

INTEGRACIÓN DE SERVICIOS EN APLICACIÓN MÓVIL PARA APRENDIZAJE UBICUO

**Mariana BRACHETTA, Julio MONETTI, Bianca SOZZI, Mercedes MUÑOZ
& Oscar LEÓN**
Ingeniería en Sistemas de Información/FR.Mendoza -UTN
{mbrachetta,julio.monetti,biancasozzi,mercedesmp1305,oleon111}@gmail.com

RESUMEN

En el trabajo se describen aspectos centrales considerados en el diseño, desarrollo e implementación de una aplicación móvil que integra geolocalización, servicios en segundo plano, tecnología de realidad aumentada (RA) y servicios de computación en la nube para lograr una experiencia concreta y medible de aprendizaje ubicuo.

Palabras clave: *Aprendizaje Ubicuo, Geolocalización, Realidad Aumentada, Mobile Cloud Computing Android, Entornos de Aprendizaje.*

CONTEXTO

Un prerrequisito al proponer una experiencia de aprendizaje ubicuo [1] es ofrecer al alumno actividades educativas situadas, es decir, disponibles en el sitio y momento donde se encuentra. Se conciben así las actividades en contexto como factor clave de todo aprendizaje [2].

El teléfono móvil integrado a los servicios en red, representa en este sentido una herramienta potencialmente adecuada para proponer a los estudiantes este tipo de actividades.

El presente trabajo se enmarca en el proyecto de investigación *PID UTN4741 (Desarrollo de un entorno basado en Cloud Computing para Aprendizaje Ubicuo)*. La integración de esta aplicación en un entorno de aprendizaje basado en *Cloud Computing*, permitirá luego a un equipo de docentes del área de Ciencias Básicas, proponer actividades de aprendizaje ubicuo y evaluar en qué medida los alumnos

adquieren conocimiento a través de los elementos que encuentran en el entorno de su vida cotidiana, es decir, llevar el aprendizaje fuera de las aulas.

Se resumen, en este trabajo los resultados alcanzados en investigación y desarrollo preliminar del proyecto, durante el cual se profundizaron los conocimientos teóricos necesarios para la implementación de la mencionada aplicación.

1. INTRODUCCION

En el presente trabajo se exponen resultados referentes a la tarea de formulación y resultados parciales obtenidos en el marco del proyecto mencionado. Se diseña y codifica una aplicación para el sistema Android [3] que hace uso de la geolocalización del alumno para presentar conceptos a través de realidad aumentada dependiendo la ubicación de este. La aplicación utiliza los servicios proporcionados por la pila de software de la arquitectura Android. Los servicios más relevantes utilizados, los cuales también se encuentran bajo estudio y análisis, son: Administrador de Ubicaciones, Administrador de Notificaciones, OpenGL|ES, Controlador de Wifi, Controlador de la Cámara, Controlador de Pantalla, Controlador de teclado. Para completar la funcionalidad deseada se utiliza:

- Para Georeferenciación: **GoogleMaps API** [5], desarrollado en el apartado 2.1, y **Servicios en Segundo Plano**, que se puede encontrar en el apartado 2.2.

- Para Realidad Aumentada: **Librería Vuforia** [6], ampliado en el apartado 2.4. y **motor gráfico Unity** [7] explicado en el apartado 2.5.

2. LINEA DE INVESTIGACIÓN Y DESARROLLO

2.1. Manejo de GPS en un entorno Android

Es requisito funcional de la aplicación poder identificar cuando un usuario (alumno), se encuentra en un determinado lugar geográfico en el que existen objetos de aprendizaje a su disposición.

La geolocalización del alumno se aborda con el uso de las librerías *Google Maps*.

Las funcionalidades de ubicación, creación de mapas, agregación de marcadores, etc., son aportadas en entornos Android por *Google Play Services*. Dado que ni los emuladores de ejecución Android disponibles, tales como *GenyMotion*, ni el entorno de desarrollo utilizado (*Android Studio*), disponen por defecto de estos servicios, los mismos deben instalarse en el SDK (Kit de desarrollo Android) de manera adicional.

