite configurar y correr tareas automáticas como puede ser compilar, correr una batería de tests, minificar el código y generar una versión distribuible del sistema o aplicación web desarrollada. Por defecto Yeoman configura ciertas tareas genéricas como para poder realizar las tareas nombradas anteriormente, pero es posible extender estas mismas modificando el archivo “gruntfile” donde en formato JavaScript se declara un módulo para NodeJs. Particularmente para este proyecto se generó una nueva tarea que permite al desarrollador tener un archivo de configuración para cada entorno de desarrollo, pudiendo así usar el backend local o remoto sin tener que hacer modificaciones en el código.

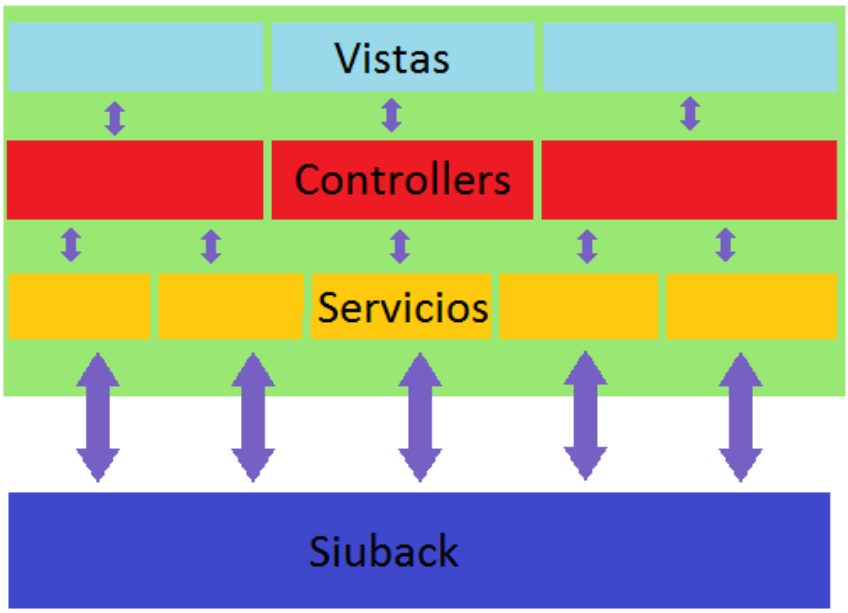
Junto a las ya nombradas, se incluyó dentro del proyecto una librería de resolución de dependencias llamada Bower [16]. Es una práctica muy común en cualquier lenguaje de programación con cierta evolución incluir un gestor de dependencias. Este tipo de librerías permiten al programador no preocuparse al momento de requerir software (frameworks o librerías) de otros desarrolladores, permitiendo que con una simple configuración se puedan incluir nuevas dependencias sólo indicando el nombre y la versión o rango de versiones que se quiere usar. Particularmente Bower utiliza un archivo de configuración llamado “bower.json”, donde en un formato json se agregan como key el nombre de la dependencia y como valor la versión o rango de versiones que se quiere de la misma. Además de sacar trabajo al programador al momento de obtener estas dependencias, permite que el código del proyecto sea más limpio y más liviano, ya que, al momento de subirlo al repositorio de código, no es necesario incluir los archivos que se resolvieron con Bower, sino que con solo subir el archivo de configuración ya se puede replicar en cualquier máquina el ambiente original donde se hicieron uso de las librerías.

Algo no menor en una aplicación es el tema visual. Dado que se seleccionó Android como sistema operativo para el desarrollo, la aplicación debería verse como una aplicación nativa. Existen varios frameworks que agilizan el estilado de una página web (Bootstrap [18], Foundation [19], entre otros). Una de ellas, y la seleccionada para este desarrollo, es materializecss [17]. Esta librería permite de una manera fácil y rápida modificar una aplicación web para seguir los estándares de Material Design de Google [20]. Hace ya unos años que Google implementó este estándar de diseño, por lo cual los usuarios

están muy acostumbrados. Ésto agrega a la aplicación en desarrollo facilidad al momento de comenzar a usarla, ya que la navegabilidad y visualización de la información es algo conocido y no rompe con la armonía del sistema.

Por último, la aplicación fue desarrollada utilizando Angularjs [12][13], un framework que implementa MVC para el desarrollo front-end y que permite crear aplicaciones de una sola página (SPA single page application). El framework permite a su vez agregar nuevas librerías que resuelven problemas comunes, como puede ser el asociar una URL a un controller y una vista y hasta manejo de proxies. Una librería muy importante, y que vale la pena nombrar, es angular-google-maps, que como el mismo nombre lo indica, fue hecha para facilitar el uso de la API de Google Maps [14], y es la utilizada al momento de mostrar los compañeros cerca en el mapa.

