

ShimiTV, a T-learning System for Android TV

Elio Quevedo, Paola Cárdenas, Magali Mejía-Pesántez, Lisette Muñoz-Guillen

Departamento de Ciencias de la Computación

Universidad de Cuenca

Cuenca - Ecuador

{elio.quevedo,paola.cardenas0108,magali.mejia,lisette.munozg}@ucuenca.edu.ec

Resumen—Interactive Digital Television (iDTV) applications allow novel mechanisms to provide services, one of which is teaching by means of virtual courses. This article presents a client – server architecture and a the development process, based on an agile methodology, of “ShimiTV”. It is an interactive application which purpose is to teach languages by a series of questions of reading, gramatical and vocabulary comprehension. The implementation is done on the emerging operating system Android TV.

Index Terms—Televisión Digital Interactiva, Ágil-TVDi, Android TV, t-learning.

I. INTRODUCCIÓN

El televisor es un medio de difusión de contenidos que cuenta con un alto número de espectadores, según el censo ecuatoriano del 2010, el 85.1 % de la población lo usa en sus hogares [1]. En este mismo año Ecuador decidió adoptar la norma *ISDB-Tb* para la difusión de contenidos de Televisión Digital Terrestre (TDT) [2]. Las aplicaciones para Televisión Digital Interactiva (TVDi) tienen requerimientos diferentes a los del software de otras plataformas tradicionales, sin embargo, no existen suficientes trabajos que guíen el proceso de desarrollo de este tipo de sistemas [3]. En el marco del proyecto de investigación “Análisis de Plataformas de Desarrollo de Aplicaciones Interactivas para Televisión Digital Terrestre”, se creó la metodología Ágil-TVDi, la cual describe de manera técnica el proceso a seguir para el desarrollo de aplicaciones para TVDi. Este artículo presenta la experiencia de desarrollo de ShimiTV, una aplicación orientada al aprendizaje de idiomas extranjeros que fue implementada en Android TV siguiendo los lineamientos de la metodología Ágil-TVDi. El documento está estructurado como sigue: en la sección II se describen trabajos relacionados a TVDi, t-learning y Android TV. La sección III describe la aplicación ShimiTV. Después, en la sección IV se presentan los componentes de la arquitectura, con su correspondiente implementación en la sección V. En la sección VI se muestran los resultados y ejecución de la aplicación. El último capítulo presenta las conclusiones y trabajos futuros.

II. TRABAJOS RELACIONADOS

Esta sección se presentan trabajos relacionados a TVDi, t-learning, y algunas aplicaciones de Android para la TVD.

II-A. Televisión Digital Interactiva y t-learning

Según [4], la TVDi puede ser considerada como la convergencia de la televisión y las tecnologías de la computación,

con características como: interactividad, personalización de contenidos y digitalización de las señales de audio y video [5]. Por tanto, TVDi ha motivado el desarrollo de nuevas formas de proveer servicios de educación a distancia [6], sirviendo como base para el concepto de t-learning en la educación y formación continua [7]. El término t-learning es usado para referirse a un subconjunto del e-learning, asociado a un aprendizaje basado en la interactividad con la televisión. [8]. A diferencia del aprendizaje centrado en la computadora, el t-learning supone el uso de contenidos que potencian el aspecto audiovisual y agregan funcionalidades de interactividad [6]. Además, las aplicaciones de TVDi tienen requerimientos diferentes a aquellas orientadas a dispositivos tradicionales como la computadora o dispositivos móviles. Algunos estudios de TVDi focalizan su estudio en guías para el desarrollo de aplicaciones de TVDi. En [9], se explica el desarrollo de una aplicación que no interrumpe la programación televisiva, ejecutando tareas secundarias y con baja demanda de interactividad. [3] presenta un método de desarrollo que considera las fases de requerimientos y diseño, tomando en cuenta para esta última, los intereses de los televidentes. En el caso de [10], se propone la metodología ágil MADCE-TVD, que incluye en la fase de pre-producción consideraciones pedagógicas como el público objetivo, estrategia pedagógica y definición de contenidos.

