



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Escuela Técnica Superior de Ingeniería Informática



UNIVERSIDAD POLITÉCNICA DE VALENCIA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

DESPLIEGUE DE UNA ESTRATEGIA INTEGRAL DE T-LEARNING BASADA EN LA
INTERACTIVIDAD A TRAVÉS DE LA TDT

PROYECTO FINAL DE CARRERA

Autor

JOAQUÍN MARQUÉS ALÓS

Tutor

Jaime Lozano Calleja

SETIVAL SCV

Director

Juan Carlos Ruiz García

Departamento de Informática de Sistemas y Computadores

Curso Académico 2009-2010

Resumen

La televisión es el medio de comunicación más utilizado por la mayoría de las personas debido a su comodidad en la forma de presentar los contenidos. Es por ello por lo que hay que focalizar el interés en este medio, avanzando y presentado nuevas ideas que permitan conducir a una mejor calidad de vida.

Esto es posible con la llegada de la tecnología TDT (televisión digital terrestre), dado que ésta nos ofrece las herramientas para poder explotar y exprimir al máximo el televisor.

Este proyecto se centra en incorporar un cierto grado de interactividad entre el usuario y el televisor, permitiendo al usuario enriquecerse más allá de los propios contenidos que emiten los programas de la cadena; presentando, de una forma no intrusiva, objetos de aprendizaje en forma de bloques de teoría y preguntas, relacionadas con la temática del programa que se está emitiendo en esos momentos.

Posteriormente, se utiliza un canal de retorno que incorpora el propio decodificador para conseguir que, la propia cadena de televisión, recoja la información que haya sido enviada por el usuario, para poder ser procesada y obtener ciertos resultados.

ÍNDICE GENERAL

Capítulo 1. Introducción	15
1.1 Objetivos del proyecto	17
1.2 Contexto en el que se enmarca el proyecto	18
1.3 Estructura del documento	18
Capítulo 2. Ámbito del proyecto	20
2.1 T-learning. Desarrollo de cursos de aprendizaje para la televisión interactiva ...	20
2.2 Herramientas de autor para la generación de objetos de aprendizaje.....	21
2.3 Aplicaciones interactivas para la TDT	27
2.3.1 Empresas y grupos con desarrollos para la TDT	27
2.3.1.1 Laboratorio de TVDi de la Universidad de Vigo.....	28
2.3.1.2 Atos-Origin.....	28
2.3.1.3 SETIVAL SCV	30
2.3.2 Aplicaciones interactivas en las cadenas de televisión.....	32
2.4 Conclusiones	35
Capítulo 3. Análisis y planificación.....	37
3.1 Casos de uso.....	37
3.2 Análisis de requerimientos	43
3.3 Arquitectura del sistema.....	45
3.4 Planificación	46
3.4.1 Descomposición en fases y tareas del proyecto	47
3.4.2 Métodos de estimación del esfuerzo [16]	48
3.4.3 Estimación del esfuerzo de cada tarea	49
3.4.4 Planificación temporal. Diagrama de Gantt y diagrama de precedencias.....	51
Capítulo 4. Técnicas y herramientas a utilizar	54
4.1 Ciclo de vida considerado	54
4.2 Herramientas a utilizar para el desarrollo de la arquitectura	55
4.2.1 Generación de objetos de aprendizaje	55
4.2.1.1 XML y IMS QTI	55

4.2.2 Distribución de objetos de aprendizaje	56
4.2.3 Consumo de objetos de aprendizaje.....	57
4.2.3.1 El estándar MHP	57
4.2.3.2 Programación con Xlets.....	57
4.2.4 Seguimiento	58
4.2.4.1 Canal de retorno en la TDT.....	59
4.2.4.2 Tecnologías disponibles para el canal de retorno.....	59
4.3 Conclusiones	60
Capítulo 5. Desarrollo del trabajo	61
5.1 Generación de paquetes de contenido	61
5.1.1 Base de datos de objetos de aprendizaje	66
5.2 Creación de un curso	69
5.2.1 Creación de los recursos teóricos	70
5.2.2 Creación de los recursos prácticos.....	70
5.2.3 Creación del índice	72
5.2.4 Obtención de los recursos teóricos y prácticos para la creación de un curso.....	74
5.2.4.1 Extracción de palabras clave	76
5.2.4.2 Obtención de recursos teóricos y prácticos a partir de un número definido de palabras clave.....	78
5.2.4.2.1 Ponderación de las palabras clave.....	78
5.2.4.2.2 Búsqueda de objetos de aprendizaje en la base de datos	79
5.2.4.2.3 Establecer marcas de tiempo en un objeto de aprendizaje.....	80
5.3 Envío de la aplicación consumidora de los cursos al STB	80
5.3.1 Generación de un Transport Stream [13]	81
5.3.2 Multiplexación y envío de la aplicación al STB.....	82
5.4 Aplicación consumidora de los cursos.....	83
5.5 Creación aplicación web para mostrar los informes	90
Capítulo 6. Pruebas. Validación y verificación.....	95
6.1 Pruebas unitarias	97
6.1.1 Creación de los cursos.....	97
6.1.1.1 Generación de objetos de aprendizaje mediante la herramienta de autor	97

6.1.1.2 Generación de objetos de aprendizaje mediante la extracción de palabras clave	97
6.1.1.3 Creación de los cursos a partir de los objetos de aprendizaje.....	98
6.1.2 Distribución de objetos de aprendizaje	100
6.1.3 Consumo de objetos de aprendizaje.....	101
6.1.3.1 Canal de retorno.....	101
6.1.4 Seguimiento	102
6.2 Pruebas de integración	103
6.3 Pruebas de aceptación	103
6.4 Problemas y soluciones	110
6.5 Conclusiones	110
Capítulo 7. Conclusiones del proyecto	111
Bibliografía.....	114

ÍNDICE DE FIGURAS

Figura 1.1. Alfabetización digital en la población española.	15
Figura 2.1. Editor de contenidos de la herramienta de autor Vértice.	22
Figura 2.2. Generación de contenidos de la herramienta de autor Vértice.	23
Figura 2.3. Herramienta de autor AUTORE.	24
Figura 2.4. Editando una actividad de texto con JClic autor.	25
Figura 2.5. Menú principal de la herramienta de autor Reiven.	26
Figura 2.6. Generación de un recurso multicontenido en Reiven.....	26
Figura 2.7. Evolución del porcentaje de descodificadores vendidos con MHP sobre el total.....	27
Figura 2.8. Sistema de pago de tasas y tributos para el Excmo. Ayuntamiento de Alcázar de San Juan desarrollado por Atos-Origin.....	29
Figura 2.9. Servicio de búsqueda de empleo militar para el Ministerio de Defensa desarrollado por Atos-Origin.....	29
Figura 2.10. Servicio de búsqueda de empleo público para el Ministerio de Administraciones Públicas desarrollado por Atos-Origin.....	29
Figura 2.11. Servicio de observatorio de precios para el Ministerio de medio Ambiente y Medio Rural y Marino desarrollado por Atos-Origin.....	30
Figura 2.12. Servicio de alertas por ola de calor para el Ministerio de Sanidad y Consumo desarrollado por Atos-Origin.....	30
Figura 2.13. Imagen de una pregunta de la aplicación TDT-Reiven.....	31
Figura 2.14. Imagen de una ventana emergente de la aplicación TDT-Reiven.....	32
Figura 2.15. Imagen de la aplicación empleaT.	33
Figura 3.1. Diagrama de casos de uso del diseñador de contenidos.	37
Figura 3.2. Diagrama de casos de uso del operario.	38
Figura 3.3. Diagrama de casos de uso del espectador.	38
Figura 3.4. Arquitectura del sistema a desarrollar.	46
Figura 3.5. Método PERT.	49
Figura 3.6. Precedencia entre tareas.....	51
Figura 3.7. Diagrama de Gantt.	53
Figura 4.1. Modelo de ciclo de vida en cascada.....	54
Figura 4.2. Ejemplo de la especificación IMS-QTI.	56

Figura 4.3. Ciclo de vida de un Xlet.	58
Figura 5.1. Esquema de generación de un paquete de contenidos.....	62
Figura 5.2. Generación de un recurso multicontenido en Reiven.....	62
Figura 5.3. Generación de una pregunta en Reiven.....	63
Figura 5.4. Generación de un paquete de contenidos en Reiven.	65
Figura 5.5. Bloque de teoría de un paquete de contenidos en Reiven.	65
Figura 5.6. Bloque de ejercicios de un paquete de contenidos en Reiven.	66
Figura 5.7. Diagrama de clases de la Base de datos de objetos de aprendizaje.	66
Figura 5.8. Estructura de directorios y ficheros de un curso.	69
Figura 5.9. Fichero xml de ejemplo de un recurso teórico.....	70
Figura 5.10. Fichero xml de ejemplo de un recurso práctico con una pregunta simple.71	
Figura 5.11. Fichero xml de ejemplo del índice de un curso.....	72
Figura 5.12. Obtención de los recursos teóricos y prácticos para la creación de un curso.	76
Figura 5.13. Recorrido ineficiente de un conjunto de palabras.....	77
Figura 5.14. Recorrido eficiente, previa ordenación, de un conjunto. de palabras.....	77
Figura 5.15. Paquete TS.....	82
Figura 5.16. Generación, multiplexación y envío de un TS al STB.....	83
Figura 5.17. Visualización de un recurso teórico en la televisión.	84
Figura 5.18. Visualización de un recurso práctico del tipo relacionar elementos en la televisión.	85
Figura 5.19. Visualización de un recurso práctico del tipo menú desplegable con opciones en la televisión.	86
Figura 5.20. Visualización de un recurso práctico del tipo selección entre una lista de elementos en la televisión.....	87
Figura 5.21. Menú principal para la navegación por un curso en la televisión.....	88
Figura 5.22. Navegación por los recursos teóricos del curso en la televisión.....	89
Figura 5.23. navegación por los recursos prácticos del curso en la televisión.	90
Figura 5.24. Menú principal de la aplicación web.....	92
Figura 5.25. Imagen de la aplicación web que muestra el número de participantes totales.....	92
Figura 5.26. Imagen de la aplicación web que muestra el número de participantes en cada programa.....	93

Figura 5.27. Imagen de la aplicación web que muestra el número de programas en los que ha participado cada usuario.	93
Figura 5.28. Imagen de la aplicación web que muestra el número de preguntas contestadas por cada usuario.	94
Figura 6.1. Modelo en V.	97
Figura 6.2. Creación de un curso a partir de los recursos generados mediante la herramienta de autor.	99
Figura 6.3. Creación de un curso a partir de los recursos generados mediante el extractor de palabras clave.	100
Figura 6.4. Método de comprobación de utilización del canal de retorno.....	102

ÍNDICE DE TABLAS

Tabla 3.1. Caso de uso: Generar curso manualmente.	39
Tabla 3.2. Caso de uso: Obtener curso automáticamente.....	40
Tabla 3.3. Caso de uso: Extraer palabras clave.	40
Tabla 3.4. Caso de uso: Crear curso.	41
Tabla 3.5. Caso de uso: Enviar aplicación y curso.	41
Tabla 3.6. Caso de uso: Contestar pregunta.	42
Tabla 3.7. Caso de uso: Ver los recursos teóricos y realizar los recursos prácticos del curso.	43
Tabla 3.8. Caso de uso: Recibir informes.	43
Tabla 3.9. Resumen de requerimientos.	45
Tabla 3.10. Planificación temporal de las fases del proyecto.	478
Tabla 6.1. Prueba de aceptación: Generar curso manualmente.....	104
Tabla 6.2. Prueba de aceptación: Obtener curso automáticamente.....	105
Tabla 6.3. Prueba de aceptación: Enviar aplicación y curso.	106
Tabla 6.4. Prueba de aceptación: Contestar pregunta.....	107
Tabla 6.5. Prueba de aceptación: Ver los recursos teóricos y realizar los recursos prácticos del curso.....	108
Tabla 6.6. Prueba de aceptación: Ver informes.	110

Capítulo 1. Introducción

Actualmente la demanda creciente de formación tanto en el ámbito personal como en el laboral, exige la búsqueda de nuevos mecanismos para facilitar el acceso al conocimiento. Debido al escaso tiempo del que dispone la sociedad para la formación, las soluciones de e-learning¹ están sufriendo un importante auge.

La principal dificultad que limita el acceso al e-learning es la llamada "brecha digital" o, lo que es lo mismo, la distancia entre aquellos que están alfabetizados digitalmente y que por muchas razones no han podido "subirse al carro tecnológico" y aquellos que no están alfabetizados ni pueden hacerlo, por otro sin fin de razones. Este fenómeno limita el acceso a los sistemas de e-learning tradicionales y no permite garantizar un acceso universal a la formación online (tan solo el 44% de la población tiene conexión a internet en España).

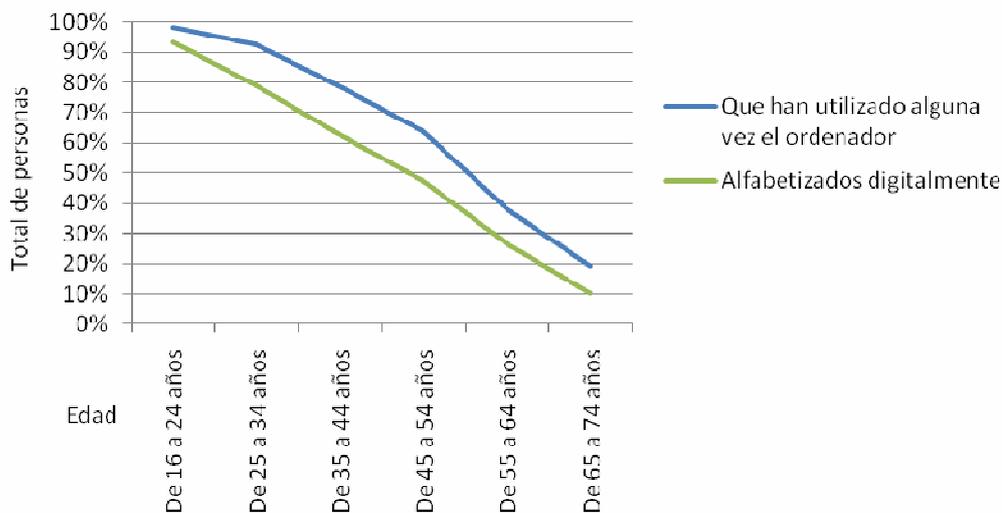


Figura 1.1. Alfabetización digital en la población española.

*Entendiendo como alfabetizados digitalmente a las personas capaces de realizar tareas sencillas como manejar ficheros o copiar y pegar datos.

Fuente: Instituto Nacional de Estadística

¹ *e-learning*: sistema de educación electrónico o a distancia en el que se integra el uso de las tecnologías de la información y otros elementos pedagógicos (didácticos) para la formación, capacitación y enseñanza de los usuarios o estudiantes en línea [1].

El objetivo del proyecto Conectra-TDT es el de utilizar la tecnología TDT para acercar la formación de manera general al mayor porcentaje de población posible. Sin embargo, la emisión de objetos de aprendizaje² educativos a través de televisión está limitada por los intereses de las cadenas emisoras y las guerras de audiencia.

Por otra parte la programación educativa o divulgativa, principalmente ofrecida por las televisiones públicas, ofrece un modelo de formación unidireccional. Siendo los espectadores meros receptores de información sin capacidad de interactuar con los contenidos que ofrece el programa.

El hecho de que la televisión se limite a ofrecer contenidos sin la participación del receptor tiene los días contados con la llegada de la tecnología TDT. Gracias a esta tecnología es posible desarrollar aplicaciones e incluirlas en la señal de la emisión para posteriormente ejecutarlas y visualizarlas en el televisor del usuario y, de esta forma, conseguir la interactividad a través de las aplicaciones.

Una vez hecho esto, es posible ir más allá mediante la conexión del decodificador de la señal de TDT a Internet y, con esto, conseguir el envío de información del espectador a la propia cadena de televisión, con las ventajas que esto podría conllevar.

Una de las partes importantes de este proyecto consiste en la generación de objetos de aprendizaje, es aquí donde entran en juego las llamadas herramientas de autor. Con estas herramientas es posible la creación de objetos de aprendizaje para luego ser utilizados en distintas plataformas.

Por otra parte, No es preciso que los objetos de aprendizaje que se ofrecen al usuario deban ser preparados de antemano, sino que podrían ser asignados automáticamente a partir de un repositorio de objetos de aprendizaje en función del contenido del programa que se esté emitiendo en ese momento y, por lo tanto, generando objetos de aprendizaje que se adecúen al contenido y temática del programa.

Una vez disponemos de los objetos de aprendizaje, necesitamos una solución que nos permita hacérselos llegar al espectador por medio de la televisión, recurriendo para ello a las aplicaciones interactivas para TDT. Estas aplicaciones constituyen una interfaz que permite mostrar por pantalla los distintos objetos de aprendizaje a través del decodificador de TDT donde se almacenan y ejecutan.

² *Objetos de aprendizaje*: Elementos para la instrucción, aprendizaje o enseñanza que pueden ser reutilizados en múltiples aplicaciones.

Todo lo comentado anteriormente, cambiaría radicalmente la forma de ver la televisión, en cuanto a que estaríamos ofreciendo herramientas que permitirían al espectador desarrollar la información que recibe más allá de los propios contenidos que emiten actualmente los programas de televisión, a la vez que se estaría aportando una forma de ocio para los usuarios.

Estos mecanismos permitirían la fidelización por parte de los espectadores a ciertos tipos de programas y, en consecuencia, el incremento de la audiencia con las ventajas que esto conlleva.

Esta tecnología abre un gran abanico de posibilidades en lo referente a aplicaciones que se pueden desarrollar para poder interactuar con el usuario y cambiar, de este modo, la figura pasiva que afronta el receptor actualmente frente al televisor.

Para poder llevar a cabo estas soluciones se hace necesario la utilización de un tipo de decodificador especial distinto a los decodificadores que poseen actualmente el gran porcentaje de la población en sus hogares.

Este decodificador incorpora el estándar MHP (Multimedia Home Platform) que permite la recepción y ejecución de aplicaciones interactivas en el televisor.

Los decodificadores de TDT que existen actualmente en el mercado, en su amplia mayoría, no cumplen con los requisitos que se necesitan para poder llevar a cabo todo lo citado anteriormente y, esto es debido, a la poca presencia de aplicaciones desarrolladas que resulten interesantes para el usuario.

1.1 Objetivos del proyecto

El proyecto, por tanto, dispone los siguientes objetivos fundamentales:

- Posibilitar el acceso al aprendizaje a todo el conjunto de la población a través de una reducción de costes considerable en relación con los métodos tradicionales, ya que los cursos podrán ser difundidos a través de plataformas que no supongan ningún coste adicional al ciudadano.
- Establecer una arquitectura de generación, emisión, uso y seguimiento de contenidos pedagógicos integrada a través de la TDT. Para ello se estudiarán las posibles fórmulas de interacción con el usuario útiles para el aprendizaje y que fomenten la participación. Además se desarrollarán estrategias de seguimiento del aprendizaje

de los alumnos-espectadores mediante el canal de retorno que ofrecen los últimos modelos de MHP (Multimedia Home Plattform).

- Investigar nuevas fórmulas de interactividad entre emisor de objetos de aprendizaje para la televisión y los espectadores mediante el diseño de nuevos modelos de interactividad con el usuario no intrusivos que permitan enriquecer los contenidos emitidos por TDT sin eclipsarlos.
- Asociar de forma automática o semiautomática objetos de aprendizaje educativos a la señal emitida a través de TDT en función de la descripción asociada al programa, extraída de la guía electrónica del programa, EPG (Electronic Program Guide), facilitando la capacidad de ofrecer objetos de aprendizaje educativos partiendo de un repositorio de objetos de aprendizaje.

1.2 Contexto en el que se enmarca el proyecto

Este trabajo ha sido financiado por la PYME valenciana SETIVAL SCV, a través de un convenio de prácticas de empresa firmado con la Universidad Politécnica de Valencia.

El interés de este proyecto por parte de SETIVAL SCV viene dado por los proyectos realizados anteriormente en la empresa relacionados con el aprendizaje a distancia a través de distintas plataformas como pueden ser los ordenadores o los teléfonos móviles y que, con la realización de este proyecto, se ampliaría este rango a una nueva plataforma como es la televisión.

1.3 Estructura del documento

Este documento se organiza conforme a las pautas que se muestran a continuación.

En el capítulo 2 realizaremos una introducción sobre las herramientas de autor para la generación de objetos de aprendizaje, así como las aplicaciones interactivas para la TDT, realizando un estudio sobre el estado en el que se encuentran estas herramientas y aplicaciones y presentando los distintos grupos y empresas que han llevado a cabo su desarrollo. Presentaremos también una solución existente similar a la que se va a desarrollar en el presente proyecto, que engloba tanto a las herramientas de autor, como a las aplicaciones interactivas, para la generación y presentación de objetos de aprendizaje.

El capítulo 3 introduce los requerimientos que debe incorporar el sistema junto con los casos de uso a desarrollar y la planificación del proyecto incluyendo las fases y tareas. Por último, se muestra la arquitectura que presentará el sistema con una breve explicación de cada uno de los módulos que lo conforman.

En el 4, tras presentar la arquitectura en el capítulo anterior, se describen las herramientas que se van a emplear para la construcción de los distintos módulos que conforman la arquitectura del sistema en base al modelo de proceso seleccionado.

En el capítulo 5 presentaremos los detalles técnicos de la solución propuesta: Conectra TDT. Centrándonos en los detalles sobre cómo ha transcurrido la implementación y cómo se han ido realizando las distintas tareas.

El capítulo 6 se centrará en las pruebas realizadas al sistema para comprobar su correcto funcionamiento.

Y para finalizar, En el capítulo 7 se mostrarán las conclusiones del proyecto, ofreciendo ideas de futuro que resultarían interesantes incorporar atendiendo a la línea seguida en este proyecto final de carrera.

Capítulo 2. Ámbito del proyecto

En este capítulo conoceremos algunas soluciones que existen tanto para la generación de objetos de aprendizaje mediante una herramienta de autor, como las aplicaciones interactivas para la TDT capaces de consumir estos objetos de aprendizaje, así como las soluciones existentes que engloban estos procedimientos.

Por último, realizaremos un estudio sobre las posibilidades de adaptación de estas herramientas a nuestro proyecto, centrándonos en nuestras necesidades.

2.1 T-learning³. Desarrollo de cursos de aprendizaje para la televisión interactiva

Para poder ofrecer formación al espectador a través de la televisión, es necesario el diseño de un sistema que permita generar contenidos y posteriormente permita visualizarlos, generando en consecuencia, lo que se denomina una solución integral de t-learning.

Un ejemplo de solución integral de t-learning es el proyecto ELU[19] (Enhanced Learning Unlimited) desarrollado en la universidad de Ljubljana, en Eslovenia.

Este proyecto aborda la idea, planteada en la introducción de este documento, de realizar cursos formativos para ser presentados a través de la televisión.

El objetivo del proyecto ELU consiste en investigar formas de aumentar el uso de la televisión para las actividades de aprendizaje a través de la adaptación y mejora de los métodos utilizados para e-learning a través del PC.

Para presentar al espectador los contenidos de los cursos, utilizan aplicaciones interactivas. Definen dos tipos de elementos de aprendizaje, estos son:

³ *t-learning*: aprendizaje interactivo y personalizado a través de la pantalla del televisor, gracias a la evolución tecnológica que se está produciendo en la televisión, la Televisión Digital Terrestre dispone de recursos tecnológicos que permitirán que los espectadores puedan acceder, a través del mando a distancia, a diferentes servicios educativos especialmente creados para este tipo de medio [5].

- Páginas multimedia
- Juegos

Las páginas multimedia tienen una funcionalidad de presentación. En ellas es posible incorporar, además de texto, elementos multimedia como pueden ser figuras, video o audio. El único factor restrictivo en términos de contenido es el tamaño de los elementos multimedia utilizados, debido a que los recursos de los STB⁴ son limitados.

Otro elemento de aprendizaje utilizado en el proyecto ELU son los juegos, con los que pretenden desarrollar las capacidades y promover el conocimiento entre una amplia audiencia. Son utilizados para realizar preguntas a los usuarios, dando lugar a la participación activa del espectador en el curso.

Se propone también la incorporación de un profesor virtual, implementado como una superposición en la secuencia de video o en los juegos y las páginas multimedia. Se trata de un instructor que ofrece cursos e instrucciones, responde a preguntas de los espectadores y proporciona retroalimentación y evaluaciones de curso. Es guiado por un sistema experto que le permite comportarse de manera autónoma, siendo capaz de interactuar con el espectador de una forma sensata y productiva.

Para la generación de contenidos hacen uso de una herramienta de autor desarrollada por el consorcio ELU, que se encarga de organizar los contenidos en base a una estructura de directorios.

2.2 Herramientas de autor para la generación de objetos de aprendizaje

Las herramientas de autor son programas que nos permiten crear de una manera sencilla nuestros propios materiales formativos. Sirven para crear objetos de aprendizaje, tutoriales, ejercicios...