A continuación, podemos ver la arquitectura organizada en capas.



Vistas

Las vistas son páginas en lenguaje HTML pero con la particularidad de tener ciertos tags de Angular. Éstos últimos permiten que se puedan renderizar los datos obtenidos desde el controller. Existe una vista por cada una de las páginas que se quieren mostrar, y una extra donde está el contenedor de Angular, llamada index.html, el cual será nuestro punto de entrada a la aplicación.

Como se mencionó previamente, este tipo de aplicaciones no cambia de página cada vez que se accede a una nueva, sino que es una aplicación de una sóla página, por lo que lo único que se modifica es el contenedor de la aplicación, evitando así que el usuario tenga que esperar que cargue todo cada vez que accede a un nuevo link, y para el caso de nuestra aplicación, permite que la navegación sea más cercana a una aplicación nativa.

Controllers

Los controllers son los encargados de responder a cada una de las URLs de la aplicación. Para cada controller existe una vista asociada, la cual se pintará luego de completar las acciones necesarias. Estos exponen también métodos por los cuales la vista puede volver a comunicarse con ellos, permitiendo generar llamadas a partir de eventos de la vista.

Éstos son los encargados de comunicar las vistas con los servicios, encapsulando así la lógica de la aplicación y permitiendo cierto grado de abstracción al trabajar con la vista.

Servicios

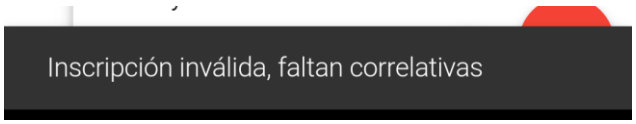
Por último, se encuentran los servicios. Éstos son los encargados de comunicarse con sistema backend. Al igual que los servicios de SiuBack, estos están organizados por recursos (servicio de materias, servicio de usuario, servicio de historia académica), siendo así más fácil identificar cual es el que responde a lo que estamos necesitando.

Manejo de Errores

Dado que el sistema SiuBack está preparado para retornar diferentes códigos http de acuerdo a su resultado, permite a la aplicación móvil tener un manejo más controlado de los errores.

Actualmente, para los casos comunes, se muestra un mensaje emergente con el detalle de error. En el caso que el problema sea de autenticación o token inválido o vencido, agregado al mensaje visual, se borran todos los datos previamente almacenados que son usados para mantener la sesión en la aplicación, enviando al usuario a la pantalla de iniciar sesión.

Por ejemplo, en caso de intentar generar una inscripción a una materia que no tiene todas sus correlativas, el siguiente mensaje aparecerá:



Notificación Toast por falta de correlativas

SECCIÓN 4

Análisis de Uso

Durante el desarrollo, la aplicación fue probada en un ambiente preparado para ello. Con el objetivo de conocer la opinión de usuarios reales potenciales sobre la experiencia en el uso de la aplicación, sus reacciones, opiniones y sugerencias, se llevó a cabo una actividad a modo de focus groups distribuida en diferentes sesiones según la disponibilidad de los usuarios.

El grupo de estudio incluyó 15 alumnos de edades variadas, donde más del 50% estaban dentro de la franja de 25 a 30 años, dejando cerca de un 15% para los menores a 25 y un 35% para los mayores a 30 años.

Dado el ámbito donde fue elaborada esta aplicación, la gran mayoría de los alumnos que probaron la misma pertenecen a la Facultad de Informática de la Universidad Nacional de La Plata, conformando el 75% de los totales encuestados. El restante 25% fue completado por alumnos de Arquitectura, Periodismo e Informática de la Unicen (Universidad Nacional del Centro de la Provincia de Buenos Aires). Todos estos encuestados fueron previamente seleccionados debido a que debían cursar en una universidad que actualmente implemente un sistema SIU Guaraní, con el fin de poder probar las funcionalidades que usan con regularidad en el sistema desarrollado.

Para la prueba inicial se distribuyó el código empaquetado en un archivo con extensión ‘apk’, junto a un usuario y contraseña creados en la implementación del backend y a una encuesta de uso.

