Especificaciones para manejo de fuentes de Simuladores

Índice:

1. Objetivo 4
2. Simuladores – Estructura y Compilación del objeto de aprendizaje 5
2.1 Estructura de carpetas del Simulador 5
2.2 Estructura y compilación de un proyecto de simulador genérico 8
2.3 Importación del proyecto simulador en entorno de desarrollo integrado (IDE) Adobe Flex Builder 3 9
2.4 Descripción de la estructura de un proyecto de simulador 12

1. Objetivo

El presente documento tiene por objetivo definir ciertos aspectos en cuanto a la estructura de archivos y directorios de los simuladores susceptibles de ser traducidos y adaptados.

Pretende además servir de introducción para realizar la construcción del proyecto de simuladores, por lo que se darán instrucciones acerca de cómo compilar el simulador o alguna de sus partes

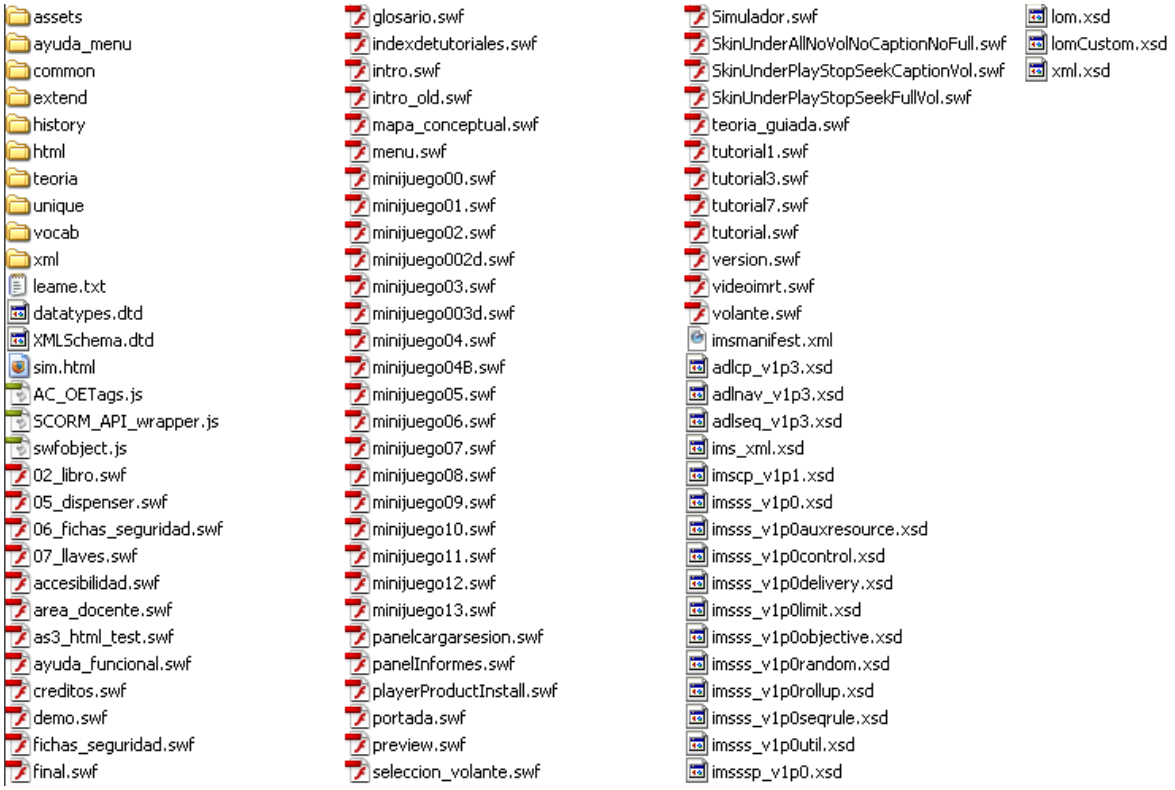
Por otra parte, este documento intenta poner al descubierto una serie de aspectos técnicos en lo referente a la estructura de los simuladores (de lo enunciado surge su ámbito técnico)

2. Simuladores – Estructura y Compilación del objeto de aprendizaje

Repasando brevemente, tenemos que los simuladores son Objetos de Aprendizaje y como tal tiene una organización de directorios y archivos que procederemos a explicar

2.1 Estructura de carpetas del Simulador

La imagen que se adjunta a continuación corresponde a la estructura de ejemplo (considérese genérica) de un simulador:



Antes de pasar a la descripción de carpetas del proyecto, haremos notar que muchos de los archivos/carpetas mostrados en la captura anterior son generados como parte del proceso de empaquetado del producto simulador. Por un lado as carpetas que resultan del empaquetado **son unique, vocab, extend y common**, dichas carpetas resultan del proceso de empaquetado. Además de las mencionadas anteriormente existen archivos xml generados en la base de esta carpeta, en general son descriptores de estructura xml (como podemos apreciar estos archivos son los de extensiones .dtd y .xsd). Dichos archivos son necesarios para su procesamiento según SCORM. Existe además una archivo de catalogación importante el mismo es el **imsmanifest.xml** que es un descriptor del empaquetado para plataforma SCORM y que además tiene

una enumeración de los archivos que conforman el Objeto de aprendizaje, por lo que si en una futura actualización se agregan/eliminan archivos de del mismo, el mismo debe actualizado para reflejar dichos cambios.
-La carpeta assets

Contiene la mayor parte de recursos multimedia que el simulador necesita para funcionar, se organizan por medio de carpetas de la siguiente manera:

- animaciones: Esta carpeta contiene archivos .swf o .flv con las animaciones y videos utilizados en el simulador
- audios: Contiene los archivos de sonido del simulador, a su vez puede tener carpetas para organizar el contenido
- biblioteca: Contiene archivos .swf que contienen los diferentes recursos compilados necesarios para la ejecución del simulador, entre los mas comunes podemos citar las escenas, fondos, fuentes (escenan.swf , biblioteca.swf , son los mas comunes de encontrar)
- imagenes: como su nombre lo indica contiene imágenes en distintos formatos que pueden ser utilizados tanto por el simulador como por los diferentes mapas conceptuales, actividades, etc. y a su vez puede tener carpetas para organizar el contenido.
- Otras carpetas: puede además contener otras carpetas según los desarrolladores hayan considerado conveniente para organizar los contenidos.

-La carpeta html contiene librerías de estilos para html utilizados para organizar algunos contenidos

-La carpeta teoria
Por su parte contiene los contenidos utilizados cada uno de los apartados y tiene cada uno carpetas que le representan (salvo img que es una carpeta que contiene imágenes compartidas)

- accesibilidad
- area_docentes
- ayuda_funcional
- conceptos_auxiliares
- teoria_guiada
- tutorial
- img

Las carpetas anteriormente mencionadas contienen los recursos de cada aparatado y en los archivos xml se almacenan los contenidos textuales

-La carpeta xml

Contiene todos los archivos de configuración por cada ítem de programación que necesitara de la misma, resultando 2 archivos especialmente importantes: recursos.xml y simulador.xml. Estos son los archivos principales en

configuración del simulador donde recursos.xml contiene las descripciones y ubicaciones de cada elemento que el simulador necesite para su funcionamiento, mientras que simulador.xml contiene la configuración relativa al funcionamiento de la maquina de estados que maneja el simulador en el desarrollo del mismo.

-Minijuegos, pantallas, teoría y otros contenidos externos:

Por otra parte cada juego presentado en el simulador tiene su correspondiente archivo .xml de configuración y su nomenclatura es minijuego[n].xml donde [n] es el sufijo que corresponde al minijuego de esta forma el minijuego10 tendrá como archivo de configuración a minijuego10.xml

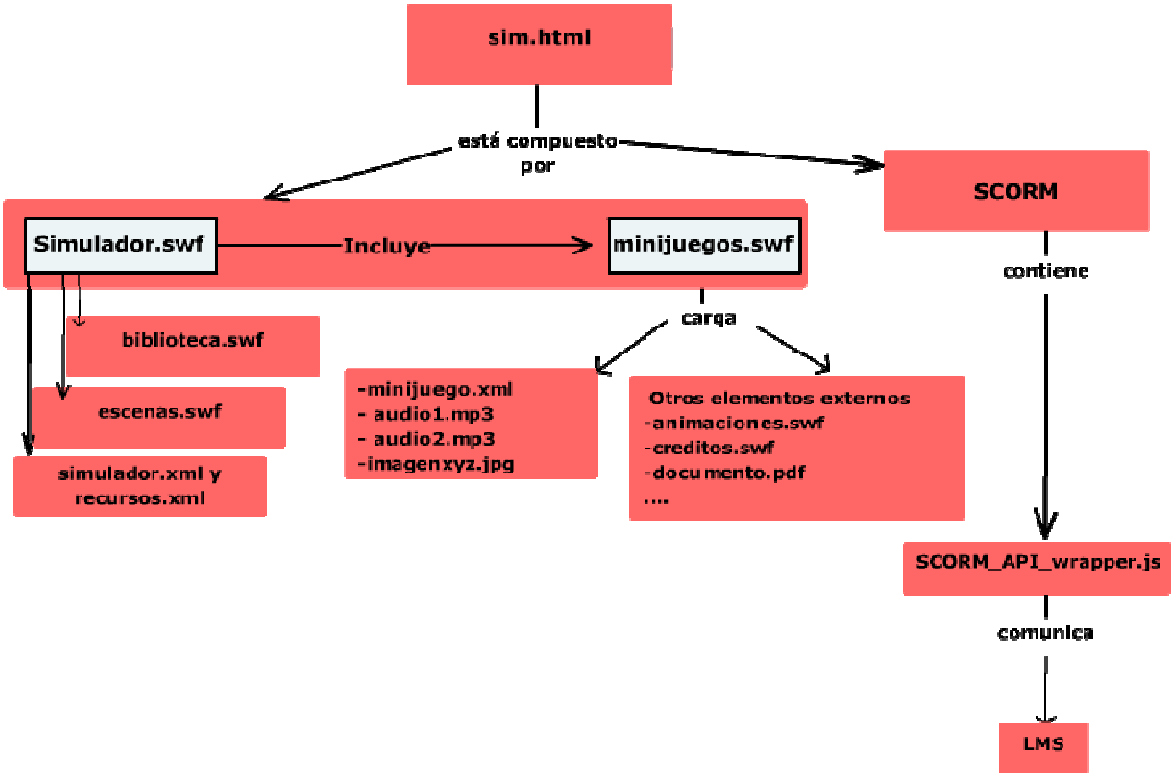
- "SCORM_API_wrapper.js"
Es el archivo de código javascript que permite implementar la trazabilidad con SCORM 2004 y es el que el simulador utiliza para hacer le manejo de esta trazabilidad

