

Análisis, diseño e implementación de una aplicación móvil de ortografía en ITESA

HERNÁNDEZ-DOMÍNGUEZ, Carmín†*, RODRÍGUEZ-FLORES, Jazmín, MARTINEZ-MIRELES, Josue, GARCÍA-MÁRQUEZ, Marco

Recibido: 9 de Agosto, 2017; Aceptado 23 de Noviembre, 2017

Resumen

Objetivos, metodología: Desarrollar una aplicación móvil que fortalezca la ortografía en alumnos del ITESA Para desarrollar la Aplicación se utilizó una metodología clásica de desarrollo de software

Contribución: En el presente documento se describe el análisis, diseño, codificación e implementación de una aplicación que fortalezca la ortografía de los alumnos de nivel superior mediante la ingeniería de software y los frameworks utilizados así como también la navegación y diseño de las pantallas. Por último se detalla el proceso implementación de la aplicación móvil y los alcances de la misma.

Aplicación Móvil, Ortografía, Nivel Superior

Abstract

Objectives, methodology: Develop a mobile application that strengthens spelling in students of ITESA To develop the application was used a classic methodology of software development

Contribution: In this paper the analysis, design, coding and implementation of a program to strengthen the spelling of students in higher level by software engineering and frameworks used as well as navigation and design screens described. Finally the implementation process of the mobile application and the scope of it is detailed.

Mobile Application, Spelling, College

Citación: HERNÁNDEZ-DOMÍNGUEZ, Carmín†*, RODRÍGUEZ-FLORES, Jazmín, MARTINEZ-MIRELES, Josue, GARCÍA-MÁRQUEZ, Marco. Análisis, diseño e implementación de una aplicación móvil de ortografía en ITESA. Revista de Simulación Computacional 2017. 1-2: 8-16

† Investigador contribuyendo como primer autor.

* Correspondencia al autor (email: chernandez@itesa.edu.mx)

Introducción

La ortografía tiene una importancia medular dentro de un idioma, debido a que posee el carácter de normalizador y fijador de la lengua además es un instrumento esencial de cohesión y unidad del idioma Español. Además, permite establecer una secuencia histórica en la formación de un idioma. En los planos social y personal, el dominio de las convenciones ortográficas facilita la comunicación eficaz de los mensajes al eliminar ambigüedades semánticas, léxicas y sintácticas, una correcta ortografía mejora la comprensión entre lectores y escritores.

Las malas prácticas de ortografía, sobre todo por parte de alumnos de nivel superior se asocian directamente a un deficiente uso del léxico, poca lectura y desatención en el proceso de escritura; hechos que están relacionados con la capacidad de autorregulación que tienen los estudiantes de sus propios procesos mentales.

Lo anterior fue revelado por el Instituto Nacional para la Evaluación (INNE) gracias a un estudio realizado por Backhoff (2003) denominado “La ortografía de los estudiantes”, en el que se conocieron los errores más frecuentes que tienen los mexicanos que cursan este nivel de estudios.

Entre los errores recurrentes se pueden encontrar la omisión, adicción o sustitución de letras además de segmentación de palabras y problemas para entender y emplear la acentuación.

Debido a lo anterior, el dominio de la ortografía es una habilidad muy apreciada en la comunicación; un bajo nivel ortográfico implica un bajo nivel de formación y cultura de un individuo y como tal, es un indicador que va asociado a la valoración de su vida académica, profesional y laboral de las personas.

Se propone entonces el desarrollo de una app dirigida a estudiantes de nivel superior, que les permita fortalecer la práctica diaria de semántica, léxica y sintáctica. Mediante un juego por medio de la superación de niveles con base en palabras, consonantes y puntuación la cual muestre un entorno amigable y atractivo para el usuario, además de que la codificación y diseño se realice en un OSS lo cual permita que la app sea gratuita a la hora de descargar.

El desarrollo se basará en un entorno multiplataforma que permita la carga de esta app en el sistema operativo móvil Android en versiones 2.1 hasta 4.1 para su correcto funcionamiento. A continuación se mencionan las características específicas de la app.

Aplicación Móvil

Software desarrollado para correr bajo Smartphones. Está diseñada para educar, entretener o ayudar en la vida diaria a sus usuarios/consumidores. Estas funcionalidades que antes sólo estaban disponibles en PC ahora pueden ser ejecutadas en los dispositivos móviles (Avilés, 2011).