De las pruebas surgieron los siguientes requerimientos con respecto al dispositivo. Para poder utilizar la aplicación sin problemas es necesario lo siguiente:

- Android 5.0 o posterior
- Aplicación WebView de Android actualizada a la última versión (https://play.google.com/store/apps/details?id=com.google.android.webview)
- Localización de alta precisión activada (utilizando los sensores GPS)

Dado que la aplicación aún no está oficialmente subida al Store de Google, es necesario también habilitar las “Orígenes desconocidos”, opción que permite que se puedan instalar aplicaciones que no provengan del Store oficial de Google. Esta opción puede encontrarse bajo Ajustes -> Seguridad -> Orígenes desconocidos.

El cuestionario que ofrece un modo de guía las actividades propuestas, contenía las siguientes preguntas:

Formulario de Análisis de Uso

SIU Guarani

*Obligatorio

¿Que edad tenés? *

- ○ 18-25
- ○ 25-30
- ○ 30 o más

¿De qué facultad sos? *

Tu respuesta

¿Tu facultad usa SIU Guarani? *

- ○ Si
- ○ No
- ○ No se

Uso de SIU Guarani

¿Con que frecuencia usas SIU Guaraní? *

	1	2	3	4	5	
Nunca	○	○	○	○	○	Todos los días

¿Cuales son las funciones que usas habitualmente del sistema? *

- ☐ Ver la historia académica
- ☐ Inscribirme a cursadas
- ☐ Inscribirme a exámenes
- ☐ Completar encuestas
- ☐ Otros: _____

¿Ingresaste alguna vez al SIU Guaraní desde tu celular? *

- ○ Si
- ○ No

¿Considerás útil una aplicación diseñada específicamente para celulares? *

- ○ Si
- ○ No

Historia académica

¿Es acorde el icono que aparece en el menú de aplicaciones? *

- ○ Si
- ○ No
- ○ Otro: _____

¿Pudiste iniciar sesión? *

- ○ Si
- ○ Con dificultad
- ○ No

¿Pudiste ver tu historia académica? *

- ○ Si
- ○ Con dificultad
- ○ No

¿Qué te parece que puede mejorar para llegar a la historia académica?

Tu respuesta

Cada elemento de la historia académica tiene información ¿pudiste accederla? *

- ○ Si
- ○ Con dificultad
- ○ No
- ○ No aplica

En caso de no haber notado lo anterior ¿como se podría mejorar para llamar la atención a verlo?

Tu respuesta

Inscripciones

¿Lograste crear algún tipo de inscripción? *

- ○ Si
- ○ Con dificultad
- ○ No

¿Que tipo de inscripción pudiste crear? *

- ☐ Examen
- ☐ Cursada
- ☐ Ninguna

¿Cómo calificarías el proceso de crear una inscripción? *

	1	2	3	4	5	
Dificil	○	○	○	○	○	Fácil

¿Qué podría mejorar del proceso de crear una inscripción?

Tu respuesta

Una vez creadas las inscripciones ¿pudiste verlas? *

- ○ Si
- ○ No
- ○ Con dificultad

Además de verlas ¿pudiste borrar una? *

- ○ Si
- ○ No
- ○ Con dificultad

¿Cómo calificarías el proceso de borrar una inscripción? *

	1	2	3	4	5	
Dificil	○	○	○	○	○	Fácil

Ubicación

Para esta sección es necesario tener activado el sistema de ubicación

¿Viste el mapa en la sección encontrar amigos? *

- ○ Si
- ○ No
- ○ Con dificultad

¿Te ubicó correctamente? *

- ○ Si
- ○ No
- ○ Otro: _____

¿Pudiste actualizar tu estado con éxito? *

- ○ Si
- ○ No
- ○ Con dificultad

Para lograr ver un compañero cerca es necesario estar inscripto a por lo menos una materia igual. ¿Lograste coordinar con alguien para ello? *

- ○ Si
- ○ No
- ○ Otro: _____

¿Pudiste ver su estado y posición en el mapa? *

- ○ Si
- ○ No

¿Que se podría mejorar de la pantalla de ubicación?

Tu respuesta

¿Cómo calificarías la experiencia de uso de la aplicación?

