wSafe: Entorno de compilación para "SAFE" en la Web

Proyecto de Sistemas Informáticos

Autores: **Lope Javier Hidalgo, Enrique Omar Huidobro, Juan Antonio Masa**
Profesor director: **Ricardo Peña**
Facultad de Informática
Universidad Complutense de Madrid
Julio 2009

Resumen / Abstract

Resumen

El proyecto desarrollado se trata de una aplicación web que controla principalmente el compilador safe, pudiendo interactuar con otras herramientas como Isabelle, Jasmin…

La aplicación gestionará las correspondientes etapas del compilador safe o de las herramientas externas, ejecutando todas las fases, las fases deseadas o una demostración que ejecutará todas las fases con un fichero .safe establecido por defecto como entrada.

Proporciona a los ficheros generados de estas etapas una "vista" que permite visualizarlos de una manera más amigable usando para ello las transformaciones xslt que se realizan sobre los ficheros xml.

Permite también la descarga de estos ficheros en formato xml para la edición por parte de usuarios avanzados y posterior carga en el servidor para la ejecución de los ficheros modificados.

Abstract.

The developed Project is a web application that controls the safe compiler, with interaction with other tools like Isabelle, Jasmin…

The application manages the several safe-compiler stages or external tools, executing all the stages, the wanted stages or a demo that executes all the stages with a .safe file established by default like input.

It provides the generated files from these stages with a view that allows visualizing them in a friendly interface through xslt transformations made in the xml files.

As well, it allows to download the xml files for editing by advanced users and uploading for execution the modified files later.

Índice general

Capítulo 1

Introducción

1.1. Contexto del proyecto

La mayoría de los lenguajes funcionales permiten que el programador se olvide de los detalles de manejo de la memoria. El sistema en tiempo de ejecución se encarga de reservar memoria a medida que se van evaluando las expresiones del programa mientras exista memoria disponible suficiente. En caso de que se agote la memoria comienza a ejecutarse un recolector de basura que determina qué partes de la memoria están muertas y por tanto pueden ser reutilizadas. Esto implica normalmente la suspensión de la ejecución del programa original. En caso de que el recolector haya logrado recuperar suficiente cantidad de memoria se puede continuar con la ejecución, mientras que en caso contrario el programa aborta. Este modelo es aceptable en muchas situaciones puesto que los programas no se ven oscurecidos por detalles de bajo nivel sobre el manejo de la memoria. Pero en otros contextos puede no serlo por los siguientes motivos:

- El retraso introducido en la ejecución por la recolección de basura impide proporcionar la respuesta en un determinado tiempo de reacción, como se requiere en programas con respuesta en tiempo real o altamente interactivos.
- El fallo de los programas debido al agotamiento de la memoria puede provocar daños personales o económicos inaceptables a los usuarios de los programas. Este es el caso de aplicaciones críticas en seguridad.

Por otra parte, muchos lenguajes imperativos ofrecen mecanismos de bajo nivel para reservar y liberar memoria que el programador puede utilizar para crear y destruir dinámicamente estructuras de datos utilizando punteros. Estos mecanismos proporcionan al programador un control completo sobre el uso de la memoria pero son fuente de numerosos errores. Algunos problemas bien conocidos son los punteros descolgados, la compartición indeseada con efectos laterales inesperados, y la saturación de la memoria con basura.

Ni el enfoque implícito de los lenguajes funcionales ni el explícito de los imperativos proporciona una lógica clara con la que el programador pueda razonar sobre el consumo de memoria. En particular, es difícil anticipar, o demostrar, una cota superior de la cantidad de memoria que garantiza la ausencia de fallos en la ejecución debidos al agotamiento de la memoria.

En el Departamento de Sistemas Informáticos y Programación de la Universidad Complutense de Madrid, el profesor Ricardo Peña dirigió un proyecto coordinado con las universidades de Castilla La Mancha, Politécnica de Valencia y Málaga, continuado posteriormente con otro llamado STAMP, conjunto con la Universidad Politécnica de Valencia y la Universidad de Almería. En el seno de estos proyectos, el grupo de la UCM trabaja en el área del *Código con Demostración Asociada* (en inglés *Proof Carrying Code* o PCC). En este contexto han desarrollado un enfoque semiexplícito del manejo de la memoria definiendo un lenguaje funcional llamado *SAFE* que tiene diversas facilidades para el manejo de la memoria, tales como la destrucción explícita y la ausencia de un recolector de basura. El compilador de SAFE está prácticamente terminado y consta de numerosas etapas que generan ficheros intermedios que pueden ser consultados por el programador.

1.2. Objetivos del proyecto

El objetivo de este proyecto es desarrollar una herramienta web que constituya una interfaz gráfica para trabajar con el compilador de SAFE comentado en el apartado anterior. Dicha interfaz permitirá a los usuarios que accedan a ella a través de la red, ejecutar individualmente las etapas del compilador, analizar los resultados obtenidos en cada una de las fases que lo componen y ejecutar cualquier programa implementado en dicho lenguaje.

El sistema debe permitir cierto nivel de interacción a usuarios experimentados, tales como la modificación de los ficheros intermedios y debe ofrecer la posibilidad de realizar un acceso concurrente por parte de distintos usuarios.

1.3. Plan de trabajo

A continuación, se expone una descripción de cada uno de los capítulos que componen este documento:

- En el capítulo 2 comentaremos a grandes rasgos los puntos fuertes del proyecto SAFE así como las principales características de este lenguaje.
- El capítulo 3 lo dedicaremos exclusivamente a dar una descripción detallada de las fases que componen el compilador de SAFE y los archivos de entrada y salida que éstas generan.
- El capitulo 4 tendrá como fin analizar en detalle las tecnologías web utilizadas para el desarrollo del proyecto, exponiendo los pros y los contras de cada una de ellas, así como las razones que nos han llevado a decantarnos por su uso.

- En el capítulo 5 abordaremos de manera rigurosa las principales características del sistema wSafe. En primer lugar, nos centramos en detallar el diseño de la interfaz de usuario y sus principales funciones. Seguidamente, comentaremos todas las opciones importantes del sistema, y más concretamente, las opciones asociadas a cada fase del compilador y las asociadas a los archivos generados por cada una de ellas.

- En el capítulo 6, se ofrece un completo manual de usuario, que explica todos los detalles acerca del uso de la aplicación y sus principales funciones.

- Por último, en el capítulo 7 trataremos las conclusiones obtenidas durante el desarrollo del proyecto.