Mediante estas herramientas podemos generar objetos de aprendizaje y empaquetarlos de forma que puedan ser consumidos en plataformas como los ordenadores y, últimamente, en los teléfonos móviles.

A continuación podemos observar algunas de estas herramientas facilitando algunas capturas para poder apreciar y entender mejor su funcionamiento:

⁴ *STB*: Las siglas provienen de SET-TOP-BOX , cuya traducción literal al español es aparato que se coloca encima del televisor, es el nombre con el que se conoce el dispositivo encargado de la recepción y opcionalmente decodificación de señal de televisión analógica o digital (DTV), para luego ser mostrada en un dispositivo de televisión [18].

▪ Vértice

Se trata de una herramienta de autor que permite generar objetos de aprendizaje multimedia, cursos, tutoriales, ...

Consiste en una herramienta de pago y presenta las siguientes características:

- Dispone de un sencillo editor de páginas
- Permite el uso de plantillas
- Incorpora un gestor multimedia
- Es compatible con las especificaciones scorm 1.2
- Facilita la personalización gráfica
- Es posible exportar el resultado final a cualquier plataforma o sitio web

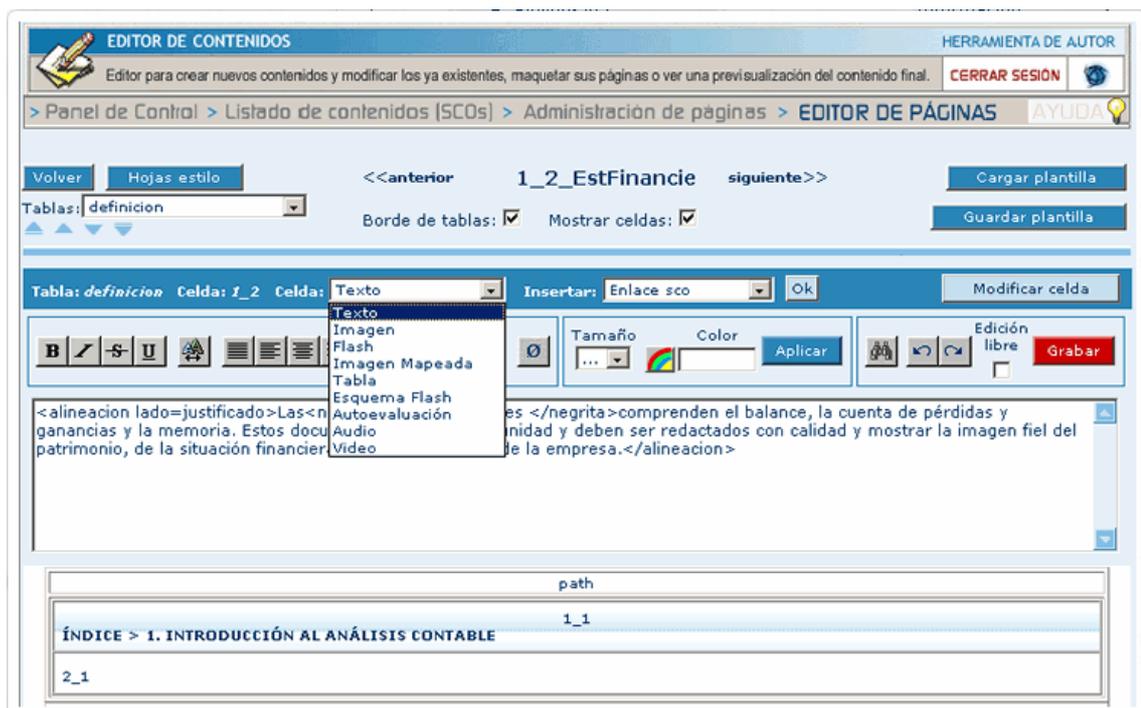


Figura 2.1. Editor de contenidos de la herramienta de autor Vértice.

GENERACIÓN DE CONTENIDOS HERRAMIENTA DE AUTOR
 Creación de contenidos SCORM mediante la agrupación de diferentes SCO's. CERRAR SESIÓN

Nuevo Contenido Importar metadata

Seleccione el tipo de contenido:

General

Título

Descripción

Palabras clave (separadas por espacios)

MetaMetadata

Esquema del metadata

Technical

Formato (Seleccione todos los archivos incluidos en el SCO)

Rights

¿Contenido de Pago?

Copyright y restricciones

Guardar Cambios

Figura 2.2. Generación de contenidos de la herramienta de autor Vértice.

- **AUTORE**

Diseñada y construida en el Campus Virtual de la Universidad del País Vasco, herramienta de código abierto que permite ser utilizada por el usuario a través de un navegador.

Autore surge de la necesidad de crear materiales docentes de e-learning para su utilización en entornos virtuales de aprendizaje (como el Campus Virtual de la UPV-EHU).

	titulo	tamaño	modificado	estado	orden
<input type="checkbox"/>	Virtual Educa 2006		2006-06-19 08:56:12	privado	▲ ▼
<input type="checkbox"/>	HAUR JOLASTOKIAK		2008-04-09 18:29:01	concurso	▲ ▼
<input type="checkbox"/>	Administración en el Perú	5.6 MB	2010-03-06 07:25:49	privado	▲ ▼
<input type="checkbox"/>	Definición de objeto de aprendizaje	1.5 MB	2009-08-07 21:19:40	privado	▲ ▼
<input type="checkbox"/>	Tutoriales		2006-10-10 16:20:01	privado	▲ ▼
<input type="checkbox"/>	atfolder.2007-04-10.0978174532		2007-04-10 16:58:17	privado	▲ ▼
<input type="checkbox"/>	Educacion Virtual		2007-04-10 21:44:50	privado	▲ ▼
<input type="checkbox"/>	atfolder.2007-04-10.1111878912		2007-04-10 16:58:31	privado	▲ ▼
<input type="checkbox"/>	Prueba ccm		2007-04-19 12:01:27	privado	▲ ▼
<input type="checkbox"/>	Online Educa 2007		2007-05-04 12:54:02	privado	▲ ▼
<input type="checkbox"/>	Metodos Cuantitativos de Investigación		2007-05-16 19:37:10	privado	▲ ▼
<input type="checkbox"/>	docu_007	926.6 kB	2007-05-24 13:05:38	privado	▲ ▼
<input type="checkbox"/>	Prueba AUTORE	1.1 MB	2007-05-26 00:37:08	privado	▲ ▼
<input type="checkbox"/>	Pruebas	801.9 kB	2007-05-29 13:44:26	privado	▲ ▼
<input type="checkbox"/>	atfolder.2007-05-30.7014096765		2007-05-30 01:18:21	privado	▲ ▼
<input type="checkbox"/>	Curso de guitarra popular		2007-06-15 17:00:16	privado	▲ ▼
<input type="checkbox"/>	Izenburu gabekoa	750.2 kB	2007-06-19 11:57:47	privado	▲ ▼
<input type="checkbox"/>	Sin título	750.2 kB	2007-06-22 09:48:20	privado	▲ ▼
<input type="checkbox"/>	asignatura Farmacia		2007-07-04 14:45:27	privado	▲ ▼
<input type="checkbox"/>	atfolder.2007-07-16.9282795719		2007-07-16 11:08:48	privado	▲ ▼
<input type="checkbox"/>	Pruebas		2007-07-17 15:10:12	privado	▲ ▼
<input type="checkbox"/>	bonito		2007-09-19 04:12:07	concurso	▲ ▼

Figura 2.3. Herramienta de autor AUTORE.

- **JClíc**

Se trata de una herramienta para la creación, realización y evaluación de actividades educativas multimedia.

JClíc está desarrollado en la plataforma Java, es un proyecto de código abierto, se distribuye bajo los términos de la Licencia Pública General de GNU (GPL) y funciona en diversos entornos y sistemas operativos.

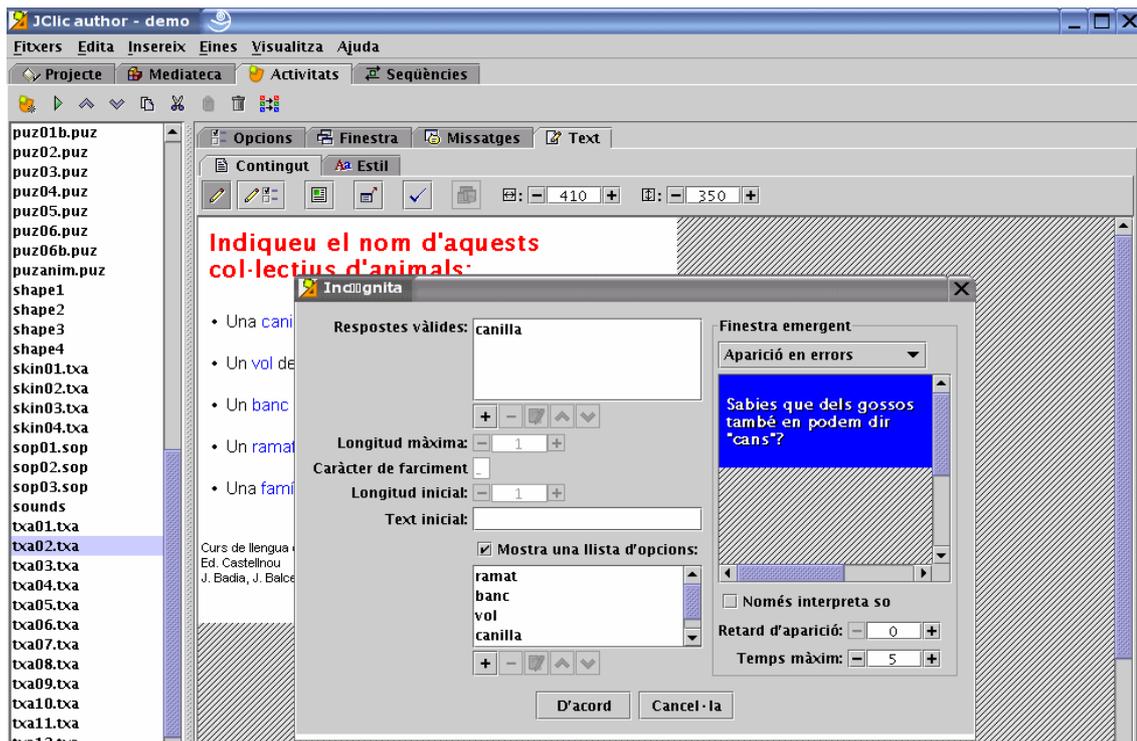


Figura 2.4. Editando una actividad de texto con JClic autor.

▪ Reiven

Reiven es una herramienta de autor para la enseñanza a distancia, que fue desarrollado por la empresa SETIVAL SCV.

Permite construir una biblioteca de contenidos teóricos y prácticos. A partir de éstos, podemos crear paquetes con estos objetos de aprendizaje ubicuos que podrán ser consumidos en plataformas como equipos de sobremesa, portátiles y teléfonos móviles.



Figura 2.5. Menú principal de la herramienta de autor Reiven.

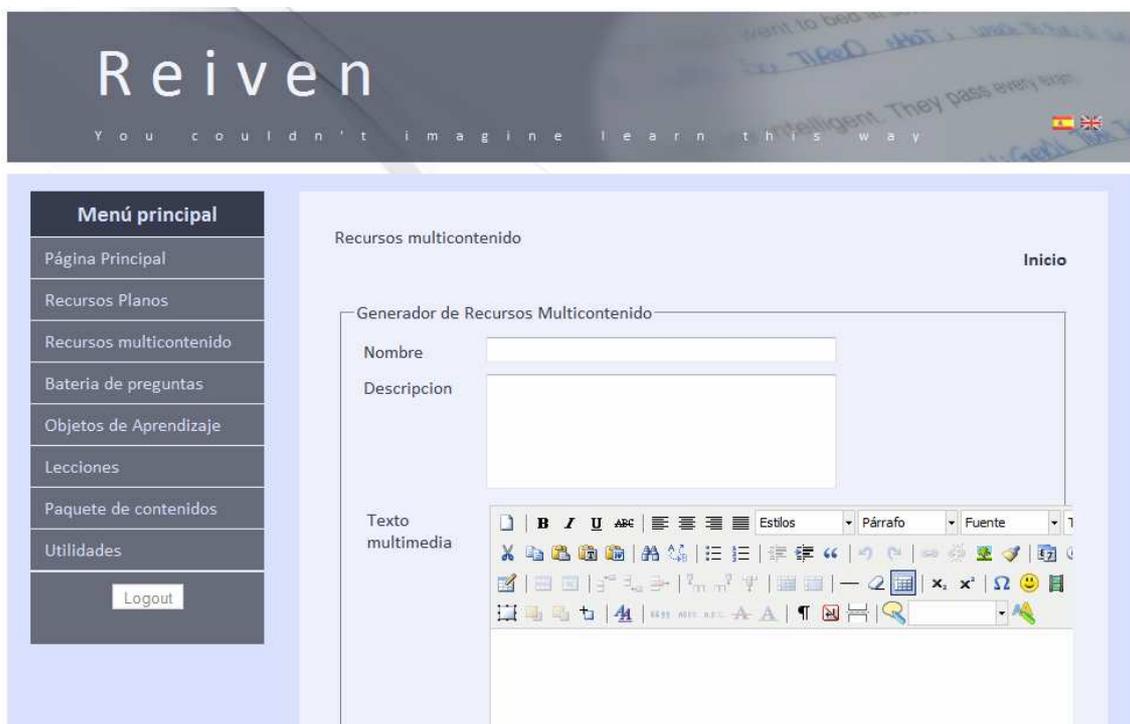


Figura 2.6. Generación de un recurso multicontenido en Reiven.

2.3 Aplicaciones interactivas para la TDT

El campo de las aplicaciones interactivas para TDT se encuentra todavía en un estado incipiente y existen pocas aplicaciones que se encuentren a disposición de los usuarios, debido principalmente a la reciente incorporación de la tecnología TDT. Cabe recordar que hasta este mismo año no se va a producir el apagón analógico.

Por otra parte, un amplio porcentaje de los usuarios no pueden acceder a estas aplicaciones, debido a que es necesario un decodificador especial que permita la ejecución de aplicaciones interactivas.

La mayoría de los equipos integrados que se ofertan en el mercado español no soportan el estándar MHP que permite acceder a los servicios interactivos ofrecidos por los radiodifusores.

Sólo 3.920 decodificadores de los vendidos entre diciembre de 2005 y febrero de 2007 incorporaban MHP frente a los más de 2,6 millones que fueron adquiridos en total [2].

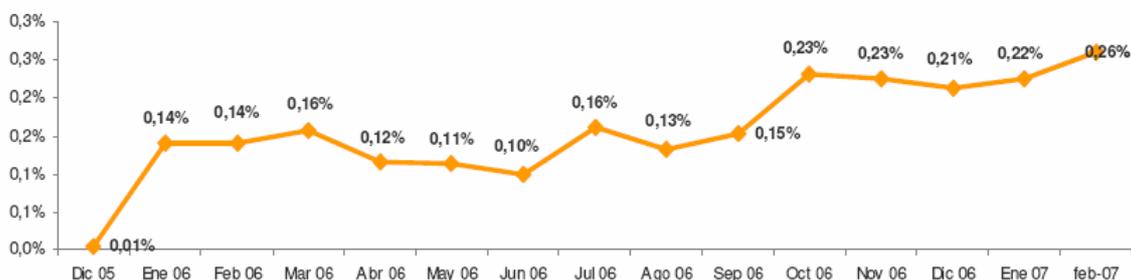


Figura 2.7. Evolución del porcentaje de decodificadores vendidos con MHP sobre el total.

Fuente: GFK España Febrero- 2007

2.3.1 Empresas y grupos con desarrollos para la TDT

Existen distintos grupos de investigación y empresas que desarrollan soluciones para televisión digital interactiva.

2.3.1.1 Laboratorio de TVDi de la Universidad de Vigo

En el Laboratorio de TVDi se han desarrollado múltiples proyectos de investigación aplicada, centrados en el estudio y desarrollo de software para decodificadores según la norma Multimedia Home Platform (MHP).

Cabe destacar los siguientes proyectos [3]:

1. Proyecto MHP: Dedicado al desarrollo de un receptor de TV digital interactiva acorde a la norma Multimedia Home Platform (MHP)
2. Proyecto AVATAR: Dedicado al desarrollo de recomendadores inteligentes y personalizados de contenidos de TV digital interactiva y servicios telemáticos implementados sobre televisión digital interactiva mediante aplicaciones MHP.
3. Proyecto PASSEPARTOUT : Dedicado al desarrollo de un prototipo de Home Media Center de próxima generación, capaz de ejecutar aplicaciones MHP y de acceder a múltiples fuentes de contenido a través de redes de banda ancha e inalámbricas.

2.3.1.2 Atos-Origin

Atos-Origin es una compañía internacional de servicios de tecnologías de la información.

Atos-Origin viene trabajando en distintos proyectos de televisión digital interactiva entre los que destacan:

- Sistema de pago de tributos mediante TPV virtual en televisión digital interactiva para el Excmo. Ayuntamiento de Alcázar de San Juan.



Figura 2.8. Sistema de pago de tasas y tributos para el Excmo. Ayuntamiento de Alcázar de San Juan desarrollado por Atos-Origin.

- Servicio de búsqueda de empleo militar para el Ministerio de Defensa.



Figura 2.9. Servicio de búsqueda de empleo militar para el Ministerio de Defensa desarrollado por Atos-Origin.

- Servicio de búsqueda de empleo público para el Ministerio de Administraciones Públicas.



Figura 2.10. Servicio de búsqueda de empleo público para el Ministerio de Administraciones Públicas desarrollado por Atos-Origin.

- Servicio de observatorio de precios para el Ministerio de medio Ambiente y Medio Rural y Marino.



Figura 2.11. Servicio de observatorio de precios para el Ministerio de medio Ambiente y Medio Rural y Marino desarrollado por Atos-Origin.

- Servicio de alertas por ola de calor para el Ministerio de Sanidad y Consumo.

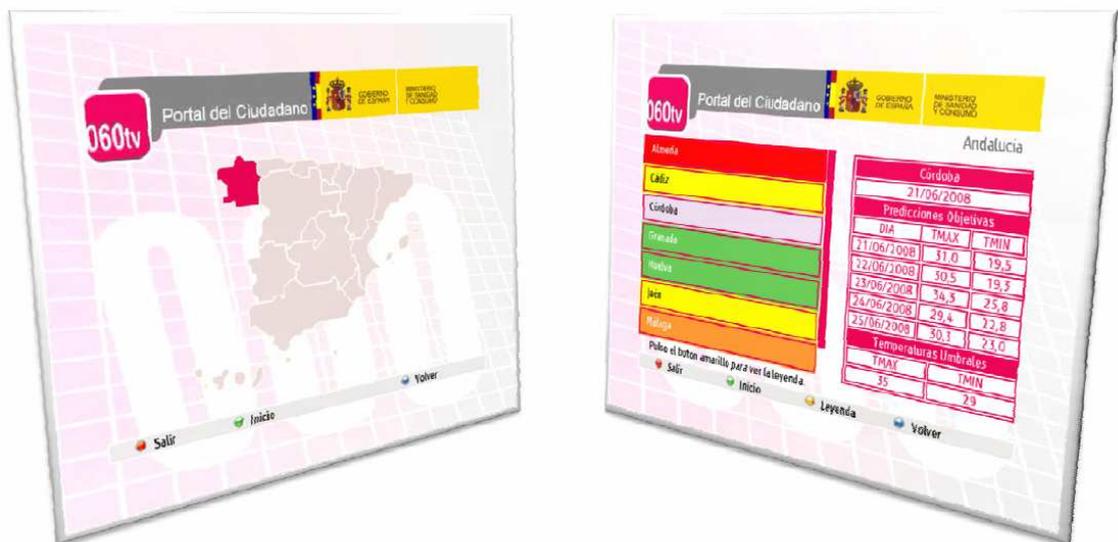


Figura 2.12. Servicio de alertas por ola de calor para el Ministerio de Sanidad y Consumo desarrollado por Atos-Origin.

2.3.1.3 SETIVAL SCV

TDT-Reiven es una plataforma perteneciente a un proyecto final de carrera [4] a partir de la colaboración entre SETIVAL SCV y la Universidad Politécnica de Valencia, que consta de un visor de objetos de aprendizaje docentes para la TDT que proporciona la funcionalidad de realizar distintos tipos de ejercicios provenientes de un curso.

Se trata de una aplicación para t-learning desarrollado sobre MHP que funciona de manera no interactiva, es decir, no proporciona una visión dinámica de la aplicación sino que su contenido es global y estático para todo el mundo.

En la figura 2.13 y 2.14 podemos observar algunas capturas de la aplicación.



Figura 2.13. Imagen de una pregunta de la aplicación TDT-Reiven.

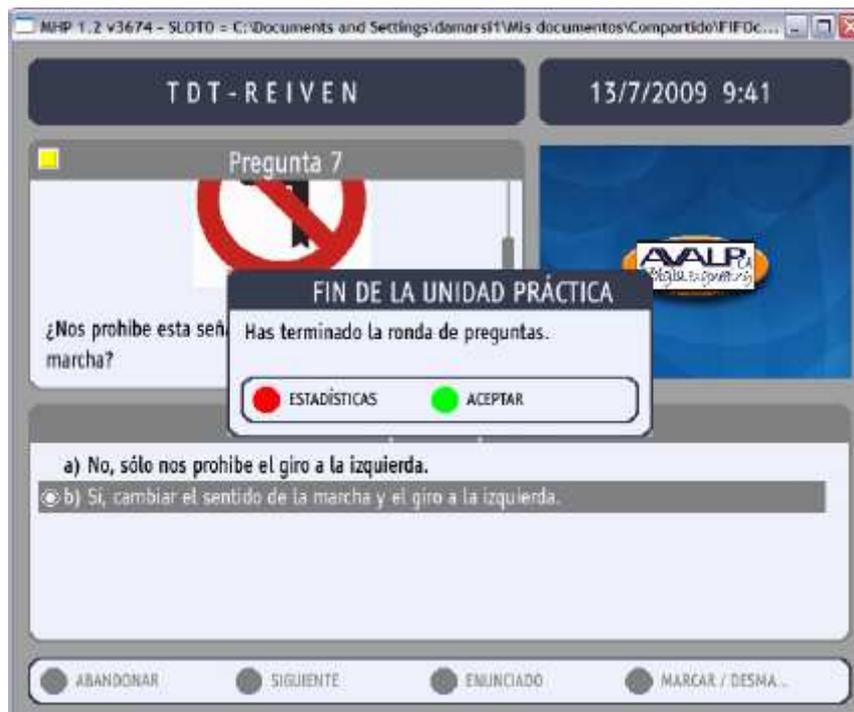


Figura 2.14. Imagen de una ventana emergente de la aplicación TDT-Reiven.

2.3.2 Aplicaciones interactivas en las cadenas de televisión

En la actualidad podemos encontrar ya servicios interactivos en cadenas de televisión.

Entre ellas se encuentra RTVE donde podemos encontrar aplicaciones básicas como la guía de programación y noticias sobre tráfico, meteorología y bolsa y un teletexto digital. Además de la aplicación EmpleaT que permite la búsqueda de empleo a través del televisor. En ella se listan ofertas de trabajo pudiendo filtrarlas por sector, comunidad autónoma y nivel formativo. Diariamente difunde más de un millar de ofertas de empleo en los servicios de TDT de TVE y se encuentra vinculada al programa "Aquí hay trabajo", de La 2. Se muestra una captura de pantalla de la aplicación en la figura 2.15.



Figura 2.15. Imagen de la aplicación empleaT.

En otras cadenas nacionales y autonómicas podemos encontrar también servicios MHP [17]. Podemos destacar la televisión pública catalana que dispone de una larga experiencia en Televisión Digital Terrestre. En Mayo de 2002 lanzó cuatro aplicaciones MHP:

- Lanzadora básica
- Tickers
- EPG
- El tiempo



En los último meses de 2003 y los primeros de 2004 se llevó a cabo el proyecto Micromercats de TDT o Prueba Piloto, donde uno de sus objetivos fue poner a disposición de usuarios reales los contenidos interactivos para conocer su aceptación y usos, para ello se instalaron 70 receptores MHP en 70 hogares en el área metropolitana de Barcelona, con la participación de unos 230 usuarios. Entre las distintas aplicaciones que se pusieron a disposición de los usuarios se encuentran las siguientes:

- Noticias: aplicación que muestra por pantalla los titulares y una sinopsis de noticias procedentes de distintos portales.



- El temps: aplicación que contiene la información meteorológica más actualizada, con previsiones, temperaturas y mapas.



- Formatos publicitarios interactivos: aplicaciones con anuncios publicitarios.



También podemos encontrar aplicaciones interactivas en Telemadrid gracias a la colaboración con otras empresas, en el desarrollo e implementación de aplicaciones MHP.