-"swfobject.js"
Es el archivo de código javascript que maneja la incorporación del swf del simulador en html

-Sim.html

Contiene el simulador. En caso de que se encuentre en un LMS se encarga de abrir cerrar la comunicación con el mismo.

- Simulador.swf
Interfaz del OA que contiene toda La navegación por El mismo. Carga los archivos de animaciones (en la carpeta assets/animaciones), librerías (assets/biblioteca) y textos necesario (en la carpeta xml), además de los ficheros de audio de la carcasa y los elementos externos.



2.2 Estructura y compilación de un proyecto de simulador genérico

Esta parte del documento tiene como objetivo introducirle en el proceso de compilación de un simulador.

Para realizar la compilación de un simulador sólo bastará con abrir el archivos simulador.fla ubicado en el directorio raiz que se entrega con el proyecto, con Adobe Flash profesional CS4 y ejecutar Menú Control → Probar Película, el proceso puede tardar unos minutos pero finalmente dejará un archivo compilado en la carpeta bin-debug (de no existir la misma deberá crearla dentro de la carpeta raíz del proyecto) con el nombre Simulador.swf.

Con esto usted habrá compilado solamente el módulo principal del simulador pero no el resto contenidos independientes que el simulador pueda cargar en tiempo de ejecución. Pero si lo que si usted necesita compilar sólo una parte del simulador que no comprenda **simulador.swf** se le recomienda utilice el entorno de desarrollo integrado tal como Adobe Flex Builder 3 con el que usted editar más cómodamente el simulador. Dicho proceso se le explicará en el siguiente apartado.

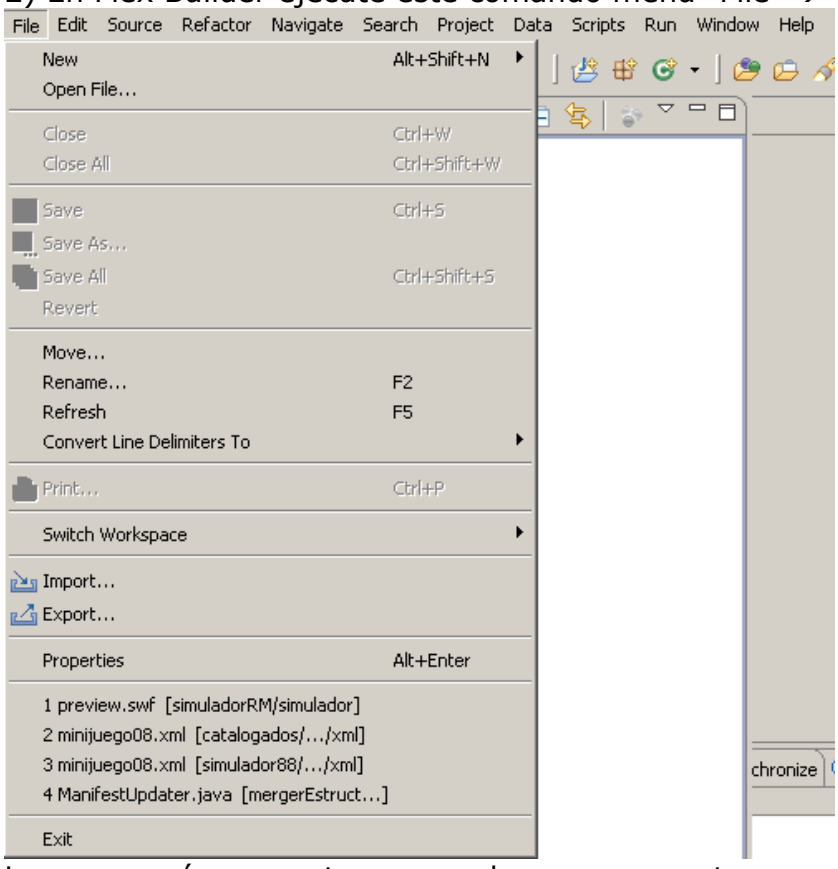
2.3 Importación del proyecto simulador en entorno de desarrollo integrado (IDE) Adobe Flex Builder 3

