

AMBIENTE DE APRENDIZAJE MÓVIL DE APOYO A LA EDUCACIÓN A DISTANCIA

Gisela Torres de Clunie y Clifton E. Clunie T.
Universidad Tecnológica de Panamá
Ave. Ricardo J. Alfaro s/n, Apdo. 0819-07289, Panamá

RESUMEN

El desarrollo de la computación móvil y su aplicación en la educación, hizo surgir un nuevo enfoque de educación a distancia, denominado aprendizaje móvil, el cual facilita a los estudiantes el aprendizaje en cualquier momento y desde cualquier lugar. Este trabajo presenta el diseño de un ambiente de aprendizaje móvil, que facilita a los usuarios de los cursos virtuales, de la Universidad Tecnológica de Panamá, interactuar de forma eficiente, flexible y transparente en un ambiente colaborativo y personalizable de interacción y alertas, por medio de *smartphones*, *tablets* y reproductores multimedia con sistema operativo Android.

PALABRAS CLAVE

Educación a distancia, aprendizaje móvil, computación móvil, *web services*.

1. INTRODUCCION

El desarrollo de la computación móvil ha promovido la creación de aplicaciones móviles orientadas al campo educativo, favoreciendo el surgimiento de una nueva categoría de educación a distancia denominada “aprendizaje móvil”, conocida comúnmente como m-learning. De acuerdo con (Georgiev et al., 2004), este enfoque educativo viabiliza el aprendizaje de los estudiantes desde cualquier lugar y en cualquier momento, mediante conexiones de redes inalámbricas.

Moodle es un sistema de administración de cursos que permite a los profesores la creación de escenarios para el aprendizaje en línea. De acuerdo con (Felizardo et al., 2007), las principales fortalezas de Moodle son las herramientas de comunicación, creación y administración de objetos de aprendizaje. Mientras tanto, a pesar de las ventajas que proporciona Moodle, así como otras plataformas educativas; las mismas no brindan un ambiente colaborativo de interacción y alertas para dispositivos móviles con sistema operativo Android, lo cual constituye el principal objetivo de nuestro trabajo. Android es una plataforma de software *open source*, basada en Linux, creada específicamente para dispositivos móviles.

La siguiente sección describe algunos trabajos relacionados referentes para nuestra propuesta. La sección 3 describe la arquitectura del ambiente de aprendizaje móvil, que permite la integración de servicios heterogéneos en distintos escenarios de trabajo de forma flexible y aplicando reutilización; discute los componentes de la arquitectura del ambiente y define las principales funcionalidades de la misma. La sección 4, describe la implementación de los *web services* incorporados en el ambiente de aprendizaje. La sección 5, presenta la personalización de las alertas y la sección 6 discute el modelo de localización. Para finalizar, en la sección 7, se presentan las conclusiones y trabajos futuros.

2. TRABAJOS RELACIONADOS

Diversos artículos discuten los avances de la tecnología móvil como un complemento para la educación a distancia. Esta sección presenta una breve descripción de algunos trabajos que presentan similitud con el proyecto que estamos realizando.

En (Colazzo, 2003), se presenta una adaptación de un sistema de administración de cursos para satisfacer las necesidades de usuarios móviles, utilizando tecnología de *web services* para definir la interfase de comunicación entre el CMS y sus extensiones móviles. Luego, en (Wains and Mahmood, 2008), encontramos la propuesta de un *framework* para la integración de un ambiente de aprendizaje móvil con un ambiente de *e-learning* donde, a partir de la disponibilidad de salas de chat basado en SMS, se promueve el aprendizaje de los estudiantes mediante la utilización de foros y debates para discutir sus problemas, pensamientos e ideas; se realizan llamadas de voz y transmisiones de radio y televisión. Mientras tanto, la mayor parte del contenido puede accederse a través de las conexiones a Internet de los celulares, sin ninguna adecuación de los objetos de aprendizaje.