En 2003 participó en el proyecto ACTUA donde se pusieron en el aire cinco aplicaciones durante los meses de junio y julio. Este proyecto se enmarcó dentro de una campaña de promoción de la TDT en la que intervinieron algunos centros comerciales en los que se instalaron centros de demostración.

Entre las aplicaciones que se instalaron podemos destacar las siguientes:

- Farmacias de guardia
- Aeropuerto de Madrid en Barajas
- Tráfico



La televisión de Galicia también se suma al campo de la tecnología MHP, que en colaboración con la universidad de Vigo, lleva realizando investigaciones desde marzo de 2004.

En Febrero de 2005, TVG incluyó en sus emisiones varias aplicaciones MHP en pruebas con cobertura en todo el territorio gallego. Entre ellas destacamos cuatro aplicaciones con el objetivo de mostrar información.

- Sorteos
- El tiempo
- Programación TVG
- Olladas, aplicación que muestra en tiempo real las cámaras web de vistas que TVG tiene distribuidas por toda Galicia.



2.4 Conclusiones

Hemos visto las distintas soluciones que existen tanto para la generación de objetos de aprendizaje mediante una herramienta de autor, como las aplicaciones interactivas para la TDT capaces de consumir estos objetos de aprendizaje, así como una solución existente que engloba estos procedimientos.

Como hemos podido ver en el capítulo, existe un gran interés por parte de empresas y cadenas de televisión en incorporar aplicaciones interactivas en sus emisiones, y nos hemos dado cuenta, que en lo referente al tipo de aplicaciones interactivas educativas o de aprendizaje,

son prácticamente nulas, aumentando las posibilidades de mercado del producto resultante de este proyecto.

De entre las distintas soluciones que existen para la generación de objetos de aprendizaje, ninguna cumple con nuestro criterio de generar objetos de aprendizaje para el despliegue en una televisión, todas ellas se centran en plataformas como el ordenador y, algunas de ellas en plataformas como teléfonos móviles.

Es por ello por lo que se hace necesario crear una herramienta de autor que permita generar objetos de aprendizaje para la TDT desde cero, o bien, adaptar una de las herramientas existentes para que cumpla con esta funcionalidad.

Nos hemos decantado por la segunda opción, la de adaptar una herramienta de autor existente para que cumpla con nuestras necesidades, y hemos elegido la herramienta de autor Reiven, debido a que es la herramienta desarrollada por SETIVAL SCV.

TDT-Reiven, nos ofrece una solución de t-learning para la navegación sobre un curso. Sin embargo, esta aplicación no cumple con algunos aspectos, como por ejemplo que la aplicación ofrezca una mínima intrusión visual respecto al video visionado. Por esto, se ha pensado en la posibilidad de realizar ciertas modificaciones, previo estudio de su funcionalidad, e incluirla en nuestro proyecto.

Capítulo 3. Análisis y planificación

A lo largo de este capítulo se identificarán los requerimientos y los casos de uso a lo que deberá responder la solución a desarrollar. Además de la arquitectura que tendrá el sistema.

Al final se presenta la planificación de tareas con sus tiempos asociados para cada una de ellas, incluyendo el diagrama de Gantt, y se expondrán las conclusiones que se han obtenido del capítulo.

3.1 Casos de uso

Los casos de uso son una forma de visualizar el comportamiento de los distintos módulos que componen un sistema, así como las interacciones del sistema con el usuario o con otro sistema.

Los casos de uso nos ayudan a describir qué es lo que el sistema debe hacer desde el punto de vista del usuario.

En las figuras 3.1, 3.2 y 3.3 se muestran los diagramas con los distintos casos de uso para el diseñador de contenidos, el operario y el espectador respectivamente, identificados para el sistema:

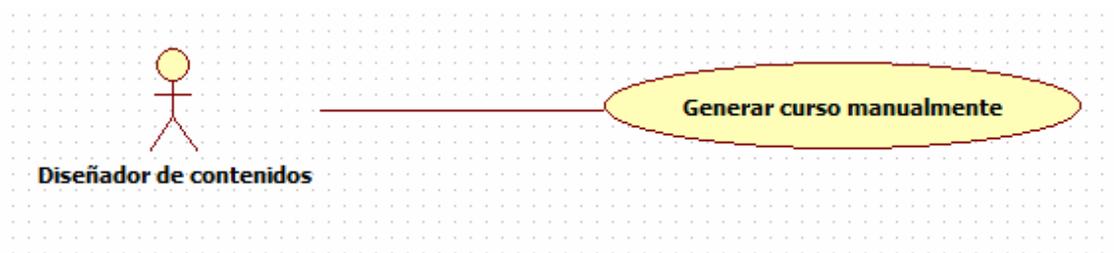


Figura 3.1. Diagrama de casos de uso del diseñador de contenidos.

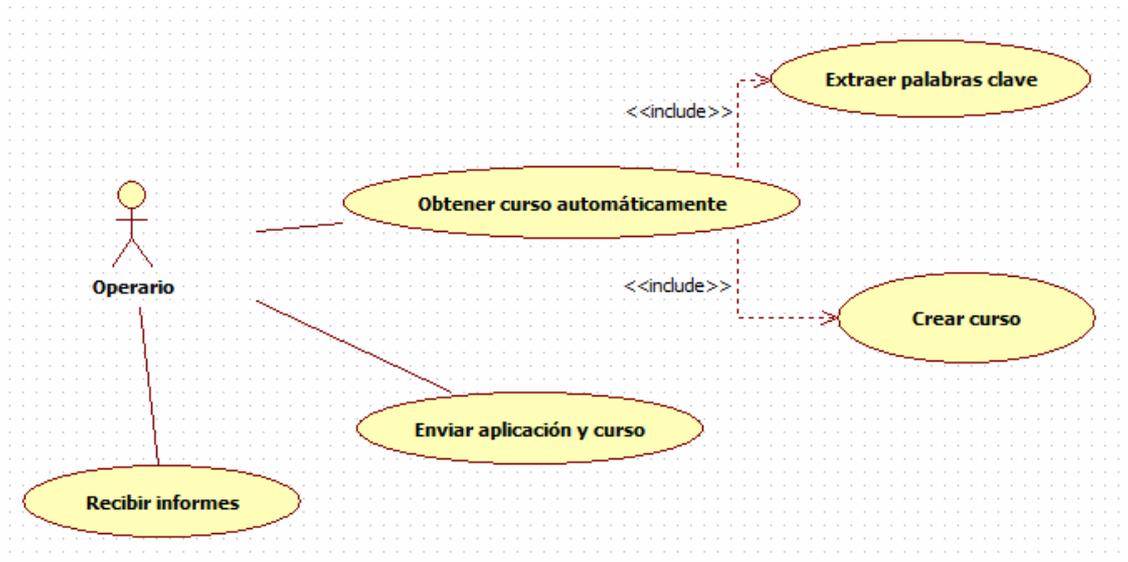


Figura 3.2. Diagrama de casos de uso del operario.

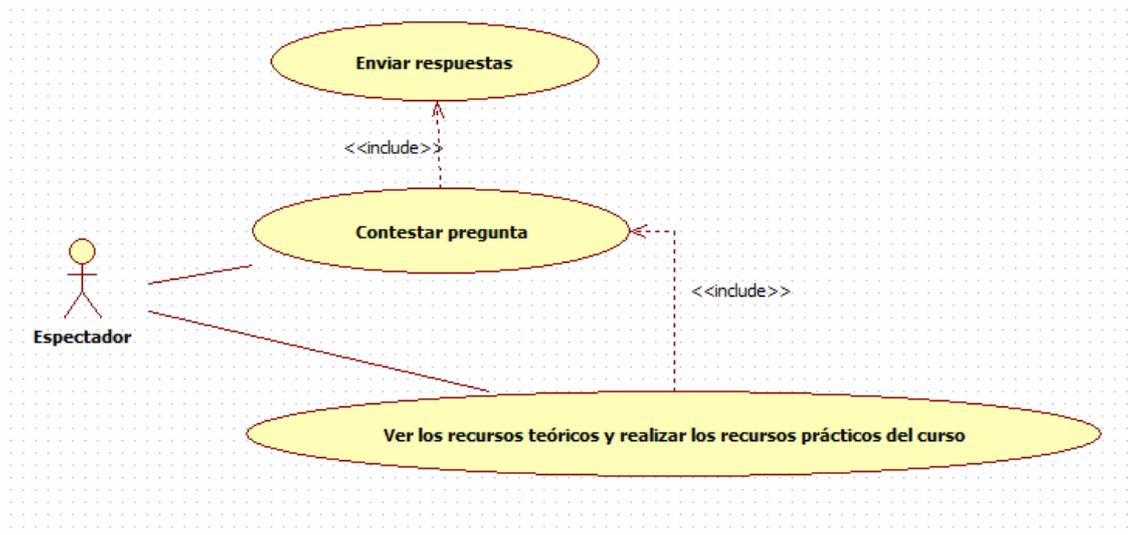


Figura 3.3. Diagrama de casos de uso del espectador.

Las tablas 3.1 a 3.8 muestran las plantillas para cada uno de los casos de uso.

Caso de uso: Generar curso manualmente
Precondiciones: El operario se ha identificado en la herramienta de autor

Postcondiciones: El curso queda almacenado en la base de datos	
INTENCIONES DEL USUARIO	OBLIGACIONES DEL SISTEMA
1. El operario se identifica en la herramienta mediante login y password	2. El sistema valida tanto el login como el password y muestra la herramienta de autor
3. el operario genera un paquete de contenidos rellenando los campos necesarios para ello	4. El sistema almacena el paquete de contenidos en la base de datos
EXTENSIONES SÍNCRONAS	
#1: Si en 2 los valores introducidos no se encuentran en la base de datos, el sistema muestra un mensaje de error y pide de nuevo al usuario que se identifique	
EXTENSIONES ASÍNCRONAS	
Ninguna	

Tabla 3.1. Caso de uso: Generar curso manualmente.

Caso de uso: Obtener curso automáticamente	
Precondiciones: El sistema recibe la EPG del programa, con la descripción, el código y la duración del mismo	
Postcondiciones: El sistema genera un curso	
INTENCIONES DEL USUARIO	OBLIGACIONES DEL SISTEMA
1. El operario introduce la descripción, el código y la duración del programa	2. El sistema comprueba que no existe ningún curso en la base de datos asociado al código de programa
	3. El sistema CALL "Extraer palabras clave"
	4. El sistema CALL "Crear curso"
EXTENSIONES SÍNCRONAS	

#1: Si en 2 el sistema encuentra un curso en la base de datos asociado al código de programa, lo devuelve y el caso de uso termina con éxito
EXTENSIONES ASÍNCRONAS
Ninguna

Tabla 3.2. Caso de uso: Obtener curso automáticamente.

Caso de uso: Extraer palabras clave	
Precondiciones: El sistema recibe la descripción del programa	
Postcondiciones: El sistema obtiene un conjunto de palabras clave	
INTENCIONES DEL USUARIO	OBLIGACIONES DEL SISTEMA
	1. El sistema extrae las palabras clave a partir de la descripción del programa
EXTENSIONES SÍNCRONAS	
Ninguna	
EXTENSIONES ASÍNCRONAS	
Ninguna	

Tabla 3.3. Caso de uso: Extraer palabras clave.

Caso de uso: Crear curso	
Precondiciones: El sistema recibe un conjunto de palabras clave	
Postcondiciones: El sistema genera un curso	
INTENCIONES DEL USUARIO	OBLIGACIONES DEL SISTEMA
	1. El sistema realiza una búsqueda exhaustiva en la base de datos a partir del conjunto de palabras clave

	2. El sistema crea un curso relacionado con el conjunto de palabras clave, lo devuelve y el caso de uso termina con éxito
EXTENSIONES SÍNCRONAS	
Ninguna	
EXTENSIONES ASÍNCRONAS	
Ninguna	

Tabla 3.4. Caso de uso: Crear curso.

Caso de uso: Enviar aplicación y curso	
Precondiciones: El sistema recibe la aplicación y un curso	
Postcondiciones: La aplicación junto con el curso se encuentran cargados en el decodificador de TDT	
INTENCIONES DEL USUARIO	OBLIGACIONES DEL SISTEMA
1. El operario introduce los ficheros de la aplicación y del curso junto con la señal de video	2. El sistema multiplexa la aplicación con el curso y la señal de video y lo envía al decodificador mediante tramas
EXTENSIONES SÍNCRONAS	
#1: Si en 2 se produce algún error en el envío, el caso de uso termina en fallo	
EXTENSIONES ASÍNCRONAS	
Ninguna	

Tabla 3.5. Caso de uso: Enviar aplicación y curso.

Caso de uso: Contestar pregunta	
Precondiciones: el curso que incluye la aplicación dispone de al menos una pregunta	
Postcondiciones: la respuesta queda almacenada para su posterior envío	
INTENCIONES DEL USUARIO	OBLIGACIONES DEL SISTEMA
1. El espectador selecciona la respuesta y pulsa el botón "ok"	2. El sistema almacena la respuesta
EXTENSIONES SÍNCRONAS	
Ninguna	
EXTENSIONES ASÍNCRONAS	
En cualquier momento el espectador puede volver a ver la emisión en pantalla completa pulsando el botón "back"	

Tabla 3.6. Caso de uso: Contestar pregunta.

Caso de uso: Ver los recursos teóricos y realizar los recursos prácticos del curso	
Precondiciones: -	
Postcondiciones: -	
INTENCIONES DEL USUARIO	OBLIGACIONES DEL SISTEMA
1. El espectador pulsa la tecla "menú" para acceder a los contenidos del curso	2. El sistema muestra los recursos teóricos y prácticos que contiene el curso
EXTENSIONES SÍNCRONAS	
Ninguna	
EXTENSIONES ASÍNCRONAS	
En cualquier momento el espectador puede volver a ver la emisión en pantalla completa pulsando el botón "back"	

Tabla 3.7. Caso de uso: Ver los recursos teóricos y realizar los recursos prácticos del curso.

Caso de uso: Recibir informes	
Precondiciones: -	
Postcondiciones: -	
INTENCIONES DEL USUARIO	OBLIGACIONES DEL SISTEMA
1. El operario accede a la aplicación web que se encarga de mostrar los informes.	2. El sistema muestra los informes filtrados según las preferencias del usuario.
EXTENSIONES SÍNCRONAS	
Ninguna	
EXTENSIONES ASÍNCRONAS	
Ninguna	

Tabla 3.8. Caso de uso: Recibir informes.

3.2 Análisis de requerimientos

En este apartado se identifican los requerimientos del proyecto que posteriormente se simplifican en una tabla resumen.

El análisis de requerimientos permite especificar el papel que desempeña un sistema y su comportamiento. Permite definir las características y atributos que tendrá el sistema. El análisis de requerimientos nos permitirá identificar los problemas y sus causas y buscar soluciones para resolverlos.

A continuación se detallan los requerimientos que debe de cumplir nuestro sistema.

El sistema deberá incorporar una herramienta que permita **generar objetos de aprendizaje**, tanto teóricos como prácticos, y empaquetarlos en un curso, pudiendo asociar este curso a un programa de televisión, incorporando distintos parámetros.

El sistema será capaz de buscar en la base de datos un curso que se encuentre asociado a un código de programa, el cual obtendrá a partir de la EPG del propio programa. En el caso de que no se obtenga ningún curso para un programa en cuestión, debido a que no ha sido generado previamente mediante la herramienta de autor, el sistema **generará automáticamente un curso** a partir de un número definido de palabras clave extraídas de la descripción del programa.

Una vez preparados los objetos de aprendizaje, se necesita una aplicación para poder consumirlos y presentarlos por pantalla. Esta aplicación deberá ofrecer una mínima intrusión visual respecto al video visionado, a la vez que deberá ser intuitiva en lo que respecta a la navegación a través de ella.

El sistema deberá ser capaz de **multiplexar** la aplicación encargada de consumir los objetos de aprendizaje junto con los propios objetos de aprendizaje y la señal de televisión y, una vez hecho esto, enviarlo todo al decodificador TDT para que se encargue de ejecutar y **visualizar por pantalla**.

La aplicación deberá proporcionar los mecanismos necesarios para poder **enviar las contestaciones** de los usuarios a las preguntas proporcionadas en el curso, a una base de datos que se encargue de almacenarlas.

Se debe proporcionar una herramienta web que permita **visualizar diferentes informes** a partir de las respuestas proporcionadas por los usuarios.

Id	Nombre	Descripción
R.0	Generación de objetos de aprendizaje	El sistema deberá incorporar una herramienta que permita generar objetos de aprendizaje y asociarlos a un programa
R.1	Asignador de objetos de aprendizaje automático	
R.1.1	Obtención del curso	El sistema será capaz de buscar un curso que esté asociado a un código de programa
R.1.2	Generación del curso a partir de la extracción de palabras clave de la EPG del	El sistema generará automáticamente un curso a partir de un número definido de palabras

	programa	clave
R.2	Aplicación para el consumo de objetos de aprendizaje	Se dispondrá de una aplicación que permita consumir los objetos de aprendizaje por pantalla, ofreciendo una mínima intrusión visual respecto al video visionado.
R.3	Multiplexor de aplicación y objetos de aprendizaje	El sistema deberá ser capaz de multiplexar la aplicación y los objetos de aprendizaje y enviarlo todo al decodificador de TDT
R.4	Envío de contestaciones	La aplicación deberá proporcionar los mecanismos necesarios para poder enviar las contestaciones de los usuarios
R.5	Presentación de resultados	Se debe proporcionar una herramienta web que permita visualizar diferentes informes

Tabla 3.9. Resumen de requerimientos.

3.3 Arquitectura del sistema

En este apartado se presenta la arquitectura que tendrá el sistema, junto con una explicación para cada módulo que la compone.

La arquitectura a desarrollar la podemos dividir en cuatro partes, relacionadas entre sí, que se describen a continuación:

1. Una **herramienta de autor** que permita generar paquetes de contenido que incluirán una serie de bloques de teoría y una serie de preguntas para posteriormente ser consumidos por una aplicación. Más adelante profundizaremos más sobre esta y el resto de herramientas.
2. Una aplicación que sea capaz de **consumir los paquetes de contenido** o cursos sin interferir en la visión del programa y que sea capaz de enviar las respuestas de los usuarios, en el caso de las preguntas, a un servidor.
3. Un **asignador de objetos de aprendizaje automáticos**, es decir, un programa que se encargue de obtener la EPG (guía electrónica del

programa) de un programa en cuestión y se encargue de buscar un paquete de contenidos (un curso), generado por la herramienta de autor, que esté relacionado con la temática del programa. Una vez obtenido, tendrá que multiplexar tanto la señal de emisión (video y audio), como la aplicación descrita anteriormente junto con un curso.

4. Por último, diseñar una solución que permita **obtener resultados** a partir de las contestaciones que proporcionen los usuarios a la hora de realizar los cursos, tales como: el número de respuestas correctas de una determinada pregunta, el número de participantes, etc...

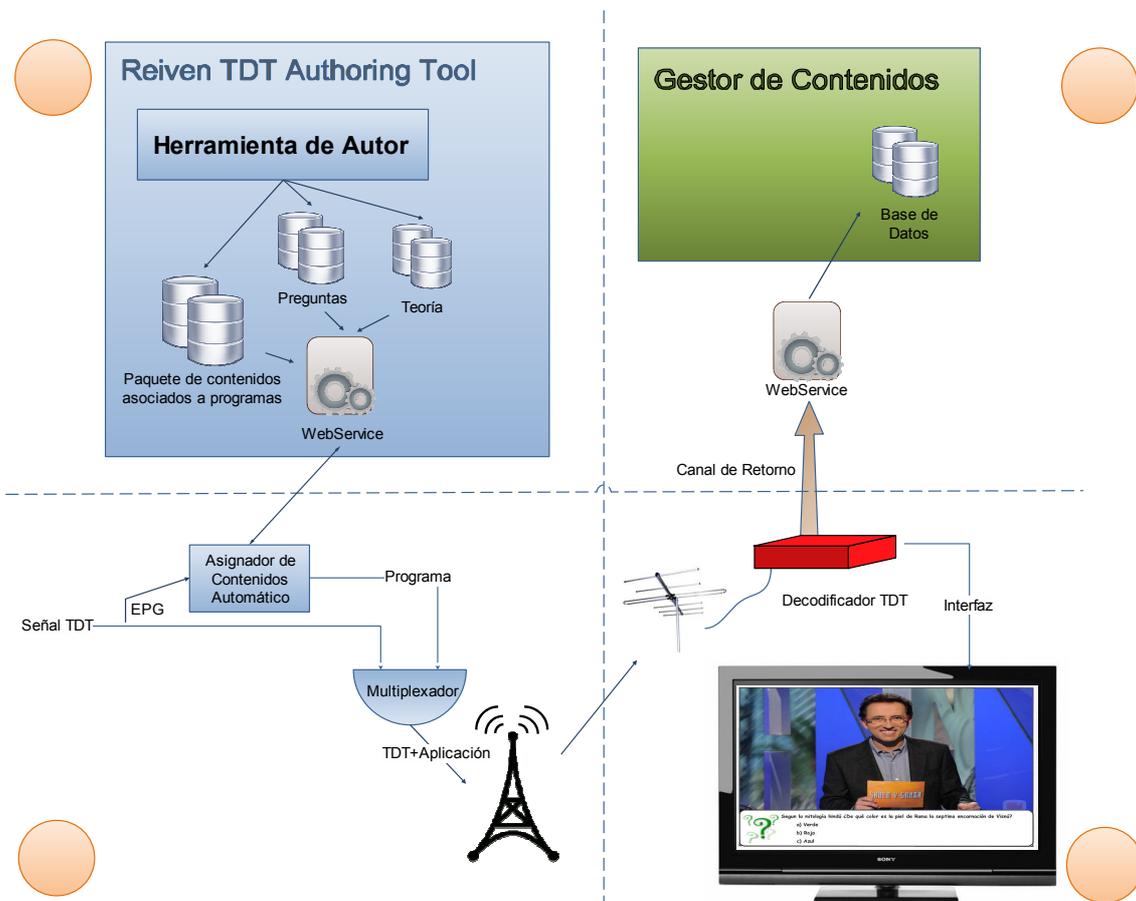


Figura 3.4. Arquitectura del sistema a desarrollar.

3.4 Planificación

Este apartado presenta la planificación para el proyecto, con las tareas que se deberán realizar para cada fase del proyecto incluyendo los

tiempos estimados de las mismas. También se muestra el diagrama de Gantt del proyecto.

La fecha de comienzo del proyecto será el 1 de Diciembre de 2009 y la fecha de terminación será en marzo de 2010.

FASES	Diciembre	Enero	Febrero	Marzo
Requerimientos	■			
Análisis	■			
Diseño		■		
Codificación		■	■	
Pruebas				■

Tabla 3.10. Planificación temporal de las fases del proyecto.

3.4.1 Descomposición en fases y tareas del proyecto

En este punto procedemos a identificar las distintas fases del proyecto junto con las tareas que hay que realizar para cada una de ellas:

- Fase de análisis:
 - Familiarización con el lenguaje PHP.
 - Estudio de la generación de paquetes de contenido mediante la herramienta de autor Reiven.
 - Estudio sobre cómo conectar a una base de datos MySQL con PHP.
 - Familiarización con XML y estudio del estándar IMS QTI.
 - Estudio del estándar MHP y programación con xlets.
 - Estudio del envío y ejecución de aplicaciones en el decodificador de TDT.
 - Documentación y estudio del canal de retorno.
- Fase de diseño:
 - Diseño de los componentes a modificar en el módulo de generación de paquetes de contenido.

- Diseño del módulo de obtención de cursos.
- Diseño de la interfaz de la aplicación consumidora de los cursos.
- Diseño de la aplicación web encargada de mostrar los informes.
- Fase de codificación:
 - Implementación caso de uso: Generar curso manualmente.
 - Implementación caso de uso: Extraer palabras clave.
 - Implementación caso de uso: Crear curso.
 - Implementación caso de uso: Obtener curso automáticamente
 - Implementación caso de uso: Enviar aplicación y curso.
 - Implementación caso de uso: Contestar pregunta.
 - Implementación caso de uso: Ver los recursos teóricos y realizar los recursos prácticos del curso.
 - Implementación caso de uso: Ver informes.
- Fase de pruebas:
 - Realización de pruebas unitarias y depuración de cada uno de los módulos que componen el sistema.
 - Realización de pruebas de integración del sistema.
 - Realización de pruebas de aceptación del sistema.