Teniendo Adobe Flex Builder 3 (sino puede conseguirlo desde la página de Adobe) inicie la aplicación, ésta le preguntará cuál quiere sea su directorio de trabajo, elija alguno conveniente (en nuestro caso D:/workspace para simplificar y a modo de ejemplo pero puede ser cualquier otro)

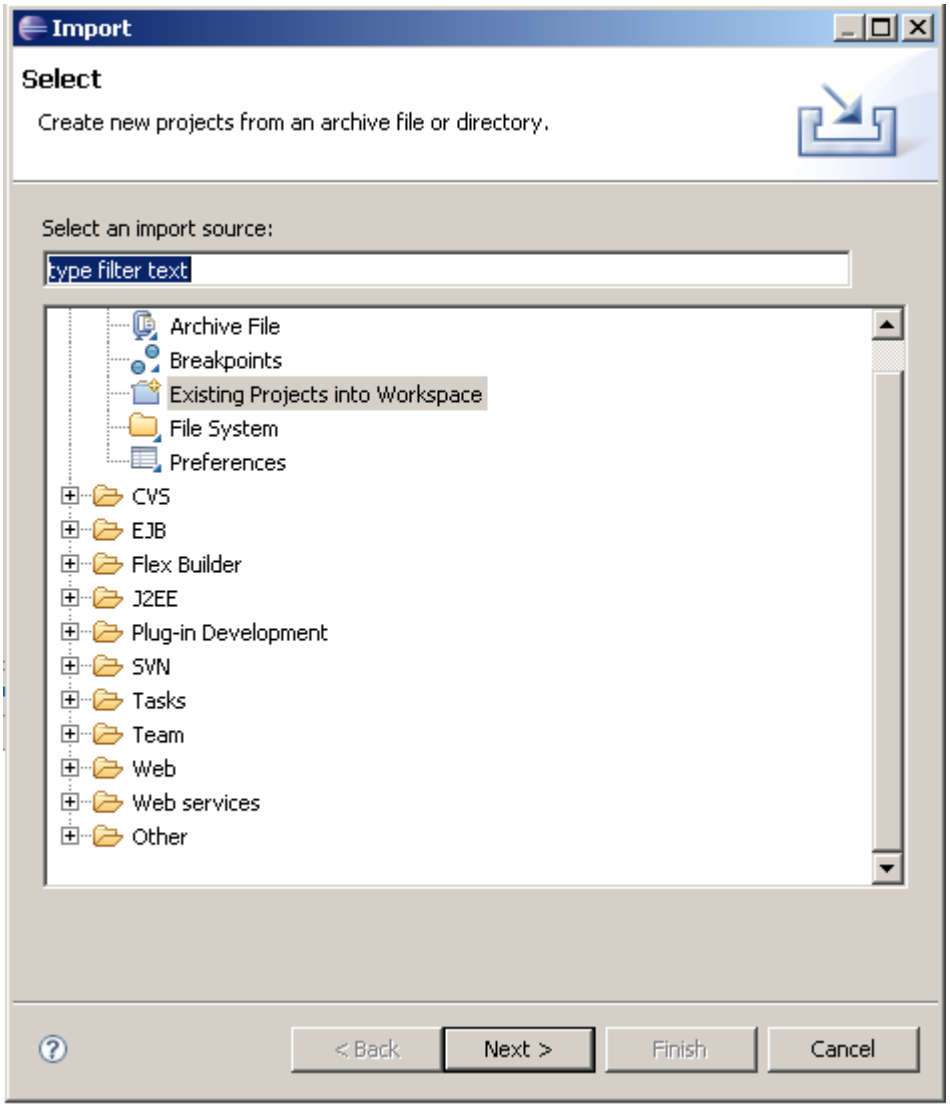
El procedimiento para importar y tener configurado un proyecto de simulador con Adobe Flex Builder es el siguiente:

1) Tome la carpeta del proyecto y cópiela en el directorio de trabajo de Flex Builder (de esta forma si su directorio de trabajo es d:/workspace y decide editar el simulador 72 el directorio raiz de proyecto para este simulador será d:/workspace/sim72 si ha nombrado la carpeta de proyecto como sim72 por supuesto) previamente a haberlo descomprimido si el proyecto el ha llegado e un formato comprimido.