II-B. Android TV

En el año 2010 iniciaron las investigaciones asociadas a la instalación y ejecución del sistema operativo Android sobre plataformas embebidas para televisores [11]. Una de estas propuestas se expone en [12], en donde se importaron las librerías de Android en un decodificador de televisión digital con Sistema Operativo Linux, se incluyeron funcionalidades para cambio de canales, almacenamiento y reproducción de videos pero con un bajo rendimiento. En [13] se propone Android4TV, una aplicación para televisores con Sistema Operativo Android, sirve como decodificador para TVDi realiza el proceso de decodificación de la señal de televisión digital y al mismo tiempo permite la ejecución de otras aplicaciones. En Junio de 2014 Google anunció el lanzamiento de Android TV como Sistema Operativo para televisores [14].

Considerando cada una de las experiencias investigadas, se puede evidenciar que Android TV a pesar de encontrarse en auge aún no cuenta con suficientes experiencias de desarrollo, es por esta razón que se ha visto la necesidad de crear

una aplicación interactiva para TVD utilizando este sistema operativo, la misma que se detalla en la siguiente sección.

III. DESCRIPCIÓN DE LA APLICACIÓN

ShimiTV (del vocablo Kichwa “Shimi” que significa “Idioma o lenguaje”) consiste en una aplicación interactiva que ofrece a los televidentes, un conjunto de lecciones que les permite aprender, practicar y reforzar sus conocimientos sobre un idioma. El modelo de enseñanza se basa en preguntas que evalúan los conocimientos del televidente en relación al vocabulario, gramática, comprensión auditiva y comprensión lectora.

III-A. Análisis y diseño

ShimiTV se planteó como un servicio “24/7”, es decir, está continuamente disponible para los televidentes. La interfaz gráfica usa toda la pantalla y es independiente de la transmisión televisiva con el objetivo de remover distracciones durante el proceso de aprendizaje de idiomas. Después de instalar la aplicación, el televidente se registra con un nombre único de usuario, correo y contraseña. Al iniciar la sesión, se muestran los cursos estudiados por el televidente, así como otros cursos disponibles. Al ingresar en algún curso, se deberá seleccionar un nivel de dificultad, el cual actúa como una agrupación lógica de lecciones. Cada lección cuenta con los siguientes tipos de preguntas: Selección múltiple que incluyen textos e imágenes, escritura y ordenamiento.

También se cuenta con evaluaciones por nivel, donde las preguntas se presentan de manera aleatoria a partir de aquellas lecciones que ya han sido estudiadas.

IV. ARQUITECTURA

Se propone una arquitectura cliente-servidor, donde el servidor está organizado en capas que administran la lógica del sistema y gestionan el acceso de múltiples clientes. La comunicación entre estos componentes requiere de un canal de retorno y autorización de acceso a los servicios HTTP mediante el protocolo OAuth. A continuación se detallan los componentes de la arquitectura:

IV-A. Servidor

El servidor gestiona el acceso a la base de datos y a los servicios expuestos mediante una interfaz REST. El servidor tiene cuatro componentes que se describen a continuación:

IV-A0a. Capa de acceso a datos: Gestiona la conexión, escritura y lectura de la información almacenada en la base de datos. Así se facilita el cambio del gestor de base de datos puesto que únicamente se requiere modificar las funciones de esta capa. También, tiene un módulo que gestiona el almacenamiento y recuperación de archivos multimedia como imágenes y audio.

IV-A0b. Capa de servicio: Usa las funciones de la capa de acceso a datos para integrar la información y gestionar la lógica de la aplicación. Aquí se realizan validaciones para confirmar que el televidente está modificando únicamente información asociada a su cuenta.

IV-A0c. Capa de servicios RESTful: Se comunica con la capa de servicios y provee las funcionalidades de la aplicación mediante una interfaz REST. Todas las peticiones a recursos retornan el mismo tipo de respuesta para facilitar el manejo de la información en la aplicación cliente.

IV-A0d. Autorización de acceso: Se propone el uso del protocolo OAuth para autorizar el acceso a los clientes. El mecanismo de ingreso implementado es por concesión de contraseña (*password grant*), es decir, el usuario envía información de su nombre de usuario y contraseña a cambio de un *token*.

IV-B. Cliente

Es la aplicación ejecutada en la televisión cuya función es administrar la comunicación con el servidor y la interacción con el televidente. Como se describe a continuación:

IV-B0a. Comunicación con el servidor: Solicita los recursos al servidor mediante peticiones HTTP que incluyen en su cabecera el *token* asociado al televidente. La creación de usuarios no requiere de esta autorización, por tanto, se debería solicitar una confirmación mediante correo electrónico. El uso de *tokens* evita que el usuario deba introducir múltiples veces sus credenciales para ingresar en la aplicación.