3.4.2 Métodos de estimación del esfuerzo [16]

Para estimar el esfuerzo existen distintos métodos. Nosotros utilizaremos el siguiente, con el que estimaremos el esfuerzo para cada una de las tareas:

- *PERT* (optimista, pesimista, más posible): este método utiliza la siguiente fórmula matemática para calcular la estimación del esfuerzo y obtiene una media:

$$\text{Estimación} = \frac{\text{Pesimista} + 4 * \text{Más probable} + \text{Optimista}}{6}$$

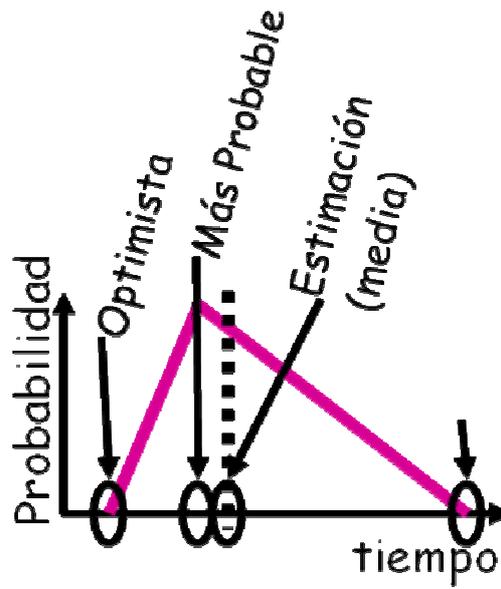


Figura 3.5. Método PERT.

3.4.3 Estimación del esfuerzo de cada tarea

A continuación se presenta la duración estimada en horas, según el método PERT comentado en el punto anterior, de cada una de las tareas:

- Fase de análisis:
 - Familiarización con el lenguaje PHP: 3 horas.
 - Estudio de la generación de paquetes de contenido mediante la herramienta de autor Reiven: 10 horas.
 - Estudio sobre cómo conectar a una base de datos MySQL con PHP: 2 horas.
 - Familiarización con XML y estudio del estándar IMS QTI: 8 horas.
 - Estudio del estándar MHP y programación con xlets: 15 horas.

- Estudio del envío y ejecución de aplicaciones en el decodificador de TDT: 20 horas.

- Documentación y estudio del canal de retorno: 11 horas.

Total: 69 horas.

- Fase de diseño:

- Diseño de los componentes a modificar en el módulo de generación de paquetes de contenido: 12.

- Diseño del módulo de obtención de cursos: 17 horas.

- Diseño de la interfaz de la aplicación consumidora de los cursos: 10 horas.

- Diseño de la aplicación web encargada de mostrar los informes: 8 horas.

Total: 47 horas.

- Fase de codificación:

- Implementación caso de uso: Generar curso manualmente: 20 horas.

- Implementación caso de uso: Extraer palabras clave: 25 horas.

- Implementación caso de uso: Crear curso: 40 horas.

- Implementación caso de uso: Obtener curso automáticamente: 10 horas.

- Implementación caso de uso: Enviar aplicación y curso: 15 horas.

- Implementación caso de uso: Contestar pregunta: 15 horas.

- Implementación caso de uso: Ver los recursos teóricos y realizar los recursos prácticos del curso: 40 horas.

- Implementación caso de uso: Ver informes: 10 horas.

Total: 175 horas.

- Fase de pruebas:

- Realización de pruebas unitarias y depuración de cada uno de los módulos que componen el sistema: 80 horas.

- Realización de pruebas de integración del sistema: 20 horas.
 - Realización de pruebas de aceptación del sistema: 5 horas.
- Total: 105 horas.

3.4.4 Planificación temporal. Diagrama de Gantt y diagrama de precedencias

Una vez hecho el análisis y el diseño del proyecto y nos disponemos a codificar las distintas tareas, primero hay que plantearse el orden entre ellas y, para ello, cabe plantearse las siguientes cuestiones:

- ¿Qué debe haberse hecho antes de esto?
- ¿Qué puede hacerse a la vez?
- ¿Que debe seguir a lo que hacemos ahora?

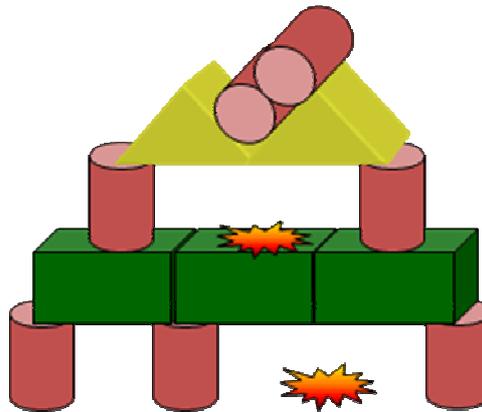


Figura 3.6. Precedencia entre tareas.

La respuesta a estas cuestiones queda reflejada en los diagramas que se explican a continuación.

El diagrama de Gantt permite resolver el problema de la programación de actividades, es decir, su distribución conforme a un calendario, de forma que se puede visualizar la duración de cada actividad, sus fechas de iniciación y terminación y el tiempo total requerido para la ejecución de un proyecto.

Consiste en una representación gráfica sobre dos ejes, en el eje vertical se representan las tareas del proyecto y en el eje horizontal se representa el tiempo.

Las siguientes características servirán de ayuda para entender el gráfico:

- Las tareas se representan mediante bloques rectangulares cuya anchura representa la duración de la misma y la altura no tiene ningún significado.
- La posición de cada bloque, según el eje horizontal del tiempo, indica el inicio y finalización de la tarea.
- Los bloques en color rojo indican el camino crítico, esto es el conjunto de tareas que tienen holgura cero, es decir, que si una de ellas se retrasa, se retrasa todo el proyecto, a diferencia de las tareas con holgura mayor que cero, cuyo valor indica el tiempo que disponemos para jugar con el inicio de la tarea, sin afectar a la totalidad del proyecto.

El diagrama de Gantt del proyecto se muestra en la figura 3.7

El diagrama de precedencias consiste en otra forma de representación equivalente al diagrama de Gantt, es la forma de representación más habitual en programas informáticos junto al diagrama de Gantt.

Este diagrama muestra las dependencias que existen entre las distintas tareas del proyecto, a la vez que muestra la fecha final del proyecto así como la flexibilidad de comienzo de las tareas, si existe, sin que afecte al tiempo total del proyecto. Incluye también el camino crítico indicando las tareas que no se pueden retrasar sin que ello conlleve el retraso de todo el proyecto.

Obviamos la representación de este diagrama debido a la similitud con el diagrama de Gantt.

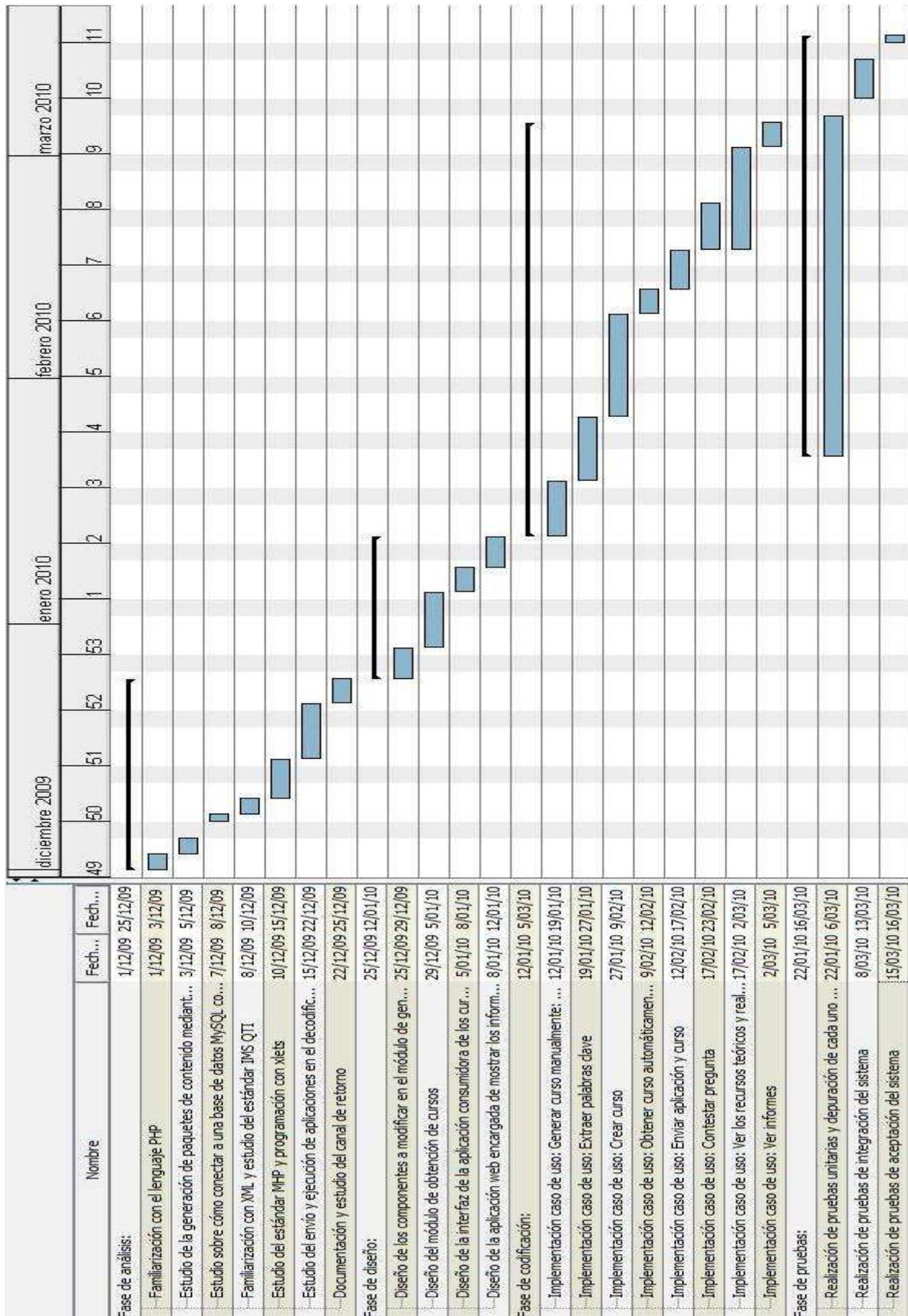


Figura 3.7. Diagrama de Gantt.

Capítulo 4. Técnicas y herramientas a utilizar

En este capítulo se introduce la metodología de desarrollo que se va a utilizar y las distintas herramientas de las que se harán uso para la realización de los distintos módulos que conforman la arquitectura del proyecto, así como los procedimientos que se deberán seguir para utilizarlas.

4.1 Ciclo de vida considerado

La metodología de desarrollo seguida es el ciclo de vida en cascada basado en el enfoque metodológico que ordena las etapas del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la anterior.

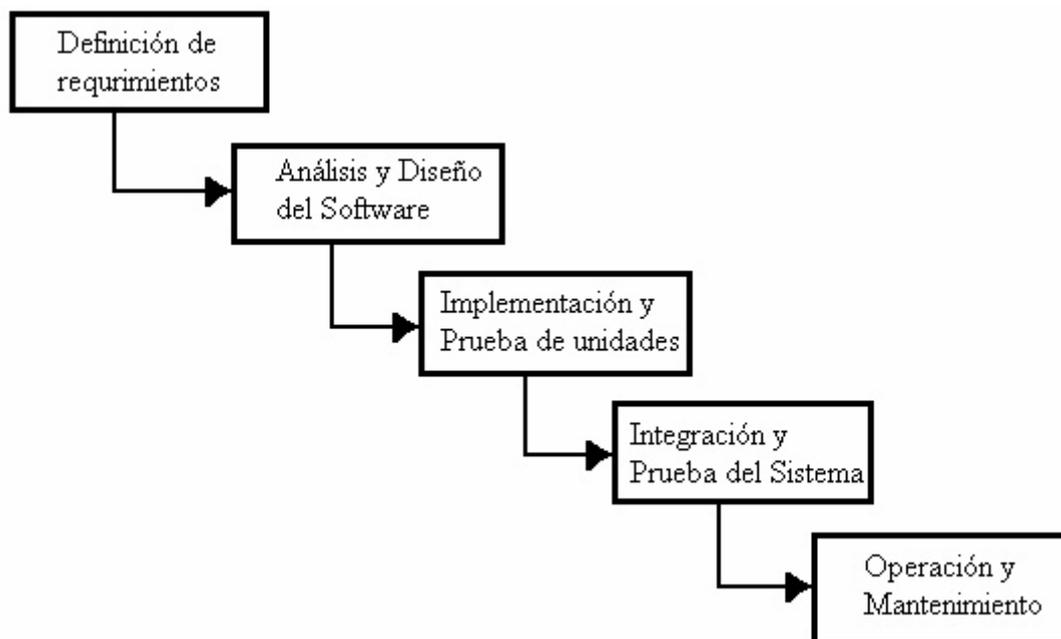


Figura 4.1. Modelo de ciclo de vida en cascada.

4.2 Herramientas a utilizar para el desarrollo de la arquitectura

En este punto se detallan las herramientas a utilizar para desarrollar los distintos módulos por los que está compuesta la arquitectura del proyecto.

4.2.1 Generación de objetos de aprendizaje

El desarrollo de este módulo se va a realizar mediante el lenguaje de programación PHP. También se va a utilizar el formato XML (Extensible Markup Language) siguiendo el estándar IMS QTI (IMS Question and test interoperability) , para empaquetar los cursos y poder, de esta forma, ser enviados a la aplicación que se encarga de consumirlos.

4.2.1.1 XML y IMS QTI

El formato XML es un lenguaje que permite compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes, permitiendo la globalización y la compatibilidad entre los sistemas.

Se basa en estructurar la información en forma de árbol, mediante la utilización de etiquetas de la forma <nombre>, las cuales engloban un pedazo de información.

La especificación IMS QTI está basada en XML y se trata de un estándar que permite la interoperabilidad entre sistemas que trabajan con objetos de aprendizaje, centrándose en la definición de un modelo para preguntas y tests, incluyendo también respuestas [6].

La idea es poder reflejar todas las condiciones de un examen o autoevaluación en un fichero XML que pueda ser ejecutable por cualquier sistema compatible con QTI.

Es por ello por lo que se decidió utilizar este estándar en la herramienta de autor Reiven a la hora de empaquetar los cursos, para, de esta forma, poder generar cursos para cualquier plataforma, que a su vez cumpla con el estándar.

Esta especificación es la que utiliza el sistema de gestión de cursos también llamado CMS (*Course Management System*) o LMS (*Learning Management System*) de Moodle, entre otros.

En la figura 4.2 se muestra un ejemplo de esta especificación:

```

<?xml version="1.0" encoding="UTF-8"?>
<assessments title="preguntas" random="0">
  <items>
    <assessmentItem identifier="490" title="Titulo de ejemplo"
      adaptative="false" timeDependent="false">
      <outcomeDeclaration identifier="SCORE" cardinality="single"
        baseType="integer">
        <defaultValue><value>0</value></defaultValue>
      </outcomeDeclaration>
      <responseDeclaration identifier="RESPONSE"
        cardinality="single" baseType="identifier">
        <correctResponse>
          <value>R0</value>
        </correctResponse>
      </responseDeclaration>
      <itemBody>
        <html><p>Enunciado de ejemplo</p></html>
        <choiceInteraction responseIdentifier="RESPONSE"
          shuffle="true" maxChoices="1">
          <prompt></prompt>
          <simpleChoice identifier="R0">Respuesta correcta
            <feedbackInline outcomeIdentifier="FEEDBACK"
              identifier="R0" showHide="show">
            </feedbackInline>
          </simpleChoice>
          <simpleChoice identifier="R1">Respuesta incorrecta
            <feedbackInline outcomeIdentifier="FEEDBACK"
              identifier="R1" showHide="show">
            </feedbackInline>
          </simpleChoice>
        </choiceInteraction>
      </itemBody>
    </assessmentItem>
  </items>
</assessments>

```

Figura 4.2. Ejemplo de la especificación IMS-QTI.

4.2.2 Distribución de objetos de aprendizaje

Para poder enviar la aplicación al decodificador de TDT se hará uso de las siguientes herramientas, todas ellas son software libre:

1. **oc-update.sh:** herramienta que permite convertir el directorio de la aplicación al formato conveniente para poder ser enviado al STB.
2. **Tscbrmuxer:** Herramienta que permite multiplexar la aplicación, una vez convertida al formato que corresponde, junto con la señal de televisión y enviar todo ello al decodificador.

Profundizaremos en el funcionamiento de estas dos herramientas en el punto 5.7.

4.2.3 Consumo de objetos de aprendizaje

De entre los distintos estándares que existen para el desarrollo de aplicaciones interactivas para la TDT se ha adoptado el estándar MHP, ya que es el estándar que se ha decidido utilizar en España y en el resto de Europa, así como en distintos países del mundo.

Será necesaria la programación con xlets, aplicaciones en Java para entornos de televisión, y es la forma de desarrollar aplicaciones definida en el estándar MHP, para el desarrollo de la aplicación que actuará como interfaz para el usuario a través del televisor y será la encargada de presentar los distintos objetos de aprendizaje, que hayan sido generados, mediante el módulo de generación de objetos de aprendizaje.

4.2.3.1 El estándar MHP

El estándar MHP (Multimedia Home Platform) define una plataforma para aplicaciones interactivas a través de la Televisión Digital Terrestre [7].

Lo que ofrece básicamente es interoperabilidad entre diferentes aplicaciones y terminales por medio un interfaz genérico entre las aplicaciones digitales interactivas y los terminales en los cuales se van a ejecutar, en nuestro caso los STB.

Para la realización de este proyecto, contamos en el laboratorio con un STB que incorpora el estándar MHP y es capaz de ejecutar aplicaciones interactivas.

Una de las diferentes posibilidades que ofrece MHP a la hora de desarrollar una aplicación es la del uso del canal de retorno para poder enviar, o en su caso recibir, información.

4.2.3.2 Programación con Xlets

Los Xlets son aplicaciones en Java para entornos de televisión [8]. Son muy parecidos a los Applets o Midlets de Java, sin embargo, existen algunas diferencias entre ellos. Una de las más destacadas consiste en que si bien varias aplicaciones pueden estar ejecutándose en un STB, sólo una de ellas puede ser visible cada vez en la pantalla, a diferencia de los Applets. Por este motivo se hace necesario que los Xlets incorporen la opción de poder ser pausados, para esperar a que termine la ejecución del Xlet que se encuentra visible en pantalla, y la opción de poder ser resumidos, es decir, que estando pausados, pasen a ejecución.

No siguen el modelo típico de las aplicaciones Java en el que cada aplicación tiene el control completo de su ciclo de vida y de los recursos de la máquina virtual sobre la que se ejecutan.

Su ciclo de vida es el que se muestra en la figura 4.3, donde se pueden apreciar los cuatro estados en los que se puede encontrar un Xlet: cargado, pausado, activo y destruido.

Estas aplicaciones son iniciadas manualmente mediante el mando a distancia del Set-Top-Box.

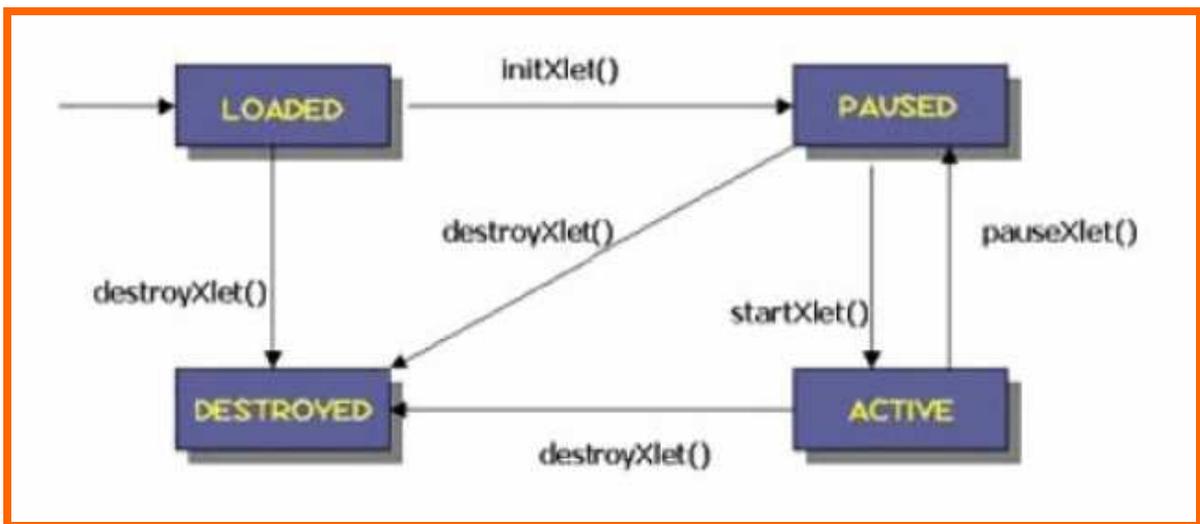


Figura 4.3. Ciclo de vida de un Xlet.

4.2.4 Seguimiento

En este apartado se identifican las herramientas y tecnologías a utilizar para poder llevar un seguimiento de las respuestas que envíen los usuarios a la hora de realizar los distintos cursos. Se plantea el uso del canal de retorno del decodificador de TDT para llevar a cabo esta tarea, a la vez que se profundiza en esta tecnología aportando información acerca de ella.

Para la construcción de este módulo, se hará uso del lenguaje de marcado HTML para poder mostrar a través de una página web, los distintos resultados a partir de la información proporcionada por los usuarios mediante las respuestas a las preguntas planteadas en los distintos programas.

Para ello, se va a emplear el canal de retorno que proporcionan los decodificadores que implementan el estándar MHP, para poder enviar las

respuestas de los usuarios al servidor donde permanece alojada la base de datos encargada de almacenar la información de los usuarios. Se ha hecho uso de la especificación proporcionada en el proyecto fin de carrera TDT-Reiven Interactivo [9], en el que se introducía la forma de poder enviar datos proporcionados por el usuario hacia el exterior.

Respecto al sistema de gestión de base de datos (SGBD) a utilizar para almacenar tanto los objetos de aprendizaje, para la elaboración de los cursos, como la información procedente de los usuarios, sobre las contestaciones que hayan proporcionado a las preguntas proporcionadas en los cursos, será MySQL, debido a que se trata de un software libre, fácil de usar y muy utilizado (se estima que en la actualidad existen más de seis millones de instalaciones por todo el mundo).

4.2.4.1 Canal de retorno en la TDT

Este mecanismo le permite al usuario poder enviar información generada por él mismo, a través de las respuestas que haya realizado en las preguntas, al exterior.

Para la realización de este proyecto se ha optado por enviar esa información a una base de datos en la que almacenar las respuestas de los usuarios para, posteriormente, tratar esa información y presentar ciertos resultados.

La TDT no incorpora el canal de retorno en su estructura de red y es necesario habilitarlo a través de otra tecnología que posibilite esta conexión.

4.2.4.2 Tecnologías disponibles para el canal de retorno

Para el envío de la información, el canal de retorno presenta tres opciones: mediante un módem, mediante la utilización de Bluetooth junto con una conexión GPRS/UMTS y mediante un cable de red Ethernet.

La utilización de un módem presenta varios inconvenientes, ya que si se emplea, el teléfono deja de ser accesible y no puede utilizarse y, por el contrario, si se está utilizando el teléfono, el canal dejará de estar accesible para las aplicaciones. Por otra parte, esta opción tiene poca capacidad, con el retardo que ello conlleva.

La opción de utilizar la tecnología Bluetooth como canal de retorno tiene la ventaja de poder ver en la televisión los archivos y fotografías del teléfono móvil, si éstos cuentan con dicha tecnología y tienen una conexión

GPRS (*General Packet Radio Service*) o UMTS (*Universal Mobile Telecommunications System*) [10].

Entre éstas dos conexiones cabe decir que la conexión GPRS posee un mecanismo de manejo de congestión mediante el cual puede continuar aceptando datos aún cuando se encuentren ocupados todos los recursos, a la vez que es capaz de establecer prioridades [11]. Este tipo de conexión se encuentra más extendida que la conexión UMTS.

Sin embargo la opción de utilizar la tecnología Bluetooth presenta también inconvenientes como puede ser la velocidad, que se encuentra en torno a los 9,6 kbps [12] y dista bastante en comparación con otras opciones como puede ser la conexión por cable de red Ethernet.

Como última opción tenemos la de conexión por cable de red Ethernet y es la que se ha optado por utilizar, ya que es la más común entre los diferentes STB's (Set-Top-Boxes) y la que mayor ancho de banda presenta.

Las alternativas al futuro canal de retorno pasarían por una de las siguientes opciones [10]:

- El uso de PLC (comunicaciones a través de la red eléctrica)
- La tecnología Wi-Fi
- La tecnología WiMAX.

El canal de retorno, pronto o tarde, tenderá a utilizar una tecnología inalámbrica para proporcionar comodidad y facilidad de uso.

4.3 Conclusiones

En este capítulo hemos visto el modelo de ciclo de vida seleccionado para la elaboración del proyecto, así como las distintas herramientas, junto con una explicación para cada una de ellas, que se pretenden utilizar para la construcción de los distintos módulos que conforman la arquitectura del sistema.

Capítulo 5. Desarrollo del trabajo

Una vez estudiadas las tecnologías que se van a utilizar para el desarrollo del sistema, tras el análisis y el diseño presentado en los puntos anteriores y previa planificación de las tareas que se van a realizar, presentamos en este capítulo la fase de codificación.