La definición planteada por CFC(2012) de una aplicación móvil es un programa que se puede descargar y acceder directamente desde su teléfono o desde algún otro aparato móvil – como por ejemplo una tablet o un reproductor MP3.

Tipos de aplicaciones móviles

- Web. Definida por (ISPAMAT, 2007) como la que necesitan de un navegador web o browser como IExplorer Mobile, Mínimo y Opera para ejecutarse. Aplicación y datos pueden residir remotamente en un servidor u obtenerse del mismo dispositivo móvil.

En cuanto a desarrollo las Web son más sencillas de programar, permite que las actualizaciones sean transparentes al usuario y el desarrollador tiene todo el control de la misma al residir en servidor.

- Nativas. (ISPAMAT, 2007) Aplicación nativa es aquella que se instala en el propio dispositivo como cualquier otra aplicación y se desarrolla mediante la utilización de un lenguaje de programación compatible con el sistema operativo del dispositivo o de un framework de desarrollo. En cuanto a desarrollo las aplicaciones nativas requieren un mayor esfuerzo de desarrollo, tanto en horas como en especialización del equipo.

El dispositivo y los lenguajes utilizados son más limitados y complejos que el entorno servidor o desktop. Siempre que sea posible un desarrollo por terceros, ya que algunos sistemas operativos móviles no lo permiten.

Es por esto que se analiza como el aprendizaje móvil podría ayudar a eliminar esa barrera y cerrar la brecha que separa el aprendizaje formal del informal. En Singapur, por ejemplo, se ha estudiado el empleo de las tecnologías móviles para facilitar el aprendizaje de los alumnos de primaria en diferentes contextos y ubicaciones.

El Proyecto Leveraging Mobile Technology for Sustainable Seamless Learning in Singapore Schools (SEAMLESS) fue un estudio longitudinal, efectuado a lo largo de tres años, pionero en el empleo de dispositivos móviles como ‘ejes de aprendizaje’ para integrar los instrumentos del aprendizaje personal y ofrecer un único lugar en el que guardar el historial de aprendizaje y los recursos de cada alumno (Looi et al., 2010; Learning Sciences Lab, 2010).

Tipos de sistemas operativos para desarrollar app móviles

- IOS. Anteriormente denominado iPhone OS es un sistema operativo móvil de Apple desarrollado originalmente para el iPhone, y que permite instalar aplicaciones de forma muy sencilla a través de la App Store.
- Android es un sistema operativo basado en Linux y orientado a dispositivos móviles, como teléfono inteligente y tablets. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en el 2005. Android tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos.
- Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Los programas están escritos en el lenguaje de programación Java.
- BlackBerry OS. Es un sistema operativo móvil desarrollado por Research in Motion para sus dispositivos BlackBerry. El sistema permite multitarea y tiene soporte para diferentes métodos de entrada adoptados por RIM para su uso en computadoras de mano. Este sistema operativo ha sido pionero en la gestión del correo electrónico y está muy orientado a un uso profesional y empresarial. Cuenta con una tienda de aplicaciones para el dispositivo llamada App World.
- Windows Phone. Es un sistema operativo móvil compacto desarrollado por Microsoft, y diseñado para su uso en teléfonos inteligentes y otros dispositivos móviles.

La mayor ventaja de este sistema de Microsoft: es la compatibilidad, facilidad de uso, integración con los sistemas Windows.

Análisis de la APP

Entorno de desarrollo y compilación

El desarrollo de la aplicación se llevó a cabo mediante *Eclipse* como entorno de desarrollo con los plugins necesarios para el desarrollo de aplicaciones Android.

Control de versiones de código fuente

Se usará un sistema de control de versiones para el código fuente de la aplicación de tipo GIT, el elegido es Bitbucket de Atlassian, ya que no obliga a hacer público el repositorio.

Persistencia de datos

Los datos de la aplicación deben almacenarse en un sistema, para ello se hará uso de un backend en el que se almacenarán los objetos necesarios.

Lenguaje de programación

Para desarrollar de forma nativa para Android se utilizará el lenguaje de programación Java, requisito imprescindible para tal fin.

Metodología de desarrollo (DCU)

La metodología a emplear para el desarrollo del proyecto es el diseño centrado en el usuario (DCU). El desarrollo de aplicaciones para dispositivos móviles supone un reto para cualquier programador acostumbrado a la programación de aplicaciones web o de escritorio (Serrano Galiana, 2010). En particular este proyecto usará una variación del tipo DCU ágil. Esta metodología combina el proceso típico del Diseño Centrado en el Usuario y Agile Software Development.