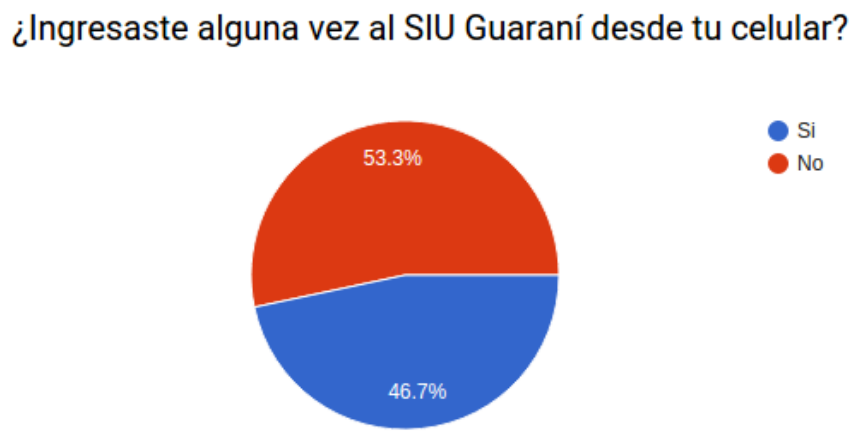
	1	2	3	4	5	6	7	8	9	10	
Muy mala	○	○	○	○	○	○	○	○	○	○	Muy buena

Respuestas

Las respuestas obtenidas se presentan a continuación:

Al revisar la frecuencia de uso se ve que más del 80% utiliza el sistema de manera moderada, accediendo por lo menos varias veces al mes, y contrastado con el uso que se le da al sistema (más del 80% lo usa para crear inscripciones o ver su historia académica) esto genera mucho tráfico en el sistema considerando la cantidad de alumnos inscriptos por facultad.

Continuando con el análisis de la encuesta, se puede ver que el grupo de alumnos que alguna vez intentaron acceder desde el celular es bastante grande:



Esto evidencia la necesidad de una aplicación específica para celulares, dado que casi el 50% de los alumnos encuestados alguna vez ingresaron al sistema desde un celular, y siguiendo con la pregunta siguiente, el 100% de los encuestados consideraron conveniente una aplicación desarrollada específicamente para estos dispositivos.

Una vez hechas las preguntas generales, se pasó a hacer las preguntas de análisis de uso. Como punto de entrada se analizó la correcta aparición de la aplicación instalada en el menú de aplicaciones del dispositivo y la correspondencia del icono al contenido de la misma, donde el 85% de los encuestados consideró acorde el icono a la aplicación instalada, el cual es un dato no menor dado que éste identifica la aplicación entre el resto y ayuda al fácil reconocimiento de la misma al momento de intentar acceder.

Una vez accedida, el 95% de los encuestados pudo iniciar sesión en el sistema, donde el restante se vio limitado por la versión del sistema operativo utilizada en su dispositivo. Este inconveniente se verá reflejado también en las próximas preguntas donde, por ejemplo, el 95% de los usuarios pudo ver correctamente la historia académica, dado que el restante 5% no pudo acceder al sistema. Una vez dentro de la historia académica, se les solicitó que investiguen las funcionalidades y sugieran cambios o mejoras a la sección, resultando en la siguiente lista:

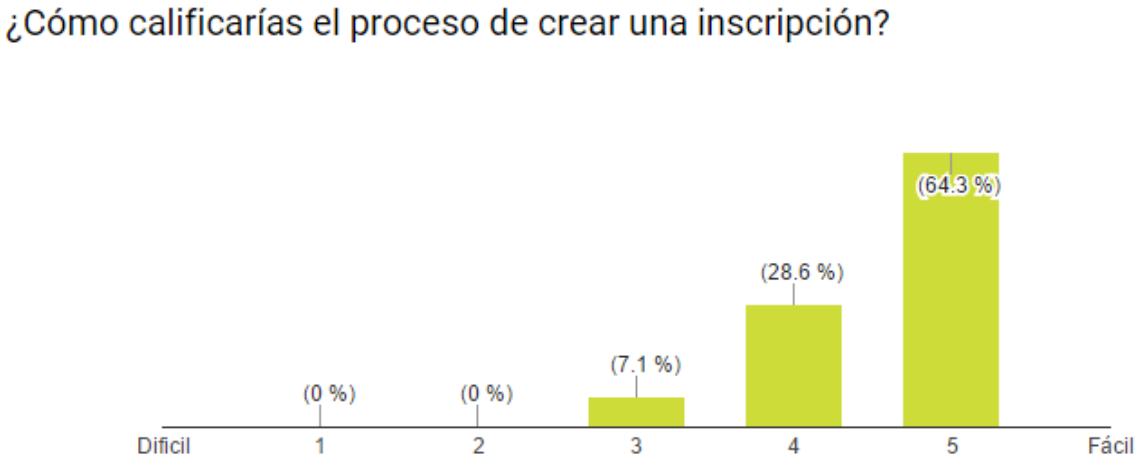
- Incluir el promedio con/sin aplazos
- Compatibilidad con celulares más viejos
- Mantener al usuario logueado luego de cerrar la aplicación

Dentro de la historia académica, el usuario puede acceder a ver la información de cada una de las materias listadas en la misma, donde el 80% de los encuestados pudo acceder a ella, y para hacer más

evidente que cada item tiene información, sugirieron mejorar las tarjetas donde se presenta cada materia. Una vez continuada la encuesta, las personas que no pudieron acceder a la aplicación y no pudieron ver su historia académica, eran descartadas automáticamente de la encuesta llevándolos al final de la misma, el resto de los encuestados siguieron completandola.