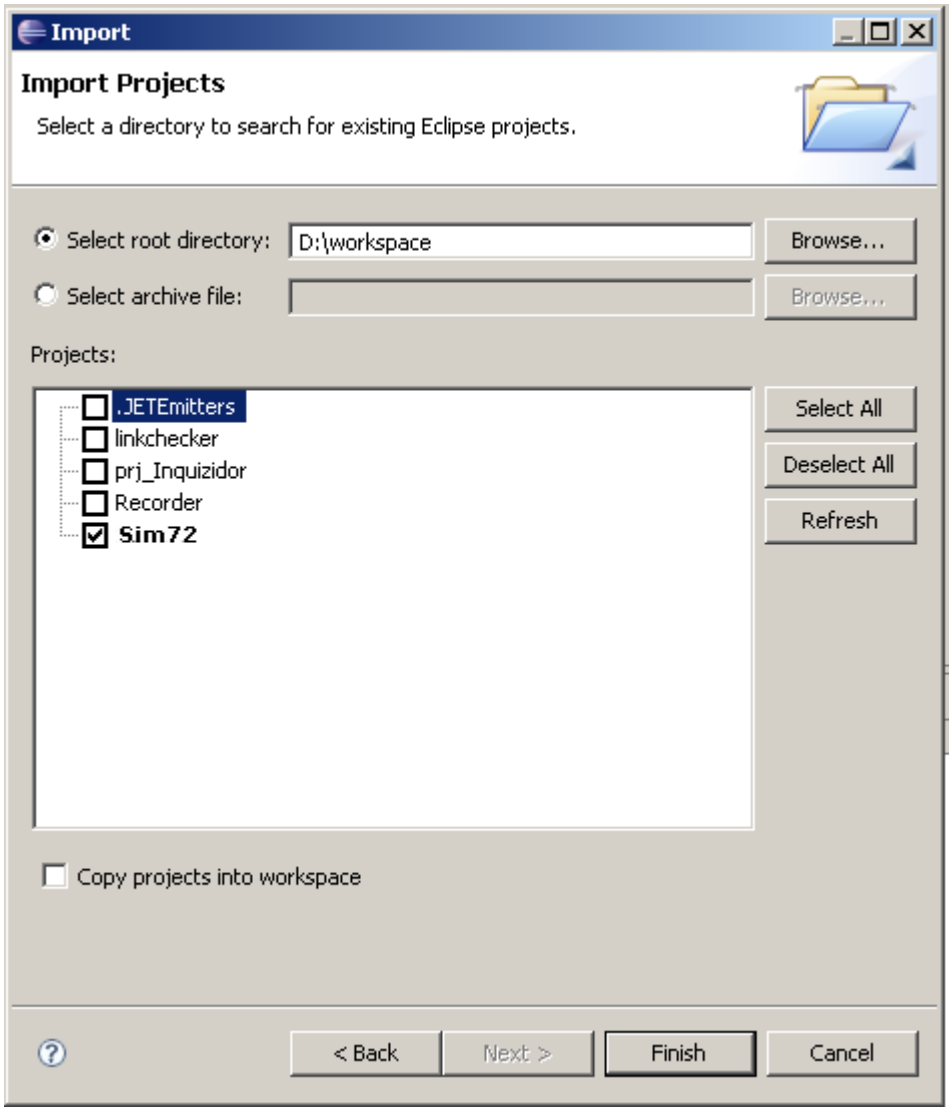
2) En Flex Builder ejecute este comando menú "File" → "Import"



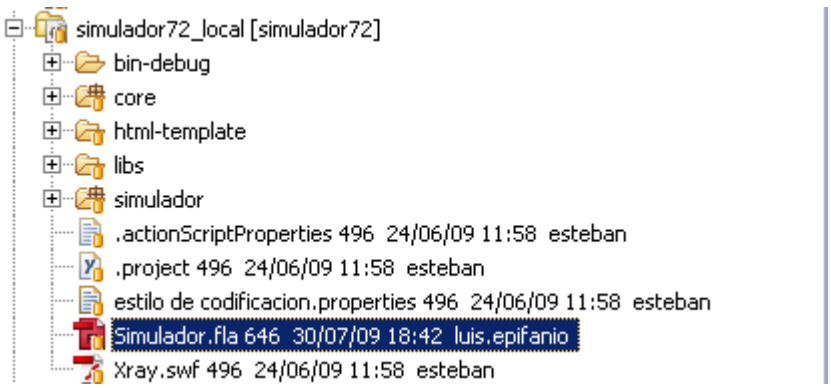
Le aparecerá una ventana como la que se muestra a continuación:



Se elije "Existing Project into workspace" y a continuación "Next" se le mostrará una ventana como esta:



Se pincha en "Finish" y obtendrá entre sus proyectos el simulador a editar:

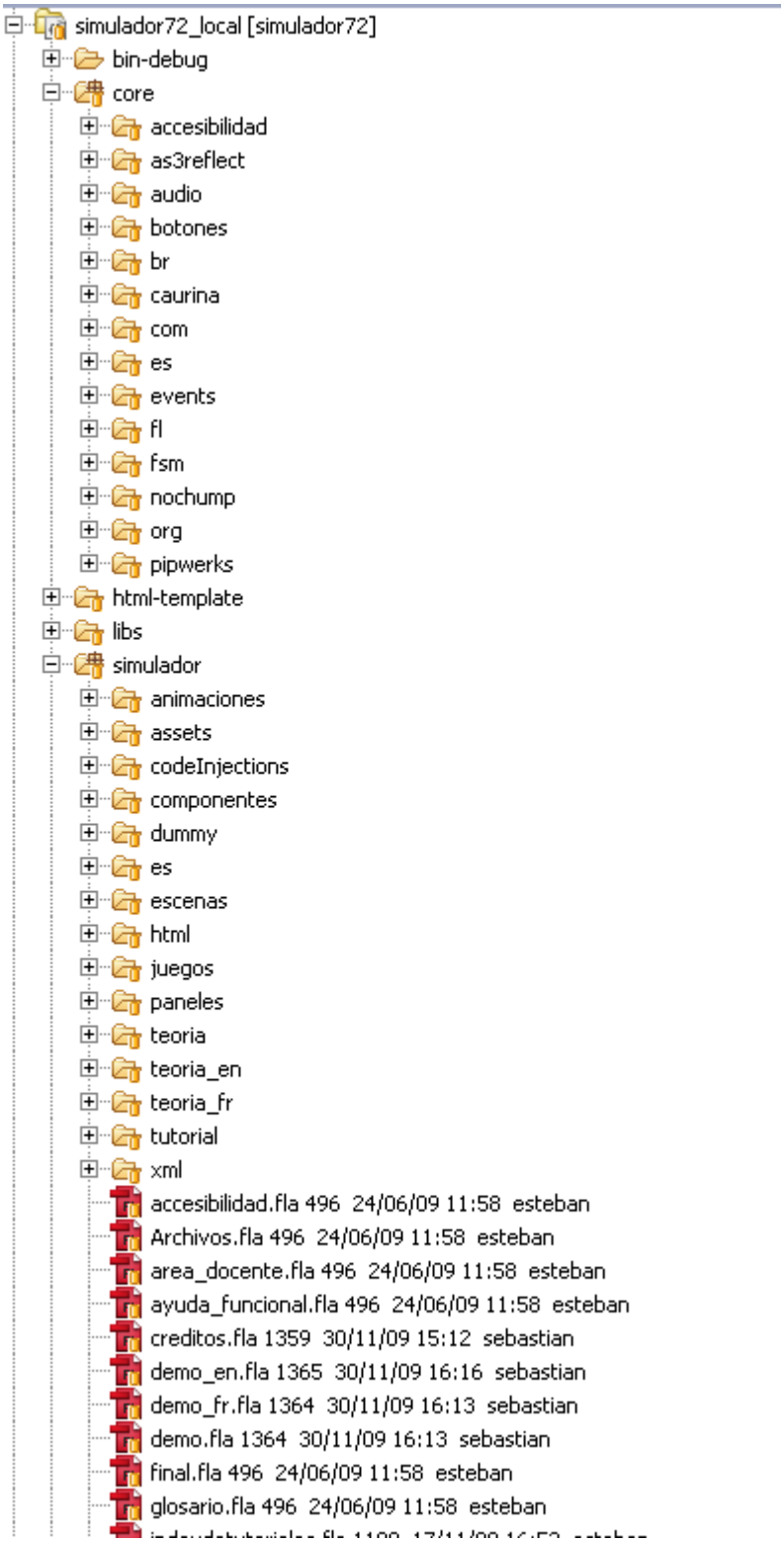


2.4 Descripción de la estructura de un proyecto de simulador

Generalmente un proyecto de simulador contiene los siguientes elementos:

1) **Simulador.fla:** Es el módulo principal del simulador y es el encargado de cargar el resto de animaciones, juegos, pantallas, etc. que maneja el simulador. Controla además los estados de la simulación, lista de tareas y lógica de simulación. Al compilar este modulo se incluyen/actualizan todos los cambios de código fuente incluido en la carpeta core (que se explicará más adelante)
2) **Xray.swf:** Es un copilado swf que permite al programador ver salidas de logging de un simulador en tiempo de ejecución y ayuda en la depuración del mismo
3) **Estilo de codificacion.properties:** Es un archivo con la configuración que contiene las directivas de cómo aplicar el mismo formato al código fuente (en caso si un nuevo programador se suma al equipo). Es un archivo accesorio pero útil si el trabajo se divide entre varios desarrolladores
4) **.project y .actionScriptProperties:** Son archivos que Flex Builder utiliza en su configuración. No deberían editarse si se desconoce su funcionamiento y esto generalmente es innecesario.
5) Carpeta **libs:** Contiene librerías .swc compiladas en caso de necesitarse o completar dependencias en la asistencia de código.
6) Carpeta **bin-debug:** Contiene los archivos resultados de la compilación (compilados) que deberán incluirse o actualizarse en la versión de prueba o entrega según la etapa de desarrollo.