La literatura técnica reporta algunos trabajos relativos a comunicaciones móviles y distribución de servicios. (Ibrahim and Zhao, 2009) presentan un modelo conceptual de *framework* para comunicación en dispositivos móviles que da soporte a la movilidad y distribución de servicios sobre múltiples dispositivos y plataformas. Muchos de los principios, entre ellos las bases de la comunicación cliente – servidor y de agentes móviles, son un referente para nuestro proyecto; pero a diferencia de nuestro trabajo, la propuesta de (Ibrahim and Zhao, 2009) está diseñada para Java Virtual Machines (JVM) y no considera un ambiente de interacción y alertas.

3. MODELO DE ARQUITECTURA DEL AMBIENTE

La Figura No.1 presenta la arquitectura del ambiente. El componente cliente representa la aplicación desarrollada para los dispositivos móviles con sistema operativo Android, mientras que el lado servidor constituye la infraestructura que realiza la integración con el ambiente Moodle.

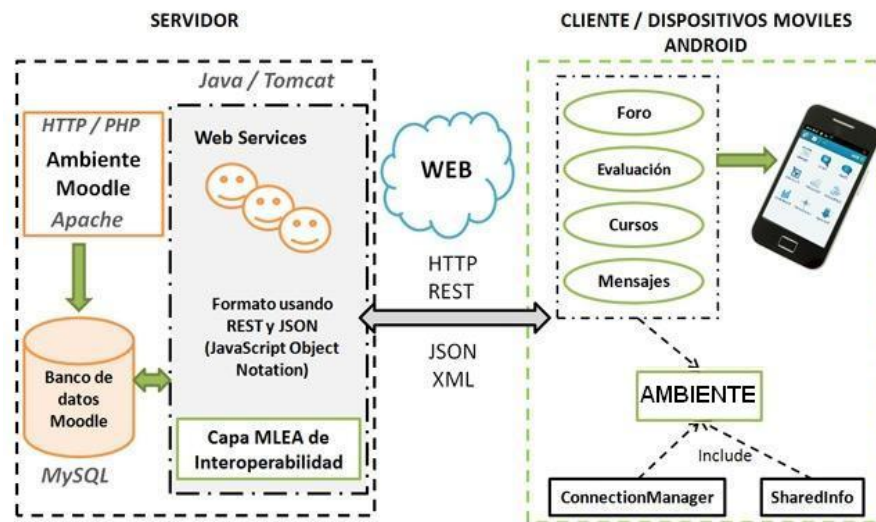


Figura 1. Arquitectura del ambiente

En el lado servidor, un conjunto de web services actúa como interfaz de comunicación entre los clientes y el servidor, resultando responsables por ofrecer a los clientes las funcionalidades integradas. Un web service es un componente de software definido por una interfaz independiente que está disponible a través de la red. Las operaciones definidas en la interfaz, realizan funciones de negocio (Hewitt, 2009). Por medio de los web services, los clientes tienen acceso a los principales recursos de Moodle, como foro, evaluación, mensajes, chat, descarga de archivos, localización, alertas, anuncios, calificación y curso, entre otros. Al recibir una solicitud, un web service accesa la base de datos de Moodle para recuperar y/o manipular los datos necesarios para responder adecuadamente a la solicitud. En el lado cliente, la implementación está basada en el patrón de desarrollo de Android: para cada pantalla de la aplicación, existe una clase Java responsable por controlar las acciones de esa pantalla. De esta forma, para cada funcionalidad provista (por ejemplo foro,

evaluación, mensajes, chat,...) existe un conjunto de pantallas y, por lo tanto, un conjunto de clases Java, representadas en la arquitectura mediante el uso de paquetes.

3.1 Componente Cliente

El componente cliente, es decir, la aplicación Android, utiliza el patrón de proyecto Facade (Gamma et al., 1995) para realizar la comunicación entre las pantallas y la clase responsable por invocar los *web services*. Cada pantalla en la aplicación Android posee una clase asociada. La clase MLEA actúa en la aplicación como una fachada para realizar la comunicación entre las clases. La Figura No. 2a, ilustra el uso de la clase MLEA, la cual es utilizada por todos los requisitos desarrollados. En la figura se puede observar que los módulos creados (por ejemplo, Curso, Foro, Mensajes) son clientes de la clase MLEA, la cual define una interfaz entre las clases ConnectionManager y SharedInfo, aislándolas del resto de la aplicación. La clase SharedInfo administra la base de datos del dispositivo móvil, donde son almacenadas las informaciones de los usuarios, tales como: datos de autenticación, identificadores de usuario conectado y de curso escogido, así como datos de caché. La figura solamente presenta algunos de los módulos desarrollados. Los módulos que conforman la arquitectura son: login, foro, evaluación, mensajes, chat, descarga de archivos, localización, alertas, anuncios, calificación, escoger curso, encuesta, evaluación del foro, estadísticas del foro y visualizar usuarios conectados.