El sistema Android provee dos mecanismos alternativos para la geolocalización: `FINE_LOCATION` y `COARSE_LOCATION`, para georeferenciar con distintos niveles de precisión la ubicación del dispositivo. El uso de estos mecanismos se configura en el manifiesto de la aplicación (archivo *AndroidManifest.xml*). Por defecto se utiliza el mecanismo de `FINE_LOCATION` (alta precisión con uso de GPS), puesto que se torna necesaria la ubicación precisa del alumno en un momento dado. En caso de no disponerse el servicio en tiempo de ejecución el sistema accede automáticamente a la modalidad `COARSE_LOCATION` (menor precisión con uso de datos de la conexión Wi-Fi o de la red de telefonía celular). Para acceder al GPS, se requieren permisos de usuario provistos en tiempo de instalación de la aplicación [8].

2.2. Servicios en segundo plano

Cuando el alumno ingresa en un ámbito geográfico donde existen objetos de aprendizaje accesibles (polígono de interés), es preciso notificar de tal evento, dado que es un

requerimiento de la aplicación llamar la atención del estudiante, para proponerle actividades de aprendizaje diseñadas para un sitio y contexto determinado. Para implementar el servicio de notificación se utilizan los servicios en segundo plano de Android. Esta característica, propia del sistema operativo, permite la ejecución ininterrumpida de un proceso de la aplicación que reacciona ante la ocurrencia de un evento [9].

Existen varias opciones para implementar tareas de segundo plano en Android:

- *AsyncTask*
- *Thread*
- *Handler*
- *Service*

Todas las opciones antes mencionadas están materializadas a través de clases Java provistas en el SDK de Android. Se realizaron pruebas comparando las diferentes alternativas, optándose por el uso de *Service* para dar solución a los requisitos funcionales descriptos. Un *Service* constituye un componente de la aplicación, que permite la ejecución de tareas de larga duración en segundo plano, sin la presencia de una interface gráfica activa. Con su uso se logra que el servicio esté permanentemente en funcionamiento, esperando que se produzca el evento. Existen dos tipos de *Service*:

- Servicio iniciado
- Servicio de enlace

La diferencia entre ambos radica en el tipo de vinculación que existe entre el servicio y el resto de los componentes de aplicación o incluso su conexión con aplicaciones de servidor. En función de los requisitos funcionales definidos, se optó por un servicio iniciado. En particular, el servicio permanece de forma indefinida *escuchando* los cambios de ubicación y reacciona ante el ingreso del dispositivo al polígono de interés, y notificando finalmente al alumno de tal situación.

Para permitir al servicio *escuchar* los cambios de ubicación en tiempo real se implementa la interface Java *LocationListener* [10]. Para generar el mensaje de notificación al alumno

se utilizan las clases de Android *Builder* y *NotificationManager*, junto a otras utilizadas para configurar el comportamiento de la notificación. Mediante la notificación se le presentan al alumno contenidos educativos disponibles en el contexto en que se encuentra, preparados especialmente por el equipo docente para promover actividades de aprendizaje oportunas y situadas.

Se evaluaron dos opciones alternativas para determinar cuando el usuario ingresa al polígono de interés:

Primera Opción: Definir la zona de interés como una circunferencia, ubicando un punto central mediante coordenadas y haciendo comparaciones dentro de un radio. Esta opción resulta conveniente porque es sencilla de implementar y requiere menor capacidad de almacenamiento de datos. Su desventaja radica en la menor precisión con que se determina el polígono de interés, el cual queda necesariamente inscrito dentro de la circunferencia.

Segunda Opción: Definir la zona de interés como un polígono uniendo un conjunto de puntos. La ventaja de esta opción es su mayor precisión para definir el área donde ubicar los objetos de aprendizaje (Por ejemplo: el ámbito de una habitación, el *buffete* o el campo de juego universitario). Su desventaja radica en la necesidad de guardar mayor cantidad de coordenadas y requerir mayor capacidad de cómputo para determinar el ingreso del usuario en la zona de interés.

Dado que tanto la disponibilidad de memoria como las capacidades de procesamiento pueden ser limitadas en un dispositivo móvil, se decidió que para los objetivos de la experiencia la primera opción resulta más conveniente. Así, cuando el alumno ingresa en el área de interés definida y visualiza la notificación correspondiente, la aplicación activa una segunda fase que hace uso de la tecnología de RA [4] para ofrecer al alumno actividades que aprovechan aspectos del ámbito real donde se encuentra; y lo enriquecen con contenidos especialmente

preparados para favorecer situaciones de aprendizaje significativo.