Además de los capítulos anteriormente citados, incluimos una serie de apéndices, referenciados desde los capítulos, donde adjuntamos la estructura de los ficheros generados, listados de código de la implementación, así como algunos ejemplos de interés.

Capítulo 2

El proyecto SAFE

2.1. El lenguaje SAFE

SAFE es un lenguaje funcional polimórfico de primer orden similar a Haskell o ML en cuanto a su sintaxis, en el que el programador coopera de forma moderada con el sistema de manejo de la memoria proporcionando cierta información sobre el uso que se pretende hacer de las estructuras de datos. Por ejemplo, puede indicar que una estructura de datos particular no se va a necesitar más y que consecuentemente se puede destruir de forma segura y reutilizar la memoria que ocupa. También permite el control de la compartición entre distintas estructuras de datos. La gestión de la memoria se realiza mediante regiones. Los beneficios de este enfoque son fundamentalmente que no es necesario un recolector de basura y que es posible razonar más fácilmente sobre la cantidad de memoria que consume un programa.

Pero aun más importante que las facilidades concretas que se le ofrecen al programador, es la definición de un sistema de tipos que garantiza que el uso de las mismas se hace de forma segura. Este sistema de tipos garantiza que no se crean punteros descolgados en la memoria. La definición del sistema de tipos incluye un análisis de compartición definido mediante interpretación abstracta que permite determinar qué variables están en peligro por el hecho de destruir una estructura de datos.

2.2. Sintaxis del lenguaje

SAFE es un lenguaje impaciente donde la compartición se impone usando variables en las aplicaciones de función y de constructor. La sintaxis amarga de SAFE es el lenguaje núcleo que resulta de la compilación de un lenguaje de mayor nivel llamado *Full-Safe.* En la figura 2.1 se muestran las reglas que componen la sintaxis *Core-Safe*.

El modelo de memoria de SAFE está basado en regiones donde se crean las estructuras de datos. Una región es una colección de celdas y una celda almacena exactamente una aplicación de constructor. En Full-Safe las regiones son implícitas y son inferidas cuando *Full-Safe* se *desazucara* en *Core-Safe*.

$$
\begin{array}{lll}
prog \rightarrow \overline{data_i}^{\,n} \; ; \; \overline{dec_j}^{\,m} \; ; \; e & & \{\textit{Core-Safe} \text{ program}\} \\
data \rightarrow \textbf{data}\ T\ \overline{\alpha_i}^{\,n}\ @\ \overline{\rho_j}^{\,m} = \overline{C_k\ \overline{t_{ks}}^{\,n_k}\ @\ \rho_m}^{\,l} & & \{\text{recursive, polymorphic data type}\} \\
dec \rightarrow f\ \overline{x_i}^{\,n}\ @\ \overline{r_j}^{\,l} = e & & \{\text{recursive, polymorphic function}\} \\
e \rightarrow a & & \{\text{atom: literal } c \text{ or variable } x\} \\
\quad \mid x\ @\ r & & \{\text{copy data structure } x \text{ into region } r\} \\
\quad \mid x! & & \{\text{reuse data structure } x\} \\
\quad \mid a_1 \oplus a_2 & & \{\text{primitive operator application}\} \\
\quad \mid f\ \overline{a_i}^{\,n}\ @\ \overline{r_j}^{\,l} & & \{\text{function application}\} \\
\quad \mid \textbf{let}\ x_1 = be\ \textbf{in}\ e & & \{\text{non-recursive, monomorphic}\} \\
\quad \mid \textbf{case}\ x\ \textbf{of}\ \overline{alt_i}^{\,n} & & \{\text{read-only case}\} \\
\quad \mid \textbf{case!}\ x\ \textbf{of}\ \overline{alt_i}^{\,n} & & \{\text{destructive case}\} \\
alt \rightarrow C\ \overline{x_i}^{\,n} \rightarrow e & & \{\text{case alternative}\} \\
be \rightarrow C\ \overline{a_i}^{\,n}\ @\ r & & \{\text{constructor application}\} \\
\quad \mid e & &
\end{array}
$$

Figura 2.1: Sintaxis *Core-Safe*

La ocupación y desocupación de regiones están asociadas a la invocación de funciones: una *región de trabajo* está ocupada cuando se entra en la llamada de la función y desocupada cuando se sale de ella. Todas las estructuras de datos asignadas a dicha región se perderán.

Dentro de una función se deben construir las estructuras de datos, pero éstas también deben ser destruidas usando un *patrón destructivo,* denotado con el símbolo !, que desocupa las celda correspondiente al constructor más externo. Como ejemplo, se muestra a continuación la función *append* en Full-Safe:

```
appendD []!     ys  = ys
appendD (x:xs)! ys  = x : appendD xs ys
```

Otra posibilidad es destruir parte de la estructura de datos y re-usar el resto en el resultado, como se observa en la función *split*:

```
splitD 0 zs!       = ([], zs!)
splitD n []!       = ([], [])
splitD n (y:ys)!   = (y:ys1, ys2) where (ys1, ys2) = splitD (n-1) ys
```

La zs! De la derecha indica que la lista zs no se continua destruyendo sino que se reutiliza para devolver el resultado.

Como se ha indicado anteriormente, la sintaxis *Core-Safe* pretende ser el núcleo de un lenguaje de mayor nivel similar a Haskell. La encargada de soportar este lenguaje de alto nivel es la sintaxis *Full-Safe*. Si sólo se contara con la sintaxis *Core-Safe* como medio de implementación, la programación en SAFE resultará muy tediosa. Por ello, es necesario una sintaxis que ofrezca más facilidades al programador, la sintaxis *Full-Safe*, con el fin de simplificar el trabajo al mismo. Posteriormente, mediante las

transformaciones de *desazucaramiento*, los programas escritos a partir de la sintaxis *Full-Safe* se convertirán en su equivalente *Core-Safe*.

En la sintaxis *Core-Safe* únicamente son posibles las definiciones de función, sin embargo en la sintaxis *Full-Safe* se permite la definición de declaraciones de tipo y de datos. Las primeras serán de gran utilidad para especificar los tipos de los datos empleados así como el de las funciones. Las segundas ofrecerán al programador la posibilidad de crear sus propias estructuras para almacenar información.