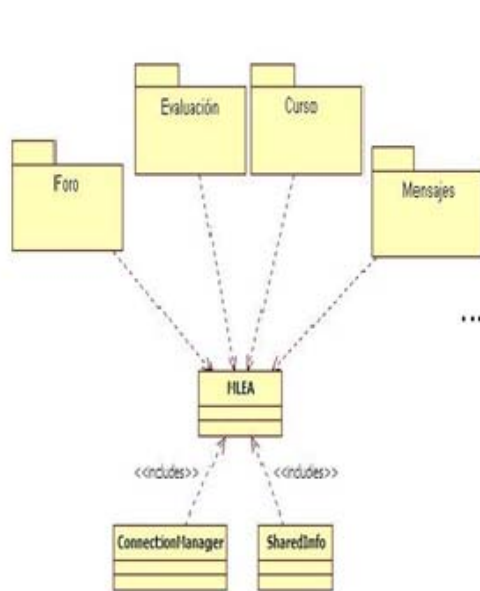


Figura 2a. Vista del componente cliente

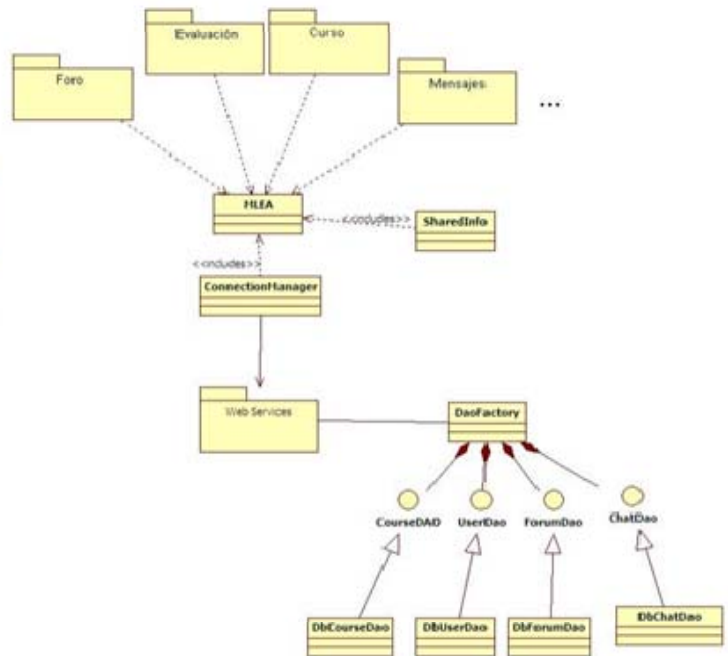


Figura 2b. Vista del componente servidor

3.2 Servidor

El componente servidor utiliza dos patrones de diseño para responder a las solicitudes de los clientes. Primeramente, los *web services* utilizan el patrón de proyecto DAO (*Data Access Object*) (Alur et al., 2003), para acceder y manipular las informaciones en la base de datos de Moodle. Para cada tipo de dato que será utilizado, existe una interfase DAO que indica las operaciones que se pueden realizar con este tipo. La capa del modelo de la aplicación posee los siguientes tipos de datos, cada uno con una interfase DAO específica: login, foro, evaluación, mensajes, chat, descarga de archivos, localización, alertas, anuncios, calificación, escoger curso, encuesta, evaluación del foro, estadísticas del foro y visualizar usuarios conectados. Para aumentar la flexibilidad de la aplicación, las clases DAO no son instanciadas directamente por los *web services*; en vez de esto, es utilizada una fábrica para la construcción de las clases DAO. Esta práctica

corresponde al uso del patrón de proyecto FactoryMethod, el cual garantiza que toda la aplicación utilice las DAO apropiadas para la configuración escogida. La Figura No.2b presenta el uso de los patrones de diseño, donde el conjunto de web services utiliza la DaoFactory para crear instancias de las interfaces, las cuales son utilizadas para acceder y manipular las informaciones en la base de datos de Moodle.

