

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA
GRADO EN INGENIERÍA DEL SOFTWARE

**AMPLIACIÓN TV-LEARNING: APRENDIZAJE A
DISTANCIA CON SMART TVS**

**EXTENSION TV-LEARNING: DISTANCE LEARNING WITH
SMART TVS**

Realizado por
JESÚS JALDO ASENJO

Tutorizado por
DANIEL GARRIDO MÁRQUEZ

Departamento
LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JULIO 2015

Fecha defensa:
El Secretario del Tribunal

Resumen

Uno de los aparatos eléctricos más usados en los hogares del mundo es la televisión. Durante años se ha usado como un medio de comunicación, en el cuál podías enterarte de noticias ocurridas en el mundo, además de poder pasar tu tiempo libre viendo películas o series.

Pero en la actualidad, con la llegada de Internet, las televisiones también cambiaron y ahora son aparatos inteligentes y capaces de conectarse a Internet. En poco se parecen los primeros modelos con los modelos actuales, tanto por aspecto, como por funcionalidad.

En este Trabajo Fin de Grado, junto con mi anterior Proyecto Fin de Carrera, hemos conseguido realizar dos aplicaciones para las televisiones Smart TV's de Samsung.

Las aplicaciones están inspiradas en los cursos de enseñanza de autoescuela, en los que podemos ver material para el aprendizaje y realizar test.

En el PFC realizamos la aplicación para el alumnado, para poder estar matriculado en cursos y realizar test. También realizamos una aplicación web para el ordenador, en la que el docente podía crear esos cursos y visualizar a sus alumnos.

En el TFG, hemos dotado al docente también de una aplicación de Smart TV's (además de la que ya disponía para PC), en la cual podrá ver los cursos creados por él, y hacer un seguimiento de los alumnos en sus cursos.

Palabras claves: Televisión, Internet, Smart TV, Samsung.

Abstract

Television is one of the most used equipment in the world. For years it has been used as a media which has offered many different contents such as worldwide news, sport, films or series.

Nowadays Internet has arrived to TV, and that has changed it. Televisions are now intelligent and capable to use Internet. TVs have also changed its appearance and functionality, current models are very different from first models.

In this final project, and also in my previous project about this same subject, we have managed to make two applications for Samsung Smart TVs.

This new applications that we have created are inspired in driving school courses, which let you see learning material and make test.

In my previous Project we made an application for students, to be enrolled in courses and to make test. We also did a web application for computers in which teachers could create courses and view their students.

In this Project we have also created an application for the teacher (besides the PC application) in which they can see the courses they have created and they can keep track of students in their courses.

Keywords: Television, Internet, Smart TV, Samsung.

Índice

1.	Introducción.....	5
1.1.	Motivación.....	5
1.2.	Objetivos	5
1.3.	Metodologías y Tecnologías de Desarrollo	7
1.4.	Fases de Trabajo	7
1.5.	Materiales.....	7
1.6.	Estructura de la Memoria.....	8
2.	Entorno Tecnológico.....	9
2.1.	Tecnologías y Herramientas Web.....	9
2.1.1.	Servlets	9
2.1.2.	JSP	11
2.1.2.1.	Tags JSP.....	11
2.1.2.2.	Directivas.....	12
2.1.2.3.	Acciones.....	12
2.2.	Tecnologías del Servidor	12
2.2.1.	Servicios Web	13
2.2.2.	Base de Datos con WampServer	13
2.3.	Tecnologías TV	13
2.3.1.	Samsung Smart TV.....	14
3.	Especificación del Sistema.....	17
3.1.	Requisitos Funcionales.....	17
3.2.	Requisitos No Funcionales.....	18
3.3.	Actor del Sistema	19
3.4.	Casos de Uso del Sistema	19
3.5.	Matriz de Trazabilidad	25
3.6.	Diagrama de Entidades	26
3.7.	Diagramas de Secuencia.....	27
4.	Diseño de la Aplicación.....	29
4.1.	Modelo de Datos	29
4.2.	Arquitectura de Desarrollo.....	34
4.2.1.	Arquitectura de Desarrollo Web	34
4.2.2.	Arquitectura de Desarrollo de Televisión	36
5.	Implementación	37
5.1.	Implementación Web.....	37

5.1.1.	Descripción Web	37
5.1.2.	Módulo src/modelo.....	37
5.1.3.	Diseño Web.....	39
5.1.4.	Servlets	40
5.2.	Implementación Televisión	41
5.2.1.	Descripción Sistema Televisivo	41
5.2.2.	Estructura del Código	42
6.	Conclusiones	45
6.1.	Problemas Encontrados.....	45
6.2.	Conclusiones.....	45
6.3.	Limitaciones.....	45
6.4.	Líneas Futuras	46
Apéndice A: Manual de Instalación		47
A.1.	Requisitos Hardware	47
A.2.	Requisitos Software	47
A.3.	Instalación.....	47
A.3.1.	Instalación de la Base de Datos	47
A.3.2.	Despliegue de la Aplicación	48
A.3.3.	Despliegue de Smart TV.....	48
A.3.4.	Configurar las Opciones de la Aplicación	48
Apéndice B: Manual de la Aplicación.....		51
B.1.	Iniciar Sesión	51
B.2.	Pantalla Principal.....	52
B.3.	Ver Mis Datos.....	52
B.4.	Ver Curso	53
B.5.	Ver Material	54
B.6.	Ver Test	54
B.7.	Ver Alumno	55
B.8.	Ver Test Realizados Por Alumno	56
Bibliografía.....		57
Índice de Ilustraciones		59

1. Introducción

En este capítulo se presenta la motivación y el objetivo del trabajo, así como las distintas fases por las que se ha pasado durante su desarrollo.

1.1. Motivación

La principal motivación de este Trabajo Fin de Grado ha sido la ampliación a mi Proyecto Fin de Carrera, dotándola de una nueva aplicación Smart TV.

El actual acceso desde distintos dispositivos tales como televisiones, portátiles, ordenadores de sobremesa, tablets, teléfonos móviles, a las diferentes aplicaciones y contenido de la red hacen que haya querido investigar en la tecnología de las televisiones y aplicaciones web.

El proyecto tuvo la intención de facilitar al usuario el modo de hacer cursos a través de su televisor. Por ejemplo, me inspiré en los cursos de autoescuela, en los cuales podemos estar matriculados. Podríamos ver el diferente material subido por nuestro profesor en nuestro curso, además de poder realizar los diferentes test relacionados con ese curso, todo esto desde nuestro televisor. Sin embargo, el profesor si debía registrarnos a través de un dispositivo como el portátil, dichos profesores podrían crear distintos cursos y matricular a sus diferentes alumnos en sus cursos. En dichos cursos el profesor podrá subir material y crear test para que el alumno pueda comprobar su nivel en los determinados temas. Un alumno podría estar matriculado en varios cursos a la vez con diferentes profesores.

Este trabajo tiene la intención de facilitar al usuario docente el modo de visualizar sus cursos, los alumnos matriculados en sus cursos y los test realizados por estos alumnos, todo esto desde su televisor.

1.2. Objetivos

El objetivo de este Trabajo Fin de Grado es ampliar mi proyecto Fin de Carrera de Ingeniería en Informática de Gestión, llamado TV-Learning: Aprendizaje a distancia con Smart TVs. Este proyecto consistió en una aplicación para Smart TV (de la marca Samsung) para enseñanza a distancia, en la que el alumnado disponía de una aplicación Smart TV y el docente de una aplicación para PC.

Con esta ampliación hemos pretendido desarrollar también una aplicación Smart TV para la docencia.

Este enfoque es, por sí mismo, una innovación en el aprendizaje a distancia, puesto que la persona puede, desde su casa y manejando dispositivos de uso cotidiano para todo el mundo (un ratón, un mando a distancia y un televisor) realizar todo tipo de tareas de aprendizaje.

La principal ventaja del sistema a desarrollar, estará en la facilidad de uso, y en definitiva, la familiaridad con la que cualquier persona puede utilizar esta aplicación, desde su televisión. En este caso, el docente podría hacer un fácil seguimiento de sus cursos, alumnos, etc. desde casa.

El sistema ya constaba de las siguientes partes gracias al Proyecto Fin de Carrera.

- Creación de material de aprendizaje. Esta creación será realizada por el docente encargado del curso. Se le permitirá subir material, (imágenes) así como texto, de manera que el alumno tendrá después a su disposición este material. La creación del material será realizada en un PC, incorporándose dicho material a los contenidos del curso.
- Consulta del material de aprendizaje: mediante una aplicación ejecutada en la Smart TV, el alumno inscrito en el curso podrá consultar el material realizado por el docente.
- Evaluación de los estudiantes: una vez consultado el material por el estudiante, éste podrá realizar diversos test también preparados por el docente en el PC.
- Todas las partes que afectan directamente al docente: creación del material, gestión de los cursos, seguimiento de alumnos, etc. están incluidas en una sola aplicación PC que podrá ser utilizada por el docente.
- En el caso de los estudiantes, dispondrán de la aplicación para Smart TV, desde donde podrán consultar el material, realizar los test o comprobar sus calificaciones.

El nuevo sistema consta además de las siguientes partes en una nueva aplicación Smart TV para el docente.

- El docente podrá visualizar sus datos y cursos en los que imparte docencia.
- El docente visualizará el material de aprendizaje y los test de sus cursos.
- El docente podrá consultar los datos de los alumnos matriculados en sus cursos, mediante una aplicación ejecutada en un dispositivo Samsung Smart TV.

- El docente podrá comprobar las respuestas que han realizado sus alumnos en los diferentes test de los cursos impartidos por dicho docente.

1.3. Metodologías y Tecnologías de Desarrollo

Se seguirá un proceso de desarrollo ágil iterativo en el que se generarán pequeñas versiones. Para el desarrollo del trabajo se utilizarán distintos diseños UML para la gestión de la base de datos. Para la creación de los servicios web se utilizará Java EE y para el desarrollo de la aplicación de televisión Samsung SDK.

1.4. Fases de Trabajo

Como se ha mencionado en el apartado anterior, el trabajo se realizará de forma iterativa, en cada iteración se podrán identificar las siguientes fases:

- Desarrollo de prototipos para evaluar las tecnologías empleadas.
- Especificación de las nuevas funcionalidades que se añadirán en cada iteración.
- Desarrollo de los distintos servicios web que permitan la comunicación del servidor con los dispositivos de TV.
- Implementación de las funcionalidades para la Smart TV.
- Integración de las diferentes funcionalidades.
- Pruebas y documentación.

1.5. Materiales

En la medida de lo posible se ha hecho uso de recursos de libre distribución: MyEclipse como entorno de desarrollo integrado (IDE), Java como lenguaje de programación, Wampserver como gestor de la base de datos y MagicDraw UML como herramienta de especificación de UML y algunas tecnologías como:

- JavaServer Pages (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java.

- Samsung Smart TV SDK como framework de programación para la televisión.

1.6. Estructura de la Memoria

La memoria de este trabajo está dividida en seis capítulos, con diversos subapartados en los que se describe el trabajo realizado durante cada capítulo. A continuación el contenido de cada capítulo:

- Capítulo 1: **Introducción** - Es el capítulo en el que nos encontramos. Aquí se presenta la motivación y el objetivo del trabajo, así como las distintas fases por las que se ha pasado durante su desarrollo.
- Capítulo 2: **Entorno tecnológico** - En este capítulo se describen las tecnologías utilizadas durante el desarrollo del trabajo así como la forma de integrar las mismas dentro de la arquitectura de desarrollo.
- Capítulo 3: **Especificación de la aplicación** – En este capítulo se describen el conjunto de requisitos de los que se compone el trabajo y los análisis del sistema.
- Capítulo 4: **Diseño de la aplicación** – A partir de la elaboración de los análisis funcionales de los casos de uso se procede a la elaboración del diseño de la aplicación.
- Capítulo 5: **Implementación** – En este capítulo se presentará el diseño de arquitectura e implementación de los distintos componentes necesarios para crear el trabajo propuesto.
- Capítulo 6: **Conclusiones y líneas futuras** – Se describe los problemas que se han encontrado a lo largo del desarrollo de la aplicación, unas conclusiones finales y para terminar unas líneas sobre una ampliación para el futuro de la aplicación.

Adicionalmente se incluyen diferentes Anexos de los manuales de instalación y uso, así como la Bibliografía y el índice de ilustraciones:

- **Manual de instalación** – El manual de instalación del sistema se describe en este capítulo.
- **Manual de usuario** – Se describe el manual de usuario en este capítulo.
- **Bibliografía** – Contiene las referencias usadas durante el desarrollo del trabajo.
- **Índice de ilustraciones** – Contiene la referencia de las ilustraciones que se incluyen en la memoria.

2. Entorno Tecnológico

En este capítulo se describen las principales tecnologías que se han utilizado para desarrollar cada uno de los componentes del sistema final, así como las herramientas de desarrollo que se necesitaron.