   A continuación se detallarán las carpetas core y simuladores que resultan las de mayor importancia ya que contienen la mayoría de los módulos del simulador sino todos. Por otra parte estas carpetas al contener código fuente y ser un proyecto de Actionscript3 (AS3) se organizan jerárquicamente en carpetas/paquetes que son los que se detallaran, previo a esto mostramos una captura de sus paquetes/directorios principales


simulador72_local [simulador72]
bin-debug
core
accesibilidad
as3reflect
audio
botones
br
caurina
com
es
events
fl
fsm
nochump
org
pipwerks
html-template
libs
simulador
animaciones
assets
codeInjections
componentes
dummy
es
escenas
html
juegos
paneles
teoria
teoria_en
teoria_fr
tutorial
xml
accesibilidad.fla 496 24/06/09 11:58 esteban
Archivos.fla 496 24/06/09 11:58 esteban
area_docente.fla 496 24/06/09 11:58 esteban
ayuda_funcional.fla 496 24/06/09 11:58 esteban
creditos.fla 1359 30/11/09 15:12 sebastian
demo_en.fla 1365 30/11/09 16:16 sebastian
demo_fr.fla 1364 30/11/09 16:13 sebastian
demo.fla 1364 30/11/09 16:13 sebastian
final.fla 496 24/06/09 11:58 esteban
glosario.fla 496 24/06/09 11:58 esteban

7) Carpeta **core:** Contiene el código común a la mayoría de los simuladores y contiene mayormente código reutilizable a nivel de simuladores o bien código frecuentemente utilizado.

   Paquete accesibilidad: Contiene clases as3 relativas a las características accesibles del producto software

   Paquete as3reflect: Contiene clases que facilitan la ejecución dinámica de métodos por configuración xml (se utiliza cuando en el simulador debe ejecutarse alguna porción de código según una condición/estado dada del simulador y no por código).

   Paquete audio: Contiene clases relativas al manejo de audio, es frecuentemente utilizado y probado. Rara situación justificaría su mantenimiento

   Paquete botones: Contiene clases relativas al manejo de diferentes tipos de botones según presente transiciones y estados (estado over, out, clic, etc.)

   Paquete br.com.stimuli.*: Contiene clases relativas al manejo de precarga de los contenidos del simulador. Más conocido como BulkLoader. Código frecuentemente utilizado y probado. Rara situación justificaría su mantenimiento

   Paquete caurina.*: Contiene clases relativas al manejo de transiciones de flash en animaciones, color, línea de tiempo y demás. Código frecuentemente utilizado y probado. Rara situación justificaría su mantenimiento.

   Paquete com. blitzagency.*: Contiene clases relativas al manejo de logging de la aplicación (asóciese con Xray). Código frecuentemente utilizado y probado. Rara situación justificaría su mantenimiento

   Paquete com. robertjpayne.*: Contiene clases relativas al renderizado de escenas 3D mediante técnica de 'vista libre'. Sólo algunas escenas de algunos simuladores hacen uso efectivo del mismo y se lo adjunta como código útil.

   Paquete es.oneclick.*: Contiene clases relativas al manejo de eventos del simulador (**events**), clases administradoras principales (**manager**) que centralizan mucho código crítico como manejo de sesiones, interactividad de usuarios, carga, impresión, logging, temporización; también hay código de pantallas (la de inicio que sufre mayormente cambios estéticos y no programáticos) y utilidades varias (**ui y utils** respectivamente).Contiene código que es estructuralmente común a

varios simuladores.

Paquete events.*: Contiene clases relativas al manejo de otros eventos específicos de cada versión del simulador

Paquete fl.*: Es el paquete fl de adobe flash, se ha colocado para ayudar en la asistencia de código. No requiere mantenimiento ya que el SDK de flash provee uno está a modo de asistencia de código y depuración.

Paquete fsm.*: Contiene clases relativas al manejo de estados que hace el simulador a partir de a especificación realizada en simulador.xml. Este paquete contiene el manejo de la máquina de estados finitos (Finite State Machine: fsm).

Paquete nochump.*, org.papervision3d.*, org.collada.*: Contiene clases relativas al renderizado de 3d para aquellas escenas que hagan uso de la misma.

Paquete pipwerks.*: Contiene clases relativas al manejo de la interacción entre el simulador y el modulo SCORM de los sistemas LMS para los que fue pensado.

8) Carpeta **simulador:** Contiene todos los archivos específicos para el simulador en curso y en ella están la mayoría de los recursos explicados en el inciso 2.1. De forma análoga a que fue explicada la carpeta core se hará con la estructura de esta:

   Paquete animaciones.*: Contiene clases utilizadas en el desarrollo de aquellas animaciones que requieran en mayor o menos medida ajustes de programación.