Cabe mencionar que la obtención de requisitos se obtuvo mediante la utilización de la planilla IEEE 830 Requerimientos de Software y de varias pruebas realizadas a los usuarios directos, lo cual arrojó la información que a continuación se muestra:

Historia del documento

En la tabla 1 se muestran los aspectos generales que definen la ingeniería de software para la toma de requerimientos, basado en el plantilla IEEE 830.

Fecha	Versión	Comentarios
15 Enero 2016	0.1	Versión inicial
05 Junio 2016	1.0	Revisada por el equipo

Tabla 1 Aspectos generales de la toma de requerimientos.
Fuente: Elaboración propia

Requerimientos software

Esta sección entrega un resumen de todos los requerimientos de alto nivel del software del sistema. Los requerimientos de testing, de interfaz y otros son igualmente definidos.

Requerimientos funcionales

- FSR1 [La app mostrará una presentación]
- FSR2 [La app mostrará una introducción]
- FSR3 [La app mostrará mediante niveles el avance de la práctica]
- FSR4 [La app mostrará ejemplos que muestren la realización de la práctica]
- FSR5 [En cada práctica además se debe mostrar como se debe realizar al seleccionarla.]

Requerimientos de testing

Los requerimientos de testing para cada uno de los requerimientos funcionales definidos en la sección anterior permiten validar claramente los requerimientos del software. Al igual que los requerimientos funcionales, los requerimientos de testing tienen un identificador único (Áviles E., 2011).

- ST1 [Realzar el primer clickeo que muestre la presentación]
- ST2 [Realizar el primer clickeo que muestre la introducción]
- ST3 [Contestar al menos 4 prácticas las cuales permitan indicar el nivel avanzado]
- ST4 [Elección de cada práctica en donde se muestre una explicación de la realización de esta]

Matriz requerimientos funcionales vs Requerimientos de testing

En la tabla 2 se muestra la especificación en la matriz que mapea los requerimientos funcionales con los requerimientos de testing.

Requerimiento funcional	Requerimientos de test			
	ST1	ST2	ST3	ST4
FSR1	X	X		X
FSR2		X		X
FSR3	X	X		X
FSR4	X	X		X
FSR5		X	X	

Tabla 2 Matriz de requerimientos funcionales vs testing.
Fuente: Elaboración propia

Requerimientos de Calidad

En esta sección se identifican todos los requerimientos de calidad que han sido especificados por el cliente. Para cada requerimiento de calidad se debe especificar lo siguiente:

ISSN 2523-6865

ECORFAN® Todos los derechos reservados

- Escala** Dimensión de la medición
- Prueba** Como se realizará la medición
- Peor Caso** El peor valor aceptable (bajo este valor se considera falla)
- Plan** Valor planificado
- Autoridad** Quien valida el requerimiento

Diseño de la APP

Arquitectura cliente

En la figura 1 se muestra la arquitectura de la app, vista desde tres capas.

La aplicación se puede dividir en diferentes capas:

1. La **capa de presentación** formada por los recursos textuales y la interfaz gráfica.
2. Los **permisos** que permiten a la aplicación acceder a diferentes características del dispositivo.
3. La **lógica de aplicación** formada por las diferentes clases que dan forma a la aplicación.



Figura 1 Arquitectura cliente

Esta separación de capas es la habitual en las aplicaciones para Android.

Codificación de la APP

Durante esta fase se ha realizado la codificación de la aplicación mediante el seguimiento del método de desarrollo ágil, con base en la creación de múltiples iteraciones del producto hasta llegar al resultado final.

HERNÁNDEZ-DOMÍNGUEZ, Carmín, RODRÍGUEZ-FLORES, Jazmín, MARTINEZ-MIRELES, Josue, GARCÍA-MÁRQUEZ, Marco. Análisis, diseño e implementación de una aplicación móvil de ortografía en ITESA. Revista de Simulación Computacional 2017

Las diferentes iteraciones se han almacenado en un gestor de versiones GIT. El código se ha desarrollado con la ayuda del IDE Eclipse y ADT (Android development tools).

En la figura 4 se muestra la presentación de la app, donde se puede observar el título de la primer versión de la app.