2.1. Tecnologías y Herramientas Web

Para el desarrollo de las páginas Web se utilizaron básicamente dos tecnologías: las páginas se desarrollaron en JSP (Java Server Pages) y para la comunicación de estas páginas con el servidor se utilizaron Servlets. El objetivo de utilizar estas tecnologías era el de mantener por separado cada capa siguiendo el patrón de diseño MVC (Modelo, Vista, Controlador).

2.1.1. Servlets

Los Servlets son una tecnología de la plataforma de Java para la mejora y ampliación de los servidores Web. Los Servlets proporcionan un componente, independiente de la plataforma, para la creación de aplicaciones basadas en Internet, sin las limitaciones de rendimiento de los programas de Interfaz de entrada común (CGI).

Los Servlets tienen acceso a toda la familia de APIs Java, incluida la API JDBC para acceder a bases de datos empresariales. Los Servlets también pueden acceder a una biblioteca específica HTTP de llamadas y recibir todos los beneficios de la madurez del lenguaje Java, incluyendo la portabilidad, rendimiento, reutilización.

El ciclo de vida de un Servlet se divide en los siguientes puntos:

1. El cliente solicita una petición a un servidor vía URL.
2. El servidor recibe la petición.
 - a) Si es la primera, se utiliza el motor de Servlets para cargarlo y se llama al método `init()`.
 - b) Si ya está iniciado, cualquier petición se convierte en un nuevo hilo. Un Servlet puede manejar múltiples peticiones de clientes.
3. Se llama al método `service()` para procesar la petición devolviendo el resultado al cliente.
4. Cuando se apaga el motor de un Servlet se llama al método `destroy()`, que lo destruye y libera de los recursos abiertos.

Podemos ver este ciclo de vida de una manera más visual en la Ilustración 1.

Ciclo de Vida

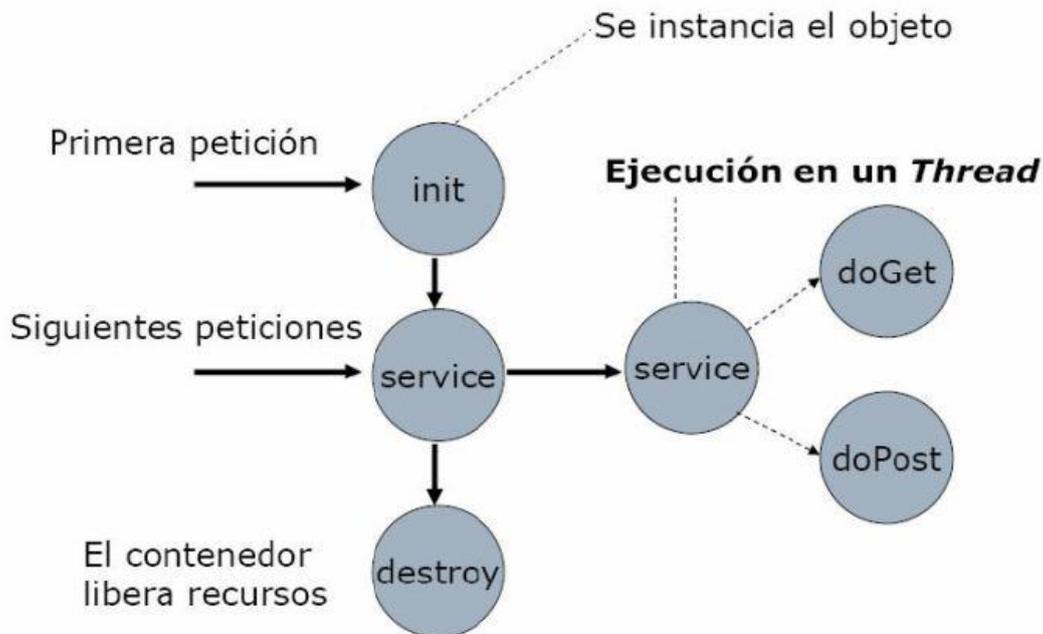


Ilustración 1. Ciclo de Vida de un Servlet

Existen dos métodos dentro de los Servlet:

- **doGet()** se llama después de una consulta de tipo HTTP GET. Generalmente esto ocurre cuando un usuario pulsa en un enlace o cuando se escribe directamente en la barra del navegador.
- **doPost()** se llama después de una consulta de tipo HTTP POST.

Los dos métodos se llaman desde la implementación por defecto del método **service()** que se encuentra en la clase base *HttpServlet*.

Normalmente el navegador pide siempre páginas mediante GET y puede enviar datos bajo las dos formas (GET y POST). Las dos diferencias son que:

- GET está restringido a nivel de medida de los parámetros enviados al servidor.
- El envío de datos mediante GET puede cuestionar la seguridad debido a que los parámetros se añaden a la URL.

2.1.2. JSP

La tecnología JSP es una extensión de la tecnología Servlet creada para apoyar la creación de páginas HTML y XML. Java Server Pages (JSP) permite desarrollar rápidamente y mantener con facilidad páginas Web. Como parte de la familia de tecnologías Java, la tecnología JSP permite el desarrollo rápido de aplicaciones Web de forma independiente a la plataforma utilizada. La tecnología JSP separa la interfaz de usuario de generación de contenidos, permitiendo a los diseñadores cambiar el diseño general de la página, sin alterar el contenido dinámico subyacente.

El uso de esta tecnología tiene varios beneficios como:

- **Extender el lenguaje JSP:** Se puede ampliar el lenguaje JSP con “etiquetas simples de manejadores”, que utilizan una nueva, más simple y más amplia, API de extensión de etiqueta. Esto estimula el creciente número de bibliotecas que se pueden usar, las etiquetas reutilizables disponibles, y que a su vez reduce la cantidad de código necesario para crear aplicaciones Web eficientes.
- **Fácil de escribir y mantener:** La JavaServer Pages Standard Tag Library (JSTL) es una librería de etiquetas de expresión que está ahora integrado en la tecnología JSP y ha sido actualizado para soportar funciones. El lenguaje de expresión ahora se puede utilizar en lugar de las expresiones scriptlet.

2.1.2.1. Tags JSP

Son zonas de código Java dentro de la propia página. Se utilizan etiquetas:

```
< % ... % >
```

- **Expresión:** se utiliza para integrar el valor de la expresión Java dentro del código, por ejemplo el valor de una variable recibida.

```
< % = expresión Java % >
```

- **Código:** utilizado para ejecutar código Java.

```
< % código Java % >
```

2.1.2.2. Directivas

Son indicaciones que se le dan al motor de Servlets acerca de los parámetros generales de ejecución para esa página.

```
< % @page atributo = "valor" % >
```

Los principales atributos que se pueden usar son:

- Cuando se necesite importar una librería externa.

```
< % @page import = "java.util" % >
```

- Para indicar el formato de la página (se pueden añadir varios atributos a la vez).

```
< % @page contentType = "text/html" pageEncoding = "UTF-8" % >
```

- También se pueden incluir otros ficheros dentro de la página JSP antes de que vaya a ser procesada.

```
< % @include file = "cabecera.html" % >
```

2.1.2.3. Acciones

- Incluir una página HTML o JSP como parte de la página actual.

```
< % jsp : include page = "cabecera.html" % >
```

- Direccionar a otra página.

```
< % jsp : forward page = "pagina.jsp" % >
```

2.2. Tecnologías del Servidor

Para el resto de componentes del sistema, se utilizó un denominador común, que fue el servidor. La función de este servidor es la de unificar todas las peticiones que se hicieran desde los distintos clientes, sirviendo a la vez de plataforma de acceso a otros servicios externos y de comunicación interna entre los clientes. Para el desarrollo de estas comunicaciones se utilizaron servicios Web.

2.2.1. Servicios Web

Los servicios Web son una tecnología que utiliza un conjunto de estándares y protocolos que permiten intercambiar información entre distintas aplicaciones. Las aplicaciones no tienen por qué estar desarrolladas en el mismo lenguaje de programación y ni siquiera en la misma plataforma para realizar el intercambio de información.

Las ventajas de utilizar los servicios web, son entre otras:

- Fomentar los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Aportar una interoperabilidad entre distintas aplicaciones independientemente del software o plataformas utilizadas.

2.2.2. Base de Datos con WampServer

WampServer es un entorno de desarrollo web para Windows con el que podrás crear aplicaciones web con Apache, PHP y bases de datos MySQL database. También incluye PHPMyAdmin y SQLLiteManager para manejar tus bases de datos fácilmente.

Las principales características es que provee a los desarrolladores con los cuatro elementos necesarios para un servidor Web: un Sistema Operativo (Windows), un manejador de base de datos (MySQL), un software para servidor Web (Apache) y un software de programación script Web (PHP (generalmente), Python o PERL), debiendo su nombre a dichas herramientas. WAMP incluye, además de las últimas versiones de Apache, PHP y MySQL, versiones anteriores de las mismas, para el caso de que se quiera testear en un entorno de desarrollo particular.

2.3. Tecnologías TV

Para el desarrollo del sistema, se ha usado la tecnología que proporciona Samsung Smart TV y su SDK de desarrollo.

2.3.1. Samsung Smart TV

De forma simple podemos definir el SDK Samsung Smart TV como una herramienta que proporciona Samsung para realizar aplicaciones para sus televisiones con tecnología Smart TV de forma sencilla. En primer lugar veremos de qué se componen.

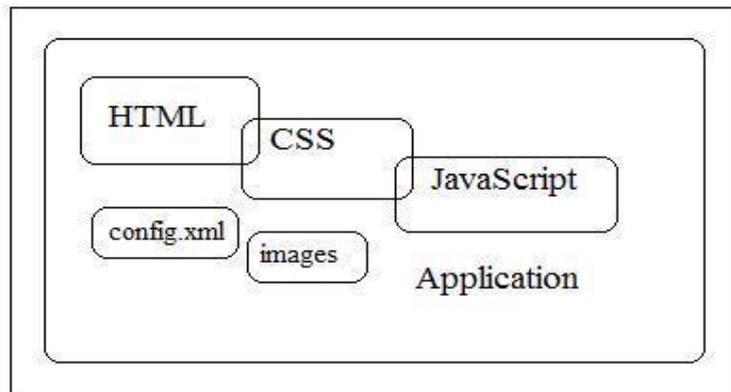


Ilustración 2. Composición de una Aplicación Samsung Smart TV

De forma simple podemos definir a una aplicación Smart TV como una página web que se ejecuta en el navegador web de la televisión. Se compone de archivos HTML, Javascript, CSS, un fichero de configuración e imágenes. Además de esto se puede usar el SDK de Samsung para usar librerías propias de Samsung.

Dentro del SDK existen tres naturalezas de proyectos:

- **Básicos:** En JavaScript, con objetos propios y librerías propias del SDK. Funciona con el concepto de que una aplicación se divide en n escenarios (vistas).
- **JavaScript:** En JavaScript, libertad de utilizar sus librerías o las tuyas propias, es más similar a un desarrollo web clásico.
- **Flash:** En ActionScript 2.0.

Las aplicaciones de Samsung Smart TV consisten como hemos dicho anteriormente, en una combinación de HTML, CSS, JavaScript, XML, imágenes y otros archivos. El Samsung Smart TV SDK organiza los archivos en los proyectos, proporcionando un panel explorador de proyectos y documentos, y una navegación por pestañas para poder acceder a los documentos del proyecto con facilidad. Se pueden crear automáticamente aplicaciones en blanco, con todos los archivos que una aplicación básica necesita.

Después de crear una aplicación, se puede probar con el validador de HTML y el Samsung Smart TV SDK emulador, que recrea el ambiente de la televisión en el ordenador.

El Samsung Smart TV SDK permite editar intuitivamente los aspectos visuales de las aplicaciones. El modo visual muestra una vista previa de la aplicación. Se puede hacer clic y arrastrar los componentes comunes de aplicaciones tales como botones, casillas de verificación, etc. En la ventana de edición, se puede cambiar el tamaño y organizar los componentes e inspeccionar las propiedades específicas de cada componente en el panel de Propiedades.



Ilustración 3. Samsung Smart TV SDK

3. Especificación del Sistema

En este capítulo se describen los requisitos funcionales y no funcionales del sistema, así como los casos de uso y diagramas más relevantes que se obtuvieron durante la fase de análisis de la aplicación.

3.1. Requisitos Funcionales

La aplicación deberá ofrecer a los usuarios las siguientes funcionalidades:

RF01. Acceder de forma privada al sistema – El sistema debe ser capaz de permitir el acceso de los usuarios a la aplicación.

RF02. Desconectar del sistema – El sistema debe ser capaz de poder cerrar sesión de los usuarios conectados en la aplicación.

RF03. Buscar usuarios – El sistema debe ser capaz de poder buscar usuarios matriculados en los cursos.