Por ejemplo la version Core-Safe de append sería:

```
append xs ys @r = case ! xs of
          [ ] -> ys
          x : xx ->  let yy = appendD xx yy @ r in
                     let zz = x : yy @ r
                     in zz
```

Capítulo 3

El compilador de SAFE

3.1. Fases del compilador

Las fases que componen el compilador de SAFE pueden verse en la figura 3.1. La primera fase la constituye el **Analizador Léxico**, cuyo parámetro de entrada es una lista de caracteres, correspondiente al programa fuente que el usuario desea compilar. La unidad más pequeña que maneja este analizador es el carácter.

El objetivo es obtener una lista de unidades léxicas (ULs) que será la unidad utilizada en la siguiente etapa: el **Analizador Sintáctico**. Dichas ULs pueden ser identificadores, operadores infijos, palabras reservadas, símbolos especiales del lenguaje, etc. El resultado obtenido tras realizar el análisis sintáctico es un Árbol de Sintaxis Abstracta.

La siguiente fase se denomina **Renombrador/Comprobación de la Semántica:** El análisis implementado en las fases posteriores supone que todas las variables ligadas tienen diferentes nombres, así pues cada variable se reemplaza por una nueva. Además, se comprueban las restricciones contextuales del lenguaje, por ejemplo, que todas las variables se usan dentro de su mismo ámbito, que todas las funciones y constructoras de datos se llaman con el número adecuado de argumentos, etc.

Durante la **Inferencia de Tipos Hindley-Milner** se decora cada expresión, constructora de datos y definición de función con la información de tipos inferida en esta fase. El Árbol de Sintaxis Abstracta se decora también con información sobre las regiones de cada una de las estructuras de datos que lo componen. En particular, se determina si una estructura de datos dada es o no local a la actual llamada de función.

A continuación se realizan una serie de **Transformaciones Core-Safe.** En esta fase el programa *Safe* original se traduce en una versión *Core-Safe desazucarada* y semánticamente equivalente. Además, se preservan las decoraciones de tipos inferidas en fases previas.

La fase siguiente es el **Análisis de Compartición:** Dadas dos variables pertenecientes a la misma función, se calcula si las respectivas estructuras de datos que son apuntadas por ellas en tiempo de ejecución comparten posiciones de memoria. El programa se decora con esa información de compartición, la cual es necesaria en la siguiente fase.

Por último, se realiza el **Análisis de Destrucción.** En esta fase se infiere un tipo seguro para el programa origen, garantizando que no se generen punteros descolgados como consecuencia de una sentencia **case!.**

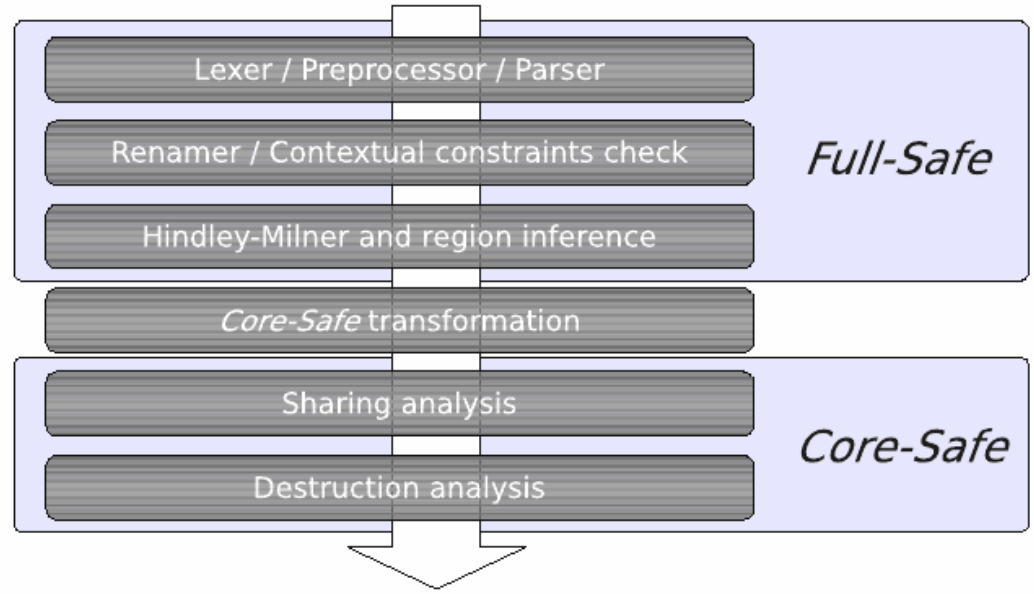


Figura 3.1: Fases del compilador

Además de las fases mostradas en la figura 3.2, existen otras fases que son dignas de mencionar:

La fase de **Generación de Certificados,** que toma el resultado del análisis de destrucción y genera una demostración matemática en un formato que puede ser procesado por el asistente de demostraciones Isabelle/HOL.

Las fases de **generación de código** para la máquina SVM (Safe Virtual Machine) y para la máquina JVM (Java Virtual Machine) y de **reglas de reescritura de términos** para la herramienta de reescritura MU-Term.

Capítulo 4

Tecnologías web utilizadas

4.1. HTML

Comenzaremos este capítulo de "Tecnologías web utilizadas" hablando de HTML, ya que hoy en día hablar de una página o de una aplicación web, es prácticamente sinónimo de hablar de HTML.

HTML son las siglas de *HyperText Markup Language* (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes, o enlaces (links) que conducen a otros documentos o a otras fuentes de información (por ejemplo bases de datos) que pueden estar en tu propia máquina o en máquinas remotas de la red.

Por supuesto, la estética de los documentos escritos en HTML no se limita a texto sencillo; HTML consigue todos los efectos que habitualmente se pueden producir con un moderno procesador de textos: negrita, cursiva, distintos tamaños y fuentes, tablas, párrafos tabulados, sangrías, incluso texto y fondo de página de colores, y muchos más. No obstante, lo más correcto es dejar la presentacion del documento (colores, tamaños, estilos...) a las hojas de estilos, *CSS*, que veremos más adelante. Con HTML también podemos incluir *scripts*, como por ejemplo *Javascript*, el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

El diseño en HTML aparte de cumplir con las especificaciones propias del lenguaje debe respetar unos criterios de accesibilidad web, siguiendo unas pautas, o las normativas y leyes vigentes en los países donde se regule dicho concepto. Se encuentra disponible y desarrollado por el W3C a través de las Pautas de Accesibilidad al Contenido Web 1.0 WCAG, aunque muchos países tienen especificaciones propias como España con la Norma UNE 139803.