Dentro de las carpetas que se nombran como **drawable** encontramos las imágenes de las que hace uso la aplicación para las diferentes resoluciones de dispositivo. En la aplicación se ubicarán los pines y el icono de la aplicación, esto se puede ver en la figura 2.

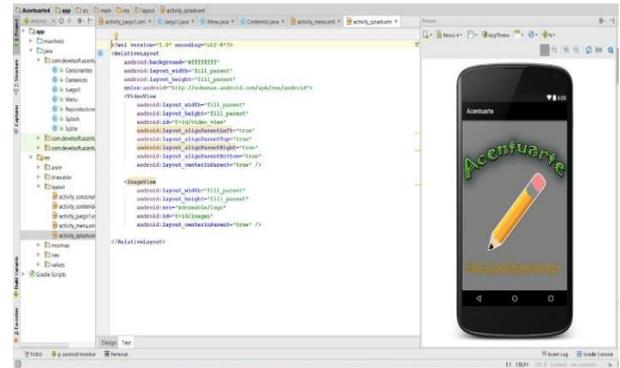


Figura 4 Pantalla con el título de la primera versión de la app

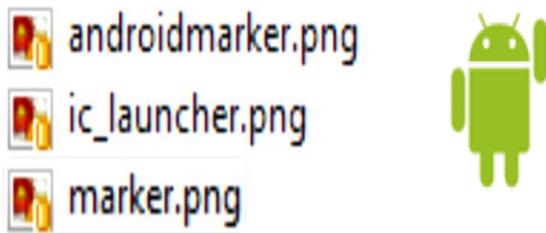


Figura 2 Pines e icono de la app

Implementación de la APP

Gestión de código

A continuación se muestran las vistas de la aplicación y su descripción, cabe mencionar que el desarrollo de estas vistas aportarán un panorama amplio para una posterior versión.

A nivel local se ha usado el plugin IDE Eclipse EGIT. En la Figura 3 se muestra la codificación realizada mediante un simulador para mostrar la primer implementación de imágenes y el botón que da inicio al juego. En la figura 6 se muestra el código de la primer pantalla donde se puede observar el entorno gráfico del simulador del móvil.

En la figura 5 se muestra el icono que referencia a la app dentro de la vista principal del dispositivo móvil, además se observa la pantalla principal y la indicación “pulse para comenzar” cabe mencionar que mientras estos eventos ocurren se reproduce el video implementado, posterior a esto se muestra la pantalla con el menú principal en el cual se puede elegir sobre el tema a practicar, ya sea “Consonantes con dificultad”, “Acentuación” o “Puntuación”. Cabe mencionar que cada tema fue estudiado para brindar la mejor enseñanza a los estudiantes de nivel superior.

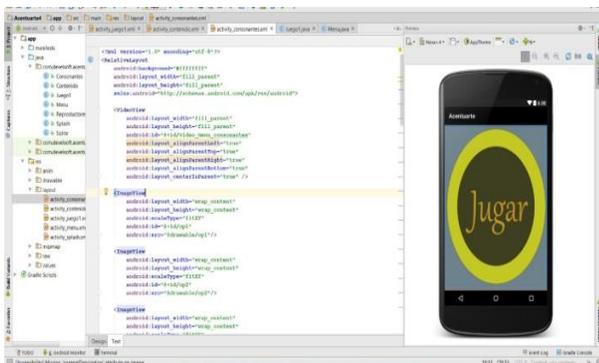


Figura 3 Primer pantalla de la app



Figura 5 Presentación de la aplicación

En la figura 6 se muestran las pantallas de la introducción a cada uno de los ejercicios, la explicación y ejemplos del uso de los temas de la app y se observan las consonantes sobre las que se puede practicar.



Figura 6 Introducción y elección

En la figura 7 se muestran los ejercicios propuestos para la práctica de ortografía además se muestra la calificación obtenida y en forma de corazones el nivel obtenido.



Figura 7 Pantallas de preguntas

Pruebas Unitarias

Las pruebas unitarias se realizan mediante el uso de Android Junit Test una extensión del conocido framework Junit adaptado a Android. El código fuente se programa de forma manual.

Las subclases principales sobre las que se suelen programar las pruebas unitarias son *ActivityInstrumentationTestCase2*, diseñada para realizar pruebas funcionales, y *ActivityUnitTestCase*, diseñada para realizar test de actividades Android de forma aislada.

Las clases creadas para realizar las pruebas se han realizado extendiendo las subclases nombradas.