RF04. Ver usuarios – El sistema debe ser capaz de poder mostrar los usuarios en la aplicación.

RF05. Buscar curso – El sistema debe ser capaz de poder buscar cursos en la aplicación.

RF06. Ver curso – El sistema debe ser capaz de poder mostrar el curso de docencia en la aplicación.

RF07. Buscar material de aprendizaje – El sistema debe ser capaz de poder buscar materiales de aprendizaje de un curso determinado.

RF08. Ver material de aprendizaje – El sistema debe ser capaz de poder mostrar el material de aprendizaje de los diferentes cursos.

RF09. Buscar test – El sistema debe ser capaz de poder buscar test de un curso determinado.

RF10. Ver test – El sistema debe ser capaz de poder mostrar los test de los cursos.

RF11. Buscar test realizado - El sistema debe ser capaz de poder buscar test realizados por un alumno determinado en un curso.

RF12. Ver test realizado – El sistema debe ser capaz de poder mostrar el test realizado por un alumno en un curso.

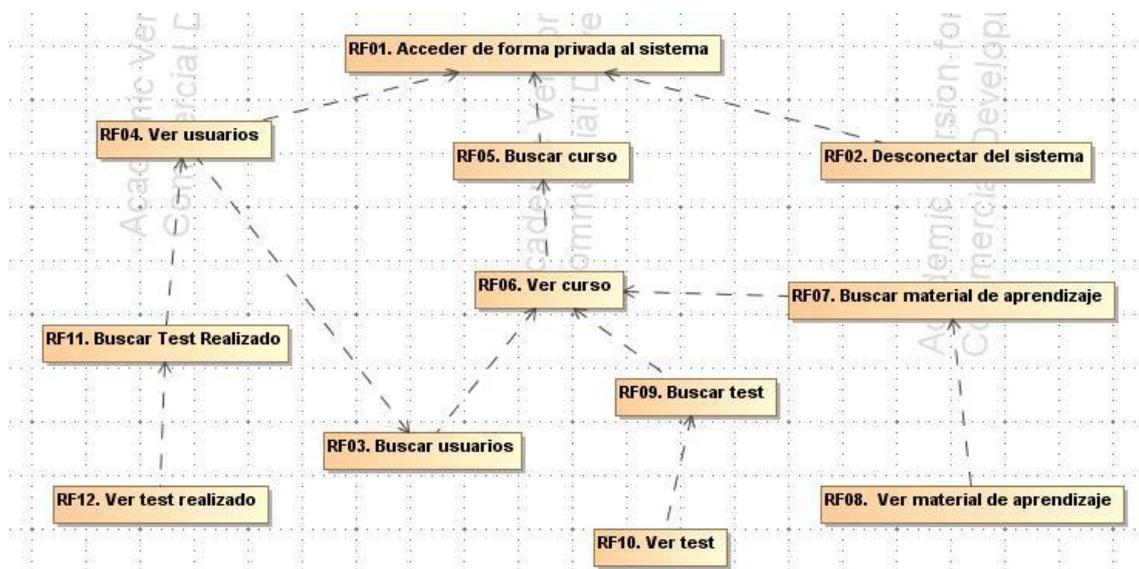


Ilustración 4. Diagrama de Requisitos Funcionales

3.2. Requisitos No Funcionales

Entre los requisitos no relacionados con la funcionalidad del sistema se destacan los siguientes:

RNF01. Usabilidad – El sistema ofrecerá una interfaz gráfica intuitiva y amigable que permita a los usuarios un fácil manejo y un aprendizaje de utilización rápido.

RNF02. Sistema operativo Smart TV – La aplicación debe poder ejecutarse en las televisiones Samsung Smart TV.

RNF03. Robustez – El sistema debe estar preparado para responder y controlar las entradas incorrectas de los usuarios.

RNF04. Extensibilidad – El sistema debe ser fácilmente extensible. Se le debe poder añadir nuevas funcionalidades de una manera relativamente fácil o mejorar las ya existentes sin afectar a su funcionamiento.

3.3. Actor del Sistema

En el sistema existirá un único actor debido a que la parte del alumnado se realizó en el PFC.

- **Usuario TV** – dentro del “Usuario TV” estará el profesorado. Estos podrán consultar sus datos, acceder a sus cursos en los que tienen alumnos matriculados, consultar los test y los materiales de sus cursos, consultar a sus alumnos y ver los test realizados por estos alumnos.

3.4. Casos de Uso del Sistema

A partir de los requisitos funcionales se identificó el sistema, del que se extrajeron los siguientes casos de uso. Para cada uno de estos casos de uso se va a dar una descripción con el siguiente formato:

Identificador:	Identificador del caso de uso
Título:	Título del caso de uso
Descripción:	Describe brevemente el caso de uso
Prerrequisito:	Indica si el caso de uso tiene algún prerrequisito
Escenario principal:	
Pasos que describen el funcionamiento normal del caso de uso	
Escenario alternativo:	
Posibles alternativas al funcionamiento normal del caso de uso	

En la Ilustración 5 podemos ver los distintos casos de uso del sistema:

3. Especificación del Sistema

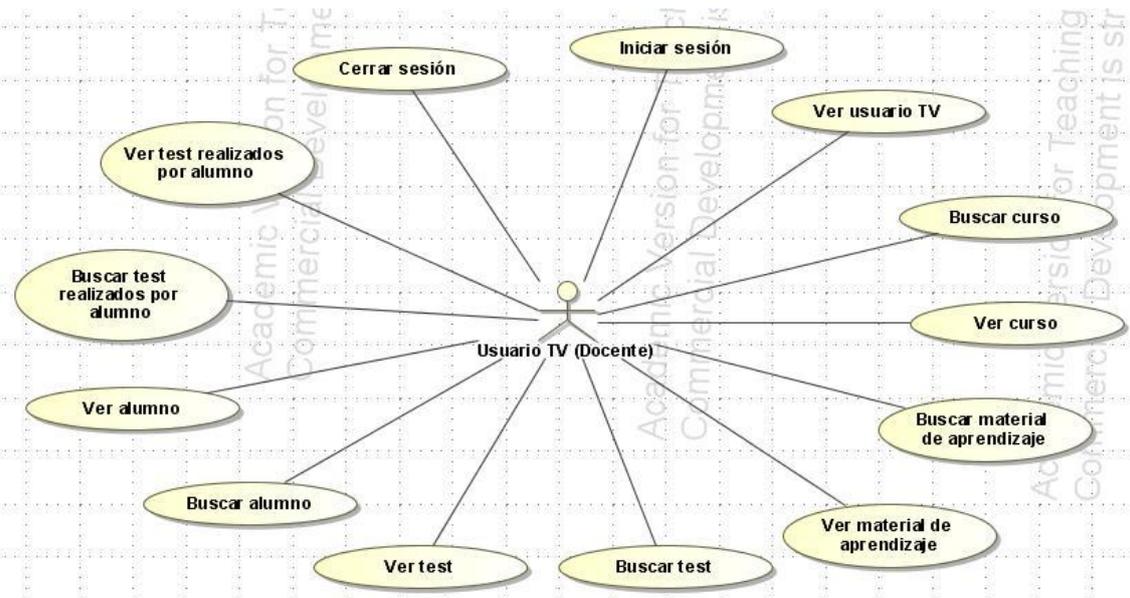


Ilustración 5. Casos de Uso

Identificador:	CU01
Título:	Iniciar sesión
Descripción:	Permite al Usuario TV acceder al sistema introduciendo su usuario y su contraseña
Escenario principal:	<ol style="list-style-type: none"> 1. El sistema muestra el formulario de entrada 2. El usuario introduce su nombre de usuario y contraseña, y pulsa el botón "Iniciar sesión" 3. El sistema valida la información del usuario 4. El sistema muestra la pantalla de inicio
Escenario alternativo:	<ol style="list-style-type: none"> 4.1. El sistema redirecciona a otra pantalla que le muestra un mensaje de error si el nombre de usuario o la contraseña son incorrectos

Identificador:	CU02
-----------------------	------

Título:	Cerrar sesión
Descripción:	Permite al Usuario TV cerrar la sesión
Prerrequisito:	Se ha cumplido el CU01
Escenario principal:	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón de “Apagar” en el mando a distancia 2. El sistema apaga la aplicación 	

Identificador:	CU03
Título:	Ver Usuario TV
Descripción:	Permite al Usuario TV ver su información
Prerrequisito:	Se ha cumplido el CU01
Escenario principal:	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Mis Datos” 2. El sistema muestra los datos personales del docente 	

Identificador:	CU04
Título:	Buscar curso
Descripción:	Permite al Usuario TV buscar un curso creado por él mismo
Prerrequisito:	Se ha cumplido el CU01
Escenario principal:	
<ol style="list-style-type: none"> 1. El sistema muestra en un desplegable los cursos en que el usuario es docente 2. El usuario elige el curso del que quiere ver el contenido 	
Escenario alternativo:	

2.1. El usuario no contiene ningún curso
--

Identificador:	CU05
Título:	Ver curso
Descripción:	Permite al Usuario TV visualizar el contenido de su curso
Prerrequisito:	Se ha cumplido el CU04
Escenario principal:	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Entrar” 2. El sistema valida el curso seleccionado, y muestra una pantalla con una pequeña descripción del curso y su nivel, además se pueden ver los nombres de los test que contiene el curso, los de los materiales pertenecientes a ese curso y los nombres de los alumnos matriculados en ese curso 	

Identificador:	CU06
Título:	Buscar material de aprendizaje
Descripción:	Permite al Usuario TV buscar un material creado por él mismo
Prerrequisito:	Se ha cumplido el CU05
Escenario principal:	
<ol style="list-style-type: none"> 1. El sistema muestra en un desplegable los materiales que contiene el curso en el que se encuentra 2. El usuario elige el material del que quiere ver el contenido 	
Escenario alternativo:	
2.1. El curso en el que se encuentra no contiene ningún material	

Identificador:	CU07
Título:	Ver material de aprendizaje
Descripción:	Permite al Usuario TV visualizar el material de aprendizaje
Prerrequisito:	Se ha cumplido el CU06
Escenario principal:	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón "Entrar" 2. El sistema valida el material seleccionado, y muestra una pantalla con los datos y la imagen del material de aprendizaje 	

Identificador:	CU08
Título:	Buscar test
Descripción:	Permite al Usuario TV buscar un test creado por él mismo
Prerrequisito:	Se ha cumplido el CU05
Escenario principal:	
<ol style="list-style-type: none"> 1. El sistema muestra en un desplegable los test que contiene el curso en el que se encuentra 2. El usuario elige el test del que quiere ver el contenido 	
Escenario alternativo:	
2.1. El curso en el que se encuentra no contiene ningún test	

Identificador:	CU09
Título:	Ver test
Descripción:	Permite al Usuario TV visualizar un test
Prerrequisito:	Se ha cumplido el CU08

Escenario principal:
<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Entrar” 2. El sistema valida el test seleccionado, y muestra las preguntas con sus respectivas respuestas

Identificador:	CU10
Título:	Buscar alumno
Descripción:	Permite al Usuario TV buscar a un alumno matriculado en su curso
Prerrequisito:	Se ha cumplido el CU05
Escenario principal:	
<ol style="list-style-type: none"> 1. El sistema muestra en un desplegable los alumnos que están matriculados en el curso en el que se encuentra 2. El usuario elige el alumno del que quiere ver la información 	
Escenario alternativo:	
2.1. El curso en el que se encuentra no contiene ningún alumno	

Identificador:	CU11
Título:	Ver alumno
Descripción:	Permite al Usuario TV ver la información de un alumno matriculado en su curso
Prerrequisito:	Se ha cumplido el CU10
Escenario principal:	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Entrar” 2. El sistema valida el alumno seleccionado, y muestra los datos del alumno, además se pueden ver los nombres de los test que ha realizado el alumno en ese curso 	

Identificador:	CU12
Título:	Buscar test realizados por alumno
Descripción:	Permite al Usuario TV buscar un test realizado por un alumno en su curso
Prerrequisito:	Se ha cumplido el CU11
Escenario principal:	
<ol style="list-style-type: none"> 1. El sistema muestra en un desplegable los test realizados por ese alumno correspondientes al curso en el que nos encontramos 2. El usuario elige el test del que quiere ver las respuestas respondidas por el alumno 	
Escenario alternativo:	
2.1. El alumno no ha realizado ningún test del curso	

Identificador:	CU13
Título:	Ver test realizados por alumno
Descripción:	Permite al Usuario TV visualizar las diferentes respuestas a un test realizado por un determinado alumno en un curso
Prerrequisito:	Se ha cumplido el CU12
Escenario principal:	
<ol style="list-style-type: none"> 1. El usuario pulsa el botón "Entrar" 2. El sistema valida el test seleccionado, y muestra una pantalla con las veces realizadas ese test y las respuestas escogidas por el alumno 	

3.5. Matriz de Trazabilidad

Con objeto de verificar que todos los requisitos funcionales del sistema se encuentran cubiertos por algún caso de uso se presenta la siguiente matriz de casos de usos por requisitos.