Durante este capítulo se presentará la forma de construcción de cada uno de los módulos que constituirán Conectra-TDT, explicando detalladamente el procedimiento seguido para su realización.

5.1 Generación de paquetes de contenido

Un paquete de contenidos no es más que un conjunto de recursos teóricos y/o prácticos agrupados en un "paquete" y asociado a un programa de televisión en concreto.

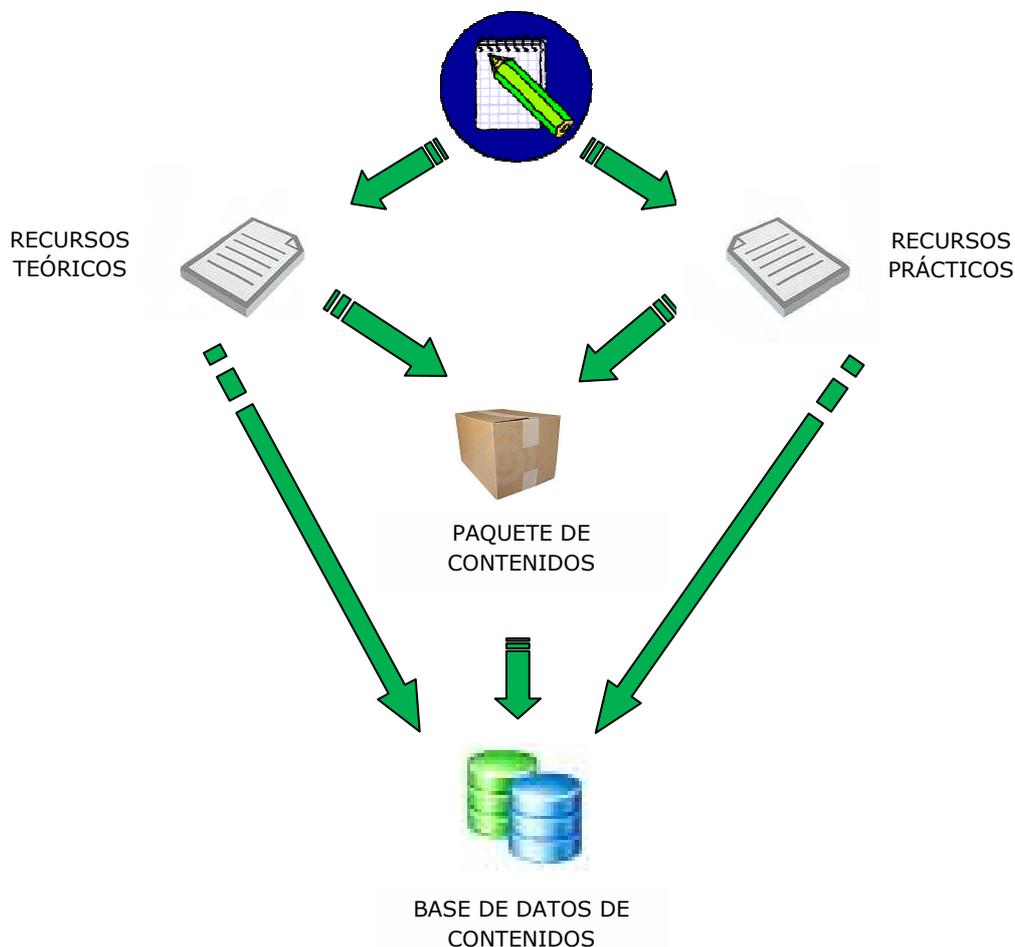


Figura 5.1. Esquema de generación de un paquete de contenidos.

A continuación se muestran dos capturas correspondientes a la generación de recursos teóricos y prácticos en Reiven:

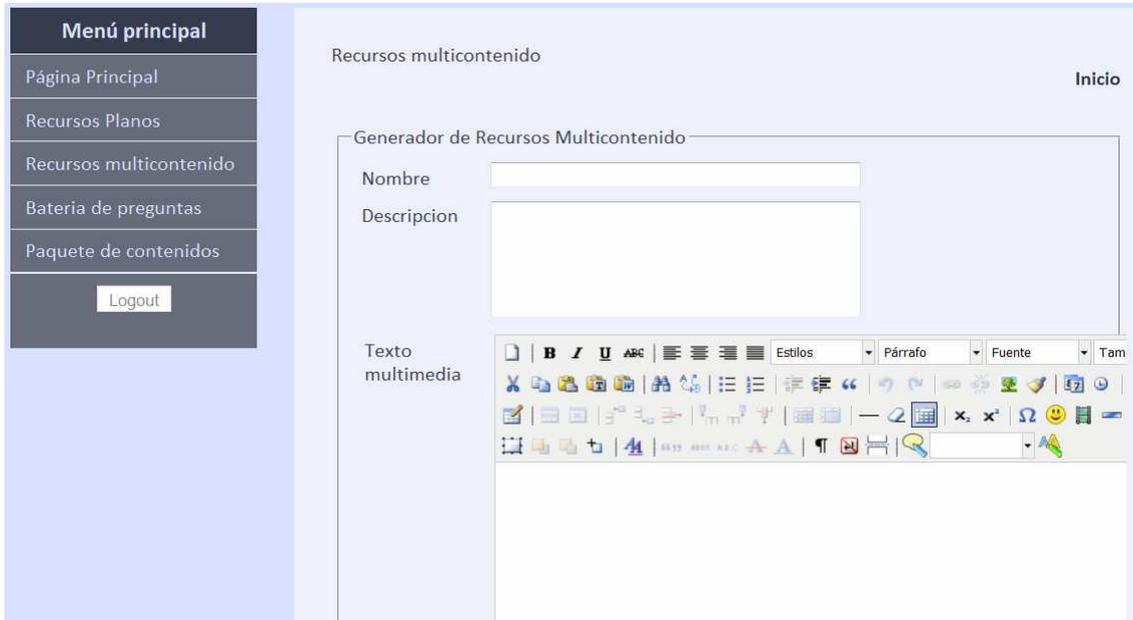


Figura 5.2. Generación de un recurso multicontenido en Reiven.

Figura 5.3. Generación de una pregunta en Reiven.

Los contenidos teóricos constan de un título, descripción y un contenido multimedia en hipertexto (Fig5.2).

Los contenidos prácticos se construyen como baterías de preguntas, cada una de estas preguntas dispone de un enunciado y un mecanismo de respuesta. Las preguntas pueden ser de tipo selección simple, múltiple, ordenar,...

Para la generación de paquetes de contenido se ha utilizado un módulo de la herramienta Reiven⁵, para adaptar los contenidos generados a las características propias de un servicio de t-learning. Se ha modificado ampliando el número de funcionalidades. Éstas son:

1. **Marcas de tiempo:** A la hora de añadir un objeto de aprendizaje teórico o práctico al paquete, se le añade a cada objeto de aprendizaje, un campo en el que introducir el minuto en el que se quiere que aparezca dicho recurso en la emisión del programa. (Fig.5.5-1 y Fig.5.6-1)
2. **Feedback opcional:** Añadir un campo de selección para los objetos de aprendizaje prácticos para poder dar la opción de mostrar o no la solución correcta a las preguntas. (Fig.5.6-2)
3. **Asociación a programa:** Añadir un campo en el que poder introducir el código del programa al que será asociado el paquete de contenidos. (Fig.5.5-3)
4. **URL de recepción de respuestas:** Añadir un nuevo campo para poder introducir la URL a la que se deberán enviar las respuestas de los usuarios a las preguntas del curso. (Fig.5.5-4)

⁵ La herramienta de autor Reiven fue presentada en el capítulo 2: Ambito del proyecto.

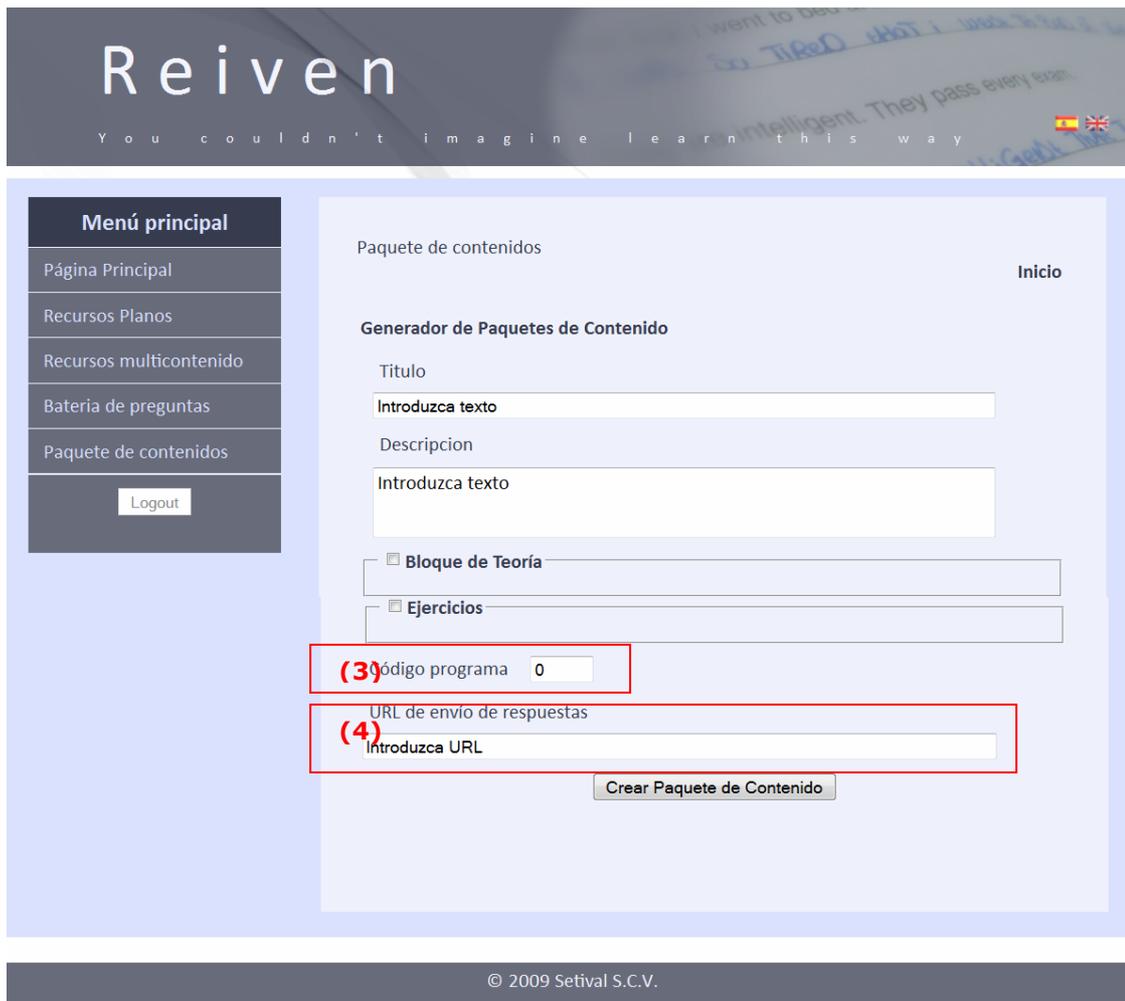


Figura 5.4. Generación de un paquete de contenidos en Reiven.

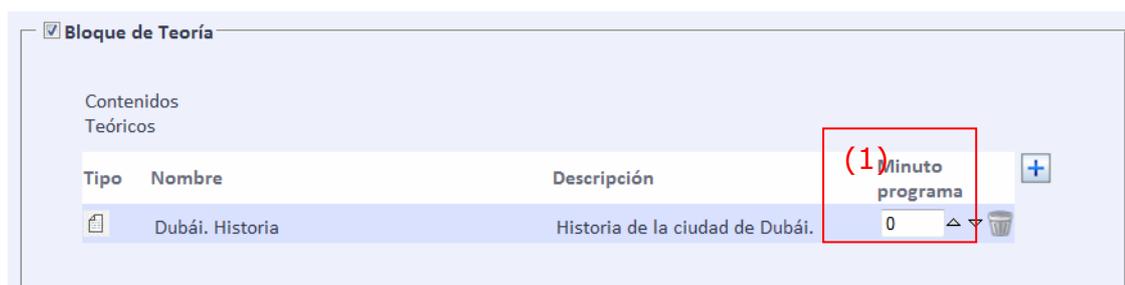


Figura 5.5. Bloque de teoría de un paquete de contenidos en Reiven.

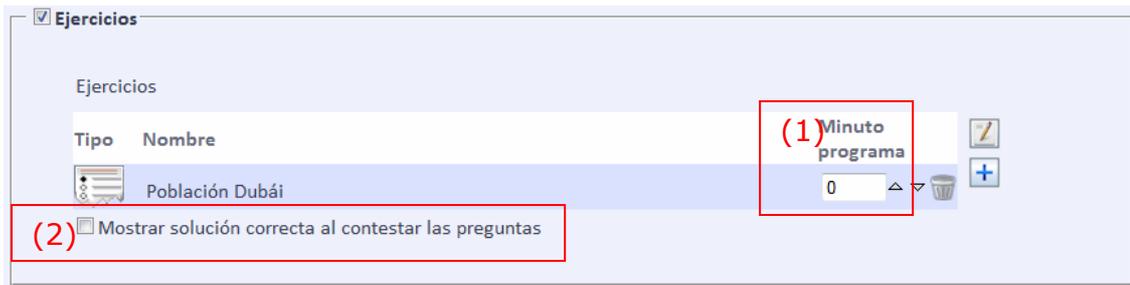


Figura 5.6. Bloque de ejercicios de un paquete de contenidos en Reiven.

5.1.1 Base de datos de objetos de aprendizaje

Para la gestión de los contenidos construidos a través de la herramienta de autor hacemos uso del siguiente modelo de datos:

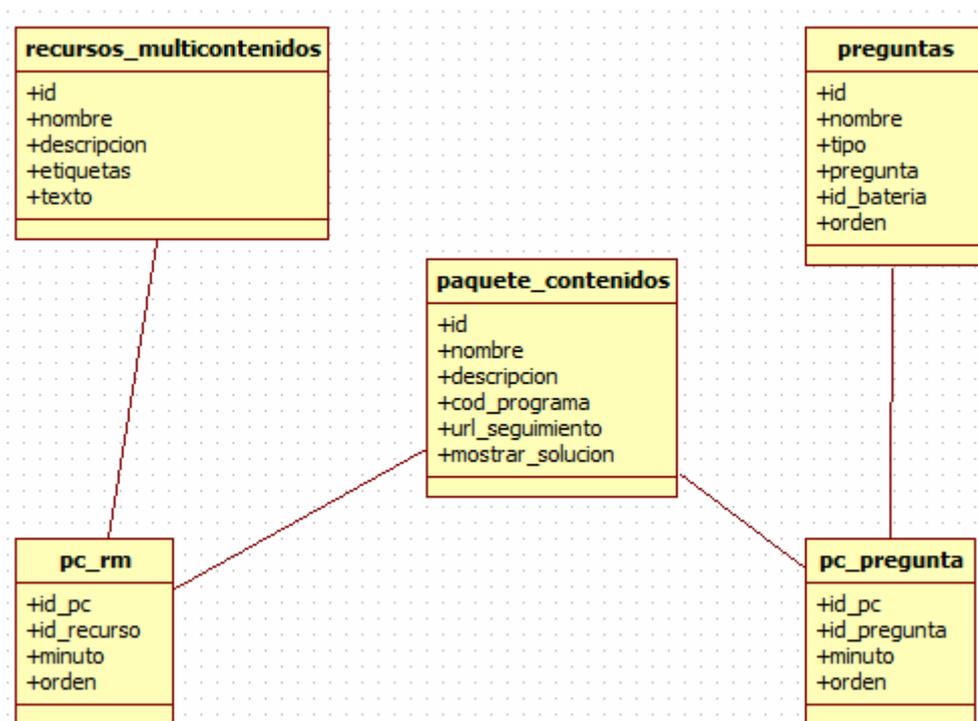


Figura 5.7. Diagrama de clases de la Base de datos de objetos de aprendizaje.

1. Los objetos de aprendizaje teóricos se almacenan en una tabla llamada "recursos_multicontenidos" con los siguientes campos:
 - id: identificador del recurso teórico.

- nombre: nombre del recurso teórico.
- descripción: pequeña descripción sobre lo que trata el recurso teórico.
- etiquetas: para poder asociarle unas etiquetas que identifiquen el contenido del recurso.
- texto: contenido del recurso

2. Los objetos de aprendizaje prácticos se almacenan en una tabla llamada "preguntas" que presenta los siguientes campos:

- id: identificador del recurso práctico.
- nombre: nombre del recurso práctico.
- tipo: tipo de la pregunta que puede ser:
 - 1: Selección simple, se selecciona una respuesta entre varias.
 - 2: Emparejar, se relacionan las respuestas de una columna con las respuestas de la otra.
 - 3: Desplegable, se selecciona una respuesta de entre las posibles opciones del menú desplegable.
- pregunta: contenido de la pregunta, tanto enunciado como posibles respuestas.
- id_bateria: batería de preguntas a la que se encuentra asociada la pregunta, esto se explicará a continuación.
- orden: orden de la pregunta en la batería.

Existen baterías de preguntas que engloban un conjunto de preguntas, de forma que se pueden encapsular preguntas que, por ejemplo, traten del mismo tema.

Es por ello por lo que cada pregunta tiene un campo en el que se indica la batería a la que pertenece.

3. En la base de datos también podemos encontrar una tabla llamada "paquete_contenidos" en la que se almacenan los cursos con sus respectivos parámetros. Esta tabla presenta los siguientes campos:

- id: identificador del paquete de contenidos.

- nombre: Nombre del paquete de contenidos.
 - descripción: descripción del paquete de contenidos.
 - cod_programa: código del programa al que se encuentra asociado el paquete.
 - url_seguimiento: Dirección web donde enviar las respuestas de los usuarios a las preguntas.
 - mostrar_solucion: indica si deben aparecer o no las respuestas a las preguntas.
4. Para asociar tanto los recursos teóricos como los prácticos con un paquete de contenidos, es decir, un curso, se han creado dos tablas, una para asociar los recursos teóricos con un curso y otra para asociar los recursos prácticos con un curso:
1. La tabla llamada "pc_rm" sirve para asociar los recursos teóricos con los paquetes de contenido y presenta los siguientes campos:
 - id_pc: identificador de un paquete de contenidos.
 - id_recurso: identificador de un recurso de teoría.
 - minuto: minuto de aparición del recurso de teoría con identificador "id_recurso" en el curso con identificador "id_pc".
 - orden: orden del recurso teórico en el curso.
 2. La tabla llamada "pc_pregunta" sirve para asociar los recursos prácticos con los paquetes de contenido y presenta los siguientes campos:
 - id_pc: identificador de un paquete de contenidos.
 - id_pregunta: identificador de una pregunta.
 - minuto: minuto de aparición de la pregunta con identificador "id_pregunta" en el curso con identificador "id_pc".
 - orden: orden de la pregunta en el curso.

5.2 Creación de un curso

Un curso lo podríamos definir como un objeto de aprendizaje fruto de la agregación de recursos teóricos y/o prácticos dispuestos en un formato que permita ser leído por la plataforma consumidora de estos objetos de aprendizaje.

Los cursos han sido codificados en una estructura basada en XML, que nos permite abstraer su contenido mediante un modelo orientado a objetos.

Como ya hemos introducido un curso estará compuesto por recursos teóricos y/o recursos prácticos con sus respectivos tiempos de aparición en un determinado programa.

La manera en la que se obtienen estos recursos para la construcción del curso se presentará más adelante en el punto 5.2.4. En primer lugar detallaremos la forma en la que se estructura y codifica el curso.

Un curso presenta la estructura de directorios y ficheros como la que se muestra en la figura 5.8.

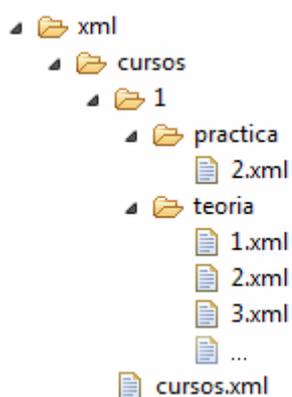


Figura 5.8. Estructura de directorios y ficheros de un curso.

La creación de los ficheros de un curso se encuentra dividida en tres procesos:

1. Codificación de los recursos teóricos: corresponde a cada uno de los ficheros xml contenidos en el directorio "teoría".
2. Codificación de los recursos prácticos: corresponde al fichero "2.xml" contenido en el directorio "práctica".

3. Creación del índice: corresponde al fichero "cursos.xml" contenido en el directorio cursos.

5.2.1 Creación de los recursos teóricos

Para cada uno de los recursos teóricos se genera un fichero xml (ver figura 5.9) compuesto por las siguientes etiquetas:

```
<?xml version="1.0" encoding="UTF-8"?>
<rm type="texto" title="Historia Dubái">
  <description>
    Establecimiento unidad monetaria
  </description>
  <html>
    <p>En 1973, Dubái se unió a los demás emiratos
    y adoptó una divisa única y uniforme: el dirham
    de los EAU. </p>
  </html>
</rm>
```

Figura 5.9. Fichero xml de ejemplo de un recurso teórico.

1. <rm></rm>: etiquetas que engloban al resto de etiquetas del documento. Presentan dos atributos:
 - a. type: indica de que tipo es el contenido del recurso, texto, imágenes, ...
 - b. title: indica el título del recurso teórico.
2. <description></description>: etiquetas que contienen la descripción del recurso.
3. <html></html>: etiquetas que contienen los distintos párrafos del contenido del recurso.

5.2.2 Creación de los recursos prácticos

Se genera un único fichero xml (ver figura 5.10) para el conjunto de recursos prácticos.

Cada uno de los recursos prácticos (preguntas) listado en dicho fichero xml está codificado siguiendo el estándar IMS-QTI 1.2.

```

<?xml version="1.0" encoding="UTF-8"?>

<assessments title="preguntas" random="0">
  <items>
    <assessmentItem identifier="490"
      title="Título de pregunta simple"
      adaptative="false" timeDependent="false">
      <outcomeDeclaration identifier="SCORE"
        cardinality="single" baseType="integer">
        <defaultValue><value>0</value></defaultValue>
      </outcomeDeclaration>
      <responseDeclaration identifier="RESPONSE"
        cardinality="single" baseType="identifier">
        <correctResponse><value>R0</value></correctResponse>
      </responseDeclaration>
      <itemBody>
        <html><p>Enunciado de pregunta simple</p></html>
        <choiceInteraction responseIdentifier="RESPONSE"
          shuffle="true" maxChoices="1">
          <prompt></prompt>
          <simpleChoice identifier="R0">
            Respuesta correcta simple
            <feedbackInline outcomeIdentifier="FEEDBACK"
              identifier="R0" showHide="show">
            </feedbackInline>
          </simpleChoice>
          <simpleChoice identifier="R1">
            Respuesta incorrecta simple
            <feedbackInline outcomeIdentifier="FEEDBACK"
              identifier="R1" showHide="show">
            </feedbackInline>
          </simpleChoice>
        </choiceInteraction>
      </itemBody>
      <modalFeedback outcomeIdentifier="FEEDBACK"
        identifier="" showHide="show">
      </modalFeedback>
      <modalFeedback outcomeIdentifier="FEEDBACK"
        identifier="" showHide="hide">
      </modalFeedback>
    </assessmentItem>
  </items>
</assessments>

```

Figura 5.10. Fichero xml de ejemplo de un recurso práctico con una pregunta simple.

Se muestran las etiquetas más importantes para la creación de un curso.

1. <assessments></assessments>: etiquetas que engloban la descripción, si existe, y las preguntas del curso.
2. <items> </items>: etiquetas que engloban todas las preguntas del curso.

3. <assessmentItem></assessmentItem>: etiquetas que contienen una pregunta del curso.
4. <responseDeclaration></responseDeclaration>: etiquetas que contienen la cardinalidad de la pregunta en su atributo "cardinality" además de la/s respuesta/s correcta/s a la pregunta/s.
5. <itemBody></itemBody>: etiquetas que contienen el enunciado de la pregunta y las posibles respuestas.
6. <simpleChoice></simpleChoice>: etiquetas que contienen una respuesta de una pregunta.
7. <modalFeedback></modalFeedback>: etiquetas que contienen la información que se le presentará al alumno, una vez haya contestado a todas las preguntas, dependiendo de las respuestas que haya dado.