Por todo ello, es necesario el uso de HTML en nuestra aplicación, ya que es el estándar utilizado y reconocido por todos los navegadores, y entre ellos los más utilizados como Mozilla Firefox, Internet Explorer etc. Independientemente de las tecnologías que utilicemos para el desarrollo de nuestra aplicación web, es HTML lo que deberemos mostrar al navegador.

4.2. CSS

Las hojas de estilo en cascada *(Cascading Style Sheets, CSS)* son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

CSS se utiliza para dar estilo a documentos HTML, separando el contenido de la presentación. Los e*stilos* definen la forma de mostrar los elementos HTML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la hoja de estilo CSS afectará a todas las páginas vinculadas a dicha hoja de estilo en las que aparezca ese elemento.

El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que le aplicaremos a:

- Un web entero, de modo que se puede definir la forma de todo el web de una sola vez.
- Un documento HTML o página. En un pequeño trozo de código en la cabecera se puede definir la forma que se aplicará a toda la página.
- Una porción del documento HTML, aplicando estilos que serán visibles en un trozo de la página.
- Una etiqueta en concreto, llegando incluso a poder definir varios estilos diferentes para una sola etiqueta. Esto es muy importante ya que ofrece potencia en nuestra programación. Podemos definir, por ejemplo, varios tipos de párrafos: en rojo, en azul, con márgenes, sin ellos...

Hoy en día existen tres versiones: CSS1 y CSS2, con CSS3 en desarrollo por el *World Wide Web Consortium* (W3C). Los navegadores modernos implementan CSS1 bastante bien, aunque existen pequeñas diferencias de implementación según marcas y versiones de los navegadores. CSS2, sin embargo, está solo parcialmente implementado en los más recientes.

Durante el desarrollo de nuestro sistema Web, separaremos el contenido de la presentación, por lo que paralelamente a la creación de páginas HTML, deberemos ir creando la correspondiente hoja de estilos, con el objetivo de definir el formato necesario para visualizar correctamente la aplicación en los distintos navegadores.

4.3. PHP

Debido a la importancia de PHP dentro de nuestra aplicación, comenzaremos está sección del capítulo 4 dando diferentes definiciones que podemos encontrar en la red:

- **www.internet-didactica.es**: "*Acrónimo recursivo de PHP Hypertext Pre-processor (pre-procesador PHP de hipertexto). Es un lenguaje de programación de libre licencia empleado comúnmente para la creación de páginas web dinámicas*".

- **www.icad.com**: "*(Hyper Text Pre-Prosessor / Personal Home Pages), Lenguaje de programación de licencia libre, empotrado dentro del HTML y ejecutado en el servidor antes de ser enviado al navegador, usado para crear paginas dinámicas (datos dinámicos)*".

- **www.portal-uralde.com**: "*Lenguaje de programación de estilo clásico, o sea, con variables, sentencias condicionales, bucles, funciones, etc. Está más cercano al JavaScript o a C. PHP se ejecuta en el Servidor*".

- **www.eseguridad.gob.mx**: "*Es un lenguaje de código abierto sumamente popular especialmente utilizado para desarrollar aplicaciones que se ejecutan en servidores Web y puede ser integrado en HTML. Las siglas son por PHP Hypertext Preprocessor*".

Ampliando estás definiciones podemos decir que PHP es el acrónimo recursivo que significa ***P**HP **H**ypertext **P**re-processor.* PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser empotrado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores.

PHP se utiliza para complementar páginas en HTML para hacer una Web más interactiva y funcional. Gracias a este lenguaje se puede desarrollar cualquier tipo de servicios a través de su sitio Web. Es por ello que hemos decidido desarrollar la mayor parte de nuestra aplicación en este lenguaje.

Un código empotrado dentro de HTML, es un código que puede encontrarse junto con código HTML. En el caso que nos ocupa, dentro de un código PHP (archivos con extensión php), podremos encontrar fragmentos de código HTML, y fragmentos de código PHP, este siempre dentro de marcas "*<?php*" y "*?>*".

Como hemos comentado anteriormente, el código en PHP se ejecuta y se interpreta en el lado del servidor, cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el *script* solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente, y se mostrará correctamente en el navegador.

Una de las ventajas más importantes de PHP es que es un software de licencia libre o software libre. El software libre ofrece total libertad a los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Por lo que es posible encontrar en la red numerosos ejemplos, librerías, etc. de las que podremos hacer uso con total libertad. Si bien, esta es una de las ventajas más importantes, existen otras ventajas para el uso de PHP, entre ellas:

- Es un lenguaje multiplataforma, es decir, es posible ejecutarlo desde prácticamente todos los sistemas operativos existentes. En nuestro caso, haremos uso de PHP bajo un servidor LINUX.

- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL. Si bien, en nuestra aplicación no haremos uso de esta ventaja.

- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones). En parte, debido a lo comentado anteriormente, por ser un software libre.

- Posee una amplia documentación en su página oficial (http://www.php.net/manual/es/), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Esto en nuestro caso, y extrapolando a cualquier aplicación desarrollada en PHP es de gran utilidad, debido a la facilidad de encontrar el significado o funcionamiento de diferentes procedimientos y variables a usar.

- Permite las técnicas de Programación Orientada a Objetos.

También podemos añadir, que es un leguaje, fácil de aprender, debido al gran parecido que posee con los lenguajes más comunes de programación estructurada, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Como desventaja principal, podemos señalar que si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable, con todo lo que eso conlleva.

Dentro de nuestra aplicación, los *scripts* PHP se encargarán de crear la página HTML que se mostrará en el navegador, esto se hará en conjunto con otros *scripts* desarrollados en lenguaje *Javascipt*, que como veremos más adelante se ejecutan en el lado del cliente. De esta forma obtendremos el control de todo lo que se ejecuta en nuestra aplicación, ya sea del lado del cliente con *Javascipt*, o del lado del servidor, con PHP. Además de esto, toda la funcionalidad de la aplicación es generada y controlada prácticamente por scripts PHP. Por poner un ejemplo, la creación del comando para ejecutar una fase determinada del compilador SAFE, así como el lanzamiento de dicho comando, y el control de errores, se controla mediante PHP en el lado del servidor.

4.4. JavaScript

JavaScript es un lenguaje de programación interpretado, utilizado principalmente en páginas web, para realizar tareas y operaciones en el marco de la aplicación cliente, con una sintaxis semejante a la del lenguaje Java.