3. Especificación del Sistema

	RF01	RF02	RF03	RF04	RF05	RF06	RF07	RF08	RF09	RF10	RF11	RF12
CU01	X											
CU02		X										
CU03				X								
CU04					X							
CU05						X						
CU06							X					
CU07								X				
CU08									X			
CU09										X		
CU10			X									
CU11				X								
CU12											X	
CU13												X

Ilustración 6. Matriz de Trazabilidad

Como se puede apreciar, no queda ningún requisito sin cubrir y todo caso de uso surge de algún requisito.

3.6. Diagrama de Entidades

A partir de los casos de uso y los requisitos, se obtiene el diagrama de entidades (Ilustración 7) a alto nivel que será ampliado en la fase de diseño de la aplicación.

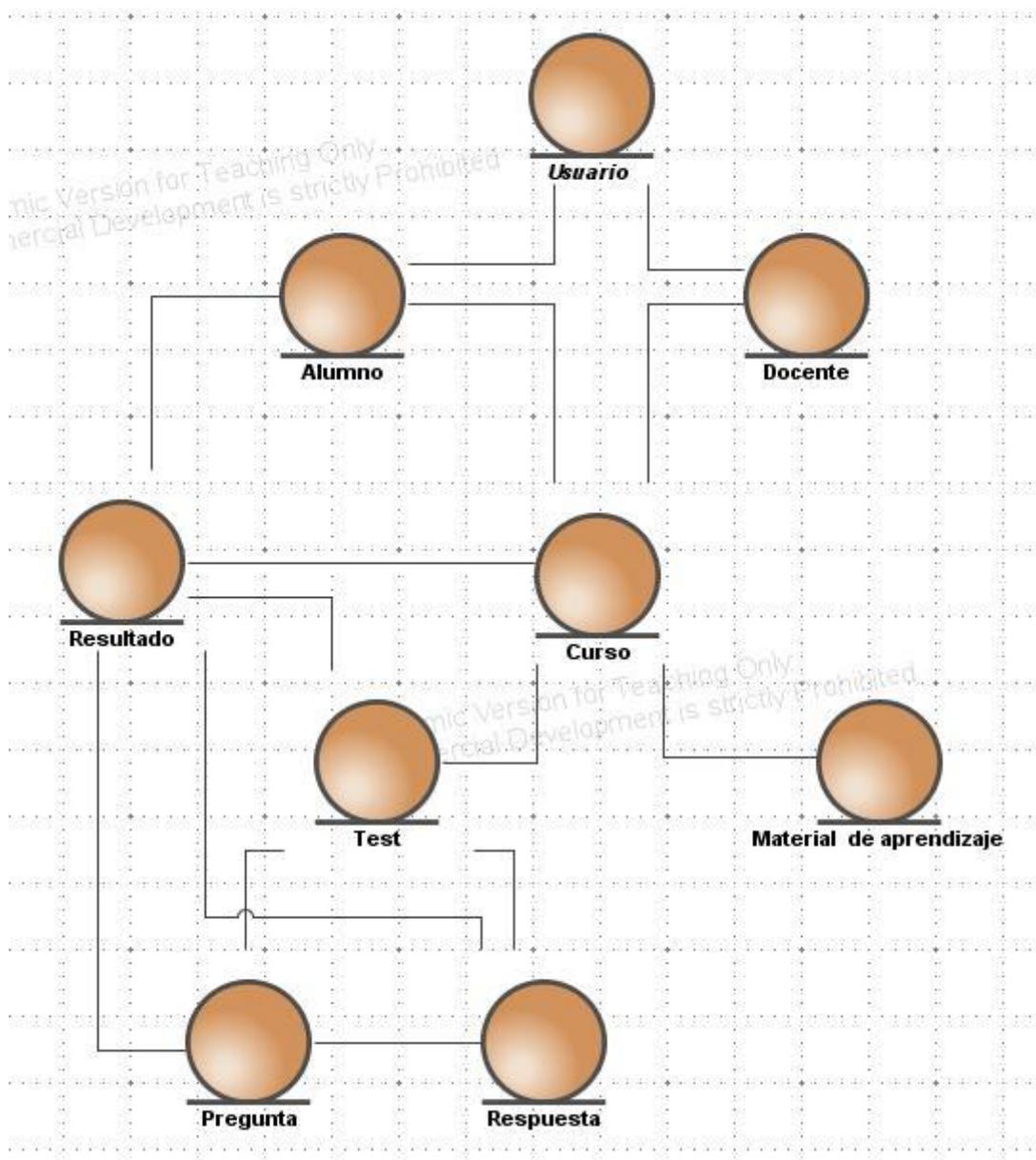


Ilustración 7. Diagrama de Entidades

3.7. Diagramas de Secuencia

Una vez obtenidas las entidades del diagrama de entidades y especificadas las distintas unidades de control, se pasó a la realización de los diagramas de secuencia. Aquí se muestran algunos ejemplos de los diagramas de secuencia.

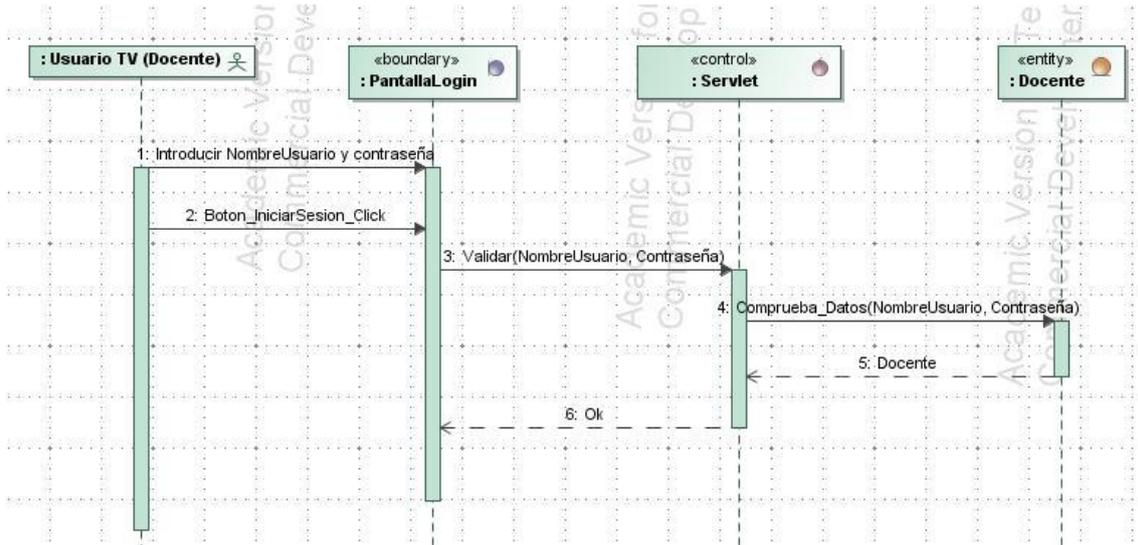


Ilustración 8. Diagrama de Secuencia CU01

En la Ilustración 8, el diagrama de secuencia simula como se ejecuta el CU01, que pertenece al CU Iniciar Sesión. Podemos observar la comunicación entre la pantallaLogin y el Servlet, el proceso de petición y su posterior respuesta.

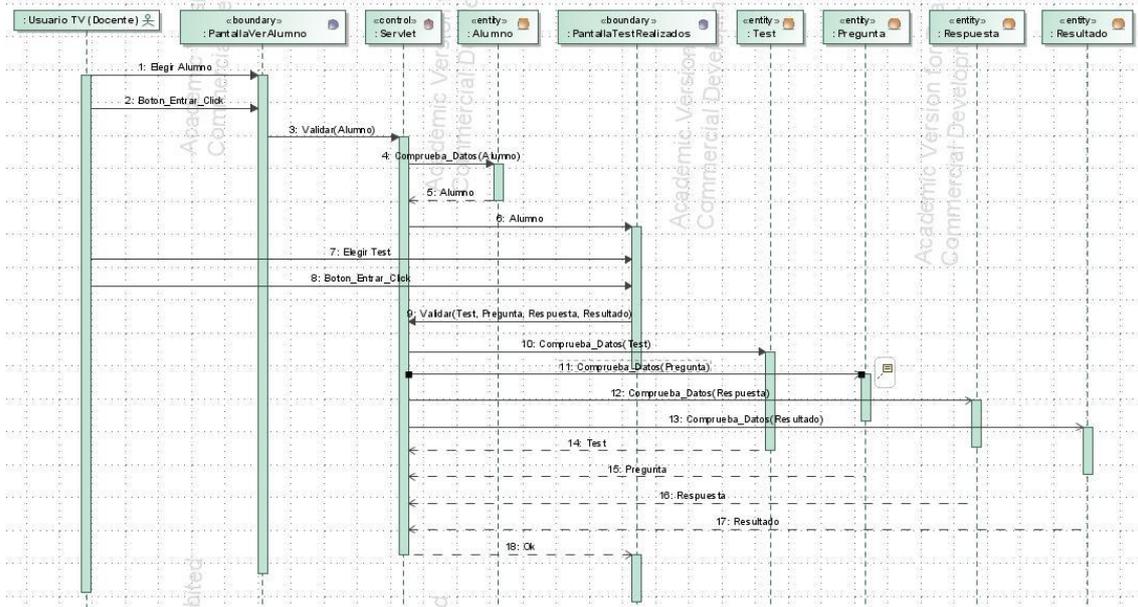


Ilustración 9. Diagrama de Secuencia CU13

Podemos encontrar como caso de uso más representativo, la visualización de los test realizados por el alumno, CU13. Por esto, en la Ilustración 9, podemos observar las peticiones en el diagrama de secuencia de la visualización de los test de un alumno. Cabe destacar la comunicación que se realiza con dos Servlets.

4. Diseño de la Aplicación

A partir de la elaboración de los análisis funcionales de los casos de uso se procede a la elaboración de la aplicación. Este diseño pasa por definir una arquitectura software que proporcione el marco de referencia necesario para la definición y posteriormente implementación del sistema. Esta arquitectura software debe establecer los fundamentos básicos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos y necesidades del proyecto.

4.1. Modelo de Datos

A partir de la documentación obtenida en la fase de análisis del sistema se procede a la elaboración del modelo de datos de la aplicación en el que almacenaremos toda la información relacionada con las entidades detectadas de nuestro sistema y que se muestran en la Ilustración 10. Cada una de estas entidades se corresponde con una tabla o vista del sistema que se explican a continuación.