4. IMPLEMENTACION DE WEBSERVICES

Para cada funcionalidad de la aplicación, existe una clase responsable por proveer el conjunto de *web services* correspondientes al requisito específico.

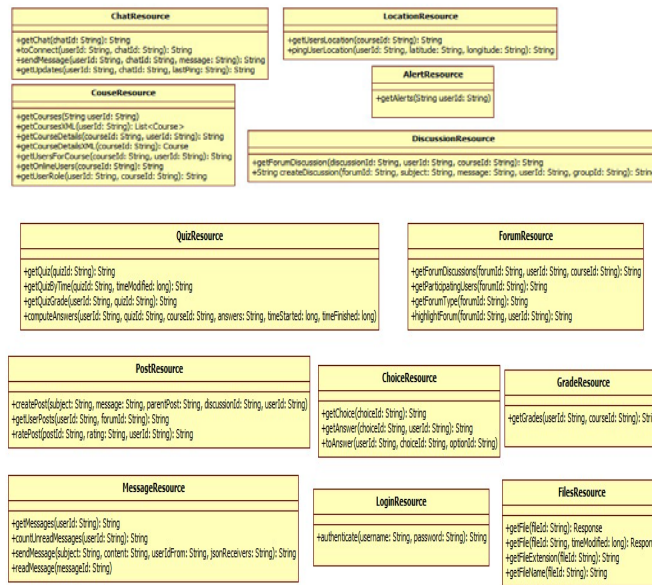


Figura .3. Especificación de los *web services*

Los *web services* fueron desarrollados utilizando el protocolo REST (*Representational State Transfer*) para *web services* (Allamaraju, 2010), pues Android no posee soporte nativo para otro tipo de protocolo. De esta forma, el desarrollo resultó más sencillo con el uso de las bibliotecas existentes en Android para la invocación de los servicios. REST emula el protocolo HTTP mediante la restricción de establecer la interfaz a un conjunto conocido de operaciones estándar como GET, POST o PUT. Además del HTTP, protocolo patrón utilizado por REST para la comunicación, se utilizó JSON (*JavaScript Object Notation*) para definir el formato de los datos, por resultar un formato liviano para el intercambio de datos computacionales, lo cual resulta en ventajas en el procesamiento de aplicaciones móviles. En la Figura No. 3, se presentan las clases implementadas, con sus respectivos *web services*. Por ejemplo, la clase *ForumResource*, contiene 4 *web services*: el primero que es listado, es el *getForumDiscussions*, este servicio recibe como parámetro los identificadores del foro, del usuario y del curso, retornando todas las discusiones que componen un foro específico.

Inicialmente, pensamos utilizar los *web services* de Moodle en sus versiones más recientes; sin embargo, identificamos que la cantidad de servicios disponibles por Moodle es muy poca y, de los mismos, apenas una pequeña parte sería aprovechable para los propósitos del proyecto. Frente a esta situación, optamos por desarrollar todos los *web services* necesarios para el funcionamiento de la aplicación.