   Paquete assets.*: Esta carpeta, cuya estructura es análoga a la homónima explicada en el punto 2.1, contiene, a diferencia de aquella, los archivos no compilados de los diferentes recursos. La versión compilada de cada archivo en esta carpeta estará por convención en la carpeta del mismo nombre ubicada en el directorio bin-debug. He aquí una breve descripción de las subcarpetas:
   a. **animaciones**: contiene todas las animaciones que carga el simulador, pueden ser archivos de Adobe Flash CS3/CS4(.fla) o bien películas formato .flv
   b. **audios**: contiene los audios disponibles para el simulador, en su mayoría son archivos formato mp3 (rara vez wav)
   c. **biblioteca**: Son archivos .fla que contienen recursos compartidos (fuentes, imágenes, incluso código as3 compilado).
   d. **extras**: se autodefine

e. **imagenes**: contiene aquellas imágenes disponibles para el simulador o sus componentes; organiza su contenido por medio de carpetas.
f. **navegador**: contiene recursos relativos al panel de navegación del simulador.

Paquete codeInjections.*: Contiene todos los métodos que serán invocados dinámicamente por el simulador según su estado y evento. Los métodos invocados se agrupan, naturalmente en clases y puede haber varias d ellas según se necesiten

Paquete codeInjections.*: Contiene todos los métodos que serán invocados dinámicamente por el simulador según su estado y evento. Los métodos invocados se agrupan, naturalmente en clases y puede haber varias d ellas según se necesiten

Paquete componentes.*: Contiene todas aquellas clases utilizadas para modelar componentes interactivos tales como listas desplegables, glosarios, galerías de imágenes, etc.

Paquete dummy.*: Este es un paquete que tiene por propósito alojar a todos aquellos prototipos o versiones de pruebas (que denominamos dummies). No contiene versiones finales de módulo ya que han sido reemplazado por los definitivos, sin embargo en el proceso de desarrollo estos prototipos han sido versionados.

Paquete es.oneclick.*: Contiene código que es específico al simulador en desarrollo. Generalmente alberga eventos específicos de este simulador, clases controladoras específicas de este simulador. El paquete ui alberga la mayoría de las interfaces de usuario utilizadas específicamente en es simulador

Paquete escenas.*: Contiene código que es específico al simulador en desarrollo. Y como su nombre indica contiene la programación de las diferentes escenas/perspectivas q posee el simulador. Además tiene la programación de todos los elementos interactivos presentes en ella. A nomenclatura de las salas Escena[i][j].as, donde **"i"** es el Numero de sala lógica (Ej. Escena1) y donde **"j"** es una letra que indica la vista de la sala por lo que una clase seria Escena1a.as. En cuanto a los elementos interactivos se organizan en paquetes de acuerdo al nombre de la sala. Así un nombre valido de paquete sería **"salacontrol"** dicho paquete tendría los objetos interactivos Ordenador.as por ejemplo

Paquete html.*: Contiene hojas de estilos utilizadas

Paquete juegos.*: Contiene las clases de cada uno de los juegos interactivos presentes en el simulador, generalmente tienen el mismo nombre que el archivo .fla que le implementa.

Paquete paneles.*: Contiene las clases que soportan los distintos paneles del simulador y se organiza por paquetes; por lo tanto habrá clases para configuración, feedback (diálogos, alerta, dialogo, etc.), interfaz, sesiones, etc.

Paquete teoria.*: Es el paquete donde se alojan todos lo contenidos externos tales como accesibilidad, teoría guiada, ayuda funcional, etc. Una nota particular acerca de este paquete es que su contenido es generado según el idioma de destino, así para adecuar el contenido al idioma inglés se generaron los paquetes **teoria_en**, **teoria_fr.**

Paquete tutorial.*: Contiene las clases implementan el apartado de tutorial.

Paquete xml.*: Contiene todos los archivos de configuración que cualquier módulo del simulador o el mismo simulador puedan necesitar para su funcionamiento. Asimismo toda Clase de contenido externo (que derive de `es.oneclick.ui.PantallaInicio`) buscara en este directorio u archivo xml de configuración con el nombre de la clase en minúsculas. De esta forma si el contenido es "PantallaTest.as" buscara en este directorio el archivo de configuración "pantallatest.xml".

Para finalizar el análisis de este directorio de la estructura del proyecto debemos mencionar que todo contenido que sea cargado por el simulador tales como juegos, mapas conceptuales, demo, glosario, finales de simulación, intro, accesibilidad, etc. Se encuentran en esta carpeta, los archivos .fla presentes en esta carpeta dejarán su archivo compilado en la carpeta bin-debug según el nombre se le configure. Debe notarse además que la clase base de cada simulador (**Simulador.as**) se encuentra en esta carpeta y es a clase base del archivo principal de compilación (**Simulador.fla**).