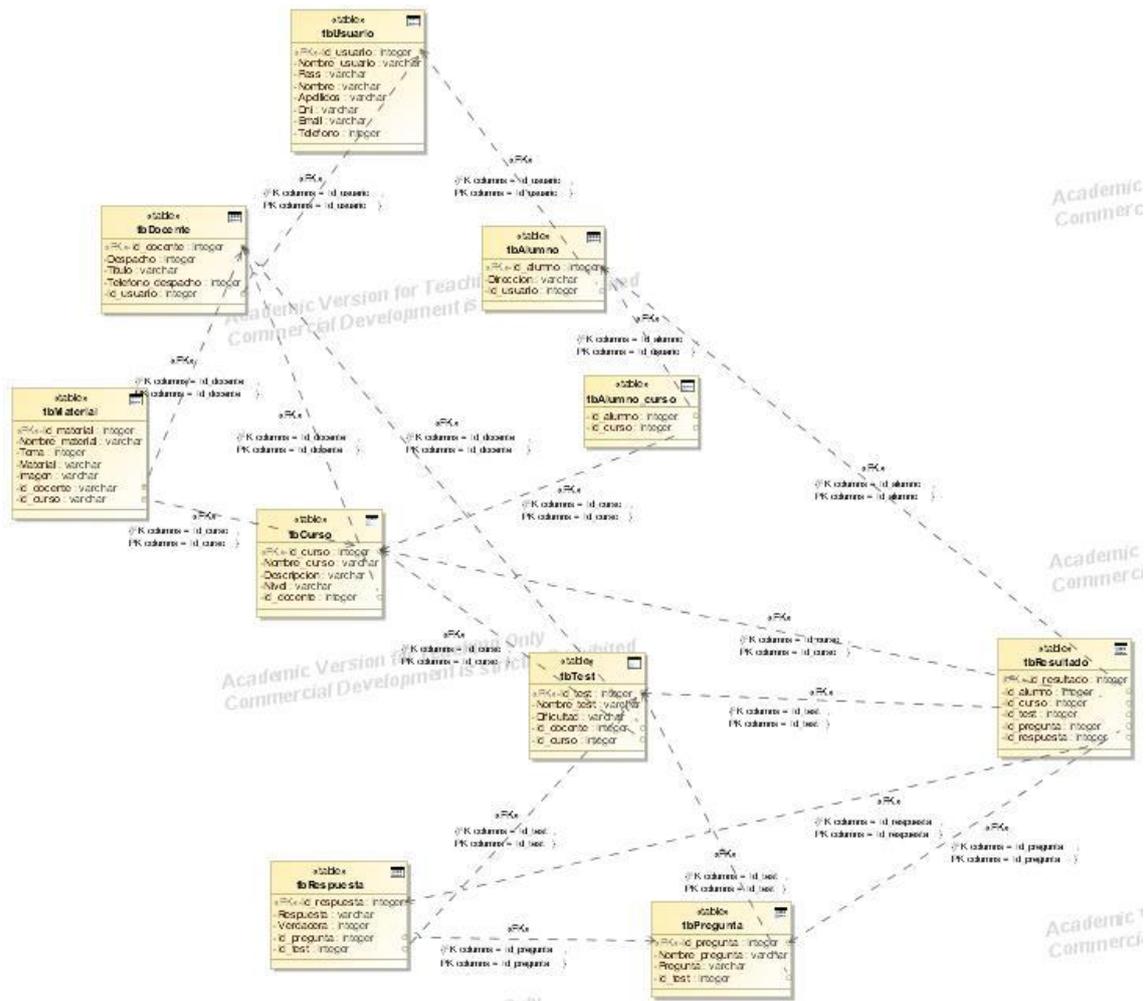


Ilustración 10. Diagrama de Modelo de Datos

Nombre:	Usuario
Descripción:	Tabla que almacena toda la información relacionada con el usuario
Columnas:	<ul style="list-style-type: none"> ▪ Id_usuario - Clave primaria (PK) de la tabla. ▪ Nombre_usuario - Nombre que el usuario usa para acceder a su cuenta. ▪ Pass – Clave del usuario. ▪ Nombre – Nombre personal del usuario. ▪ Apellidos – Apellidos del usuario.

<ul style="list-style-type: none"> ▪ DNI – Documento nacional de identidad del usuario. ▪ Email – Email del usuario. ▪ Teléfono – Teléfono de contacto del usuario.

Nombre:	Docente
Descripción:	Tabla que extiende información extra del usuario para completar la información del docente
Columnas:	
<ul style="list-style-type: none"> ▪ Id_docente – Clave primaria (PK) de la tabla. ▪ Despacho – Número de despacho. ▪ Título – Título académico que dispone el docente. ▪ Telefono_despacho – Teléfono de contacto del despacho del docente. ▪ Id_usuario – Clave foránea (FK) a la clave primaria de la tabla Usuario. 	

Nombre:	Alumno
Descripción:	Tabla que extiende información extra del usuario para completar la información del alumno
Columnas:	
<ul style="list-style-type: none"> ▪ Id_alumno - Clave primaria (PK) de la tabla. ▪ Direccion – Dirección postal del alumno. ▪ Id_usuario - Clave foránea (FK) a la clave primaria de la tabla Usuario. 	

Nombre:	Curso
Descripción:	Tabla que almacena la información de los cursos
Columnas:	
<ul style="list-style-type: none"> ▪ Id_curso – Clave primaria (PK) de la tabla. ▪ Nombre_curso – Nombre del curso. 	

<ul style="list-style-type: none"> ▪ Descripcion – Descripción básica de lo que consta el curso. ▪ Nivel – Nivel académico del curso. ▪ Id_docente – Clave foránea (FK) a la clave primaria de la tabla Docente.

Nombre:	Alumno_curso
Descripción:	Tabla que relaciona la información del alumno con la del curso
Columnas:	
<ul style="list-style-type: none"> • Id_alumno – Clave foránea (FK) a la clave primaria de la tabla Alumno. • Id_curso – Clave foránea (FK) a la clave primaria de la tabla Curso. 	

Nombre:	Material
Descripción:	Tabla que almacena la información del material de aprendizaje
Columnas:	
<ul style="list-style-type: none"> • Id_material – Clave primaria (PK) de la tabla. • Nombre_material – Nombre del material. • Tema – Número de material. • Material – Texto con todo el material. • Imagen – Nombre de la imagen que incluye el material. • Id_docente – Clave foránea (FK) a la clave primaria de la tabla Docente. • Id_curso – Clave foránea (FK) a la clave primaria de la tabla Curso. 	

Nombre:	Test
Descripción:	Tabla que almacena la información de los test
Columnas:	
<ul style="list-style-type: none"> • Id_test – Clave primaria (PK) de la tabla. • Nombre_test – Nombre del test. 	

- **Dificultad** – Dificultad del test.
- **Id_docente** – Clave foránea (FK) a la clave primaria de la tabla **Docente**.
- **Id_curso** – Clave foránea (FK) a la clave primaria de la tabla **Curso**.

Nombre:	Pregunta
Descripción:	Tabla que almacena la información de las preguntas de los test
Columnas:	
<ul style="list-style-type: none"> • Id_pregunta – Clave primaria (PK) de la tabla. • Nombre_pregunta – Nombre de la pregunta. • Pregunta – Pregunta correspondiente a un test. • Id_test – Clave foránea (FK) a la clave primaria de la tabla Test. 	

Nombre:	Respuesta
Descripción:	Tabla que almacena la información de las respuestas a las preguntas de los test
Columnas:	
<ul style="list-style-type: none"> • Id_respuesta – Clave primaria (PK) de la tabla. • Respuesta – Respuesta correspondiente a una pregunta. • Verdadera – Indica si la respuesta es verdadera o falsa (1 = verdadera; 0 = falsa). • Id_pregunta – Clave foránea (FK) a la clave primaria de la tabla Pregunta. • Id_test – Clave foránea (FK) a la clave primaria de la tabla Test. 	

Nombre:	Resultado
Descripción:	Tabla que almacena la información de los resultados a los test realizados por los alumnos
Columnas:	

- **Id_resultado** – Clave primaria (PK) de la tabla.
- **Id_alumno** – Clave foránea (FK) a la clave primaria de la tabla **Alumno**.
- **Id_curso** – Clave foránea (FK) a la clave primaria de la tabla **Curso**.
- **Id_test** – Clave foránea (FK) a la clave primaria de la tabla **Test**.
- **Id_pregunta** – Clave foránea (FK) a la clave primaria de la tabla **Pregunta**.
- **Id_respuesta** – Clave foránea (FK) a la clave primaria de la tabla **Respuesta**.

4.2. Arquitectura de Desarrollo

El diseño de la arquitectura va a tener una fuerte orientación para la separación de lógicas de negocio, servicios y capas de presentación, con objeto de permitir un alto nivel de escalabilidad y desacoplamiento.

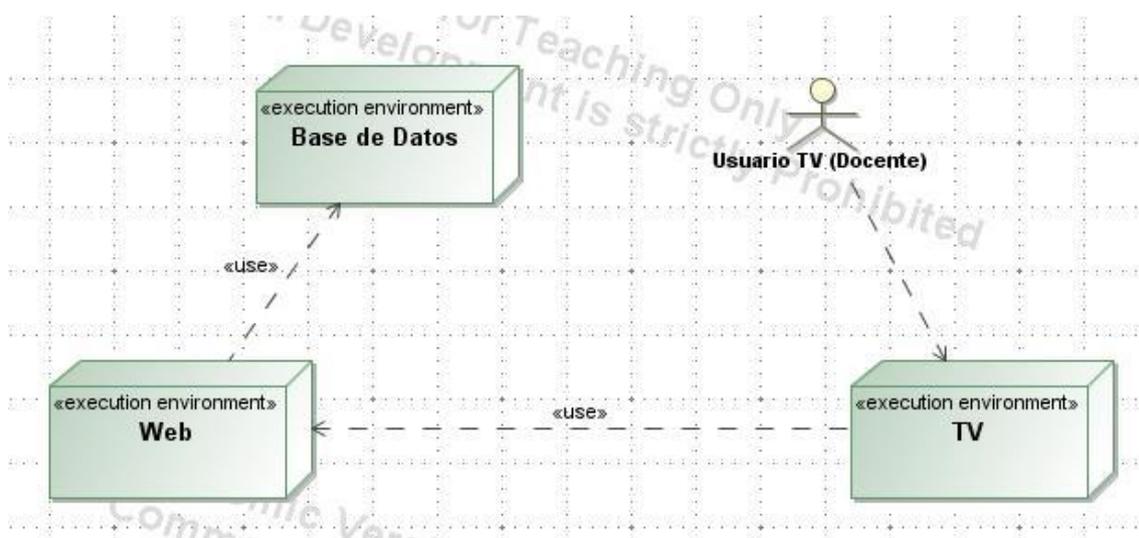


Ilustración 11. Modelo Lógico de la Arquitectura

A continuación se comentan los aspectos más destacados de cada una de las dos arquitecturas.

4.2.1. Arquitectura de Desarrollo Web

El diseño de la arquitectura Web está basado en un diseño multicapa usando la abstracción en el acceso a los datos:

- **Capa de Presentación** – Proporciona el contexto de presentación dentro de un navegador que permite acceder a los datos remotos a

través de una capa de presentación Web. Para la implementación de esta capa se ha utilizado JSP.

- **Capa de Web Services y Servlets** – Esta capa se sitúa al mismo nivel que la de presentación y encima de la de servicios. Mediante ésta, la aplicación publicará una serie de Servicios Web y Servlets con objeto de que cierta funcionalidad pueda ser accedida de manera remota por la Aplicación de Televisión.
- **Capa de Servicios** – Engloba el conjunto de clases e interfaces que resuelven la funcionalidad del sistema (lógica de negocio).
- **Capa de Integración** – Engloba las clases que mapean las tablas de la Base de Datos como objetos para realizar la abstracción y facilitar la manipulación de los datos de una forma mucho más sencilla.
- **Capa de Persistencia** – Engloba la comunicación con la Base de Datos utilizando JDBC.

En la Ilustración 12, podemos ver la arquitectura seguida para la implementación Web.



Ilustración 12. Arquitectura de Desarrollo Web

4.2.2. Arquitectura de Desarrollo de Televisión

Para el desarrollo de la Aplicación de Televisión se ha usado el SDK Smart TV que proporciona Samsung y que se ha comentado en el capítulo 2. Por tanto se ha utilizado la arquitectura que proporciona dicha plataforma de Samsung y que básicamente se compone de la aplicación que se ejecuta por encima de una capa llamada Application Manager y que ésta a su vez es ejecutada sobre el navegador de la televisión llamado Maple Browser.

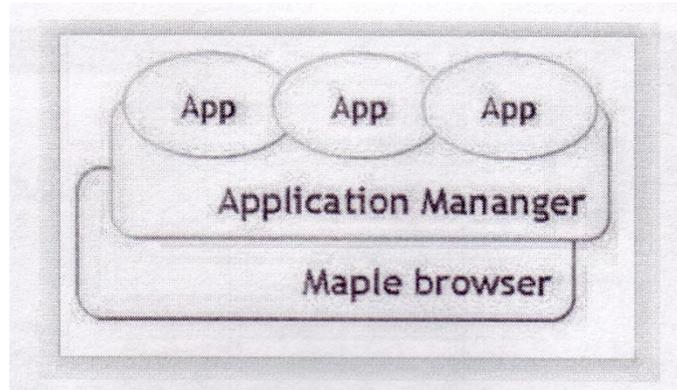


Ilustración 13. Arquitectura Samsung Smart TV

5. Implementación

En este capítulo se presenta la implementación resultante de los distintos componentes necesarios para crear el trabajo propuesto.

5.1. Implementación Web

En este apartado vamos a ver con más detalle cómo se ha implementado el sistema Web y la estructura del código.

5.1.1. Descripción Web

El sistema Web ha sido desarrollado usando MyEclipse Java Enterprise. Para la realización se ha creado un proyecto empresarial que se compone de dos módulos, un módulo web donde se ubica la propia página web (WebRoot) y un módulo donde se ubica parte de la lógica de negocio y la capa de persistencia (src/modelo).

En la Ilustración 14 podemos observar el entorno de desarrollo y podemos diferenciar claramente cada uno de los componentes comentados.



Ilustración 14. Descripción Web

5.1.2. Módulo src/modelo

El módulo SRC/modelo es el encargado de almacenar parte de la lógica de negocio y la capa de persistencia de todo el sistema.

Contiene las entidades que encapsulan las tablas de la Base de Datos siguiendo el modelo visto en la sección 4.1, como podemos observar en la Ilustración 15.

Existen dos tipos de archivos en este módulo los VO (Value Object) y los DAO (Data Access Object).