2.3. Realidad Aumentada

El concepto de RA está relacionado con cómo las personas perciben la realidad, y de qué forma la tecnología puede enriquecer dicha experiencia. La RA necesita cuatro elementos básicos:

- un **dispositivo** para capturar imágenes (cámara)
- un **elemento sobre el que proyectar las imágenes** (celular, computadora, consola de videojuegos)
- un **software de procesamiento**, capaz de interpretar la información que provee el usuario a través de su dispositivo y generar la información virtual correspondiente
- un **elemento “activador de realidad aumentada”**. Este concepto se refiere a la imagen ante la cual el sistema debe reaccionar (*Target*)

Se analizó el estado del arte con respecto a diferentes librerías utilizadas en proyectos actuales.

2.4. Librería Vuforia

Para el desarrollo de la aplicación se utilizó la librería Vuforia, que es una plataforma para la creación de aplicaciones de RA con soporte multiplataforma para los sistemas operativos *Mobile*.

La librería ofrece múltiples tipos de reconocimiento, tales como identificación de texto, imágenes, *multi-targets* y *cylinder-targets*. El presente trabajo aborda sólo lo relativo a reconocimiento de imágenes. Los *Image Targets* son imágenes planas sobre las cuales se presentan los objetos. Las imágenes que se utilizan como objetivo deben guardarse en una base de datos a la que Vuforia accede para poder comparar la imagen capturada por la cámara con la información almacenada y reaccionar según corresponda.

Las imágenes deben estar en formato JPG o PNG, en color o blanco y negro. Podría definirse que un *target* es adecuado o no a los

efectos de su reconocimiento, de acuerdo al número de *features* o puntos de reconocimiento que tenga la imagen. Por ejemplo, ver figura 1:

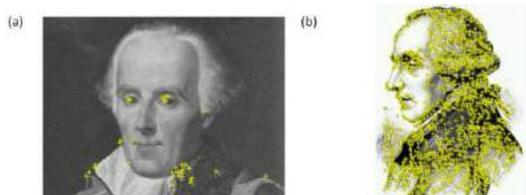


Figura 1 – Image Target. (a) Mal Target (b) Buen Target

Los *features* son los puntos en color amarillo. Como se puede observar, la primera imagen (Figura 1.a) sólo contiene unos pocos, por lo que podría ser calificada como un mal *target*, lo que quiere decir que será más difícil su reconocimiento con el dispositivo.

La librería Vuforia se complementa con el motor gráfico Unity. Este último permite generar contenido dinámico y proyectarlo sobre la imagen original.

2.5. Integración de Unity y Vuforia

Unity es un motor de videojuego multiplataforma. Permite el desarrollo de juegos para múltiples dispositivos tales como computadoras, consolas de videojuego, smartphones, smart TV, dando soporte además a los distintos sistemas operativos que cada uno de ellos puede utilizar. Unity 2017.2 provee el motor de Vuforia integrado. En caso de utilizar otra versión, el paquete puede importarse desde Unity, descargándose desde el portal de desarrollador de la página oficial de Vuforia. Una vez importado el paquete, se pueden utilizar los elementos de la librería, incluyendo *targets* predefinidos que pueden utilizarse.

Toda aplicación de Vuforia requiere una clave de licencia. Dicha licencia es gratis durante el proceso de desarrollo. Una vez que la aplicación se pone en producción es necesario actualizar la licencia a una versión paga, la cual debe registrarse en la configuración de Vuforia dentro de Unity.

Dentro de la configuración del *ImageTarget* debe establecerse la base de datos que se utilizará y qué imagen dentro de ella será el objetivo en ese caso.

El comportamiento de los objetos dentro de Unity se maneja a través de *scripts* en lenguaje C#. El ejemplo analizado hasta el momento fue la inclusión de texto sobre el objeto en el momento en que se detecta el *target*. Para ello, se debe modificar el script *Default Trackable Event Handler* y agregar el comportamiento deseado dentro de los métodos Java *OnTrackingFound()* y *OnTrackingLost()*.