IV-B0b. Interacción con el televidente: Se recomienda crear un módulo que maneje la navegación de la aplicación, es decir que cargue las pantallas y asigne los parámetros requeridos para su instanciación. Considerando que la aplicación debe ser ejecutada en dispositivos de diferentes resoluciones, se debe garantizar que los componentes de la interfaz mantengan sus proporciones. El control remoto es el principal mecanismo de interacción, por tanto se debe minimizar el número de cuadros de texto y los elementos navegables en la pantalla.

V. DESARROLLO DE LA APLICACIÓN

En esta sección se presenta el proceso de desarrollo de la aplicación ShimiTV de acuerdo a la arquitectura planteada previamente. Se describe la metodología utilizada para el desarrollo así como los materiales y herramientas.

V-A. Metodología de Desarrollo

Para el desarrollo se usó una metodología Ágil-TVDi. Esta metodología requiere la elaboración de documentos que especifiquen las actividades (lista de producto) necesarias para establecer los requerimientos, diseño e implementación de la aplicación. Las actividades se planificaron en *sprints* semanales durante un periodo de cuatro meses. Las actividades de la lista de producto incluyen: el análisis y configuración de las herramientas de desarrollo, diseño e implementación del modelo de datos, desarrollo de la lógica de los servicios REST, desarrollo de la lógica e interfaz gráfica de la aplicación cliente, consideraciones de seguridad y usabilidad.

V-B. Herramientas y Desarrollo

El servidor fue codificado en el *framework Spring Boot* en el entorno de desarrollo *Netbeans 8.1*. Este *framework* facilita la configuración, desarrollo y despliegue de interfaces web.

El acceso a las APIs se controló mediante los módulos *Spring Security* y *OAuth2*. *Spring Security* provee la interfaz *UserDetails*, que administra las cuentas de usuario. Las contraseñas fueron cifradas mediante la clase *BCryptPasswordEncoder*.

La información se almacenó en la base de datos relacional *MariaDB*. Las consultas a la misma se realizaron desde el servidor con la clase *CrudRepository* de la librería *Spring Data JPA*. Esta clase implementa operaciones *CRUD* frecuentes para aquellas clases del dominio que tenían anotaciones *JPA*.

El lado del cliente se programó con el entorno de desarrollo *Android Studio 3.0.1* con el lenguaje de programación *Java*. La comunicación con el servidor se gestionó mediante la dependencia *Spring for Android*, misma que construye las peticiones de recursos al servidor, incluyendo las cabeceras de autorización y recuperación de los *tokens*. Los *tokens* son el único mecanismo de acceso e identificación del televidente, por tanto se deben almacenar en un directorio de acceso restringido a otras aplicaciones conocido como *SharedPreferences*. El diseño de la interfaz gráfica y la interacción con el usuario se gestionó mediante los componentes de la librería *Leanback Support Library* de *Google*. Esta librería posee *widgets*, fragmentos y actividades diseñadas con los estándares de diseño de *Google* para aplicaciones de televisión. Sin embargo es difícil modificar o combinar estos componentes dentro de otros. Por esta razón fue necesario implementar fragmentos con comportamiento personalizado como el caso del menú de opciones que se muestra en la *Figura 1*.



Figura 1: RowFragment en la pantalla de selección de lecciones

Considerando que la aplicación debía adaptarse a diferentes resoluciones de pantalla, se usó el componente *Constraint Layout* para garantizar que los componentes de la interfaz mantengan sus proporciones. Finalmente, se usó la librería *Picasso* para descargar, presentar, almacenar o eliminar imágenes de tal manera que no se agoten los recursos de almacenamiento de la televisión. Los resultados de este desarrollo se muestran en la siguiente sección.

VI. DISCUSIÓN DE RESULTADOS

La aplicación fue probada en un dispositivo virtual *Android* con acceso a un servidor local instalado en el Departamento

de Investigación de la Universidad de Cuenca. Como resultado de todo el proceso de desarrollo y planificación de actividades se obtuvo la primera versión de la aplicación *ShimiTV*, esta cuenta con un curso para el aprendizaje del idioma inglés a partir del español. Este curso tiene tres lecciones de nivel básico, cada una con un cuestionario de diez preguntas. En la *Figura 2* se observan las pantallas por las que debe navegar el televidente para acceder a los cuestionarios de una lección, en (a) se muestra la pantalla de ingreso, (b) la selección de cursos, (c) la selección de niveles y (d) selección de lecciones, en esta última también es posible ingresar a las evaluaciones.