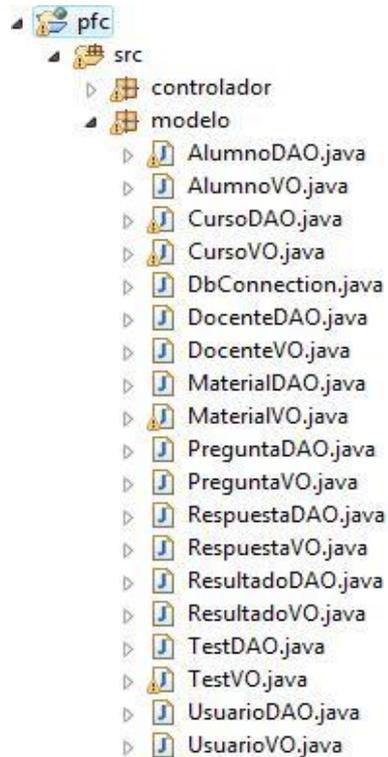


Ilustración 15. Módulo Modelo

- **Los archivos VO** - Son utilizados por DAO para transportar los datos desde la Base de Datos hacia la capa lógica de negocio y viceversa. Es una clase Java (Java Bean) que contiene:
 - Los atributos que contiene esa clase.
 - Se establecen los métodos get y set para acceder y modificar los atributos.
 - Permiten comunicar con otros JavaBeans.
- **Los archivos DAO** - es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de Datos o un archivo.

```

package modelo;

public class TestVO {

    private Integer idTest;
    private String nombreTest;
    private String dificultad;

    public Integer getIdTest() {
        return idTest;
    }

    public void setIdTest(Integer idTest) {
        this.idTest = idTest;
    }

    public String getNombreTest() {
        return nombreTest;
    }

    public void setNombreTest(String nombreTest) {
        this.nombreTest = nombreTest;
    }

    public String getDificultad() {
        return dificultad;
    }

    public void setDificultad(String dificultad) {
        this.dificultad = dificultad;
    }
}

```

Ilustración 16. Archivo VO

5.1.3. Diseño Web

Para el diseño Web se ha utilizado la tecnología JSP, acompañada de las herramientas habituales para el desarrollo Web.

En la Ilustración 17, podemos ver las clases que se han usado para el diseño Web. Como podemos observar, la Web está formada por diferentes clases .jsp, tales como la pantalla de inicio (Index), la de visualizar alumnos, la de visualizar cursos y test, etc. Los nombres de los ficheros .jsp indican que se realiza en cada clase para que sea más fácil su reconocimiento y mantenimiento. Dentro de cada .jsp se incluye un apartado que engloba el CSS dentro de estos comentarios, `<style type="text/css"> </style>`.

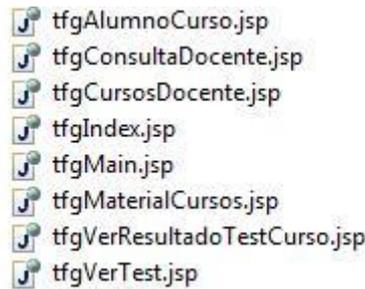


Ilustración 17. Diseño Web

5.1.4. Servlets

Las comunicaciones de todo el sistema pasan a través de los Servlets. Son usados por la Web para realizar la redirección de páginas como de igual manera por la Aplicación de Televisión.

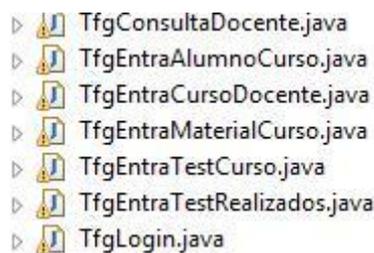


Ilustración 18. Servlets

En la Ilustración 18, podemos observar las clases que albergan a los Servlets. Como se puede ver se ha hecho una diferenciación en los nombres de los Servlets, para saber que se realiza en cada uno y que de este modo sea más fácil su reconocimiento y mantenimiento.

Los Servlet usan el método doPost(), para que no se muestren los datos en la URL. Usamos el Interface RequestDispatcher, que define un objeto que recibe peticiones y las reenvía a otro recurso:

- Servlet
- Jsp
- HTML

A continuación vamos a explicar algunas de las funciones más usadas en los Servlets:

- Usamos la variable request para obtener la Session del jsp, request.getSession().getAttribute().

- Para obtener los diferentes parámetros usamos `request.getParameter()`.
- `getServletContext().getRequestDispatcher()` - indicamos cual será el siguiente Servlet, jsp o HTML.
- `sesion.setAttribute()` – enviamos los atributos al siguiente recurso.
- `dispatcher.forward(request, response)` – cuando el Servlet lee esta línea redirecciona a la dirección indicada anteriormente.
- `sesion.removeAttribute()` – sirve para borrar atributos de la sesión.

5.2. Implementación Televisión

En este apartado se describe con más detalle cómo se ha implementado el sistema televisivo y la organización del código que se ha realizado.

5.2.1. Descripción Sistema Televisivo

El sistema televisivo se ha desarrollado con el SDK que ofrece Samsung para la realización de los proyectos para su Smart TV. El SDK ofrece una herramienta para la emulación de las aplicaciones.

Los proyectos desarrollados para la Smart TV de Samsung son aplicaciones basadas en Web, pero en lugar de necesitar de un servidor donde ubicar la aplicación lo que se hace es empaquetar el código dentro de una aplicación que es la que se ejecuta en la televisión.

Las aplicaciones son un tipo especial de páginas Web que se implementan sobre un navegador, el cual se ejecuta sobre la pantalla del televisor, como podemos observar en la Ilustración 19.

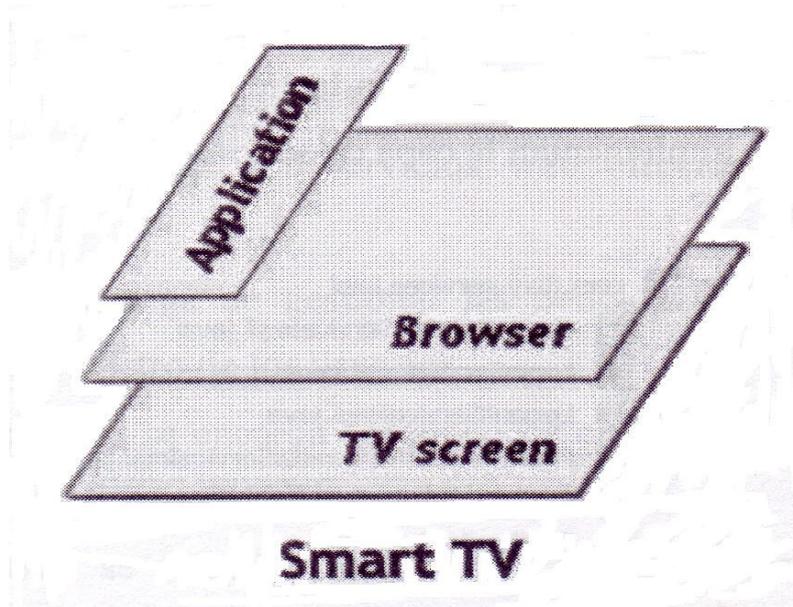


Ilustración 19. Funcionamiento Smart TV

5.2.2. Estructura del Código

La aplicación es una página Web que consiste en HTML, CSS, y JavaScript, y que se ejecuta en televisores Samsung. La página HTML muestra la estructura de la aplicación, el CSS se encarga del estilo, y el archivo JavaScript controla el comportamiento de la aplicación. Además de esto, para poder ejecutar la página en el televisor se necesita de un fichero de configuración “config.xml” que contiene información sobre el entorno operativo y la versión de la aplicación.

En la Ilustración 20, podemos ver un ejemplo de fichero de configuración.

```

<?xml version="1.0" encoding="UTF-8"?>
<widget>
  <previewjs>PreviewIME</previewjs>
  <type>user</type>
  <cpname itemtype="string"></cpname>
  <cplogo itemtype="string"></cplogo>
  <cpauthjs itemtype="string"></cpauthjs>
  <ThumbIcon itemtype="string">icon/sampleIcon_106_87.png</ThumbIcon>
  <BigThumbIcon itemtype="string">icon/sampleIcon_115_95.png</BigThumbIcon>
  <ListIcon itemtype="string">icon/sampleIcon_85_70.png</ListIcon>
  <BigListIcon itemtype="string">icon/sampleIcon_95_78.png</BigListIcon>
  <category itemtype="string"></category>
  <autoUpdate itemtype="boolean">n</autoUpdate>
  <ver itemtype="string">0.100</ver>
  <mgrver itemtype="string"></mgrver>
  <fullwidget itemtype="boolean">y</fullwidget>
  <type itemtype="string">user</type>
  <srcctl itemtype="boolean">n</srcctl>
  <ticker itemtype="boolean">n</ticker>
  <childlock itemtype="boolean">n</childlock>
  <videomute itemtype="boolean">n</videomute>
  <dcont itemtype="boolean">y</dcont>
  <widgetname itemtype="string">DefaultJSApps</widgetname>
  <description itemtype="string"></description>
  <width itemtype="string">960</width>
  <height itemtype="string">540</height>
  <mouse>y</mouse>
  <author itemtype="group">
    <name itemtype="string"></name>
    <email itemtype="string"></email>
    <link itemtype="string"></link>
    <organization itemtype="string"></organization>
  </author>
</widget>

```

Ilustración 20. Ejemplo config.xml

La estructura típica del código es similar a la que podemos ver en la Ilustración 21.

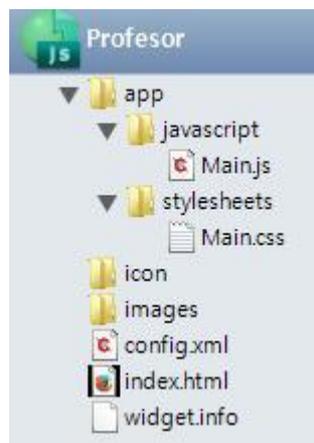


Ilustración 21. Estructura de Código Smart TV

El fichero “index.html” es el punto de acceso a la aplicación, por lo tanto es el encargado de cargar tanto el fichero de estilo como el fichero JavaScript que se necesitan.

En la Ilustración 22, podemos ver un ejemplo de un fichero de este tipo. Como podemos observar dentro de la etiqueta <head> se hacen referencia a los ficheros .js que se utilizan y al fichero de estilos “Main.css”. Los ficheros “Widget.js” y “TVKeyValue.js” siempre son necesarios, dado que uno nos permite que la página se ejecute en la aplicación y otro que se pueda utilizar el control remoto del televisor respectivamente.

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <script src="http://code.jquery.com/jquery-1.5.2.min.js" type="text/javascript"></script>
    <title>mando</title>

    <!-- TODO : Common API -->
    <script language="javascript" type="text/javascript" src="$MANAGER_WIDGET/Common/API/Widget.js"></script>
    <script language="javascript" type="text/javascript" src="$MANAGER_WIDGET/Common/API/Plugin.js"></script>
    <script language="javascript" type="text/javascript" src="$MANAGER_WIDGET/Common/API/TVKeyValue.js"></script>
    <script language="javascript" type="text/javascript" src="$MANAGER_WIDGET/Common/Util/Include.js"></script>
    <script language="javascript" type="text/javascript" src="$MANAGER_WIDGET/Common/Util/Language.js"></script>
    <script language="javascript" type="text/javascript" src="$MANAGER_WIDGET/Common/Plugin/Define.js"></script>
    <!-- TODO : Javascript code -->
    <script language="javascript" type="text/javascript" src="app/javascript/Main.js"></script>

    <!-- TODO : Style sheets code -->
    <link rel="stylesheet" href="app/stylesheets/Main.css" type="text/css">