Al igual que Java, JavaScript es un lenguaje Orientado a Objetos propiamente dicho, ya que dispone de herencia, si bien ésta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM (especificación que determina cómo los objetos -texto, imágenes, enlaces, etc.- en una página Web son representados)

Con JavaScript se pueden extender las posibilidades de las páginas Web como por ejemplo, evitar que se pueda copiar el texto de una página, mostrar botones para agregar automáticamente una página a favoritos, crear barras de *scroll*, abrir *popups*, cambiar el aspecto del puntero del ratón, rotar *banners*, validar formularios, etc.

Dentro de nuestra aplicación, JavaScript es el encargado, en conjunto con PHP, de generar el dibujo que muestra la estructura de nuestra aplicación, utilizando para ello un fichero de configuración que contiene la descripción del grafo del compilador. Además es JavaScript el encargado de controlar la correcta utilización de las diferentes

posibilidades que ofrece nuestra aplicación, esto es, controlar que hayamos seleccionado una ruta valida, que hayamos marcado un comienzo y un fin, etc.

4.5. XML

XML, siglas en inglés de *Extensible Markup Language* («lenguaje de marcas ampliable»), es un metalenguaje extensible de etiquetas. Permite definir la gramática de lenguajes específicos Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más útil y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Algunas de las ventajas más importantes de XML son las siguientes:

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.

- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan errores y se acelera el desarrollo de aplicaciones.

- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones.

La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de pedazos de información. Ejemplos son un tema musical, que se compone de compases, que están formados a su vez por notas. Estas partes se llaman *elementos*, y se las señala mediante etiquetas.

Para nuestra aplicación decidimos que todos los ficheros generados por cada fase del compilador SAFE fueran en XML. De este modo, podemos trabajar con una estructura capaz de ser reconocida y analizada, y podemos darle el formato y la presentación deseadas.

4.6. XSLT

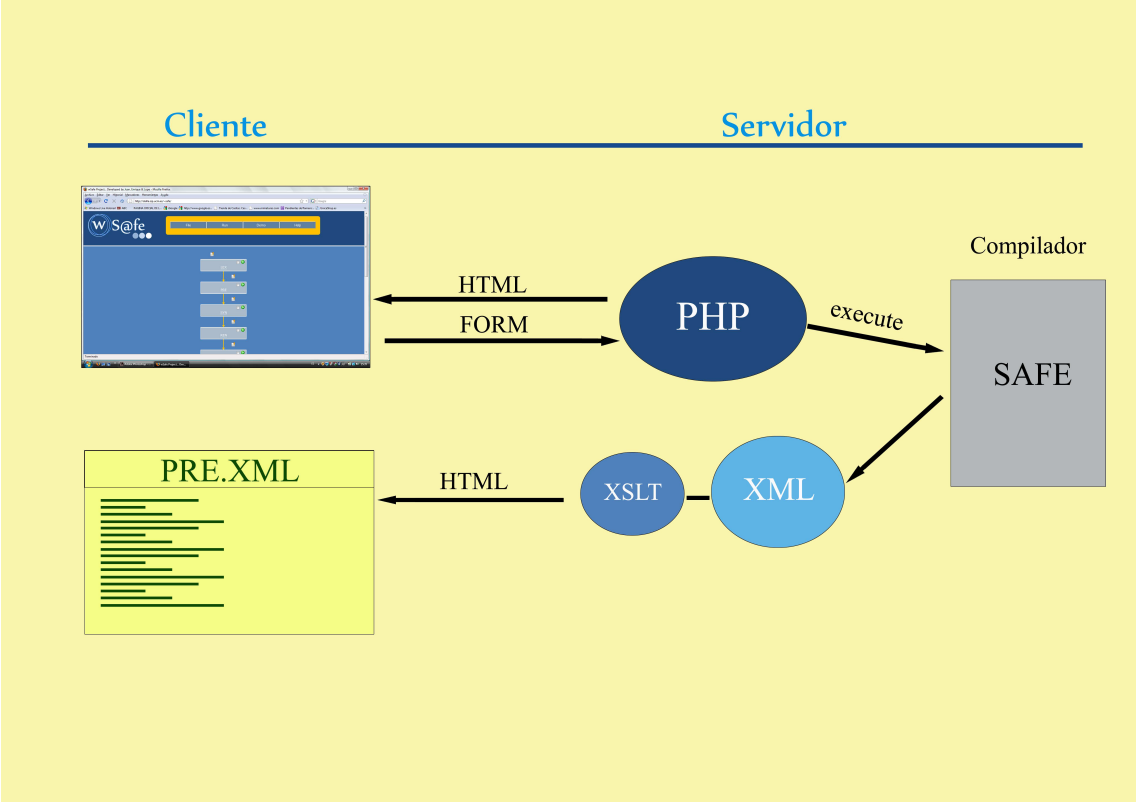
XSLT o Transformaciones XSL es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML, en nuestro caso, transformaremos los documentos generados en XML por las distintas fases del compilador SAFE en documentos HTML, para de este modo poder darles el formato y la presentación necesarias. Las hojas de estilo XSLT realizan la transformación del documento utilizando una o varias *reglas de plantilla*. Estas reglas de plantilla unidas al documento fuente a transformar alimentan un procesador de XSLT, el cual realiza las transformaciones deseadas poniendo el resultado en un archivo de salida o, como en el caso de una página web, las muestra directamente en un dispositivo de presentación tal como el monitor del usuario.

Actualmente, XSLT es muy usado en la edición web, generando páginas HTML o XHTML. La utilización de XML y XSLT permite separar contenido y presentación, aumentando así la productividad.

Como hemos comentado antes, dentro de nuestra aplicación, mediante diferentes plantillas XSLT, transformaremos los documentos intermedios del compilador SAFE (generados en XML) en documentos HTML, y de este modo, junto con las hojas de estilo podemos mostrar el contenido de dichos documentos intermedios en el navegador, con un formato más agradable y ordenado (colores, sangrado, etc.).

4.7. Nuestra aplicación

En la siguiente imagen podemos observar el ámbito de cada tecnología dentro de nuestra aplicación wSafe.