- **MainActivityTestCross:** Extiende la clase *ActivityInstrumentationTestCase2*.

Métodos de prueba implementados:

- **testTouchButton()** Realiza una prueba unitaria sobre la clase *MainActivity*, crea la actividad, localiza y simula la pulsación del botón obtener posición que contiene la actividad.
- **MainActivityTestStandard:** Extiende la clase *ActivityUnitTestCase*. Realiza las pruebas sobre la clase *MainActivity*. Métodos de prueba implementados: *testPreConditions()*. Realiza una prueba para comprobar la existencia de elementos dentro de la actividad principal. *testUI*

Resultados y discusión

La implementación de esta app mostró resultados favorables, cabe mencionar que esta app no tendrá ningún costo debido a que fue desarrollada mediante el uso de un OSS (Open Source Software) y esto genera un mayor interés por el usuario, además de que está dirigida a practicar los temas fundamentales presentados en la materia de Fundamentos de Investigación de nivel superior y como caso de estudio se seleccionó una muestra del 87% de los estudiantes del del Instituto Tecnológico Superior del Oriente del Estado de Hidalgo (ITESA) lo que permitió medir el nivel de aceptación mediante la aplicación de un test antes y después del uso de la app. En la figura 8 se muestra el test de implementación de la app.

ITESA TEST DE IMPLEMENTACIÓN "ANTES DEL USO DE LA APP"

Instituto Tecnológico Superior del Oriente del Estado de Hidalgo, ITESA
TEST DE IMPLEMENTACIÓN

Fundamentos de Investigación - Primer Semestre

Nombre del alumno: _____

Matrícula: _____ Programa Educativo: Ingeniería en Logística

Docente: M.T.I.C Carmin Hernández Domínguez Duración: 20 min

M. en C. Gonzalo Herrera Muñoz Fecha: Mayo 2016

Instrucciones: Completa con la letra B o V según sea el caso

Pár___uo Proh___ir ___divula
___lgo Su___enr Su___enr___o
___enda___i ___u___ar ___onanza

Instrucciones: Completa con la letra G o J según sea el caso

Cónju___e ___emelo Su___enr
Vi___lar Indi___esto In___enuo
In___gerir ___niete ___enr___e

Instrucciones: Completa con la letra C, S o Z según sea el caso

E___aspar E___eler E___olare___er
E___p___ón E___or___ón E___enre___a
E___piéndiso E___par___ir E___peda___ón

Instrucciones: Acentúa correctamente según sea el caso

Decimosepismo Batocesto Fluido
Torticois Zodiaco Amoniaco
Cardiaco Penido Papiro

Instrucciones: Coloca los signos de puntuación adecuados para esclarecer la lectura del siguiente texto.

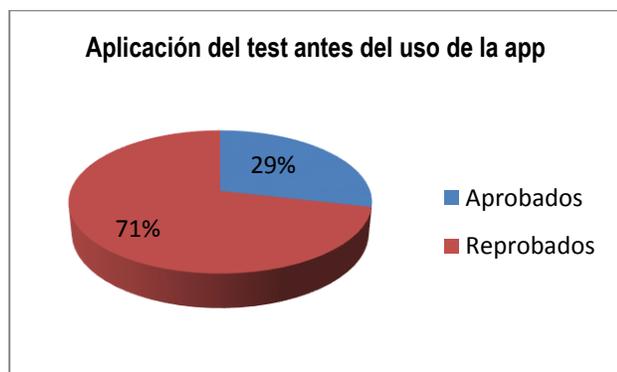
1. Aplaca sus iras mar tempestuoso
2. Cuando llegues si estás cansado acuéstate
3. Sin embargo ella que era muy coqueta siguió mirándose en el espejo
4. Así pues será conveniente adoptar una decisión
5. Ahora es de noche por tanto amanecerá

Figura 8 Test de ortografía aplicado antes y después del uso de la app

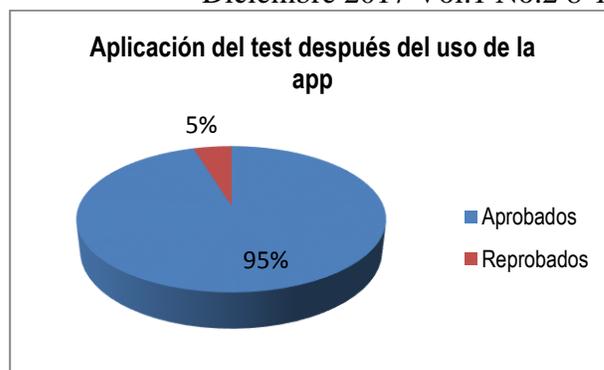
Resultados de la evaluación del test

En la gráfica 1 se muestran los resultados de la implementación del test antes del uso de la app. Donde el total de los alumnos fueron 21, de los cuales 15 reprobaron el test (71%) y únicamente 6 aprobaron (29%).