    <!-- TODO: Plugins -->
  </head>

  <body onload="Main.onLoad();" onunload="Main.onUnload();">
    <div id="welcome">Bienvenido al mundo de Samsung Smart TV!</div>
    <a href="javascript:void(0);" id='anchor' onkeydown='Main.keyDown();'></a>
  </body>
</html>
```

Ilustración 22. Ejemplo de index.html

6. Conclusiones

Uno de los últimos puntos de esta memoria debe hacer referencia a las conclusiones y a las posibles mejoras que se observan una vez desarrollada la plataforma, así como los problemas encontrados durante el desarrollo del mismo.

Este trabajo nació como ampliación a mi Proyecto Fin de Carrera. Ambos trabajos tuvieron la intención de facilitar a estudiantes la realización de diferentes cursos a través de la televisión, y a los docentes el seguimiento de dichos estudiantes en sus cursos a través de la televisión.

En primer lugar se van a describir los problemas encontrados, posteriormente se procederá a presentar un conjunto de conclusiones y en base a ellas, se detallarán un conjunto de posibles mejoras.

6.1. Problemas Encontrados

Los problemas derivados de la implementación fueron sobre todo, con el SDK de Samsung, por la dificultad de encontrar información al respecto. También con la Web ya que anteriormente a este trabajo, no había investigado mucho sobre ese entorno.

6.2. Conclusiones

Respecto a la parte personal, el desarrollo del trabajo me ha servido para seguir aprendiendo nuevas tecnologías y formas de trabajo así como poder aprender otras metodologías. También es útil para poder darse cuenta lo que uno mismo puede hacer sin ayuda de otras personas y compañeros. Tener que investigar por cuenta propia sin ayuda de un temario, como en las asignaturas cursadas a lo largo de la carrera.

6.3. Limitaciones

Aunque consideramos que la aplicación ha quedado lo suficientemente completa como para poder ser usada con las herramientas implementadas hasta el momento, no estaría de más considerar algunas mejoras que podrían hacerse posteriormente a la realización del trabajo.

6.4. Líneas Futuras

Como líneas futuras de trabajo sería interesante considerar los siguientes puntos:

- Ampliar los tipos de dispositivos en los que se puede interactuar con la aplicación. Ahora mismo se utiliza entre la Web y la televisión, podría extenderse a una aplicación móvil, para dar mayor cobertura a nuestros usuarios.
- Darle más funcionalidad al control remoto de la televisión (mando a distancia).
- Introducir en los test algunas imágenes.

Apéndice A: Manual de Instalación

En este capítulo se describe el procedimiento a seguir por el usuario final del sistema, para la instalación y puesta en marcha inicial del mismo.

A.1. Requisitos Hardware

Como referencia se indican las características del equipo que se utilizó para el desarrollo:

- Procesador: Intel Core 2 Duo 2.26GHz
- RAM: 4GB

A.2. Requisitos Software

Antes de proceder a la instalación, ha de asegurarse que el servidor tenga instalado el siguiente software:

- Servidor Web con soporte para JSP 2.1 o superior y Servlet 2.5 o superior, en nuestro caso hemos usado Apache Tomcat 6.0.
- Servidor de Base de Datos con soporte SQL. En nuestro caso hemos usado un Servidor MySQL.
- JDK (Java Development Kit) 5 o superior.
- SDK de Samsung o un televisor marca Samsung que disponga de Smart TV, cualquiera de ellos conectado a Internet.

A.3. Instalación

En este apartado se describe que herramientas deben instalarse y cómo hacerlo para el correcto funcionamiento del sistema.

A.3.1. Instalación de la Base de Datos

Para llevar a cabo la instalación de la Base de Datos necesitamos:

- Archivo “proyecto.sql”, que contiene todo el código necesario para crear la Base de Datos que necesita la aplicación.

- Conocer el identificador y la clave de un usuario con privilegios en la Base de Datos donde se desea instalar la aplicación.

Hay dos formas de instalar la Base de Datos:

- Un método es a través de la propia herramienta gráfica de administración que posea la Base de Datos. En este caso procederíamos a importar el archivo “proyecto.sql”.
- Otro método es a través de la consola de comando donde escribiríamos la siguiente sentencia:

```
C:\>mysql -u administrador* -p < proyecto.sql
```

```
Enter password:
```

*Sustituyendo administrador por el nombre de usuario correspondiente.

A.3.2. Despliegue de la Aplicación

Una vez instalada la Base de Datos, procedemos a desplegar la Aplicación en el Servidor Web Tomcat que es el que hemos usado. El método más sencillo y rápido es copiar directamente dentro de la carpeta “tomcat/Webapps”. Una vez copiado el proyecto dentro de la carpeta, reiniciar Tomcat y la aplicación se desplegará automáticamente.

A.3.3. Despliegue de Smart TV

En la aplicación de Samsung Smart TV abrimos el proyecto llamado “Profesor”. A continuación, pulsamos el botón Run Emulador, de este modo se abrirá el emulador de la televisión.

A.3.4. Configurar las Opciones de la Aplicación

Para finalizar la instalación debemos fijar unos directorios o carpetas donde serán almacenadas las imágenes.

Las carpetas son:

- La carpeta imagen debe crearse en este directorio.

```
c:\\Users\\user\\Workspaces\\MyEclipse\\pfc\\WebRoot\\
```

- La carpeta imagen también debe crearse en el directorio donde se esté ejecutando el proyecto en Tomcat, el siguiente directorio sería en mi caso particular.