Tabla 1. Servicios de Moodle

Web Service	Descripción
moodle_course_create_courses	Crea nuevos cursos
moodle_course_get_courses	Recupera los cursos existentes
moodle_enrol_get_enrolled_users	Recupera la lista de participantes del curso
moodle_enrol_get_users_courses	Recupera la lista de cursos en que un usuario participa
moodle_enrol_manual_enrol_users	Registra un usuario para participar en un curso
moodle_file_get_files	Navegar por los archivos de un curso
moodle_file_upload	Realizar el upload de un archivo
moodle_group_add_groupmembers	Agregar miembros a un grupo
moodle_group_create_groups	Crear nuevos cursos
moodle_group_delete_groupmembers	Excluir miembros de un grupo
moodle_group_delete_groups	Excluir determinado grupo
moodle_group_get_course_groups	Recupera todos os grupos definidos en un curso
moodle_group_get_groupmembers	Recupera los participantes de determinado grupo
moodle_group_get_groups	Recupera los detalles de determinado curso
moodle_message_send_instantmessages	Enviar mensajes
moodle_notes_create_notes	Crear notas
moodle_role_assign	Atribuir roles a un usuario
moodle_role_unassign	Remover rol de usuario
moodle_user_create_users	Crear usuarios
moodle_user_delete_users	Excluir usuarios
moodle_user_get_course_participants_by_id	Recupera el perfil de los participantes de un curso
moodle_user_get_users_by_courseid	Recupera los participantes de determinado curso
moodle_user_get_users_by_id	Recupera un conjunto de participantes
moodle_user_update_users	Actualizar los usuarios
moodle_webservice_get_siteinfo	Recupera algunas informaciones sobre o sitio

La Tabla No.1 presenta los servicios provistos por Moodle actualmente, destacando aquellos aprovechables para el desarrollo de la aplicación. Sin embargo, como los servicios provistos por Moodle no resultan suficientes para atender las necesidades del ambiente de aprendizaje móvil; todos los *web services* utilizados por dicho ambiente fueron implementados de manera independiente de Moodle.

5. ALERTAS PERSONALIZADAS

Una contribución importante del ambiente presentado, es la oferta de una herramienta que provee alertas personalizadas a los estudiantes y a los profesores, facilitando que éstos se mantengan actualizados sobre las actividades que se realizan en los cursos. Los usuarios que desean recibir las alertas tienen a su disposición una interfase Web por medio de la cual ellos pueden definir, para cada curso, de manera individual, los eventos que les interesa recibir las alertas. Para utilizar esta facilidad de la aplicación, los usuarios requieren ser autenticados. La Figura 4, presenta la pantalla de autenticación, en la cual los usuarios deben introducir las mismas informaciones utilizadas para autenticarse en el ambiente Moodle.



Figura 4. Pantalla de autenticación

La Figura 5, presenta la pantalla de configuración, por medio de la cual los usuarios especifican las alertas que desean recibir. Es importante destacar que los usuarios tienen a su disposición diferentes posibilidades de personalización, de acuerdo con su rol en el curso:



Figura 5. Pantalla de configuración de las alertas

Cuando se ejecutan los eventos configurados por los usuarios, éstos reciben las alertas directamente en sus teléfonos, por medio del sistema de notificación nativo del sistema operativo Android. Los teléfonos tendrán instalada una aplicación que solicitará información al servidor y, dependiendo de la respuesta, mostrará las alertas al usuario de acuerdo con las configuraciones pre-establecidas.

6. MODELO DE LOCALIZACION

El módulo de localización, cuyo diagrama presentamos en la Figura No.6, presenta un funcionamiento coherente con el funcionamiento de los otros módulos de la aplicación, presentados anteriormente. Inicialmente, es necesario descubrir la localización de los diversos usuarios. Para esto, la clase MainActivity,

responsable de controlar la pantalla inicial de la aplicación, inicializa la clase LocationService, la cual es responsable de coleccionar periódicamente la localización del usuario autenticado. Entonces, cada vez que el usuario reporta una nueva localización (actual), la clase LocationService notifica a la clase MLEA, la cual utiliza la clase ConnectionManager para guardar la localización del usuario. Para esto, la clase ConnectionManager invoca el webservice apropiado, definido en la clase LocationResource. A la vez, esta clase (LocationResource), utilizará una instancia de la interfase LocationDao (en este caso la clase DbLocationDao), la cual almacena en el banco de datos la localización recibida.