Otra de las funcionalidades claves de la aplicación y del SIU Guaraní es la creación de inscripciones, donde el 85% de los encuestados pudo crear una inscripción:

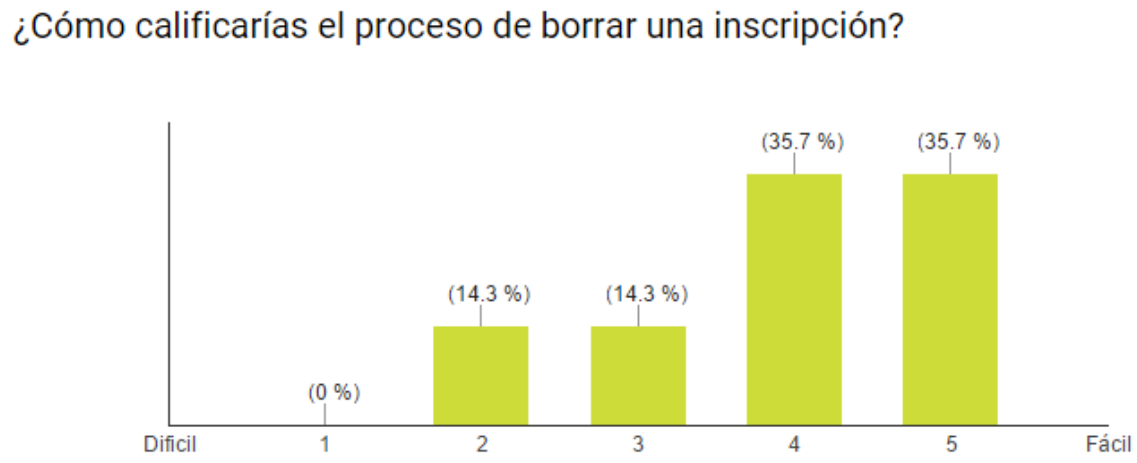
Del 85%, todos pudieron crear inscripciones a exámenes y 40% de la totalidad de encuestados pudo crear inscripciones a cursadas, resultando en la siguiente calificación del proceso de inscripción:



Al igual que para la pantalla de historia académica, a los encuestados se les pidió listar las posibles mejoras para la pantalla y proceso de inscripción, resultando en la siguiente lista de sugerencias:

- No aparecen los botones para confirmar la inscripción
- Quezas poner texto a los botones, o alguna referencia de qué significa cada icono (de examen y cursada)
- No debería listar las materias que no podés inscribirte por correlativas
- No se entiende cuales son las materias a las cuales me puedo anotar y a cuáles no por no tener las correlativas
- Búsqueda por nombre/titular

De los alumnos que lograron crear su inscripción, el 92% logró verlas sin dificultades, y el 71% pudo borrarlas, donde los restantes indicaron que por probarlo en un dispositivo con características inferiores a los requeridos no tuvieron la mejor experiencia para realizar las acciones. Gracias a estos números, la calificación del proceso de borrar inscripción fue la siguiente:



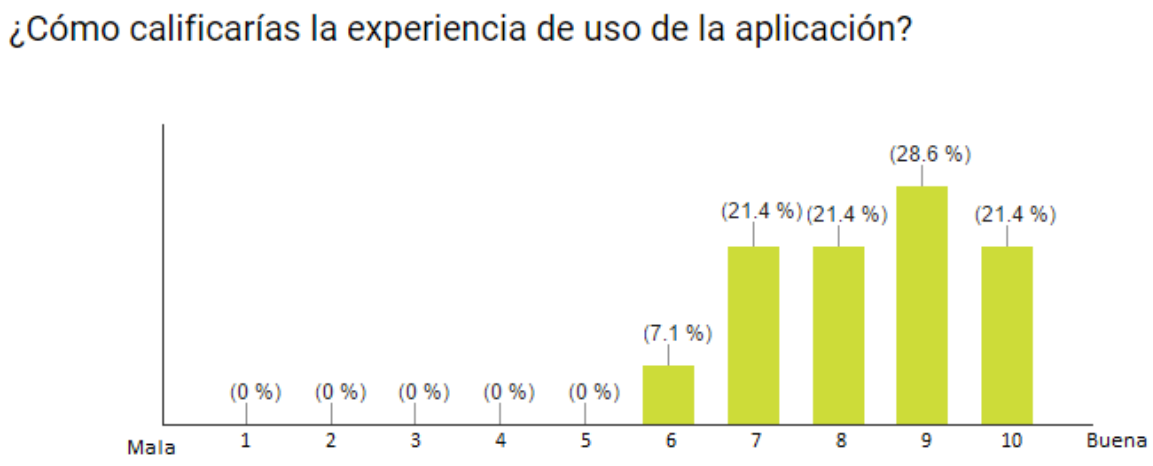
En la siguiente sección se evaluó la funcionalidad orientada a lo social, que es la ubicación de compañeros en el mapa. Esta sección fue una de las más complicadas de probar, ya que requería que los alumnos coordinen con algún compañero para verse en el mapa o avisen al encuestador para poder conectarse y aparecer junto a ellos. De todos los que probaron esta funcionalidad, el 92% pudo ver el mapa, y el 85% fue ubicado correctamente en el. Luego de consultar la razón se evidenció que el inconveniente fue el no tener la ubicación precisa de dispositivo activada, por lo que nunca pudieron ver su ubicación real marcada en el mapa.

Todos los que pudieron ver el mapa lograron también actualizar su estado, pero solo el 50% pudo coordinar con alguien para revisar la funcionalidad de ver sus compañeros en el mapa. Esta pregunta se separó para ver justamente cuantos de los que pudieron coordinar luego serían capaces de ver a la otra persona en el mapa. Luego en la siguiente pregunta se pudo ver que del 50% que pudo coordinar con alguien, el 100% logró verlos en el mapa.

Tal como se hizo en las otras secciones, ésta terminó con listado de sugerencias de mejoras:

- Aprovechar un poco más de espacio para el mapa.
- Cliquear en un compañero de la lista y mostrarlo centrado en el mapa.
- No se muestra correctamente el tema de los estados, veo solo círculos, quizás sea la versión de android.

Finalmente se pidió calificar el uso general de la aplicación, obteniendo muy buenos resultados:


¿Cómo calificarías la experiencia de uso de la aplicación?
(0 %)
(0 %)
(0 %)
(0 %)
(0 %)
(7.1 %)
(21.4 %)
(21.4 %)
(28.6 %)
(21.4 %)
Mala
1
2
3
4
5
6
7
8
9
10
Buena

Conclusión sobre la encuesta

La encuesta de usabilidad permitió medir el grado de satisfacción de los usuarios con respecto al uso de la herramienta, así como validar los supuestos iniciales respecto a la funcionalidad desarrollada y aspectos relacionados con la plataforma subyacente. Evidenció también requerimientos que no estuvieron contemplados inicialmente y que formarán parte de desarrollos futuros, principalmente la versión de Android requerida para su correcto uso. Permitió ver que el trabajo que se hizo es una necesidad (100% de los entrevistados lo considera útil) y la gente pudo usarlo con facilidad (más del 70% calificó la experiencia de uso con un puntaje mayor o igual a 8), así como también pudo fácilmente actualizar su estado y localizar algún compañero en el mapa. Éste último aspecto fue uno de los principales objetivos para enriquecer y otorgar originalidad al trabajo realizado

Conclusión

Las aplicaciones Web adaptadas para los celulares son una realidad. Organizaciones, instituciones y empresas de la más variada índole desarrollan y llevan cada día más funcionalidad hacia las plataformas móviles, dado que es un mercado en permanente expansión y la tendencia es que las personas resuelvan actividades de la vida cotidiana a través de sus teléfonos. Desde su inicio, este trabajo estuvo enfocado a aprender tecnologías orientadas al uso de celulares y cómo esta tecnología podría colaborar con el quehacer diario de los estudiantes, facilitando las tareas administrativas que inicialmente ya llevaban a cabo en forma presencial y luego a través del SIU Guaraní, que requiere de una PC conectada a Internet. SIU Guaraní es un servicio informático de gestión académica que se utiliza en las universidades nacionales de todo el país. En particular en la UNLP se encuentra instalado en todas las unidades académicas, comenzando en el año 2002.