```
c:\\Users\\user\\Workspaces\\MyEclipse\\.metadata\\.me_tcat\\webapps\\pfc\\
```


Apéndice B: Manual de la Aplicación

A continuación se describe con detalle la funcionalidad y modo de uso de la aplicación de TV que se ha desarrollado.

B.1. Iniciar Sesión

Iniciamos sesión con nuestro usuario y contraseña: para iniciar sesión en la aplicación partimos de la pantalla de inicio, introducimos nuestro usuario y contraseña, y pulsamos el botón “Iniciar sesión” y accederemos a la pantalla principal de la aplicación.

Para ello hay que utilizar los cursores del mando del televisor (Arriba, Abajo, Izquierda, Derecha y Ok), e ir moviendo el foco por los distintos campos. Primero hay que situar el foco en el campo de texto de “Usuario” y pulsar el botón “Ok” del mando. Nos aparecerá un teclado en la pantalla con el que se podrá introducir texto en el campo seleccionado. Al acabar de introducir el texto hay que pulsar “Ok” para salir del teclado y que el foco vuelva a situarse en el campo de texto. Hay que repetir el mismo proceso para introducir la contraseña.

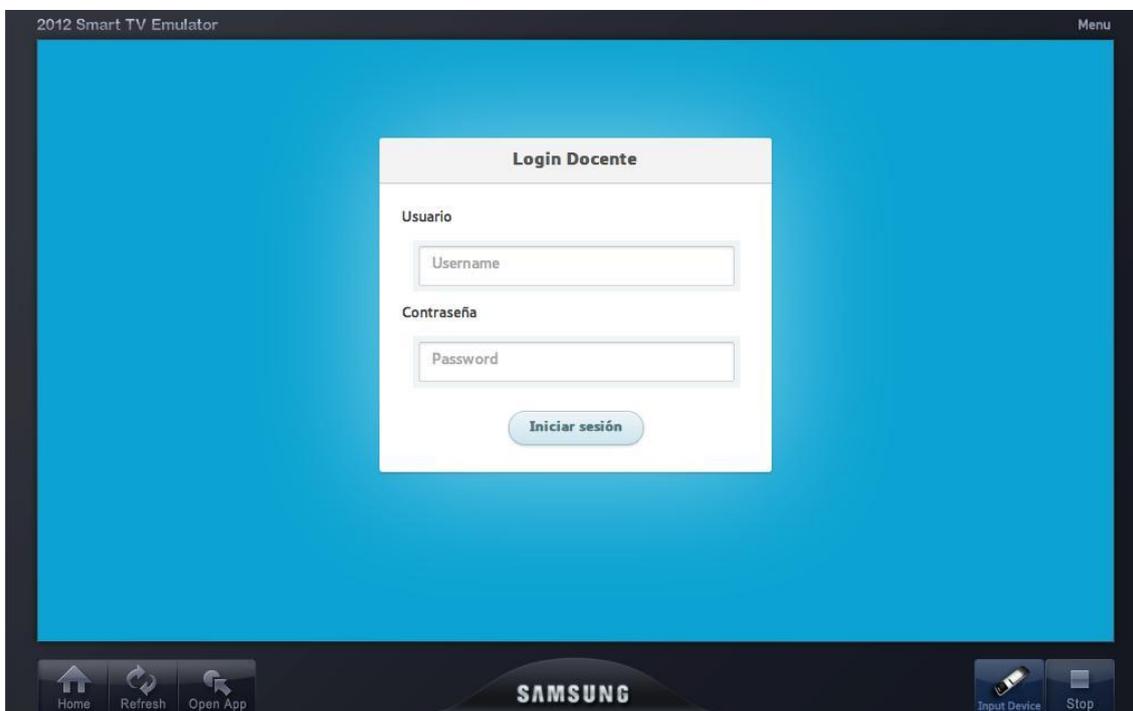


Ilustración 23. Pantalla de Inicio de Sesión

B.2. Pantalla Principal

Desde la pantalla Principal podemos elegir (en el desplegable) el curso del que queremos ver material de aprendizaje, test o algún alumno matriculado, además de consultar nuestros datos.

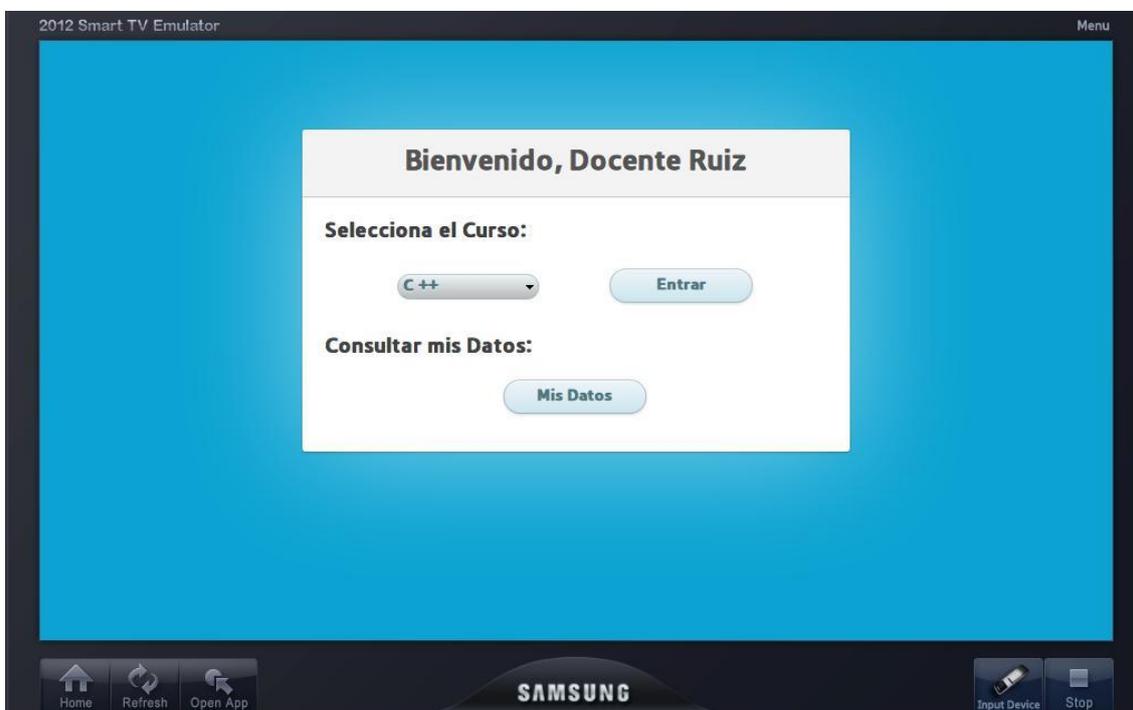


Ilustración 24. Pantalla Principal

B.3. Ver Mis Datos

Se accede desde la pantalla Principal, clicando en el botón “Mis Datos”. Nos permite consultar nuestros datos personales.

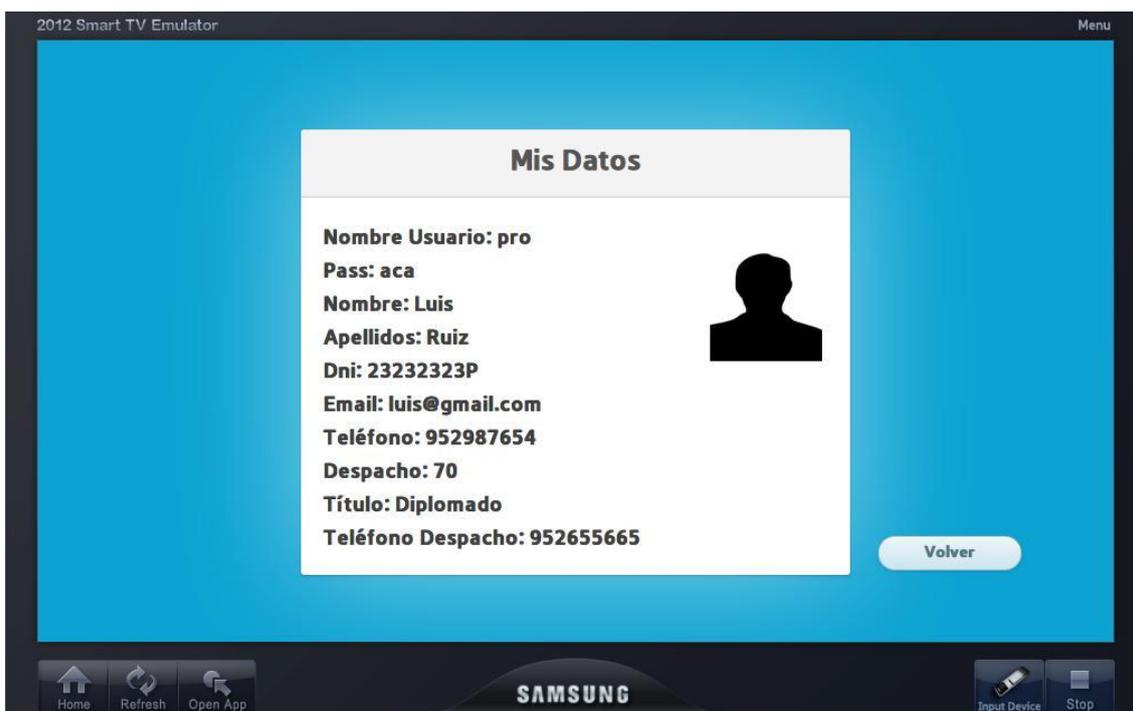


Ilustración 25. Pantalla Ver Mis Datos

B.4. Ver Curso

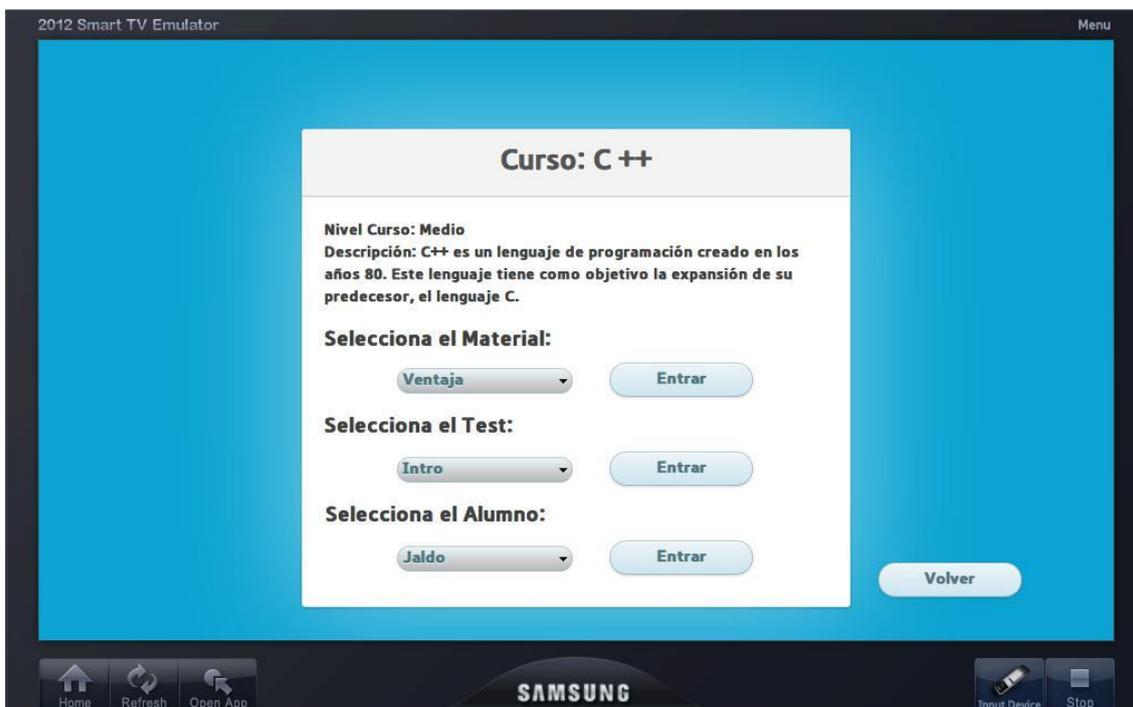


Ilustración 26. Pantalla Ver Curso

Se accede desde la pantalla Principal, eligiendo el curso que queremos ver y clicando en el botón “Entrar”. Ver un curso del que eres docente, nos permite

poder ver una pequeña descripción del curso y su dificultad, además ver el material que dispone ese curso, ver los test que dispone y ver los alumnos matriculados en ese curso.

B.5. Ver Material

Se accede desde la pantalla Ver Curso, eligiendo el material (en el desplegable) que deseamos visualizar, y clicando sobre el botón “Entrar”. Ver un material, nos permite poder ver el texto del material y si tiene incluida alguna imagen.



Ilustración 27. Pantalla Ver Material

B.6. Ver Test

Se accede desde la pantalla Ver Curso, eligiendo el test (en el desplegable) que deseamos consultar, y clicando sobre el botón “Entrar”. Ver un test nos muestra la dificultad del test, el nombre de la pregunta, la pregunta en sí, y las tres opciones de respuesta.

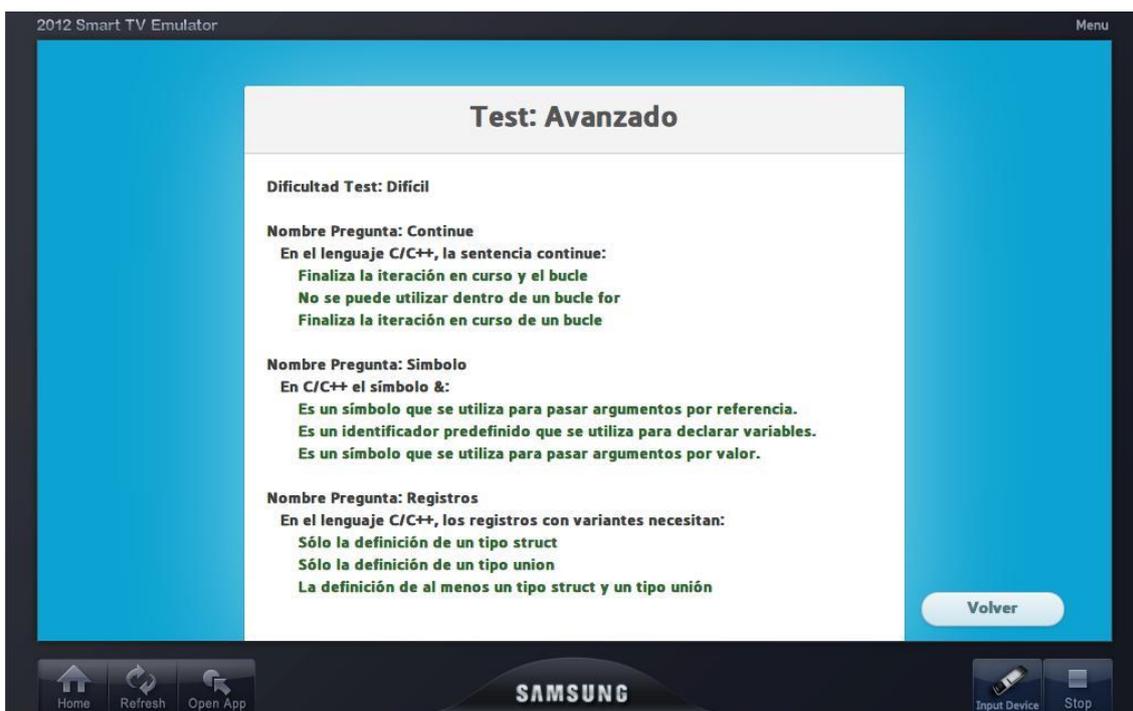


Ilustración 28. Pantalla Ver Test

B.7. Ver Alumno



Ilustración 29. Pantalla Ver Alumno

Se accede desde la pantalla Ver Curso, eligiendo el alumno (en el desplegable) que deseamos consultar, y clicando sobre el botón “Entrar”. Ver un alumno nos

muestra la información del alumno, y un desplegable con los test que ha realizado de ese curso.

B.8. Ver Test Realizados Por Alumno

Se accede desde la pantalla Ver Alumno, eligiendo el test (en el desplegable) que deseamos consultar las respuestas que ha realizado el alumno previamente, y clicando sobre el botón “Entrar”. Ver un test realizado por un alumno nos muestra las veces que ha realizado el test, y las respuestas a cada pregunta que respondió.

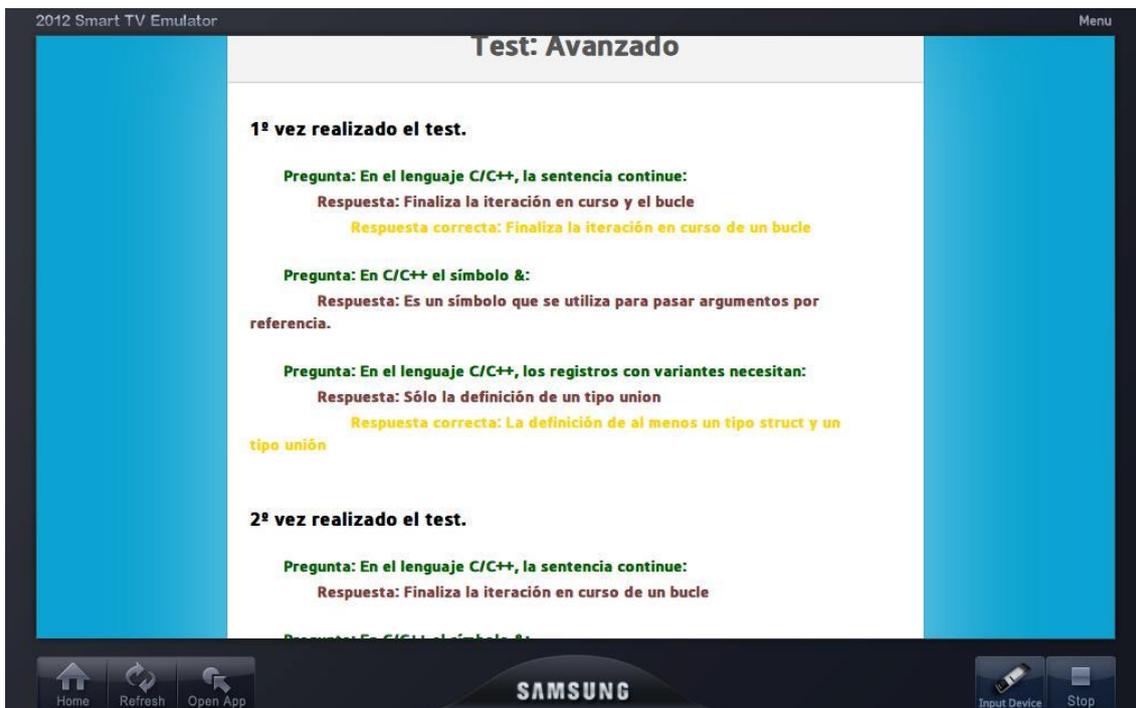


Ilustración 30. Pantalla Ver Test Realizados por Alumno

Bibliografía

- Wikipedia
 - <http://es.wikipedia.org>

- WampServer
 - <http://www.wampserver.com/en/>

- Samsung Smart TV SDK
 - <http://www.samsungdforum.com>

- Samsung TV Apps
 - <http://samsungstad.com/>

- Introducción a Java Server Pages
 - <http://www.edu4java.com/es/servlet/servlet6.html>

- Java Servlet Technology
 - <http://www.oracle.com/technetwork/java/index-jsp-135475.html>

- Java Server Pages Technology
 - <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>

- MagicDraw UML
 - <http://www.nomagic.com/>

- MySQL
 - <https://dev.mysql.com/doc/refman/5.0/es/>

Índice de Ilustraciones

Ilustración 1. Ciclo de Vida de un Servlet	10
Ilustración 2. Composición de una Aplicación Samsung Smart TV	14
Ilustración 3. Samsung Smart TV SDK.....	15
Ilustración 4. Diagrama de Requisitos Funcionales.....	18
Ilustración 5. Casos de Uso.....	20
Ilustración 6. Matriz de Trazabilidad.....	26
Ilustración 7. Diagrama de Entidades.....	27
Ilustración 8. Diagrama de Secuencia CU01	28
Ilustración 9. Diagrama de Secuencia CU13	28
Ilustración 10. Diagrama de Modelo de Datos	30
Ilustración 11. Modelo Lógico de la Arquitectura.....	34
Ilustración 12. Arquitectura de Desarrollo Web.....	35
Ilustración 13. Arquitectura Samsung Smart TV	36
Ilustración 14. Descripción Web	37
Ilustración 15. Módulo Modelo	38
Ilustración 16. Archivo VO.....	39
Ilustración 17. Diseño Web.....	40
Ilustración 18. Servlets	40
Ilustración 19. Funcionamiento Smart TV	42
Ilustración 20. Ejemplo config.xml	43
Ilustración 21. Estructura de Código Smart TV	43
Ilustración 22. Ejemplo de index.html	44
Ilustración 23. Pantalla de Inicio de Sesión	51
Ilustración 24. Pantalla Principal	52
Ilustración 25. Pantalla Ver Mis Datos	53
Ilustración 26. Pantalla Ver Curso	53
Ilustración 27. Pantalla Ver Material	54
Ilustración 28. Pantalla Ver Test	55
Ilustración 29. Pantalla Ver Alumno	55
Ilustración 30. Pantalla Ver Test Realizados por Alumno	56