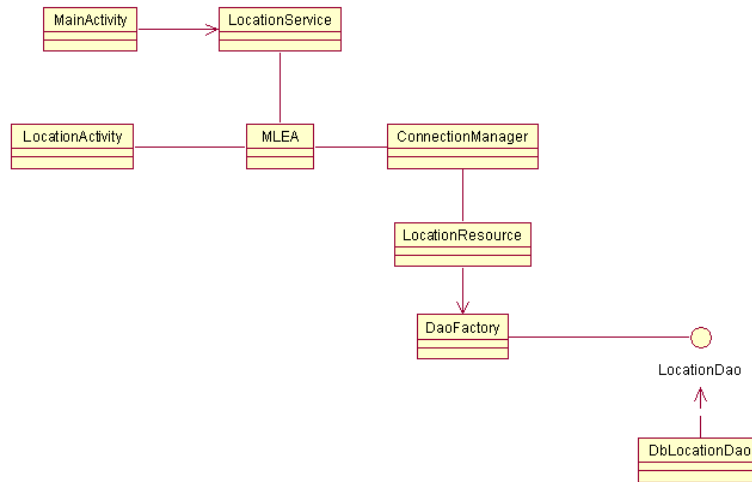


Figura 6. Modelo de Localización del ambiente móvil



Figura 7. Mapa de localización de los usuarios

Además, la clase LocationActivity es responsable por controlar la pantalla que muestra un mapa con la localización de los demás usuarios. Para esto, seguimos el mismo patrón utilizado en toda la aplicación: esta clase utiliza la clase MLEA, la cual usa la clase ConnectionManager, que a la vez invoca un web service definido en la clase LocationResource. Este web service utiliza una instancia de la interfase LocationDao, que recupera del banco de datos las localizaciones de los demás usuarios. La respuesta (es decir, la

localización de los usuarios) sigue el recorrido inverso, hasta llegar a la clase LocationActivity, que finalmente presenta al usuario el mapa de forma apropiada (ver Figura No.7).

7. CONCLUSIONES Y TRABAJOS FUTUROS

A partir de la puesta en marcha del ambiente, se recibirán los primeros beneficios del proyecto en los diversos escenarios y sedes de la Universidad Tecnológica de Panamá. La participación de todos los actores, también nos permitirá identificar situaciones y casos que orienten hacia el mejoramiento y adecuación continua de la aplicación. Un aspecto importante a destacar es que la arquitectura desarrollada hace posible que, en el futuro, se puedan desarrollar aplicaciones clientes para diferentes sistemas operativos móviles, como por ejemplo iPhone, aprovechando gran parte de la infraestructura construida y, de esta forma, se pueda alcanzar un número mayor de usuarios.

AGRADECIMIENTOS

Este trabajo es financiado por la Secretaría Nacional de Ciencia, Tecnología e Innovación - SENACYT, en el marco de la Convocatoria Pública de Fomento a la Colaboración Internacional en I+D. Los autores expresan sus agradecimientos a SENACYT y a la Universidad Tecnológica de Panamá, en Panamá, y a la Universidade do Vale do Rio dos Sinos en Porto Alegre, Brasil, por el apoyo para el desarrollo del presente proyecto.

REFERENCIAS

- Allamaraju, S.; 2010 *RESTful Web Services Cookbook: Solutions for Improving Scalability and Simplicity*. O'REILLY,
- Alur, D., et al. 2003; *Core J2EE Patterns: Best Practices and Design Strategies*. Prentice Hall,.
- Colazzo, L.; et al. 2003; Towards a multi-vendor mobile learning management system. *Proceedings of e-learn Conference*.
- Felizardo, K.; et al. 2007; O uso do Moodle no apoio de ensino de programação para alunos iniciantes. *13º Congresso Internacional de Educação a Distância*.
- Gamma, E. et al. 1995; *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional,.
- Georgiev, T.; et al. 2004; M-Learning – a new stage of e-learning. *CompSysTech '04: Proceedings of the 5th international conference on Computer systems*, pp. 1-5.
- Hewitt, E. 2009; *Java SOA Cookbook*, 1st. Edition, O'reilly.
- Ibrahim, A.; and Zhao, L. 2009; Supporting the OSGi Service Platform with Mobility and Service Distribution in Ubiquitous Home Environments. *The Computer Journal*. London. Vol. 52, Iss. 2, pp. 210-239.
- Wains, S.; and Mahmood, W., 2008; Integrating m-learning with e-learning. *SIGITE '08: Proceedings of the 9th ACM SIGITE conference on Information technology education*.