En la gráfica 2 se muestran los resultados de la implementación del test antes del uso de la app. Donde el total de los alumnos fueron 21, de los cuales 1 reprobó el test (5%) y 20 aprobaron (95%).



Gráfica 1 Aplicación del test antes del uso de la App



Gráfica 2 Aplicación del test antes del uso de la App

Entorno de ejecución

Se proporcionará un ejecutable en formato apk para poder instalar en dispositivos Android.

Discusión

La justificación de la elección de esta metodología se basa en que en uno de los entregables previos al proyecto se debía trabajar con esta metodología, por lo que resultaría contraproducente usar otro tipo de metodología, cuando se dispone de un trabajo previo. Por otra parte no se puede obviar que actualmente la experiencia de usuario y la usabilidad son imprescindibles para el éxito cualquier tipo de producto o software.

En la actualidad existen más de 700.000 aplicaciones para Android y se estima que 1.000.000 teléfonos móviles se activan diariamente.

Android es criticado por la fragmentación que sufren sus terminales al no ser soportados con actualizaciones por parte de los fabricantes, algo que se podría considerar obsolescencia programada. Esto supone problemas para los desarrolladores que deben lidiar con la retro compatibilidad entre diferentes versiones del sistema operativo.

Conclusión

Con lo antes visto se puede concluir que los dispositivos móviles están cada día más orientados a utilizarse como teléfonos inteligentes, hoy en día los usuarios de telefonía móvil requieren de aplicaciones que faciliten la vida cotidiana y de manera sincronizada en el momento que se genera la información. Tal es el caso de la aplicación presentada en este trabajo.

Referencias

- Avilés, E. (2011) Avilés, E. (Julio de 2011). *Estrategia de Desarrollo de Aplicaciones Móviles*. Recuperado el Marzo de 2012, de <http://www.slideshare.net/techmi/charla-estrategia-desarrollo-aplicaciones-mviles-universidad-girona>
- Backhoff (2003) E. Backhoff Escudero, Margarita Peón Zapata, Edgar Andrade Muñoz. (2003). La ortografía de los estudiantes de educación básica en México. 20/09/2017, de Instituto Nacional para la Evaluación de la Educación Sitio web: <http://www.inee.edu.mx/index.php/72-publicaciones/resultados-de-aprendizaje-capitulos/561-la-ortografia-de-los-estudiantes-de-educacion-basica-en-mexico>. Primera Edición. 2008. ISBN 978-968-5924-31-3
- CFC, (2012). Aplicaciones móviles: Qué son y cómo funcionan. 07/03/2012, de LA COMISIÓN FEDERAL DE COMERCIO CFC Sitio web: <https://www.consumidor.ftc.gov/articulos/s0018-aplicaciones-moviles-que-son-y-como-funcionan>
- González, F. L. (2010). González, F. L. (2010). *Aplicaciones para Dispositivos Móviles*. Universidad Politécnica de Valencia.
- ISPAMAT. (2007). Aplicación móvil: ¿Web o nativa?. 9/05/2007, de Integración de Servicios Públicos Avanzados en Municipios Altoaragoneses Sitio web: <https://ispamat.wordpress.com/2007/05/09/aplicacion-movil-¿web-o-nativa/>
- Learning Sciences Lab. (2010) Learning Sciences Lab. 2010. *Leveraging Mobile Technology for Sustainable Seamless Learning in Singapore Schools*. Singapur, Instituto Nacional de Educación, Universidad Tecnológica Nanyang. <http://isl.nie.edu.sg/projects/leveraging-mobile-technology-sustainable-seamless-learning-singapore-schools> (consultado el 5 de febrero de 2013).
- Serrano Galiana, C. (2010). Serrano Galiana, C. (2010). *Desarrollo de una Aplicación iPhone para interactuar con una vivienda domótica. Proyecto Final de Carrera. Escuela Técnica Superior de Ingeniería Informática. UPV*