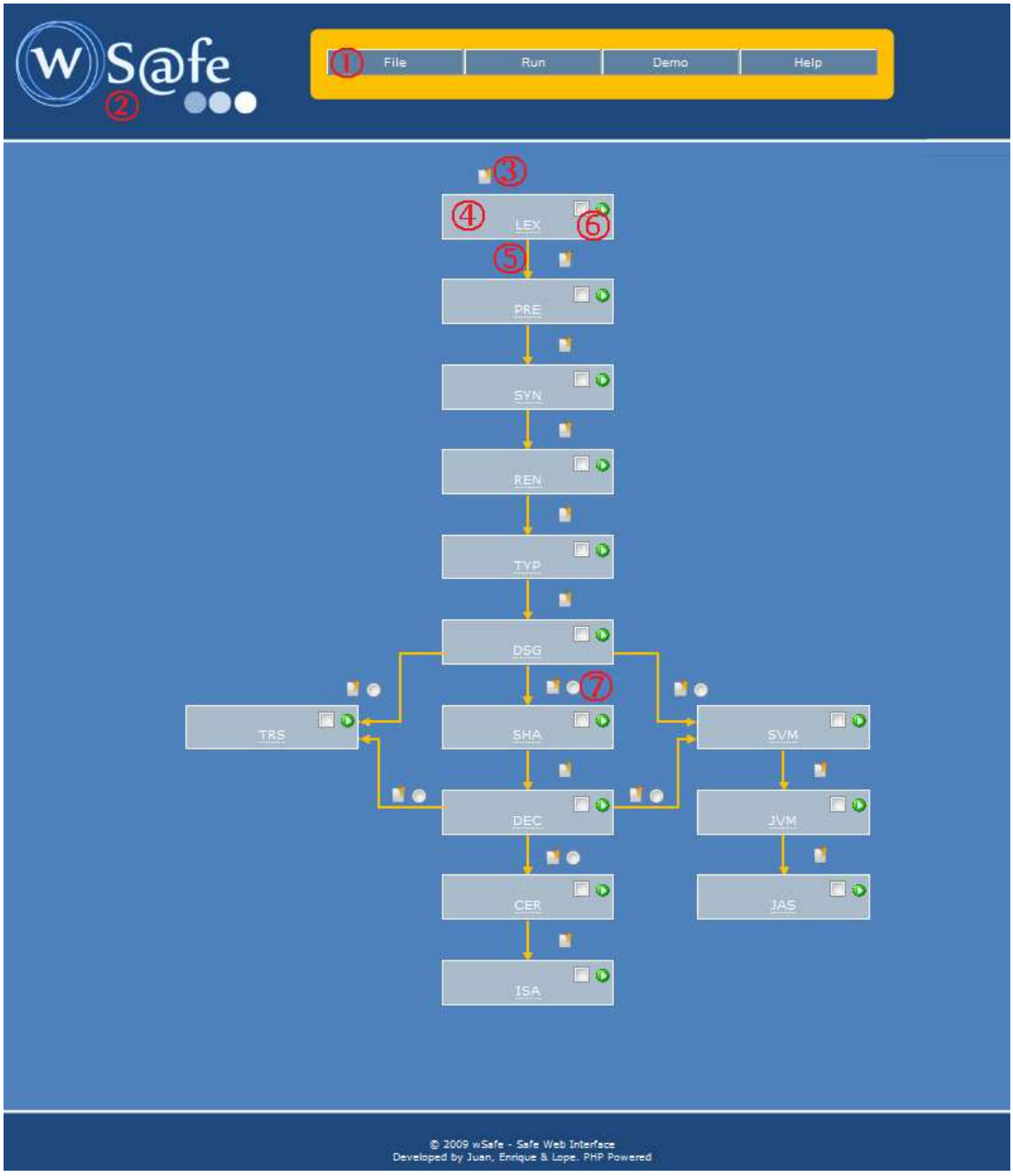
Una vez hemos introducido el link correspondiente a nuestra aplicación en el cliente (cualquier navegador), en este se recibe un **HTML**, que es creado en el servidor por el archivo **PHP** correspondiente, en nuestro caso, será el archivo "*index.php*" el encargado de crear el **HTML**. Además del **HTML** se incluye código **javascript** que será ejecutado en el cliente, y se encarga entre otras cosas de la correcta utilización de la aplicación, por ejemplo, en el caso de querer lanzar la ejecución de una ruta habiendo solo marcado el tick de inicio, nos indicará que es necesario marcar el tick de inicio y el de final. Cualquier acción que realicemos en la aplicación, mandará al servidor un formulario, que será atendido por un archivo **PHP** concreto (ejecución, reset…). En el caso de la ejecución, enviará al compilador SAFE el comando necesario para ejecutar la fase correspondiente (tomando los archivos necesarios), y este, devolverá un archivo **XML**. Este archivo **XML**, será transformado con la ayuda de plantillas **XSLT** en un **HTML**, que será el que se muestre en el cliente cuando sea solicitado.

Capítulo 5

El sistema wSafe

5.1. Diseño general de la interfaz gráfica

En la siguiente imagen podemos observar la pantalla principal de la aplicación wSafe.



En la imagen anterior encontramos numeradas las partes más importantes, su significado, es el siguiente:

1. Menú principal: Aquí encontraremos menús que afectan a la aplicación completa.
2. Logo de la aplicación.
3. Icono correspondiente al archivo inicial, generalmente será un archivo con extensión ".safe".
4. Fase. Una caja rectangular se corresponderá con cada una de las fases del compilador SAFE.
5. Arista. Las aristas son las encargadas de relacionar una fase con las antecesores y posteriores según el sentido de la flecha. Junto a dichas aristas habrá un icono que nos indica el estado actual del compilador SAFE en dicha arista.
6. Funciones asociadas a cada fase.
7. Bifurcación de caminos.

5.2. Funciones principales

Las funciones principales de la aplicación wSafe, son las asociadas con el menú principal. Entre ellas encontraremos:

Menú "File"

En el menú "File" nos encontramos con opciones generales de la aplicación.

Load inicial file: Al pinchar sobre esta opción, la aplicación nos abrirá el formulario para subir un archivo al servidor. Dicho archivo deberá ser un archivo con extensión ".safe".

Reset: Con esta opción borraremos todos los archivos generados y/o subidos al servidor y que estén asociados con nuestra sesión de usuario.

Menú "Run"

En este menú encontraremos las opciones referentes a diferentes tipos de ejecución de la aplicación.