5.2.3 Creación del índice

El índice del curso se encuentra codificado en un fichero xml (ver figura 5.11) que se genera con la siguiente estructura:

```
<?xml version="1.0" encoding="UTF-8"?>
<reiven>
  <curso id="1" title="curso"
    url="http://www.interno.setival.com/" navegar="false"
    solucion="true" cod_programa="27">
    <description>descripción del curso</description>
    <unidades>
      <ut identifier="278" minuto="2">Unidad 1</ut>
      <ut identifier="279" minuto="8">Unidad 2</ut>
      <ut identifier="280" minuto="14">Unidad 3</ut>
      <ut identifier="281" minuto="20">Unidad 4</ut>
      <ut identifier="282" minuto="26">Unidad 5</ut>
      <up identifier="2">
        <label>Preguntas</label>
        <QuestionEvent identifier="492" minuto="6"/>
        <QuestionEvent identifier="493" minuto="12"/>
        <QuestionEvent identifier="494" minuto="18"/>
        <QuestionEvent identifier="495" minuto="24"/>
        <QuestionEvent identifier="496" minuto="28"/>
      </up>
    </unidades>
  </curso>
</reiven>
```

Figura 5.11. Fichero xml de ejemplo del índice de un curso.

1. <reiven></reiven>: etiquetas que contienen todo el índice del curso.
2. <curso></curso>: etiquetas que contienen la descripción y las unidades teóricas y/o prácticas del curso. Presentan cinco atributos.
 - a. "id": identificador del curso.
 - b. "title": título del curso
 - c. "url": dirección web donde enviar las respuestas de los usuarios.
 - d. "solucion": si se encuentra a "true" indica que al contestar una pregunta debe indicarse la respuesta correcta al usuario, y si se encuentra a "false" indica que no se deben mostrar las respuestas correctas de las preguntas al usuario.
 - e. "cod_programa": código del programa al que se encuentra asociado el curso.
3. <description></description>: etiquetas que contienen la descripción del curso.
4. <unidades></unidades>: etiquetas que contienen los recursos teóricos y/o prácticos que contiene el curso.
5. <ut>: recurso teórico presente en el curso con su título. Contiene dos atributos:
 - a. "identifier": identificador del recurso teórico.
 - b. "minuto": minuto en el que tiene que aparecer el recurso teórico en el programa.
6. <up>: recursos prácticos que contiene el curso codificados en el fichero xml que indica su atributo, que es:
 - a. "identifier": identificador del fichero xml que contiene los recursos prácticos.
7. <label>: contienen el título del paquete de preguntas del curso.
8. <QuestionEvent>: recurso práctico presente en el curso. Contiene dos atributos:
 - a. "identifier": identificador del recurso práctico.
 - b. "minuto": minuto en el que tiene que aparecer el recurso práctico en el programa.

Una vez se tiene el directorio xml con sus respectivos subdirectorios y los ficheros xml generados (ver figura 5.8), se comprime todo en un fichero zip. Este fichero será el paquete de contenidos que encapsula el curso, listo para transmitirse a la aplicación desde la cual se consumirán los contenidos.

5.2.4 Obtención de los recursos teóricos y prácticos para la creación de un curso

Para la creación de un curso es necesario disponer de la descripción de un programa, así como del código y duración del mismo, todo ello extraído de la EPG del programa.

Como hemos mencionado anteriormente, la EPG consiste en la guía electrónica del programa (Electronic Program Guide), en la que podemos encontrar todos los detalles del programa en cuestión.

Tal y como muestra figura 5.12 podemos observar que existen dos formas de obtener los objetos de aprendizaje para la creación de los cursos:

1. Seleccionando automáticamente un paquete de contenidos asociado al código de programa de la emisión.
2. Generando un nuevo objeto de aprendizaje agregando contenidos a partir de la descripción del programa en emisión.

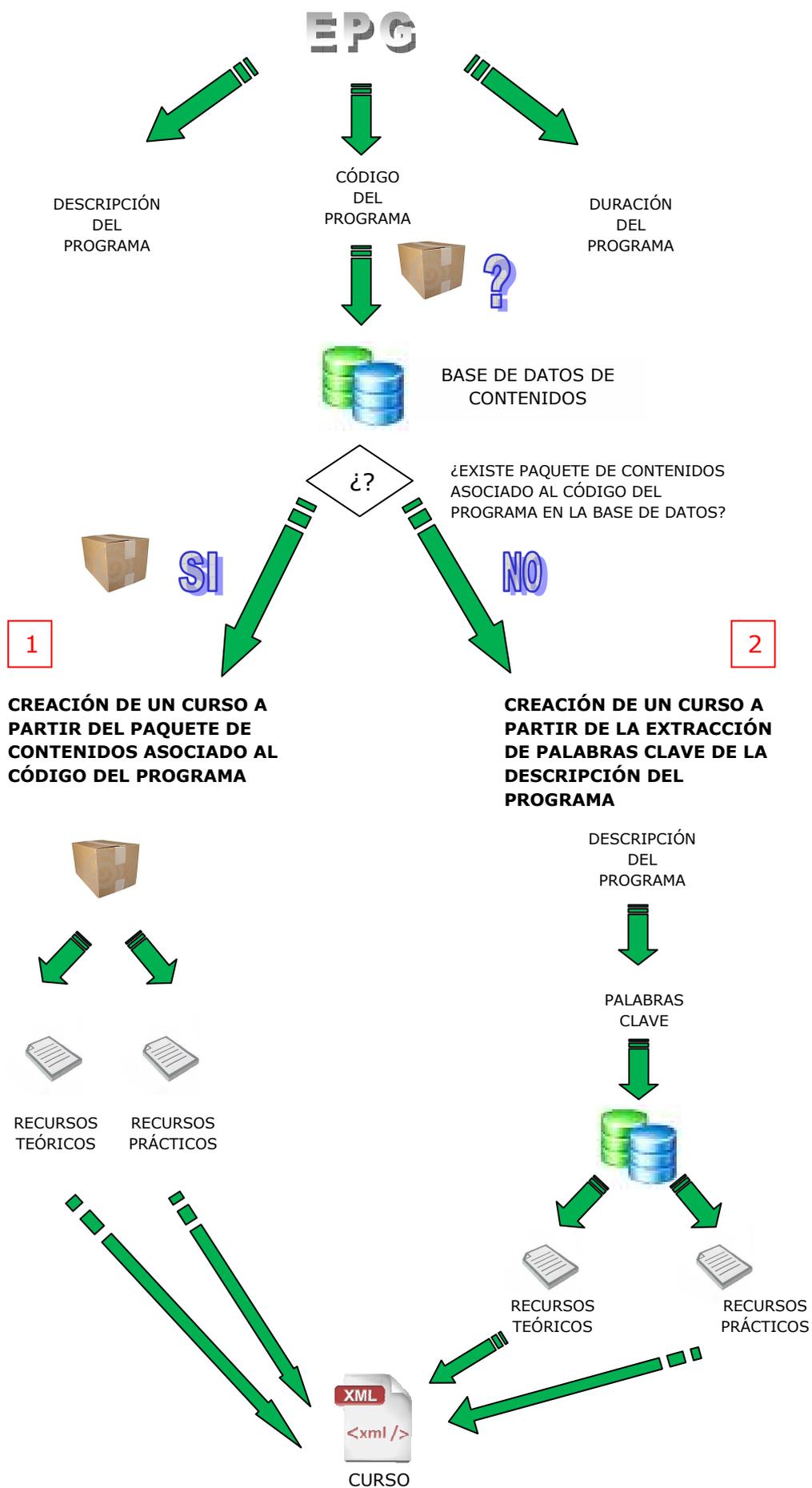


Figura 5.12. Obtención de los recursos teóricos y prácticos para la creación de un curso.

Los pasos que se siguen son dos y en este mismo orden:

1. Se comprueba si existe en la base de datos un paquete de contenidos, generado previamente mediante la herramienta de autor, asociado al código de programa que se dispone y, si es así, se obtienen todos los recursos y parámetros del paquete de contenidos y se procede a crear el curso tal y como hemos visto en los puntos anteriores.
2. Si no existe ningún paquete de contenidos asociado al código de programa del que se dispone, se procede de la siguiente forma:
 1. Se extraen las palabras clave de la descripción del programa. (Ver punto 5.2.4.1)
 2. Se obtienen recursos teóricos y prácticos de la base de datos de contenidos, a partir del conjunto de palabras clave generado en el punto anterior. (Ver punto 5.2.4.2)
 3. Se crea un curso a partir de los recursos teóricos y/o prácticos obtenidos en el punto anterior.

Los siguientes puntos del capítulo profundizan en la obtención de los recursos teóricos y prácticos siguiendo el segundo método, es decir, la creación de un curso a partir de la extracción de palabras clave de la descripción del programa.

5.2.4.1 Extracción de palabras clave

Para la extracción de las palabras clave de la EPG de un programa se ha diseñado una aplicación en PHP que se basa en calcular la frecuencia de aparición de las palabras en el texto, obviando los artículos, las preposiciones y los adverbios y fusionando los sinónimos.

Se estructura de la siguiente forma:

- 1 Se eliminan todas aquellas palabras que sean artículos, preposiciones o adverbios además de letras sueltas y signos de puntuación.
- 2 Se ordenan las palabras alfabéticamente. Este paso se realiza por cuestiones de eficiencia y las razones se detallan en el siguiente punto.

- 3 Calculamos la frecuencia de aparición de cada palabra en el texto. El hecho de haber ordenado las palabras alfabéticamente en el punto anterior permite no tener que hacer que, para cada palabra, sea necesario mirar todo el resto de palabras del texto (ver figura 5.13), sino que, al estar ordenadas, únicamente hace falta mirar la siguiente palabra y comprobar si es la misma (ver figura 5.14) y, en ese caso, aumentar en 1 la frecuencia de aparición de la palabra. Este proceso se repite para todas las palabras del texto.

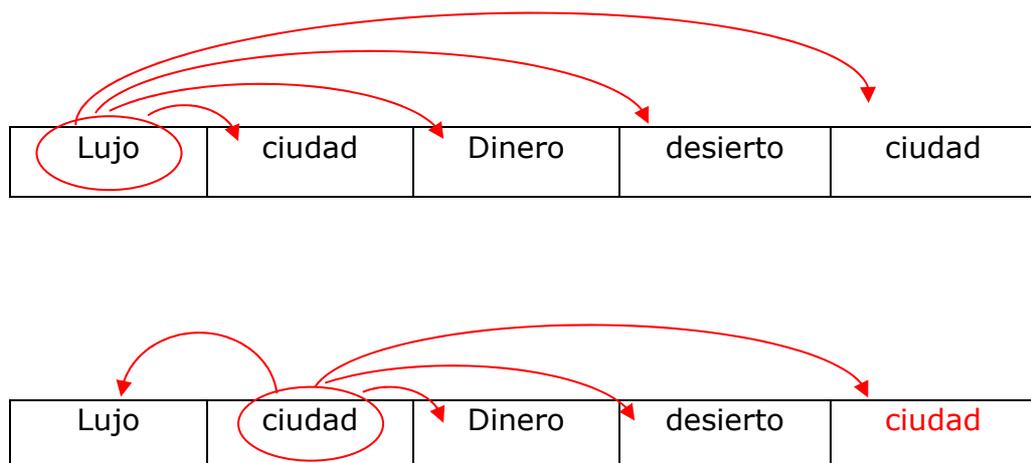


Figura 5.13. Recorrido ineficiente de un conjunto de palabras.

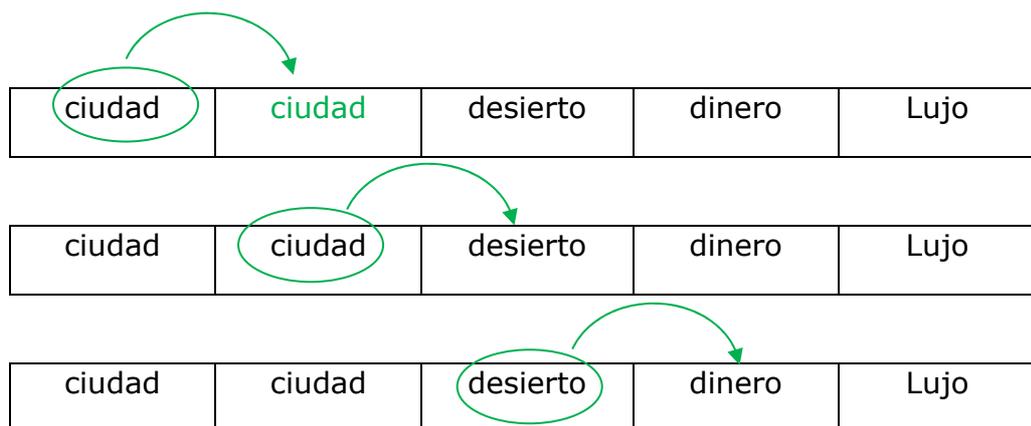


Figura 5.14. Recorrido eficiente, previa ordenación, de un conjunto. de palabras.

- 4 Fusionamos las palabras que son sinónimos, es decir, eliminamos la palabra que es sinónimo de la palabra que estamos mirando y le sumamos la frecuencia de aparición que tenía la palabra eliminada.

Para la búsqueda de sinónimos de una palabra se dispone de un fichero con palabras y un conjunto de sinónimos para cada palabra.

- 5 Por último, se coge un número determinado de palabras clave, tantas como queramos obtener, hasta un máximo del número de palabras significativas del texto. Se ha decidido obtener siete palabras clave para generar un curso. En el caso de que se obtengan menos palabras clave de las que queremos, nos quedamos con las que se encuentren y se genera un curso con ellas. Las palabras clave que obtenemos como resultado se encuentran ordenadas de mayor a menor frecuencia de aparición en el texto.

5.2.4.2 Obtención de recursos teóricos y prácticos a partir de un número definido de palabras clave

5.2.4.2.1 Ponderación de las palabras clave

Para crear un curso a partir de palabras clave, es necesario dar prioridad entre las palabras, es decir, darle un peso a cada una, para saber, de esta forma, que palabras son más significativas respecto a la descripción del programa de televisión.

Para establecer el peso de las palabras clave, en un principio se optó por que fuera el propio operario quien se encargara de ello, debido a que el operario disponía de información sobre la temática del programa de televisión y, en consecuencia, podría ponderar de una manera óptima las palabras; sin embargo, esto conllevaba en cierto modo, la desaparición de la idea de que el módulo fuera un proceso totalmente automático, además de la incomodidad de que el operario tuviera que ponderar las palabras clave una a una.

Es por ello por lo que se decidió automatizar totalmente el proceso y ponderar las palabras clave automáticamente.

Dado que las palabras clave ya vienen ordenadas de mayor a menor según la frecuencia de aparición en la descripción del programa, se decidió aprovechar esto y ponderar de forma incremental las palabras desde la última a la primera, de forma que la primera palabra, que es la que más veces aparece en el texto, es la que más porcentaje obtiene.

5.2.4.2.2 Búsqueda de objetos de aprendizaje en la base de datos

Las tablas en las que se realizan las búsquedas son dos:

- Tabla de recursos multicontenidos: tabla que contiene los objetos de aprendizaje teóricos.
- Tabla de preguntas: tabla que contiene los objetos de aprendizaje prácticos.

Se recorre el conjunto de palabras clave ya ordenadas según su peso y , para cada una de ellas, se busca en la tabla de recursos multicontenidos si aparece la palabra en el contenido de alguno de los campos que la forman y, si es así, se guarda el identificador del recurso. Una vez hecho esto se eliminan las posibles repeticiones de recursos para cada palabra clave.

Con esto obtenemos un conjunto de recursos teóricos, sin recursos repetidos, para una palabra clave, pero sólo nos quedamos con unos cuantos de ellos, dependiendo del peso que tenga la palabra. Para seleccionar el conjunto final que formarán parte del curso se ordenan los recursos teóricos y se seleccionan aquellos en los que se ha encontrado un mayor número de correspondencias con las palabras clave.

¿Cómo calculamos cuántos objetos de aprendizaje se han de coger para cada palabra clave?, bien, para ello multiplicamos el número de recursos total que necesitamos, por el peso de la palabra clave y redondeando al entero más próximo, siendo el número total de recursos necesarios, un valor arbitrario definido previamente.

En el caso en el que el número de recursos obtenidos para una palabra clave resulte insuficiente para construir un paquete de contenidos, la solución que se ha implementado es quedarnos con todos los recursos encontrados y guardarnos el número de los que faltan para sumárselo a la siguiente palabra clave, es decir, si faltan y recursos para completar los recursos que se necesitan para una palabra, los recursos que se necesitarán para la siguiente palabra serán $x + y$, siendo x los recursos necesarios para esta última palabra, y así sucesivamente hasta que no queden palabras clave.

Si llegados a este punto, es decir, que ya se han obtenido todos los recursos teóricos que se necesitaban para cada palabra clave, dependiendo de su peso, y no se ha llegado a completar el número de recursos teóricos totales necesarios, hay que implementar alguna solución para completarlos.

La solución implementada consiste en realizar una segunda pasada por el conjunto de palabras clave y mirar para cada una de ellas, empezando por la primera que es la de mayor peso, si se han encontrado

más recursos teóricos de los necesarios para la palabra en cuestión y, en ese caso, incluirlos en el resultado total y así sucesivamente para todas las palabras hasta tener todos los recursos teóricos necesarios.

En el caso de que no se encuentren tantos recursos como indica el número total de recursos necesarios, para todas las palabras clave, se obtendrán únicamente los encontrados.

Análogamente se procede de la misma forma que la descrita anteriormente para los recursos prácticos.

5.2.4.2.3 Establecer marcas de tiempo en un objeto de aprendizaje

Las marcas de tiempo nos permiten establecer el momento temporal en el que deben ser presentados cada uno de los contenidos a lo largo de la emisión del programa.

Para ello necesitamos saber la duración del programa, información que se extrae de la EPG del mismo.

A partir de la duración del programa se reparten de una forma equitativa la aparición de los distintos contenidos, tanto objetos de aprendizaje teóricos como prácticos.

Partiendo del número resultante de la división entre la duración del programa y el número de recursos totales, e incrementando en ese número tantas veces como recursos se tengan, establecemos el minuto para cada recurso. Esto quedará más claro si lo vemos con un ejemplo:

- Ejemplo:

Duración programa: 30 minutos

recursos teoría: 4

duración programa / recursos teoría: $30 / 4 = 7.5$

minuto de cada recurso: 7.5, 15, 22.5, 30

5.3 Envío de la aplicación consumidora de los cursos al STB

Los contenidos generados por la herramienta de autor que se han descrito en la sección anterior se transmiten empaquetados en la propia aplicación cliente que permitirá que dichos contenidos sean consumidos por el usuario final en su dispositivo de televisión.

Esta aplicación consumidora de los cursos se envía multiplexada sobre la señal de televisión al decodificador. Este proceso se lleva a cabo a través de los Transport Streams (TS).

Un TS es un tipo de formato de contenedor digital, especificado en el estándar MPEG-2, que contiene trozos de señal de video, audio o datos.

El objetivo de este apartado consiste en:

1. Analizar la composición y las características de un Transport Stream.
2. Mostrar cómo se multiplexa y envía la aplicación al decodificador.

5.3.1 Generación de un Transport Stream [13]

Para generar un TS, el punto de partida son los Elementary Streams (ES).

Un ES consiste en un flujo binario comprimido correspondiente a datos, video, audio, etc...

El primer paso es dividir el ES en paquetes. Estos paquetes reciben el nombre de paquetes PES y están compuestos por una cabecera y una parte de información útil. La longitud de la información útil es variable, de hasta 64 KBytes.

El segundo paso en la generación de un TS, es seleccionar los paquetes PES e insertarlos en pequeños paquetes de 188 bytes. Si ocupan menos de 188 bytes, cosa que ocurre la mayoría de las veces ya que es muy difícil que se rellenen exactamente los 188 bytes, el resto contiene datos nulos. Estos paquetes únicamente pueden contener datos de un solo paquete PES.

En la cabecera de un paquete TS podemos encontrar el campo PID. Este campo indica a qué ES pertenece el paquete TS.

En la figura 5.15 podemos ver un ejemplo gráfico de un paquete TS.

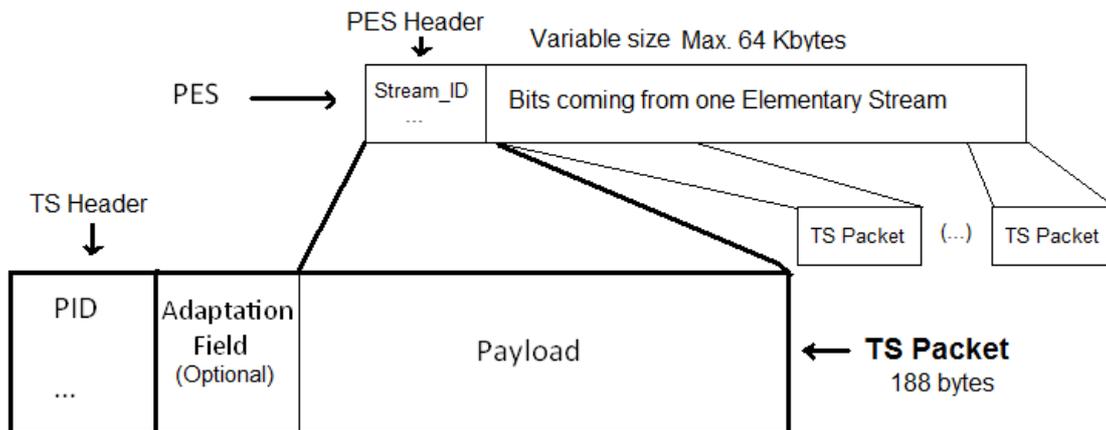


Figura 5.15. Paquete TS.

Para la generación de un TS utilizamos el sistema operativo Linux y, previa descarga del paquete, ejecutamos en la consola el siguiente script que presenta las siguientes opciones:

```
oc-update.sh object_carousel_directory association_tag module_version
                dsmcc_pid carousel_id [compress_on] [padding_on] [clean_off]
```

Un ejemplo de ejecución sería el siguiente:

```
>oc-update.sh ocdir1 0xB 1 2003 1 1 0 0
```

Donde el campo que nos interesa es "ocdir1", que representa el directorio que queremos convertir a un TS.

Para generar los TS de la aplicación primero deben compilarse todos los ficheros java, de forma que se obtienen los ficheros class, que corresponden a los ficheros java compilados y, posteriormente, se añaden todos los ficheros class a un directorio.

5.3.2 Multiplexación y envío de la aplicación al STB

Para multiplexar la aplicación junto con la señal de televisión utilizamos el programa tscbrmuxer para el sistema operativo de Linux, el cual proviene del mismo paquete que el script oc-update.sh.

La utilización del programa es la siguiente:

```
tscbrmuxer ancho-de-banda_1 archivo_1.ts ancho-de-banda_2
archivo_2.ts
```

donde en ancho de banda de cada TS indica el número de bytes que se quieren multiplexar de ese TS.

Un ejemplo de ejecución en consola sería el siguiente:

```
>Tscbrmuxer 1024 video.ts 780 audio.ts 570 aplicación.ts
```

Todo este sistema de generación de TS, multiplexación y envío al decodificador, se resume de forma gráfica en la figura 5.16.

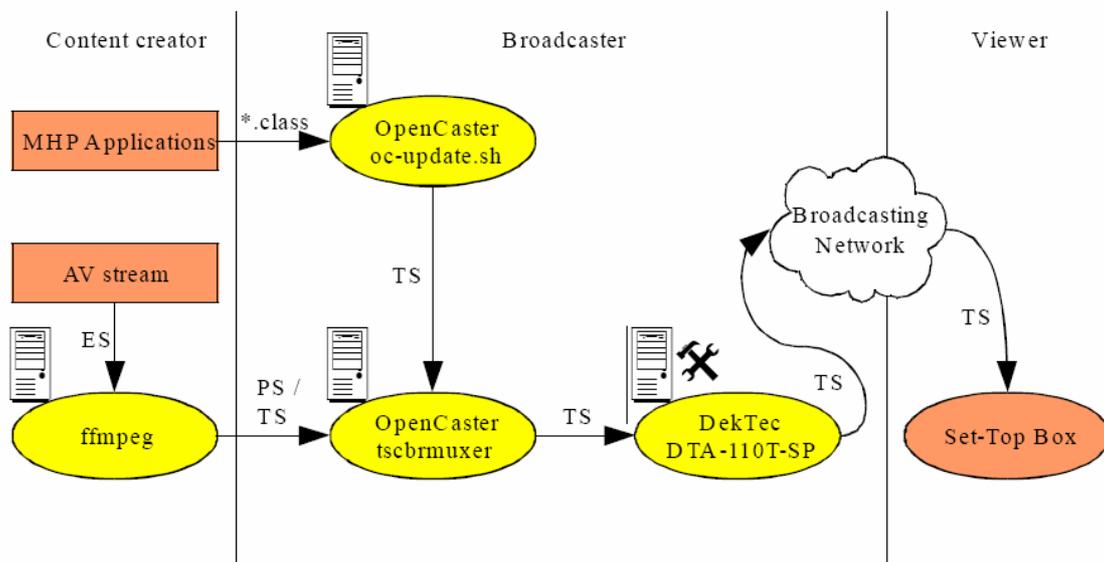


Figura 5.16. Generación, multiplexación y envío de un TS al STB.