El sistema operativo para dispositivos móviles Android es actualmente quién mayor presencia tiene, particularmente en Argentina. Es software libre y su comunidad de desarrolladores es muy activa, aunque al comienzo del desarrollo de esta tesina la documentación y los ejemplos no eran muy abundantes y la mayoría se encontraba en inglés. Dado que es una tecnología relativamente nueva y hay muchas opciones para realizar un trabajo para este tipo de dispositivos, uno de los desafíos más grandes fue el de la elección del tipo de aplicación. Dado que en la Facultad estudié programación orientada a objetos en profundidad en distintas materias y luego también seguí perfeccionandome en el trabajo, la primera elección y la más obvia era la programación nativa, pensando que la curva de aprendizaje iba a ser más corta al ser Java el lenguaje elegido por Google para programar aplicaciones Android. Dada la arquitectura con la que se hacen las aplicaciones, resultó que lo más difícil no era programar, sino el hecho de plasmar una idea en una plataforma completamente desconocida.

El trabajo de investigación realizado resultó un desafío. Requirió profundizar y mejorar actividades que en la Facultad se realizan como trabajos finales de distintas materias, aprender cómo buscar y organizar el conocimiento adquirido a lo largo de la carrera y de todo este último proceso, que es algo muy personal, generando mi propio entorno personal de aprendizaje. La herramienta Trello [22] me facilitó la gestión de tareas, donde se volcaron todos los requerimientos iniciales y los que fueron surgiendo a medida que el desarrollo iba evolucionando. Al hacer esto se pudo tener un progreso más ordenado del trabajo y mantener un historial de las cosas que se fueron haciendo.

Inicialmente para empezar a hacer cosas con Android nativo fue necesario mucho estudio, ya que la documentación permitía generar código con algo que se podía ver y probar.

Dado que la aplicación del celular no estaba funcionando aún, y para no perder la motivación, se siguió haciendo la aplicación que resultaría de backend a la aplicación móvil. En ésta área fue en la que lo aprendido en la Facultad y la experiencia laboral pudo facilitar el desarrollo, y en poco tiempo se obtuvo un servicio completo y restaría únicamente arreglar bugs que luego surgirían al momento de probarla efectivamente con la aplicación. Una vez completado esto, y para mantener el código en algún lugar en el cual se pueda consultar y descargar, se subió todo a un repositorio de código llamado BitBucket [21], donde todo el código fuente se sube a través de Git.

Al tener subido el código fuente a un repositorio y haber hecho la aplicación de backend con tecnologías orientadas a la portabilidad del desarrollo (como puede ser Maven), el tiempo de tener un entorno de desarrollo funcional se redujo mucho, pudiendo así descargar en código en cualquier computadora y poder estar trabajando en la aplicación en muy poco tiempo. Esto permitió que se

pudiera continuar con el desarrollo desde varias computadoras. Una vez completado el desarrollo de la aplicación exclusivamente Java para el manejo del backend, se continuó con la aplicación que consumiría estos servicios para presentarlos en un celular. Dado que las pruebas de la aplicación backend fueron únicamente mediante clientes REST, al conectar ambas aplicaciones comenzaron a surgir nuevos requerimientos y errores que no fueron contemplados.

La aplicación para el celular, como fue comentado, iba a ser completada con un enfoque híbrido, pero aun así quedaban varias variables por definir. Para ello fue necesario mucho estudio en el área de desarrollo en tecnologías frontend para aplicaciones web.

La integración con otras aplicaciones a través de APIs REST de comunicación requirió también de análisis y estudio específico. El resultado es la especificación completa de las operaciones que tendrá que publicar la API de SIU Guaraní para poder brindar este servicio, que es extensible para toda la comunidad universitaria.

Otro punto observado durante el desarrollo de la aplicación para el celular fue la performance. Actualmente los dispositivos móviles son muy potentes, capaces de ejecutar tareas a la par de cualquier computadora de escritorio o portátil, pero aun así quedan ciertos puntos a mejorar. Las aplicaciones híbridas son básicamente una página web renderizada en un contenedor de una aplicación nativa, pero no dejan de ser un navegador web reducido. Hoy en día se puede observar que aún los dispositivos más potentes tienen problemas al mostrar ciertos aspectos de una página web. Uno de estos aspectos son las animaciones. Una animación JavaScript requiere cierto procesamiento que una computadora puede realizar sin problemas, pero al visualizarlo en un celular puede que ésta parezca lenta. Hoy en día las aplicaciones nativas manejan estas animaciones de una manera extremadamente fluida, pero aún queda trabajo por hacer sobre las nativas. Ésta es la razón por la cual la mayoría de las aplicaciones híbridas no se lucen por sus transiciones. La aplicación desarrollada utiliza un framework que intenta simular las animaciones del sistema Android, pero a veces pueden resultar poco fluidas.