Run All: realiza una ejecución completa de la aplicación tomando como entrada el archivo con extensión *".safe".* El recorrido que seguirá la aplicación será el que

marquen los *radibuttons* asociados a las bifurcaciones existentes. Si no existen errores en la ejecución, la última fase en ejecutarse, será la última fase del camino elegido en las bifurcaciones, devolviendo el archivo correspondiente a dicha fase

Run Route: ejecución de la ruta marcada mediante los *ticks* que el usuario haya seleccionado. Es imprescindible marcar únicamente dos fases. La primera de las fases marcadas, nos indicará la fase inicial, y la segunda fase marcada, la última fase. Además tendremos que seleccionar el camino que seguirá la ejecución mediante los *radibuttons* correspondientes. Tanto en el caso que no se marquen dos fases como inicio y fin de la ejecución, como en el caso que el camino seleccionado no se corresponda con el marcado por los *ticks*, la aplicación wSafe nos avisará con un mensaje. Como entrada se cogerá el archivo correspondiente a la fase de inicio, en caso de no exisitir, se lanzará un mensaje de error. Si no existen errores en la ejecución de las fases, el último archivo generado será el correspondiente a la fase marcada como fin de la ejecución.

Nota: El inicio de la ruta a ejecutar queda definido por la menor fase marcada, y el final de la ruta coincide con la mayor, esto es debido a la ordenación de las fases.

Menú "Demo"

En este menú nos encotraremos las opciones referentes a una demostración de ejecución del compilador wSafe.

Run demo: Permite ejecutar una demostración del compilador wSafe, con unos parámetros determinados. Toma como entrada un archivo que ya existe en el servidor, y ejecuta una ruta con comienzo y fin predeterminado. El archivo utilizado para realizar la demo puede visualizarse y descargarse al igual que cualquier archivo intermedio.

5.3. Diagrama de fases

El diagrama de fases de nuestra aplicación tiene una estructura de árbol tal y como se observa en la imagen anterior. Los nodos del árbol son las diferentes fases de las que se compone el compilador SAFE. Las aristas entre los nodos, nos dan información acerca de los archivos (generados o cargados), tanto de entrada como de salida de cada fase.

Cada fase, a excepción de la primera, tendrá una artista de entrada que contendrá el archivo que se tomará como entrada en dicha fase, y una arista que contendrá el archivo generado en la salida.

Para saber el estado de cada fase en un momento determinado de la ejecución, deberemos fijarnos en el estado de la arista de salida correspondiente, como veremos más adelante.

5.3.1. Opciones asociadas a las fases

Como hemos dicho anteriormente, una fase del compilador SAFE, se corresponde con una caja rectangular, como podemos observar en la siguiente imagen:

Dentro de dicha caja podemos encontrar los siguientes componentes:

Nombre: Estas tres letras 

se corresponden con la abreviatura del nombre de la fase, también se corresponden con la extensión del archivo generado por dicha fase (que aparecerá en la arista inmediatamente posterior)

Tick: Con este icono  seleccionaremos las fases que queremos como inicio y como fin de nuestra ejecución en el modo "*Run Route*". Puesto que en el archivo de configuración existe una ordenación de las fases, para la ejecución se tomará como inicio la fase de menor ordenación, y como fin, la de mayor.

Play: Este icono  permite ejecutar una fase de manera individual, tomando como entrada el archivo correspondiente a la arista de entrada. En caso de no existir un archivo de entrada, lanzará un mensaje de error. Una vez ejecutada la fase correctamente, se generará el archivo correspondiente en la arista de salida.

5.3.2. Opciones asociadas a las aristas

Una arista conecta dos fases del compilador, y una fase posee una artista de entrada y otra de salida (a excepción de la primera fase que no posee arista de entrada, y las ultimas fases, que no poseen aristas de salida). Al ejecutar una fase determinada, en cualquier modalidad de ejecución, el compilador tomará como entrada el archivo correspondiente a la arista de entrada a dicha fase, y generará el archivo de la arista de salida.

Junto a una arista, podremos encontrarnos tres tipos de iconos, que nos indican el estado en el que se encuentra la ejecución del compilador.

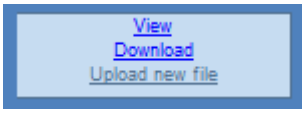
Carga: Este icono  nos indica que aún no se ha cargado un archivo en la arista, que servirá como entrada para la siguiente fase, por lo tanto, no podremos ejecutarla.

Correcto: Este icono  junto a la arista indica que el archivo asociado a dicha arista se ha cargado correctamente, dando por hecho que un archivo cargado desde una máquina local, es correcto sintácticamente. En el caso de que este icono se genere tras una ejecución de una fase determinada, nos indicará que el archivo correspondiente a la arista, generado por la fase inmediatamente anterior, se ha generado correctamente, es decir, la fase se ha ejecutado sin errores.

Error: Este icono  aparece junto a la arista asociada después de ejecutar una determinada fase, y se hayan producido errores en dicha ejecución.

Selección ruta: Este icono  lo encontramos en las bifurcaciones de las rutas, solo podremos seleccionar una de las opciones y nos indicará la ruta a seguir por la ejecución. Puede darse el caso que la ruta elegida sea incompatible con las fases seleccionadas como inicio y fin de la ejecución, en tal caso al intentar ejecutar, la aplicación wSafe, nos indicará el error.

Después de una ejecución, haciendo click sobre los iconos correspondientes a "*Correcto*" o "*Error*", podemos encontrar las siguientes opciones:



- **View**: nos permite ver el archivo generado. Este archivo será un archivo XML, que mediante la aplicación de plantillas XSLT, podemos verlo con formato (tabulaciones, colores…).
- **Download**: nos permite descargar el archivo a nuestra máquina local para modificarlo, guardarlo, etc…
- **Upload new file**: con esta opción podremos subir un nuevo archivo a la arista seleccionada. Este archivo reemplazará el ya existente para dicha arista.

Cabe destacar que la unión de las opciones Download/Upload permitiría a un usuario experimentado modificar los ficheros intermedios producidos en cada fase del compilador descargándolos a su máquina local y hacer que el compilador continuase compilando a partir de los ficheros modificados.

Capítulo 6

Manual de Usuario

6.1. Introducción

La página web permite la ejecución del compilador safe de cuatro formas diferentes:

- Ejecución tipo *Run All*
- Ejecución tipo *Run Route.*
- Ejecución individual de una fase.
- Ejecución de la demo.