Figura 2: Pantallas: a) Inicio de Sesión b) Selección de Cursos, c) Selección de Niveles y d) Selección de Lecciones.

Se implementaron cuatro tipos de preguntas: escritura, selección múltiple, selección múltiple con imágenes y ordenamiento, como se muestran en las interfaces gráficas de la *Figura 3*.

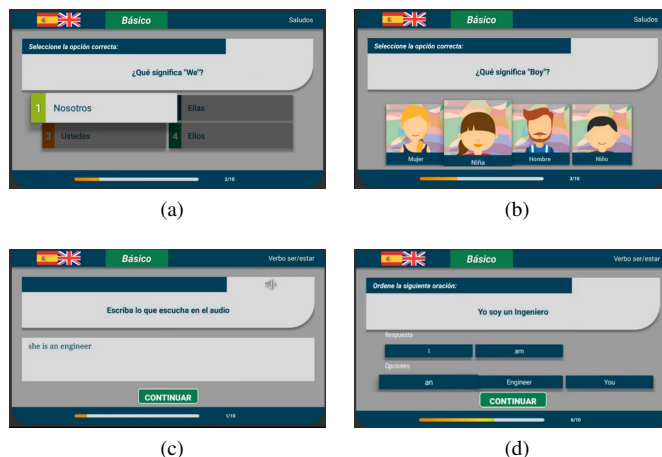


Figura 3: Preguntas: a) Selección múltiple, b) Selección múltiple con imágenes, c) Pregunta de escritura, y d) Pregunta de Ordenar.

La retroalimentación que brinda la aplicación se muestra en la *Figura 4*, en (a) se presenta la pantalla de respuesta a una

pregunta contestada por el televidente de manera correcta. En (b) se muestra una pantalla a una respuesta incorrecta con su respectiva retroalimentación.



Figura 4: Pantallas de retroalimentación. a) Respuesta Correcta, y b) Respuesta Incorrecta.

ShimiTV fue presentada como parte de una exposición realizada en la Universidad de Cuenca, dirigida a estudiantes de diferentes universidades y colegios de la región. Como parte de la exposición, los asistentes pudieron interactuar con la aplicación y dar su criterio de la experiencia adquirida, obteniendo los siguientes resultados:

- Las preguntas de escritura requirieron un mayor tiempo para ser completadas, puesto que se deben seleccionar los caracteres de la oración mediante el emulador del control remoto.
- Las preguntas con audio fueron llamativas y contribuyeron en el aprendizaje, debido a que se tenía mayor claridad en la pronunciación de las palabras.
- El diseño de las pantallas y el flujo de navegación resultó intuitivo, debido que no fue necesario guiar al usuario durante la ejecución de las actividades.

VII. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se presentó el proceso de análisis, diseño e implementación de *ShimiTV*, una aplicación de t-learning orientada al aprendizaje de idiomas extranjeros. Fue desarrollada para el Sistema Operativo Android TV usando la metodología de desarrollo Ágil-TVD. Como parte de la revisión de literatura se pudo evidenciar que no existen trabajos relacionados al proceso de desarrollo de software para Android TV, en este contexto, se contó la experiencia obtenida así como la utilidad de varios métodos y librerías propias de Android que facilitan el desarrollo de aplicaciones para TVDi.

En el desarrollo de aplicaciones interactivas, Android TV facilita la creación de las interfaces gráficas debido a su entorno integrado *Android Studio*. La disponibilidad de librerías, ayuda a la implementación de funcionalidades como el manejo y despliegue de imágenes, así como realizar peticiones al servidor de una aplicación. Por tanto es un entorno apropiado para el desarrollo ágil, facilitando cambios continuos e incrementales.