5.4 Aplicación consumidora de los cursos

Este módulo del proyecto tiene como objetivo crear una aplicación que sea capaz de mostrar en el televisor los cursos generados previamente, mostrando cada recurso teórico y práctico en el instante que le corresponda.

Partiendo de los resultados obtenidos en el proyecto TDT-Reiven se han añadido las nuevas funcionalidades requeridas para nuestro proyecto. Estas son:

- Elaboración de una nueva interfaz con menos intrusión visual respecto al video visionado.

- Aparición en pantalla de la teoría y las preguntas en el instante correspondiente a cada una de ellas, sin la intrusión visual de la aplicación en el video visionado durante el resto del tiempo.
- Envío de las respuestas aportadas por los espectadores a un servidor encargado de almacenarlas en una base de datos.

En la figura 5.17 podemos observar la aplicación en funcionamiento donde se muestra la forma de presentación de un recurso teórico por pantalla. Vemos que, tal y como se indicaba en los requerimientos, la intrusión visual de la aplicación respecto al video visionado es mínima, ofreciendo al espectador la posibilidad de dejar de visualizar el recurso teórico con simplemente apretar un botón.



Figura 5.17. Visualización de un recurso teórico en la televisión.

En la figura 5.18 aparece la forma de representar por pantalla un recurso práctico, concretamente del tipo de relacionar elementos. Cabe destacar que, al igual que con los recursos teóricos, el espectador también tiene la posibilidad de dejar de visualizar la pregunta con el simple hecho de apretar un botón.



Figura 5.18. Visualización de un recurso práctico del tipo relacionar elementos en la televisión.

Los recursos prácticos del tipo menú desplegable con opciones se visualizan tal y como se muestra en la figura 5.19.



Figura 5.19. Visualización de un recurso práctico del tipo menú desplegable con opciones en la televisión.

Otro tipo de recurso práctico es el de selección entre una lista de elementos, que podemos ver su visualización en la figura 5.20.



Figura 5.20. Visualización de un recurso práctico del tipo selección entre una lista de elementos en la televisión.

La aplicación ofrece la posibilidad de navegar a través de los recursos, pudiendo contestar a las preguntas, en cualquier momento.

Para ello ofrece una interfaz que podemos ver en la figura 5.21., la cual nos permite realizar dicha navegación de una forma muy intuitiva.

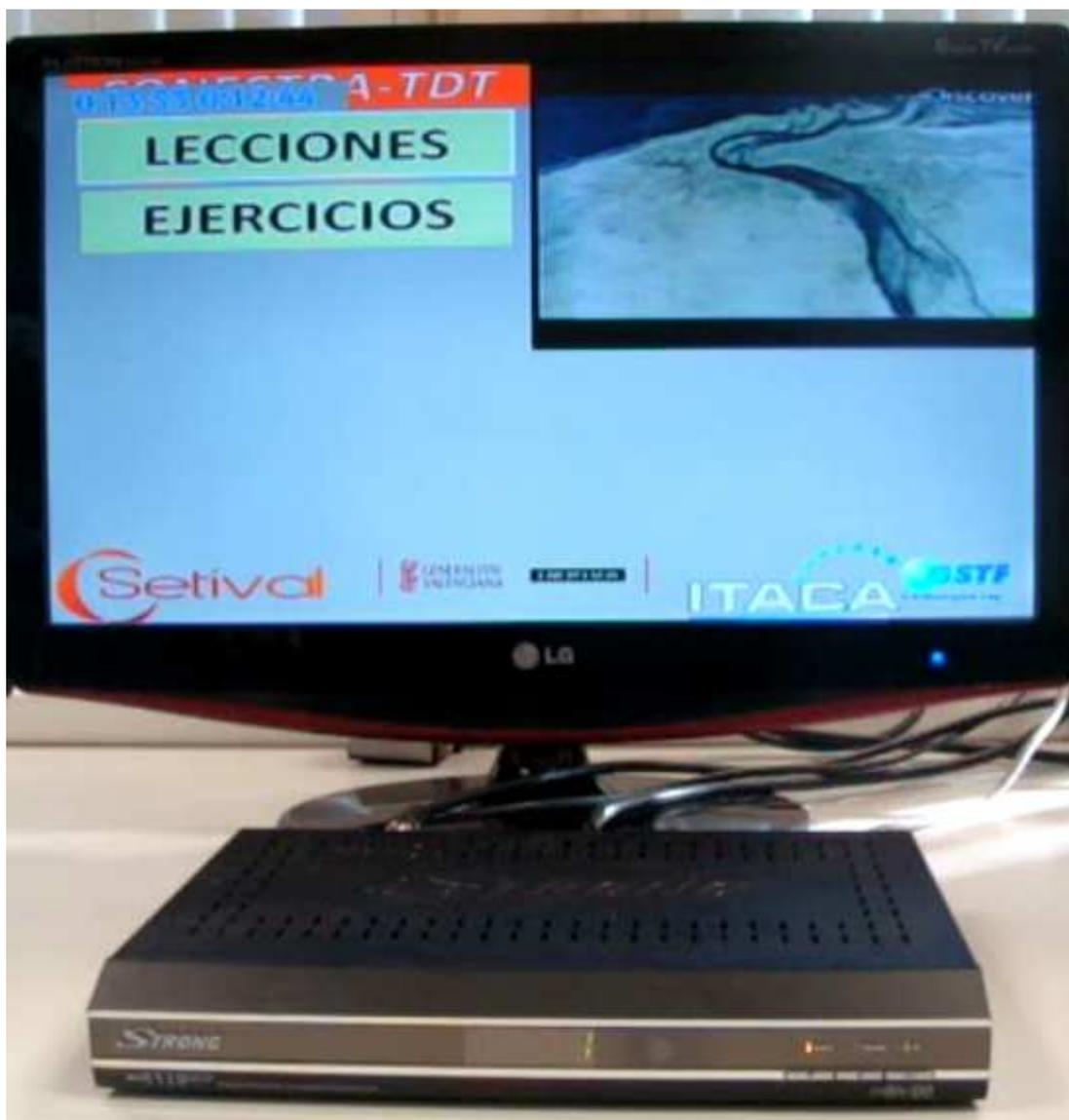


Figura 5.21. Menú principal para la navegación por un curso en la televisión.

En la figura 5.22 podemos observar la navegación por los recursos teóricos pudiendo seguir visualizando el video en una ventana en la esquina superior derecha de la pantalla.

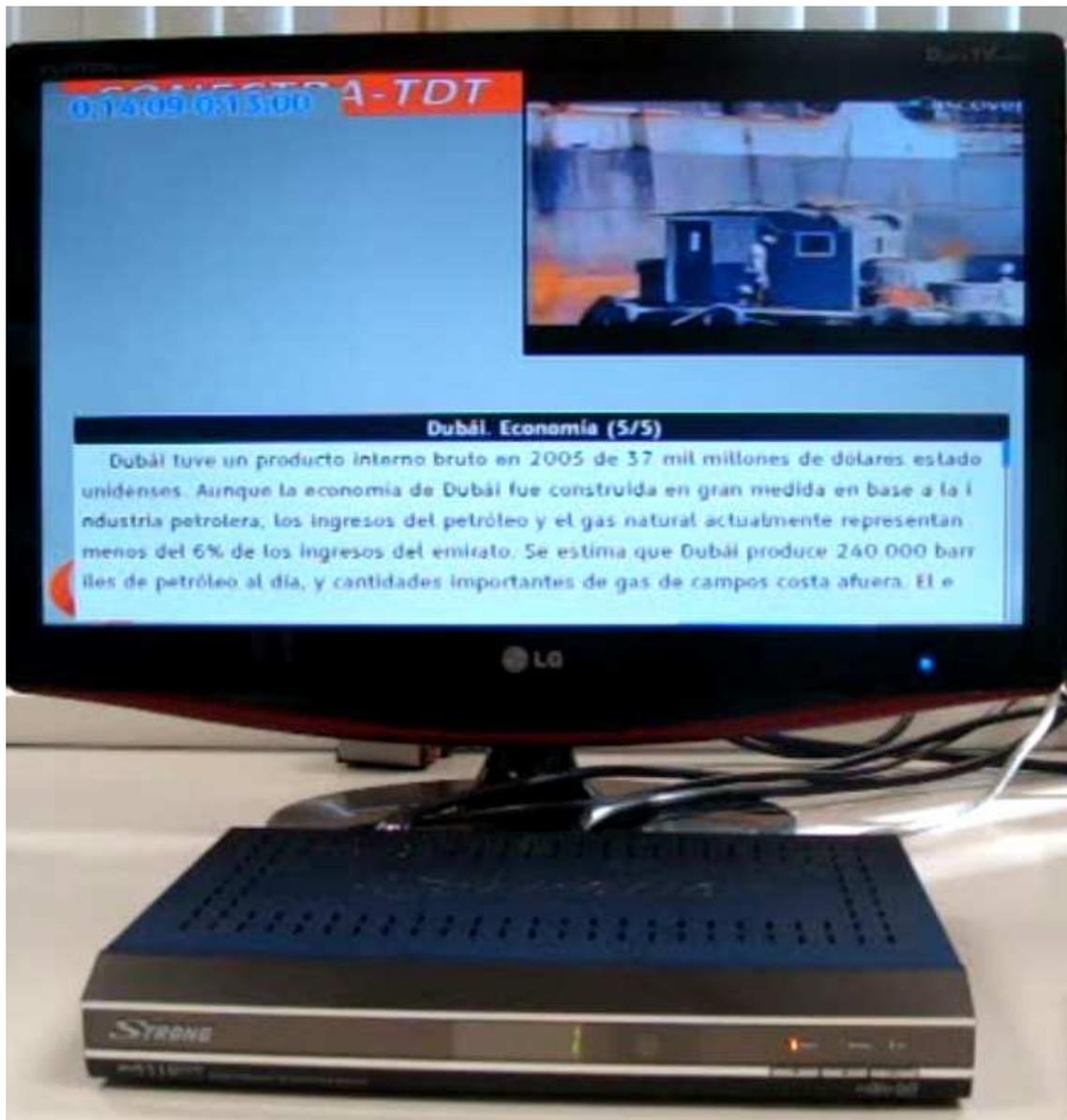


Figura 5.22. Navegación por los recursos teóricos del curso en la televisión.

Lo mismo ocurre para la navegación por los recursos prácticos donde también podemos seguir visualizando el video en una ventana. Podemos ver un ejemplo en la figura 5.23.

Añadir que el espectador en cualquier momento, también puede dejar de navegar por el curso con la simple pulsación de un botón, para seguir viendo el video en pantalla completa.

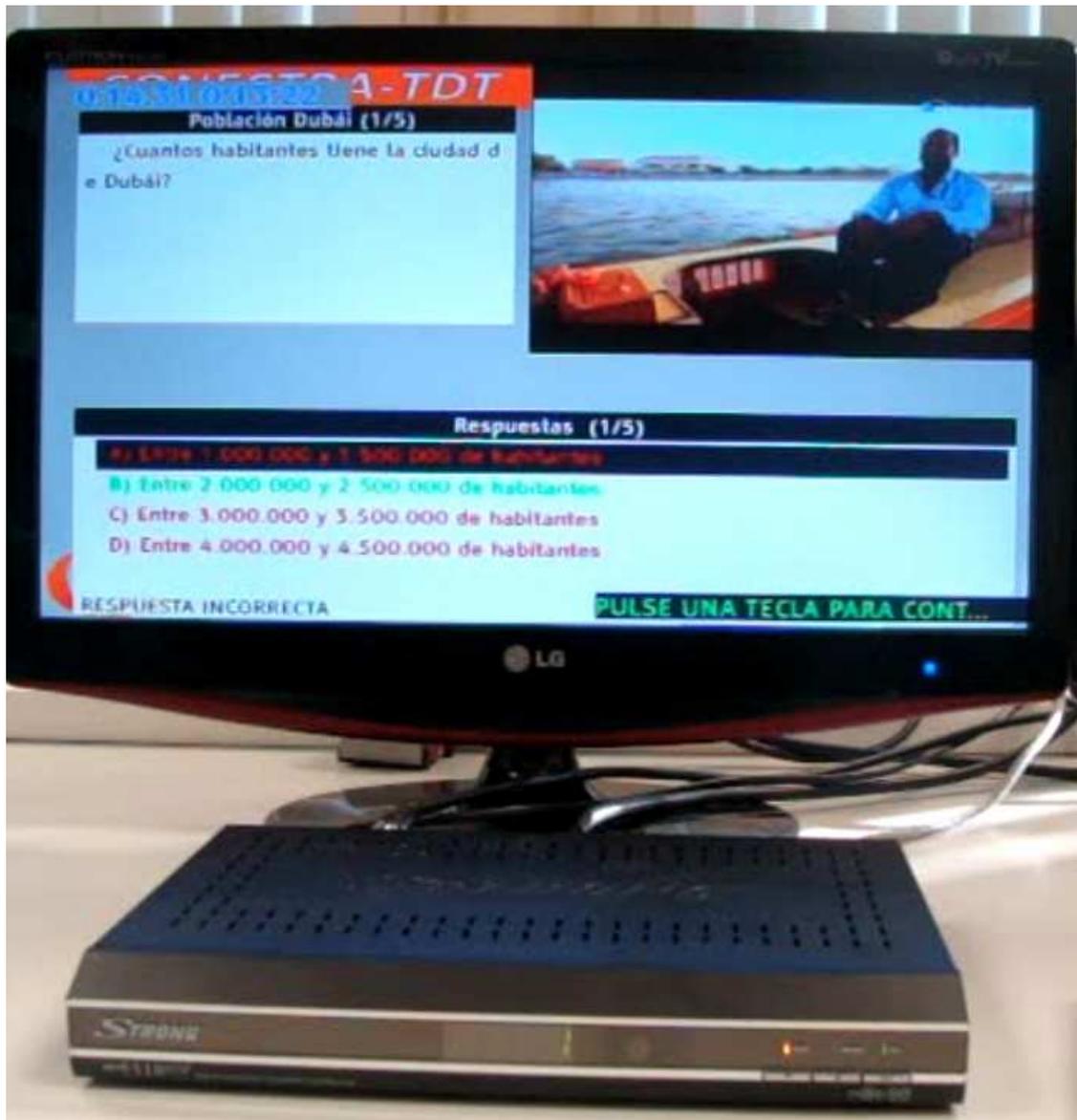


Figura 5.23. navegación por los recursos prácticos del curso en la televisión.

5.5 Creación aplicación web para mostrar los informes

Dado que los usuarios pueden enviar las respuestas que realicen a las preguntas que se le plantean en los cursos a través del **canal de retorno**, se decidió crear una aplicación web que permitiera mostrar resultados a partir de las respuestas de los usuarios.

Concretamente se trata de una aplicación web en php, la encargada de mostrar los distintos informes generados por la aplicación, y una base de datos en la que almacenar las respuestas de los usuarios.

En esta base de datos se almacenarán las siguientes características:

- Preguntas: Los identificadores de las preguntas que han aparecido en el curso de un determinado programa.
- Respuestas: respuestas proporcionadas por los usuarios para las preguntas del curso de un determinado programa.
- Programa: código del programa al que están asociadas las preguntas.
- Usuarios: identificador de los usuarios asociados a las preguntas que han contestado correspondientes a un determinado programa.

A partir de esta información, se generan distintos informes que se detallan a continuación:

1. Número de participantes totales: Número de usuarios que han participado en el conjunto de todos los programas.
2. Número de participantes por programa: Número de usuarios que han participado en cada uno de los programas.
3. Número de programas por usuario: Número de programas en total en los que ha participado cada usuario.
4. Número de preguntas por usuario: Número de preguntas contestadas en total por cada usuario.

A continuación se muestran algunas capturas de esta aplicación web:

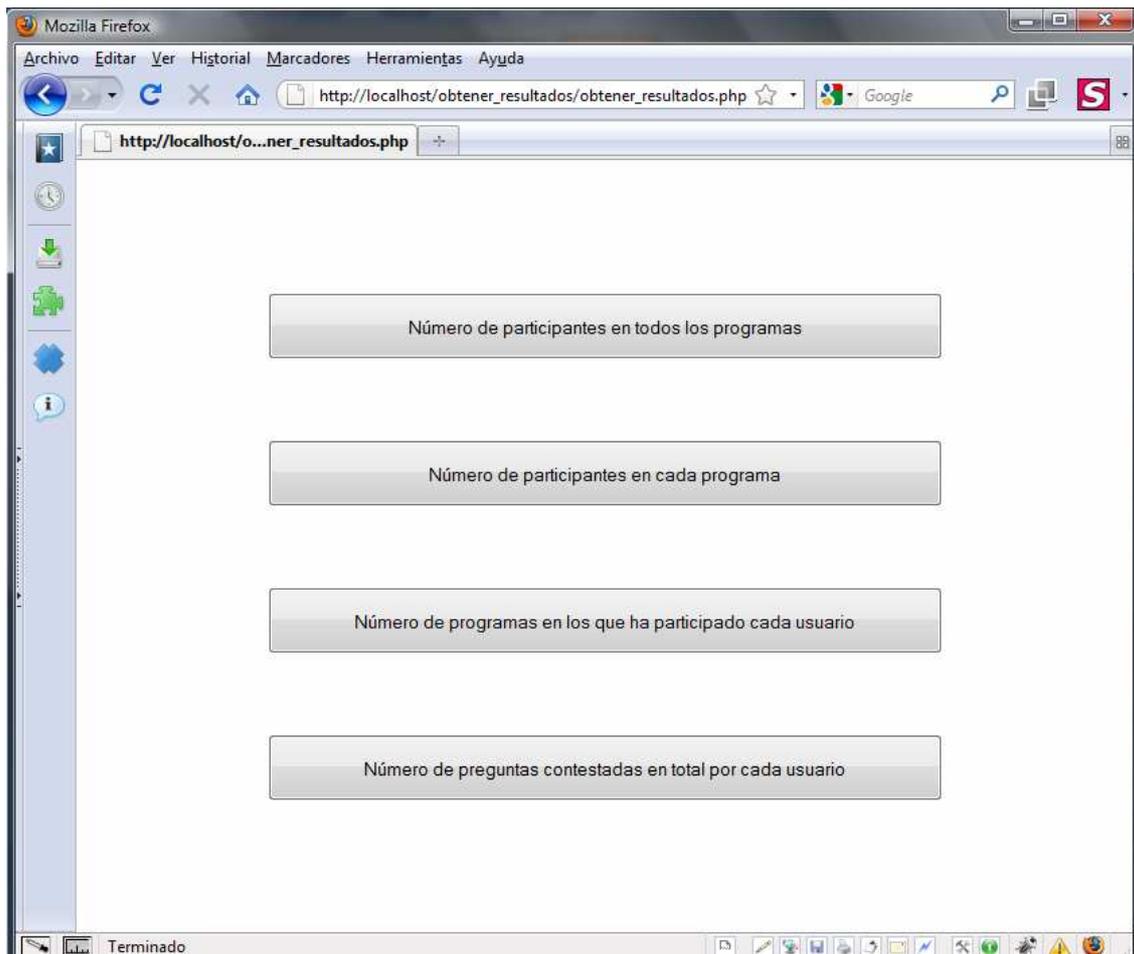


Figura 5.24. Menú principal de la aplicación web.



Figura 5.25. Imagen de la aplicación web que muestra el número de participantes totales.

Número de participantes en cada programa	
Código programa	Participantes
12	10
56	8
41	11
63	10
87	25
26	13
114	8
151	21
214	11
157	39
38	7
207	12
61	9
1222	2

Volver

Figura 5.26. Imagen de la aplicación web que muestra el número de participantes en cada programa.

Número de programas en los que ha participado cada usuario	
Usuario	Número de programas
3	6
6	6
7	9
15	7
21	12
333	1
22	9

Figura 5.27. Imagen de la aplicación web que muestra el número de programas en los que ha participado cada usuario.

Número de preguntas contestadas por cada usuario	
Usuario	Número de preguntas
3	16
6	16
7	28
15	12
21	52
333	1
22	17

Figura 5.28. Imagen de la aplicación web que muestra el número de preguntas contestadas por cada usuario.

Capítulo 6. Pruebas. Validación y verificación

La fase de pruebas consiste en realizar pruebas de todo lo que se ha ido generando durante la construcción del sistema. Durante esta fase se detectan los errores y se lleva a cabo la corrección de los mismos.

Esta fase es una de las más costosas del ciclo de vida software.

Cabe decir que en la metodología convencional utilizada para desarrollar el sistema, las pruebas se utilizan como validación del sistema desarrollado, sin embargo, en las metodologías ágiles, las pruebas juegan un papel diferente, ya que sustituyen a las especificaciones de requisitos, actuando como guía para el desarrollo del sistema.

Se realizan distintos procesos entre los que podemos encontrar:

1. Proceso de validación: Consiste en averiguar si el sistema cumple con los requisitos planificados, respondiendo a la pregunta "¿construimos el producto correcto?" [14].
2. Proceso de verificación: Consiste en verificar cada uno de los módulos que se van construyendo, asumiendo de que si lo que se está construyendo es correcto, también lo será el sistema final. Este proceso responde a la pregunta "¿construimos el producto correctamente?" [14].

Para llevar a cabo estos procesos se realizan una serie de pruebas. Analizando las pruebas desde el punto de vista de un enfoque convencional, podemos distinguir distintos tipos de pruebas para comprobar el correcto funcionamiento de un sistema, las principales son:

1. Pruebas unitarias: Consisten en realizar pruebas de cada una de las piezas de un sistema, es decir, probar por separado todos y cada uno de los módulos que componen un sistema y comprobar su correcto funcionamiento.
2. Pruebas de integración: Consisten en realizar pruebas de todo un sistema en conjunto, para comprobar que no existen errores a la hora de enlazar los distintos módulos de un sistema.
3. Pruebas de sistema: Tratan de verificar que el comportamiento del sistema satisface los requisitos planificados.

4. Pruebas de aceptación: Son realizadas por los usuarios finales del sistema para asegurarse de que el sistema cumple con sus requisitos. La diferencia entre las pruebas de sistema y las pruebas de aceptación es que las pruebas de sistema las realiza el propio desarrollador del sistema y las pruebas de aceptación las realiza el usuario final.

El proceso de validación indica que el sistema tiene que satisfacer los requisitos planificados en la primera fase, por lo que se tiene que preparar un conjunto de pruebas de aceptación que realicen esta comprobación.

Así mismo, tras la fase de requisitos se encuentra la fase de análisis, en la que se hace necesario preparar otro conjunto de pruebas, estas son las pruebas de sistema, que permiten verificar lo planificado en esta fase.

Si bajamos un nivel más, nos encontramos con la fase de diseño, de la que se tendrán que realizar sus correspondientes pruebas, las pruebas de integración.

Y por último, en la fase de codificación se tendrán que preparar distintas pruebas para comprobar el correcto funcionamiento de los distintos módulos, para ello es necesario preparar un conjunto de pruebas unitarias.

Como vemos, existe una correspondencia entre las fases del proyecto y los distintos tipos de pruebas, que da lugar al llamado "modelo en V" que podemos observar en la figura 6.1 [15].

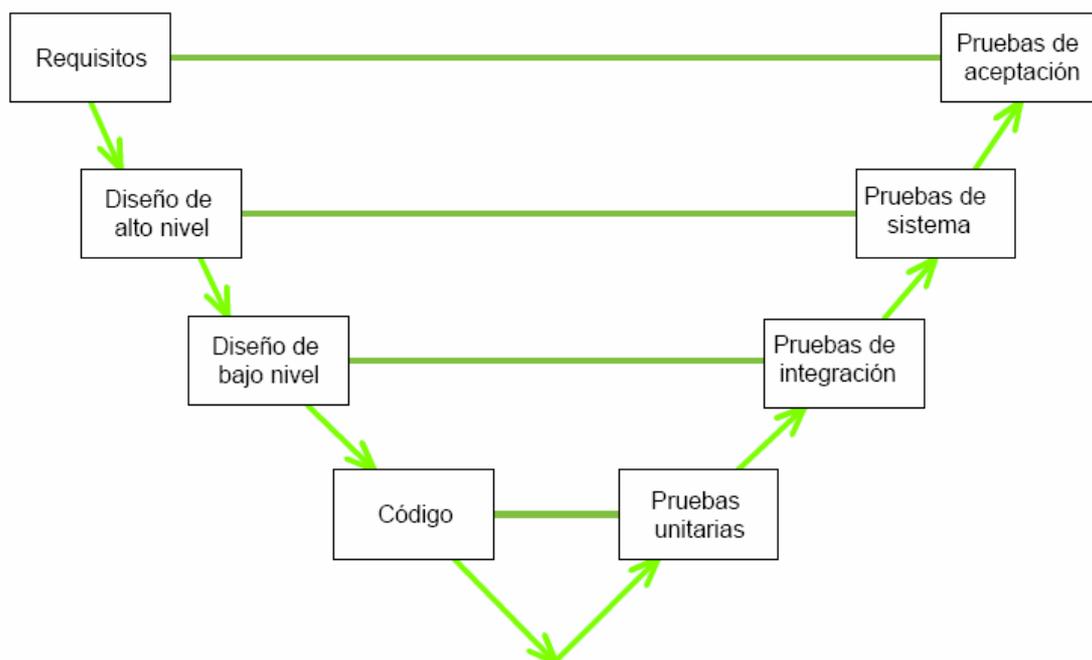


Figura 6.1. Modelo en V.