Una vez realizada la configuración correspondiente para el comportamiento deseado el resultado obtenido al ejecutar la aplicación es el siguiente:

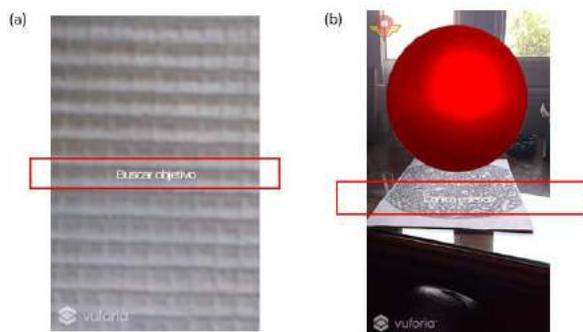


Figura 3 – Resultado de la ejecución. (a) Antes de detectar el objetivo –Texto: “Buscar Objetivo” (b) Objetivo detectado – Texto: “Cónica esférica”.

3. EMPAQUETAMIENTO Y DISTRIBUCIÓN DE LA APLICACIÓN

El desarrollo, codificado como un proyecto en el entorno Android Studio, se exporta y distribuye como un único fichero en formato ejecutable para sistema Android (*apk*).

Una vez obtenido el fichero *.apk*, el mismo debe ser firmado digitalmente para su correcta instalación y uso en dispositivos Android.

El paquete completo puede ser publicado en el repositorio *Google PlayStore* o distribuido de manera personalizada a cada uno de los usuarios. En una primera instancia del proyecto la aplicación será publicada y distribuida desde un servidor propio de la Universidad a fin de que el cuerpo docente

tenga mayor control sobre quienes acceden al mismo.

4. RESULTADOS OBTENIDOS

Se ha desarrollado un prototipo que incorpora los conceptos de geolocalización y RA para la presentación de contenido dinámicamente. De forma complementaria, el prototipo incorpora la conexión a un servidor para identificar al alumno y manejar las sesiones de usuario.

El prototipo desarrollado, permite que el equipo docente -no especializado en el área de informática- pueda visualizar el alcance y las posibilidades que el uso de estas tecnologías, adecuadamente integradas, proporciona la posibilidad de implementar experiencias concretas de aprendizaje ubicuo.

5. FORMACION DE RECURSOS

El equipo de trabajo está conformado por profesores y alumnos avanzados de la carrera *Ingeniería en Sistemas de Información* de la Universidad Tecnológica Nacional, Facultad Regional Mendoza. Los becarios se están formando en temas técnicos con la posibilidad de incorporarse a la carrera de investigador, e integrar nuevas actividades de investigación relacionadas con el presente trabajo.

6. BIBLIOGRAFIA

[1] Burbules, Nicholas. El aprendizaje ubicuo y el futuro de la enseñanza. *Encounters on Education*. Vol. 13, 2012, 3 – 14. University of Illinois, Urbana-Champaign, Estados Unidos. ISSN 1925-8992.

[2] Sagástegui, Diana. Una apuesta por la cultura: el aprendizaje situado. *Sinéctica, Revista Electrónica de Educación* N° 24. ITESO Universidad Jesuita de Guadalajara, Jalisco, México. (2004). Disponible en <http://www.sinectica.iteso.mx/index.php/SINECTICA/article/view/282>.

[3] Cruz Zapata, Belen. *Android Studio Essentials*. Packt Publishing. 2015. Birmingham, UK. ISBN 978-1-78439-720-3.

[4] Gazcón, Nicolás F, Larregui, Juan I, Castro, Silvia M. *La Realidad Aumentada*

como complemento motivacional. *Libros Aumentados y Reconstrucción 3D*. TE&ET N°17 [pág. 7-15]. RedUNCI-UNLP. 2016. ISSN 1850-9959.

[5] GoogleMaps API. Disponible en <https://developers.google.com/maps/documentation/android-api/current-place-tutorial>.

[6] Página Oficial de Vuforia. Disponible en <https://library.vuforia.com/>.

[7] Página Oficial de Unity. Disponible en <https://unity3d.com/es>.

[8] Solicitud de permisos. Disponible en <https://developer.android.com/training/permissions/requesting.html?hl=es-419>.

[9] Servicios en segundo Plano de Android. Disponible en <https://developer.android.com/guide/components/services.html?hl=es-419>.

[10] LocationListener. Disponible en <https://developer.android.com/reference/android/location/LocationListener.html>.

[11] Zapata-Ros, Miguel. Calidad en entornos ubicuos de aprendizaje. *RED. Revista de Educación a Distancia*. Número 31. Universidad de Alcalá. España.

[12] Linowes J. *Unity Virtual Reality*. Packt Publishing. Birmingham, UK. 2015. ISBN 978-1-78398-855-6.