ShimiTV busca motivar en el desarrollo de aplicaciones educativas para TVDi, donde el televidente tiene la posibilidad de adquirir nuevos conocimientos sobre un idioma, además la posibilidad de auto-evaluar lo aprendido. Los diferentes tipos de preguntas permiten que la aplicación alterne entre varias opciones, haciéndola más atractiva para el televidente, estimulando así su interacción y concentración. T-learning

debe ser aprovechado conjuntamente con la tecnología de TVDi para ofrecer contenido y herramientas de calidad a la sociedad a través de un aparato de difusión tan importante como es el televisor.

Para versiones futuras de ShimiTV se espera agregar un mayor número de lecciones y nuevos cursos de idiomas a la base de datos. Permitir la selección del idioma de la aplicación y agregar una sección adicional de vocabulario que sirva de introducción para cada lección. Finalmente se busca realizar pruebas en ambientes reales de TVDi.

AGRADECIMIENTOS

Se agradece a la Dirección de Investigación de la Universidad de Cuenca (DIUC) por el apoyo y financiamiento brindado para la realización de este proyecto.

REFERENCIAS

- [1] R. González and E. Ponce, "Proceso de la implementación de la televisión digital en el ecuador," 2012.
- [2] J. Avila, K. Palacio-Baus, M. Espinoza-Mejía, and V. Saquicela, "Sistema de recomendación de contenidos audiovisuales: Algoritmo de inferencia semántica," *Maskana*, vol. 65, no. Supl., 2016.
- [3] S. B. Buchdid, R. Pereira, and M. C. C. Baranauskas, "Creating an idtv application from inside a tv company: a situated and participatory approach," in *International Conference on Informatics and Semiotics in Organisations*, pp. 63–73, Springer, 2014.
- [4] F. Bellotti, S. Vrochidis, E. Parissi, P. Lhoas, D. Mathevon, M. Pellegrino, G. Bo, and I. Kompatsiaris, "A t-learning courses development and presentation framework,"
- [5] Y. Wu, S. Hirakawa, U. H. Reimers, and J. Whitaker, "Overview of digital television development worldwide," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 8–21, 2006.
- [6] S. L. Martorelli, S. R. Martorelli, C. V. Sanz, M. J. Abásolo Guerrero, and P. M. Pesado, "Histotv: Primeros pasos hacia una aplicación educativa para tvdi sobre histología animal," in *VI Congreso de Televisión Digital Interactiva y IV Jornadas Iberoamericanas de Aplicaciones y Usabilidad de la TVDi (JAUTI 2015)(Palma de Mallorca, España, 14 al 16 de octubre de 2015)*, 2015.
- [7] V. Bureš, J. Mikulecká, and D. Ponce, "Digital television as a usable platform for enhancement of learning possibilities for the elderly," *SAGE Open*, vol. 7, no. 2, p. 2158244017708817, 2017.
- [8] J. J. Pazos-Arias, M. López-Nores, J. García-Duque, R. P. Díaz-Redondo, Y. Blanco-Fernández, M. Ramos-Cabrera, A. Gil-Solla, and A. Fernández-Vilas, "Provision of distance learning services over interactive digital tv with mhp," *Computers & Education*, vol. 50, no. 3, pp. 927–949, 2008.
- [9] K. Chorianopoulos, "Interactive tv design that blends seamlessly with everyday life," in *Universal Access in Ambient Intelligence Environments*, pp. 43–57, Springer, 2007.
- [10] A. X. Reyes, D. E. Soto, and J. A. Jimenez, "Madce-tvd-model agile development educational content for digital television," *IEEE Latin America Transactions*, vol. 13, no. 10, pp. 3432–3438, 2015.
- [11] M. Z. Bjelica, N. M. Jovanovic, G. Velickic, and N. Teslic, "Challenges of integrating android to a tv service operators' ecosystems," in *Consumer Electronics (ICCE), 2016 IEEE International Conference on*, pp. 585–588, IEEE, 2016.
- [12] N. Kuzmanovic, T. Maruna, M. Savic, G. Miljkovic, and D. Isailovic, "Google's android as an application environment for dtv decoder system," in *Consumer Electronics (ISCE), 2010 IEEE 14th International Symposium on*, pp. 1–5, IEEE, 2010.
- [13] M. Vidakovic, N. Teslic, T. Maruna, and V. Mihic, "Android4tv: A proposition for integration of dtv in android devices," in *Consumer Electronics (ICCE), 2012 IEEE International Conference on*, pp. 437–438, IEEE, 2012.
- [14] Android, "Google i/o: Design, develop, distribute @ONLINE," 2014.