A continuación se muestran las pruebas unitarias que se realizaron para cada uno de los módulos del sistema, pasando posteriormente a mostrar las pruebas de integración. Por último, se muestran las pruebas de aceptación realizadas al sistema.

Tras la realización de las pruebas, se explican los problemas encontrados y las soluciones adoptadas para cada problema.

6.1 Pruebas unitarias

6.1.1 Creación de los cursos

Las pruebas que se realizaron para comprobar el correcto funcionamiento de este módulo consistieron en crear un conjunto de cursos de las dos formas existentes, a partir de los recursos generados mediante la herramienta de autor y a partir de los recursos generados mediante la extracción de palabras clave.

6.1.1.1 Generación de objetos de aprendizaje mediante la herramienta de autor

Para comprobar el correcto funcionamiento de esta herramienta se crearon un conjunto de cursos de tres formas distintas. La primera fue crear cursos que incluyeran únicamente bloques teóricos, la segunda fue crear cursos que incluyera únicamente preguntas y la tercera fue crear cursos que incluyera tanto recursos teóricos como prácticos.

Las tres formas funcionaron correctamente, guardándose en la base de datos los recursos teóricos y prácticos, además del resto de parámetros del curso.

6.1.1.2 Generación de objetos de aprendizaje mediante la extracción de palabras clave

Antes de comprobar los recursos obtenidos mediante este método, lo que se hizo fue comprobar el funcionamiento del extractor de palabras

clave. Para ello, se crearon distintos párrafos de texto de ejemplo y se aplicó el extractor de palabras clave para cada uno de ellos.

Pudimos comprobar que, efectivamente, se obtenían las palabras que más veces aparecían en el texto y ordenadas conforme al número de apariciones de mayor a menor, aunque nos dimos cuenta que, en algunos casos, algunas palabras clave extraídas no eran significativas en el contexto, debido a la complejidad que presenta extraer palabras clave de un texto.

6.1.1.3 Creación de los cursos a partir de los objetos de aprendizaje

Una vez comprobado el funcionamiento de la generación de objetos de aprendizaje mediante el extractor de palabras clave y mediante la herramienta de autor, se procedió a comprobar la creación de los cursos. Lo primero que se hizo fue montar un servidor en local y se creó una pequeña aplicación web en php que recogía la descripción del programa, el código del programa y la duración del mismo mediante un formulario. En las figura 6.2 y 6.3 se pueden observar dos capturas de la aplicación web que se utilizó para poder comprobar la creación de cursos mediante los dos métodos.

Código del programa	157
Duración del programa	30 minutos
Descripción del programa	<p>Dubai, la pompa y el lujo de esta ciudad del Golfo Pérsico, situada en el extenso desierto de los Emiratos Árabes, nacida gracias a los magnates del petróleo, los jeques árabes, acostumbrada al exceso, en la que incluso cuando llueve lo hace como si fuese un apocalipsis filmado en cualquier película de ciencia ficción de Hollywood. Todo lo que se pueda hacer y conseguir con dinero está en Dubai, una megaciudad construida de la nada en medio del desierto en la que se puede encontrar reflejos de cualquier rincón del mundo: desde el lujo parisino, con perfumes exclusivos a 2.500 dólares el frasco, hasta pistas de esquí y patinaje sobre hielo a 10 grados bajo cero, cuando en la calle sobrepasan la barrera de los 40 grados centígrados. Todo a capricho y gusto de jeques y príncipes, los más poderosos magnates del petróleo.</p>
<input type="button" value="Generar curso"/>	

CURSO GENERADO CORRECTAMENTE

El curso generado se encontraba ya creado y asociado al código de programa introducido.

Figura 6.2. Creación de un curso a partir de los recursos generados mediante la herramienta de autor.

Código del programa	107
Duración del programa	45 minutos
Descripción del programa	Dubai, la pompa y el lujo de esta ciudad del Golfo Pérsico, situada en el extenso desierto de los Emiratos Árabes, nacida gracias a los magnates del petróleo, los jeques árabes, acostumbrada al exceso, en la que incluso cuando llueve lo hace como si fuese un apocalipsis filmado en cualquier película de ciencia ficción de Hollywood. Todo lo que se pueda hacer y conseguir con dinero está en Dubai, una megaciudad construida de la nada en medio del desierto en la que se puede encontrar reflejos de cualquier rincón del mundo: desde el lujo parisino, con perfumes exclusivos a 2.500 dólares el frasco, hasta pistas de esquí y patinaje sobre hielo a 10 grados bajo cero, cuando en la calle sobrepasan la barrera de los 40 grados centígrados. Todo a capricho y gusto de jeques y príncipes, los más poderosos magnates del petróleo.
<input type="button" value="Generar curso"/>	

CURSO GENERADO CORRECTAMENTE

El curso ha sido generado a partir de las palabras clave de la descripción del programa.

Figura 6.3. Creación de un curso a partir de los recursos generados mediante el extractor de palabras clave.

Se crearon distintos cursos con un número variable de recursos tanto teóricos como prácticos. Se probaron a obtener cursos con distintos números de recursos, desde un solo recurso teórico y uno práctico hasta un número lo suficientemente grande de recursos de cada tipo.

Pudimos comprobar que, efectivamente, se creaban los cursos correctamente, independientemente de dicho número, comprobando que, en el caso de establecer un número alto de recursos a obtener, incluía únicamente los recursos encontrados, aunque no se llegara al número exacto.

6.1.2 Distribución de objetos de aprendizaje

Para la comprobación de este módulo se utilizó un video de ejemplo, junto con un fichero de sonido y un directorio que contenía una pequeña aplicación con Xlets ya compilada.

Primero se utilizó el script `oc-update.sh` para convertir las entradas en Transport Streams (TS) y comprobamos que se generaban, para cada uno de ellos, un fichero en formato `ts`.

Posteriormente probamos la multiplexación y el envío de estos tres ficheros al decodificador de TDT mediante el programa `Tscbrmuxer` y comprobamos que el video y el sonido se reproducían correctamente en el televisor. También vimos que se había cargado correctamente la aplicación en el decodificador y se pudo llegar a ejecutar y visualizarse por pantalla.

6.1.3 Consumo de objetos de aprendizaje

Todas las pruebas que se realizaron sobre la aplicación encargada de visualizar los cursos, se hicieron directamente sobre el dispositivo real, sin la utilización de simuladores, enviando la aplicación al decodificador y ejecutándola en el mismo.

La visualización de los cursos por pantalla se probó de la siguiente manera. Se creó un curso de ejemplo que se introdujo en el directorio de la aplicación y, tras el envío al decodificador se ejecutó y se comprobó que aparecían tanto los bloques teóricos como las preguntas en la parte inferior del televisor.

Se probó con distintos cursos y con recursos que incluían acentos e incluso con caracteres especiales y pudimos observar que se veían perfectamente sin ningún error.

6.1.3.1 Canal de retorno

Para comprobar el funcionamiento del canal de retorno primero se instaló un servidor Apache con una base de datos MySQL, encargada de almacenar las respuestas de los usuarios, en una máquina del laboratorio.

A continuación se incluyó el servicio que permite almacenar las respuestas en la base de datos.

Una vez ya tenemos la parte del servidor nos centramos en la parte cliente, que, en este caso, se trata del decodificador de TDT.

Se añadió un método a la aplicación para permitir la comunicación con el exterior (ver figura 6.4) que incluía el envío de las respuestas a las

preguntas del curso. Se estableció como dirección de envío la dirección IP del servidor creado y se incorporó en la aplicación la funcionalidad necesaria para enviar las respuestas una vez contestadas por el usuario.

Hecho esto se envió la aplicación al STB, se ejecutó y se contestaron las preguntas del curso de ejemplo creado. Una vez contestadas las preguntas, se accedió a la base de datos del servidor y se pudo comprobar que, efectivamente, las respuestas que se habían proporcionado a las preguntas se habían almacenado correctamente.

```
private static void enviar_respuestas(int id_pregunta,
    String respuesta, int cod_programa, int id_usuario){

    BufferedReader rd = null;
    try {
        String parametros = "id_pregunta="+id_pregunta+
            "&respuesta="+respuesta+
            "&cod_programa="+cod_programa+
            "&id_usuario="+id_usuario;

        URL url = new URL(
            "http://localhost/almacenar_respuestas.php?"
            +parametros);

        URLConnection conn = url.openConnection();

        rd = new BufferedReader(new InputStreamReader(
            conn.getInputStream()));

        rd.close();

    } catch (Exception e) {

        System.out.println("Web request failed");

    }
}
```

Figura 6.4. Método de comprobación de utilización del canal de retorno.

6.1.4 Seguimiento

Para comprobar el correcto funcionamiento de este módulo, ejecutamos en el navegador la aplicación en PHP, mostrando los distintos informes disponibles. Al acceder a cada uno de ellos, pudimos comprobar que se mostraban resultados, que comparamos con la base de datos y vimos que eran correctos.

6.2 Pruebas de integración

Una vez comprobado el correcto funcionamiento de todos los módulos del sistema, se realizaron diversas pruebas del funcionamiento del sistema en conjunto. Se realizaron dos tipos de pruebas, el primer tipo consistía en crear un curso manualmente mediante la herramienta de autor y el segundo tipo, en crear un curso a partir de la extracción de palabras clave de la descripción del programa.

Una vez obtenido el curso mediante los dos métodos, se añadió al directorio de la aplicación y se generó el fichero ts, que junto con los ficheros ts del video y el audio se enviaron multiplexados al decodificador.

Una vez enviado el video, el audio y la aplicación al decodificador, se ejecutó la aplicación y se mostraron correctamente todos los bloques teóricos y las preguntas.

Se contestaron las distintas preguntas del curso para comprobar el correcto almacenamiento de las respuestas utilizando el canal de retorno.

Por último, se ejecutó la aplicación web y se accedió a los distintos informes, comprobando que se habían tenido en cuenta las respuestas que habíamos proporcionado a las preguntas en la aplicación.

6.3 Pruebas de aceptación

A continuación se muestran de la tabla 6.1 a la 6.x las pruebas de aceptación realizadas al sistema.

Caso de Prueba de Aceptación	
Código: M1P1	Caso de Uso: Generar curso manualmente
Descripción de la Prueba: Comprobar que se pueden rellenar los distintos campos de un paquete de contenidos y que al guardar aparece en la lista como creado.	
Condiciones de Ejecución:	

Estar situado sobre la pestaña de "paquete de contenidos".
<p>Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Rellenar los campos de generación de un paquete de contenidos. 2. Clicar sobre "guardar". 3. Clicar sobre "paquete de contenidos".
<p>Resultado Esperado:</p> <p>El paso 2 almacena en la base de datos el paquete.</p> <p>El paso 3 muestra la lista de paquetes almacenados.</p>
<p>Evaluación de la Prueba: APTA</p>

Tabla 6.1. Prueba de aceptación: Generar curso manualmente.

Caso de Prueba de Aceptación	
Código: M1P2	Caso de Uso: Obtener curso automáticamente
<p>Descripción de la Prueba:</p> <p>Comprobar que el sistema es capaz de buscar en la base de datos un curso que se encuentre asociado a un código de programa, el cual obtiene a partir de la EPG del propio programa. En el caso de que no se obtenga ningún curso para un programa en cuestión, debido a que no ha sido generado previamente mediante la herramienta de autor, comprobar que el sistema genera automáticamente un curso a partir de un número definido de palabras clave extraídas de la descripción del programa.</p>	

<p>Condiciones de Ejecución:</p> <p>Le llega como entrada la EPG del programa con la descripción, el código del programa y la duración del mismo.</p>
<p>Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Ejecutar directamente el programa con los parámetros necesarios.
<p>Resultado Esperado:</p> <p>El paso 1 genera un curso relacionado con el programa en cuestión</p>
<p>Evaluación de la Prueba: APTA</p>

Tabla 6.2. Prueba de aceptación: Obtener curso automáticamente.

Caso de Prueba de Aceptación	
Código: M2P1	Caso de Uso: Enviar aplicación y curso
<p>Descripción de la Prueba:</p> <p>Comprobar que el sistema es capaz de multiplexar la aplicación encargada de consumir los objetos de aprendizaje junto con los propios objetos de aprendizaje y la señal de televisión y, una vez hecho esto, lo envía todo al decodificador TDT.</p>	
<p>Condiciones de Ejecución:</p> <p>Se dispone del directorio "aplicación" con la aplicación ya compilada y el directorio del curso "curso". Además se dispone de un fichero de video</p>	

"video.mpeg" y uno de audio "audio.mp3".
<p>Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Ejecutar en la consola: "oc-update.sh aplicacion 0xB 1 2003 1 1 0 0" 2. Ejecutar en la consola: "Tscbrmuxer 1024 video.ts 780 audio.ts 570 aplicaci3n.ts"
<p>Resultado Esperado:</p> <p>El paso 1 genera los ficheros: "aplicaci3n.ts", "video.ts", "audio.ts".</p> <p>El paso 2 envía la aplicaci3n junto con el video y el audio al decodificador.</p>
<p>Evaluaci3n de la Prueba: APTA</p>

Tabla 6.3. Prueba de aceptaci3n: Enviar aplicaci3n y curso.

Caso de Prueba de Aceptaci3n	
C3digo: M3P1	Caso de Uso: Contestar pregunta
<p>Descripci3n de la Prueba:</p> <p>Comprobar que la aplicaci3n visualiza una pregunta y ofrece la posibilidad de navegar entre las distintas respuestas pudiendo seleccionarlas.</p>	
<p>Condiciones de Ejecuci3n:</p> <p>Ninguna.</p>	

<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Pulsar los botones de "arriba" y "abajo". 2. Pulsar el botón "ok".
<p>Resultado Esperado:</p> <p>El paso 1 permite navegar entre las distintas respuestas.</p> <p>El paso 2 selecciona la respuesta en la que se encuentra el foco.</p>
<p>Evaluación de la Prueba: APTA</p>

Tabla 6.4. Prueba de aceptación: Contestar pregunta.

Caso de Prueba de Aceptación	
Código: M3P1	Caso de Uso: Ver los recursos teóricos y realizar los recursos prácticos del curso
<p>Descripción de la Prueba:</p> <p>Comprobar que la aplicación visualiza los distintos bloques teóricos y permite contestar a las preguntas del curso.</p>	
<p>Condiciones de Ejecución:</p> <p>Ninguna.</p>	
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Pulsar el botón "menú" 2. Seleccionar el apartado de teoría y pulsar el botón "ok". 3. Seleccionar el bloque teórico 3 y pulsar el botón "ok". 	

4. Pulsar repetidamente el botón "abajo" hasta llegar al final del texto.
5. Pulsar El botón "menú".
6. Seleccionar el apartado de preguntas y pulsar el botón "ok".
7. Seleccionar una de las respuestas para cada pregunta con los botones de "arriba" y "abajo" y pulsar el botón "ok" para cada una de ellas.

Resultado Esperado:

El paso 1 muestra una pantalla donde se puede seleccionar si se quiere ver la teoría o las preguntas del curso.

El paso 2 muestra una lista con los distintos bloques teóricos que contiene el curso.

El paso 3 muestra el bloque teórico 3.

El paso 4 permite mostrar el texto que falta por ver del bloque teórico.

El paso 5 muestra la pantalla que permite seleccionar si se quiere ver la teoría o las preguntas del curso.

El paso 6 muestra la primera pregunta del curso.

El paso 7 permite navegar por las distintas preguntas que contiene el curso contestando a cada una de ellas.

Evaluación de la Prueba: APTA

Tabla 6.5. Prueba de aceptación: Ver los recursos teóricos y realizar los recursos prácticos del curso.

Caso de Prueba de Aceptación	
Código: M4P1	Caso de Uso: Ver informes
Descripción de la Prueba: Comprobar que la herramienta web permite visualizar diferentes informes a	

partir de las respuestas proporcionadas por los usuarios.

Condiciones de Ejecución:

Ninguna.

Pasos de Ejecución:

1. Clicar sobre "Número de participantes en todos los programas".
2. Clicar sobre "Volver".
3. Clicar sobre "Número de participantes en cada programa".
4. Clicar sobre "Volver".
5. Clicar sobre "Número de programas en los que ha participado cada usuario".
6. Clicar sobre "Volver".
7. Clicar sobre "Número de preguntas contestadas en total por cada usuario".
8. Clicar sobre "Volver".

Resultado Esperado:

El paso 1 muestra el número de usuarios totales que han participado en todos los programas.

El paso 2 muestra el menú principal.

El paso 3 muestra el número de usuarios que han participado en cada programa.

El paso 4 muestra el menú principal.

El paso 5 muestra el número de programas en total en los que ha participado cada usuario.

El paso 6 muestra el menú principal.

El paso 7 muestra el número de preguntas contestadas en total por cada usuario.

El paso 8 muestra el menú principal.
Evaluación de la Prueba: APTA

Tabla 6.6. Prueba de aceptación: Ver informes.

6.4 Problemas y soluciones

En este apartado se indican los problemas que han surgido a la hora de desarrollar los distintos módulos del sistema y que soluciones se han adoptado para resolverlos.

A la hora de buscar objetos de aprendizaje en la base de datos a partir de un conjunto de palabras clave (ver punto 4.3.2), se implementó una funcionalidad añadida que consistía en que si no se encontraban recursos en una determinada tabla de la base de datos con las palabras clave disponibles, se buscaban sinónimos de cada una de las palabras clave y se volvía a buscar objetos de aprendizaje en la tabla de la base de datos con el conjunto de sinónimos de las palabras clave.

Una vez se probó esta nueva funcionalidad, se vio que para generar un curso el tiempo era considerablemente mayor y no resultaba nada práctico, por lo que se decidió eliminar la funcionalidad.

Otro problema que tuvimos fue con el canal de retorno, a la hora de enviar las respuestas al servidor. Tuvimos problemas de acceso que se solucionaron cambiando los permisos en el archivo de configuración del servidor.

6.5 Conclusiones

En este capítulo hemos visto las distintas pruebas que se han realizado en el sistema para comprobar su correcto funcionamiento, a la vez que las soluciones que se han dado para los problemas que han surgido en el desarrollo del mismo.

Capítulo 7. Conclusiones del proyecto

En el presente proyecto se ha realizado un repaso de las aplicaciones interactivas para la TDT desarrolladas hasta el momento, hemos visto que ofrecen distintas propuestas: información meteorológica, información del tráfico, sistemas de pago de tasas, de búsqueda de empleo, etc. Sin embargo, existen pocas aplicaciones desarrolladas para el campo de la formación, y éstas, no aportan contenidos docentes interrelacionados con el contenido propio de la emisión de la cadena de televisión.

Este proyecto ha dado un paso más allá en el uso de las aplicaciones interactivas para la TDT referentes a la formación, al ofrecer al espectador la posibilidad de ampliar sus conocimientos sobre algo que le interesa, sobre el contenido que se ofrece en el canal de televisión que él mismo ha seleccionado, a la vez que le permite realizar una serie de ejercicios acerca del contenido.

La solución propuesta no solo conlleva ventajas para el espectador sino que también presenta ventajas para las cadenas de televisión y productores de contenidos, debido a que, gracias a la utilización del canal de retorno, los responsables de las cadenas pueden realizar estimaciones fiables sobre el número de espectadores que participan en sus programas y el grado de implicación de éstos.

Conecta-TDT surge de la colaboración entre la Universidad Politécnica de Valencia y la empresa SETIVAL SCV para desarrollar un sistema que continuara con la misma línea que se siguió con el proyecto TDT-Reiven, correspondiente a un proyecto final de carrera, que también surgió gracias a la colaboración de las dos instituciones.

Durante la etapa de finalización, el proyecto se ha presentado en algunos eventos en los que se contaba con la participación de algunas personalidades pertenecientes a la Conselleria de Tecnología e Industria de la Generalitat Valenciana y se encuentra pendiente su presentación en el canal de televisión autonómico de la Comunidad Valenciana, Canal 9.

Los objetivos que se planificaron para el proyecto se han alcanzado con éxito, llevando al sistema a un estado en el que se encuentra preparado para ser implantado en las cadenas de televisión.

Respecto al futuro de este proyecto, quedan muchas puertas abiertas en cuanto a la posibilidad de incluir nuevas funcionalidades y mejoras que permitan llevar a cabo una mejor selección de objetos de aprendizaje que

se adecuen a la emisión del programa, así como la presentación al usuario de resultados que se consideren oportunos provenientes de la propia cadena de televisión, es decir, la utilización del canal de retorno, no solo del espectador a la cadena, sino también de la cadena al espectador.

Cabe añadir que el canal de retorno en los decodificadores de TDT puede llevar a la realización de infinidad de proyectos muy interesantes, tanto para el espectador como para las cadenas de televisión. Por poner algunos ejemplos, gracias a esta tecnología los usuarios podrían realizar encuestas a través del mando a distancia sin moverse del sofá y las cadenas de televisión podrían conocer la opinión de millones de personas acerca de un tema determinado y en cuestión de minutos presentar los porcentajes de los resultados por pantalla. Esta funcionalidad ya la llevan a cabo algunas cadenas, pero a través de internet y, por lo tanto, con mucha menos participación que si se hiciera a través del televisor.

Otro ejemplo muy interesante podría ser la participación de los espectadores en los concursos de televisión y la posibilidad de ganar premios a los acertantes que se encuentren entre los primeros en contestar a las preguntas, o bien entre los espectadores que hayan contestado correctamente a un número mayor de preguntas, con la ventaja para la cadena de televisión de ganar audiencia.

Para terminar, Podemos destacar el cambio que va a experimentar la televisión en un futuro próximo gracias a la llegada de la Televisión Digital Terrestre, convirtiéndose en un elemento más de interacción con el mundo exterior.

Bibliografía

- [1] Definición de e-learning.
<http://es.wikipedia.org/wiki/E-learning>.
- [2] Página web de la Asociación para la Implantación y el Desarrollo de la Televisión Digital Terrestre en España.
<http://www.impulsatdt.es/pdf/IT-sintonizador.pdf>.
- [3] Página web con los proyectos del Laboratorio de TV Digital Interactiva.
<http://tvdi.det.uvigo.es/es/proyectos.html>.
- [4] Daniel Martín-Albo Simón. TDT-REIVEN: Plataforma de t-learning utilizando MHP para promover el aprendizaje desde el hogar, 2009.
- [5] Definición de t-learning.
<http://es.wikipedia.org/wiki/T-Learning>.
- [6] Introducción a IMS-QTI.
<http://javierperis.wordpress.com/2008/01/12/qti-continuacion/>.
- [7] Definición de MHP.
<http://www.xataka.com/hogar-digital/mhp-servicios-interactivos-de-la-tdt>.
- [8] Introducción a los Xlets.
<http://www.cesnavarra.net/cesdigital/Lists/Noticias%20CESDigital/DispFormCES.aspx?List=5ec0dfc7-7911-470b-8b6b-71ba72783fdd&ID=23>.
- [9] Miguel Ángel Lago Ángel. TDT-REIVEN Interactivo: T-learning interactivo soportado por plataformas de desarrollo para TDT basadas en MHP, 2009.
- [10] El problema del canal de retorno en aplicaciones interactivas de televisión digital terrestre (TDT).
<http://www.cesnavarra.net/cesdigital/Lists/Noticias%20CESDigital/DispFormCES.aspx?List=5ec0dfc7-7911-470b-8b6b-71ba72783fdd&ID=19>.
- [11] En qué consiste la tecnología GPRS.
<http://www.internetmania.net/int0/int68.htm>.
- [12] Tipos de conexiones.
<http://decacross.org/blog/tag/gprs/>.
- [13] DVB-T measurements with PROMAX TV EXPLORER II+ and analysis of DVB transport streams.

http://gps-tsc.upc.es/imatge/_Xgiro/teaching/thesis/2008-2009/AnnaArias.pdf

[14] Validación y verificación de software.

<http://www.acm.org/crossroads/espanol/xrds8-3/destiny.html#ieee>

[15] El proceso de pruebas en el ciclo de vida.

<http://alarcos.inf-cr.uclm.es/doc/masi/doc/lec/parte5/polo-apuntesp5.pdf>

[16] Procesos de planificación relativos al desglose y el tiempo. Apuntes de la asignatura de Dirección y Organización de Proyectos Informáticos, Facultad de Informática de la Universidad Politécnica de Valencia.

[17] Emisiones MHP en España.

http://www.televisiondigital.es/Terrestre/ComisionSeguimientoTDT/DocumentosAprobadosForoTecnico/DocumentosForoTcnico/22GT7_Informe_situacion_MHP_Espa%C3%B1a.pdf

[18] Definición de STB.

http://es.wikipedia.org/wiki/Set-top_box

[19] Proyecto ELU.

<http://www.elu-project.com>