Una vez completados ciertos requerimientos, se comenzó con las pruebas sobre dispositivos. Los testeos de usabilidad realizados con estudiantes y recién egresados de distintas disciplinas, de la UNLP y otras universidades del país, evidenció la necesidad de brindar este servicio en producción (el 100% considera útil el desarrollo de SIU Guaraní para celulares), realizando los ajustes necesarios para que pueda ser utilizado por la mayoría de los usuarios (más del 70% calificó la experiencia de uso con puntaje mayor e igual a 8). La versión de Android y los requisitos de instalación dificultaron los testeos, agregando también que el buscar compañeros en el mapa es remoto y asincrónico, requiriendo coordinar y hacer ajustes en la aplicación para garantizar que el test sea más efectivo.

Trabajo a futuro

Actualmente el sistema no está conectado a SIU Guaraní, por lo que los datos están creados con el único propósito de realizar pruebas. Desde su inicio, la aplicación fue pensada para poder operar sobre el sistema en producción de SIU Guaraní, por lo que el paso más próximo es conectar ambas aplicaciones a través de una API expuesta por SIU Guaraní y consumida por la aplicación de backend desarrollada. Esto permitiría no sólo hacer más pruebas sino también empezar distribuir la aplicación como algo utilizable.

A modo de investigación se está realizando el mismo trabajo hecho para la aplicación híbrida, pero con tecnologías nativas, con objetivo de hacer comparaciones en performance, usabilidad y seguir investigando tecnologías. Dependiendo de las pruebas de concepto generadas por ambos tipos de aplicación, se optará cual será la versión utilizada para la primera ampliación de funcionalidades.

Una de las funcionalidades pedidas fue el módulo para profesores, el cual una vez completados la recolección de requerimientos, se podría agregar a la misma aplicación ya funcionando. Algo no menor es también mejorar las sugerencias a las funcionalidades ya existentes. Una funcionalidad que fue mencionada es no mostrar materias a las cuales uno no puede inscribirse.

Una vez completados los requerimientos obtenidos de las pruebas de concepto y los relacionados a generar una aplicación que sea de uso cotidiano, quedan ciertos requerimientos los cuales fueron excluidos de este desarrollo para no extender el trabajo innecesariamente. Las funcionalidades restantes impulsarán la parte social de la aplicación, permitiendo compartir a compañeros notas de cursadas, apuntes de estudio o incluso recomendar o calificar una cátedra, un profesor o un ayudante, e incluso compartir estas opiniones, notas o sugerencias con el mismo profesor, y así ayudar a mejorar las cátedras y su contenido.

Referencias y Bibliografía

[01] http://www.mejoresaplicacionesandroid.org/smartphone-la-evolucion/

[02] http://www.proandroid.com/la-evolucion-de-android-version-version/

[03] https://www.guarani-informatica.unlp.edu.ar/

[04] http://appdesignbook.com/es/contenidos/las-aplicaciones/

[05] http://blog.aplicacionesmovil.com/tag/tendencias/

[06] https://es.wikipedia.org/wiki/Software_libre

[07] http://blog.aplicacionesmovil.com/aplicaciones-celular/5-razones-para-el-desarrollo-aplicaciones-moviles/

[08] http://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/

[09] https://developer.android.com/sdk/index.html

[10] https://www.openshift.com/

[11] https://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/http/HttpStatus.html

[12] http://carlosazaustre.es/blog/empezando-con-angular-js/

[13] https://angularjs.org/

[14] http://angular-ui.github.io/angular-google-maps/#!/

[15] http://gruntjs.com/

[16] http://bower.io/

[17] http://materializecss.com/

[18] http://getbootstrap.com/

[19] http://foundation.zurb.com/

[20] https://www.google.com/design/spec/material-design/introduction.html

[21] https://bitbucket.org/

[22] http://trello.com/

[23] https://developers.facebook.com/

[24] https://dev.twitter.com/

[25] https://developers.google.com/maps/

[26] http://developers.mercadolibre.com/

[27] http://play.google.com/

[28] http://appworld.blackberry.com

[29] https://www.microsoft.com/en-us/windows/apps-and-games

[30] http://www.pcactual.com/articulo/laboratorio/comparativas/13412/ponemos_nota_los_sistemas_operativos_moviles_mas_populares.html