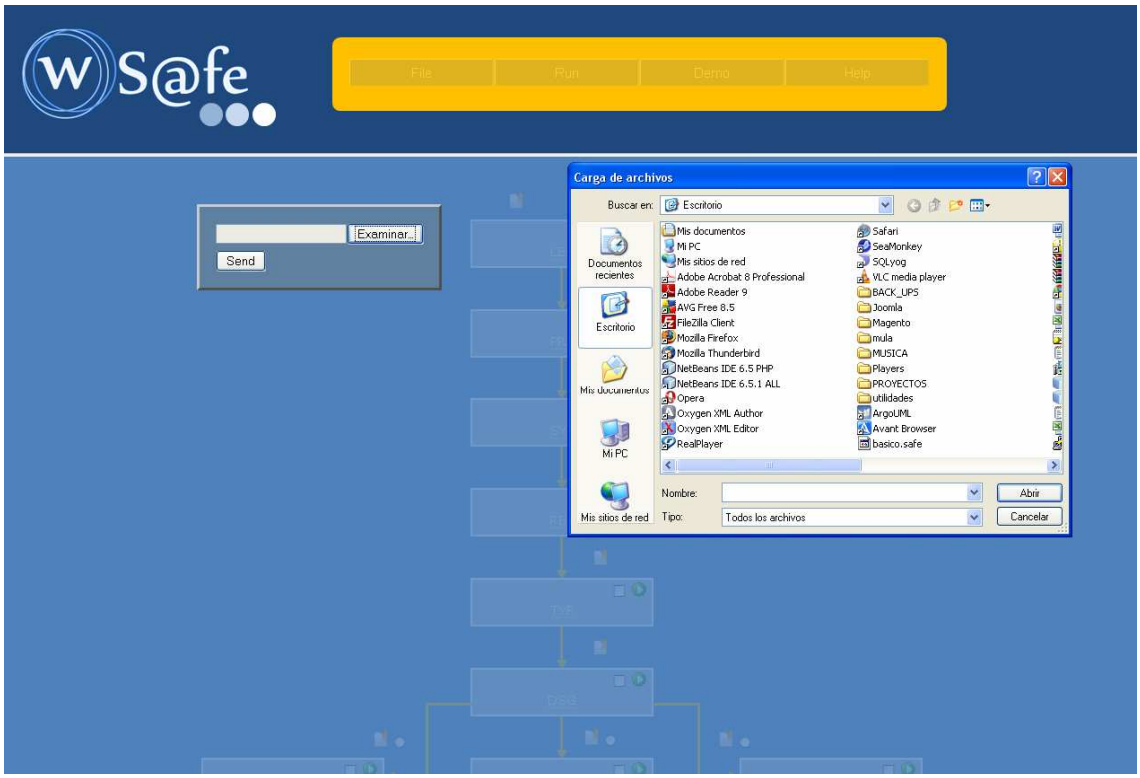
Antes de explicar detalladamente como funciona cada una, se indicará como se hace la carga de ficheros, para su posterior uso en la ejecución que se requiera.

6.2. Carga de ficheros

Cada fase cuenta sobre el rectángulo que la demarca con un icono, que dado el estado en el que se encuentre tendrá una imagen u otra.

En el caso de que todavía no se haya ejecutado la aplicación, ni que se hayan cargados ficheros, para cargar un fichero como entrada en la fase que se quiera ejecutar, se tendrá que pinchar sobre este icono  .

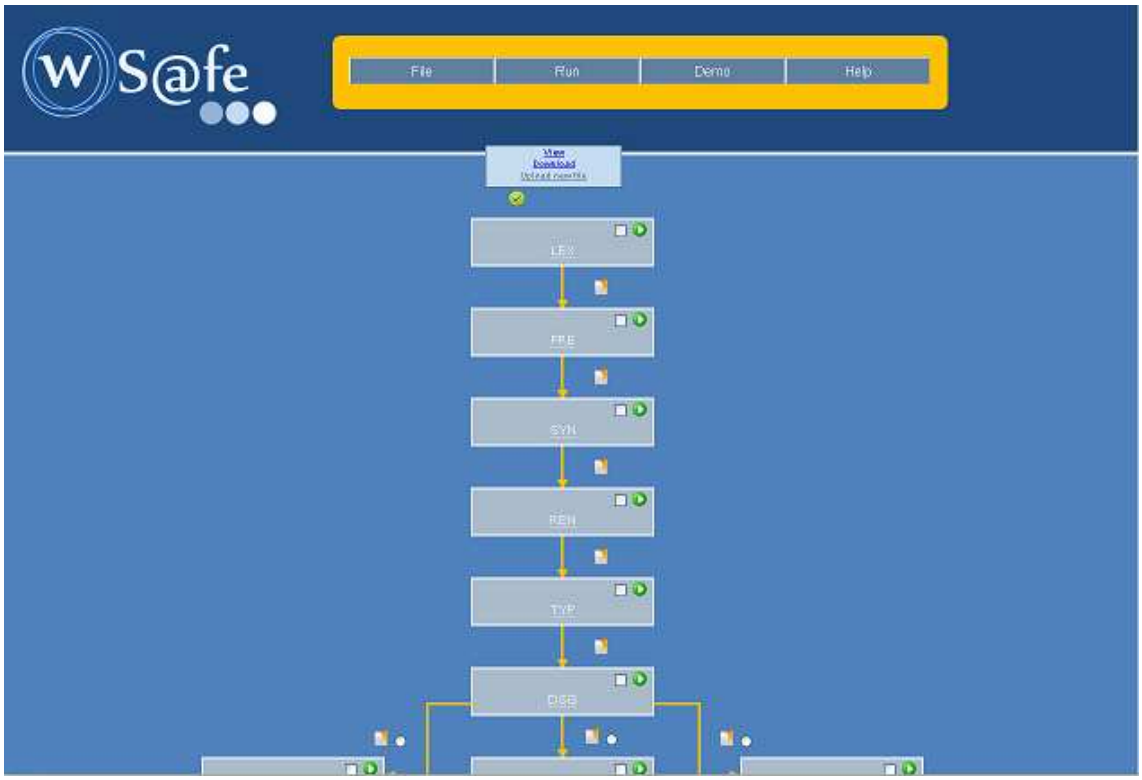
Una vez ocurrido esto, se mostrará un cuadro de diálogo que nos preguntará la ruta del archivo que se desea cargar, ubicado en la máquina local (cliente). Ahora solo debemos seleccionar el archivo y pulsar en *send*. El fichero deberá de tener como extensión las tres letras significativas (cuatro en el caso de cargar el fichero inicial) de la fase en la que lo queremos carga, de lo contrario dará un error. Es decir, en la fase primera, LEX, la entrada será un fichero con extensión .safe, en la fase PRE será .lex…



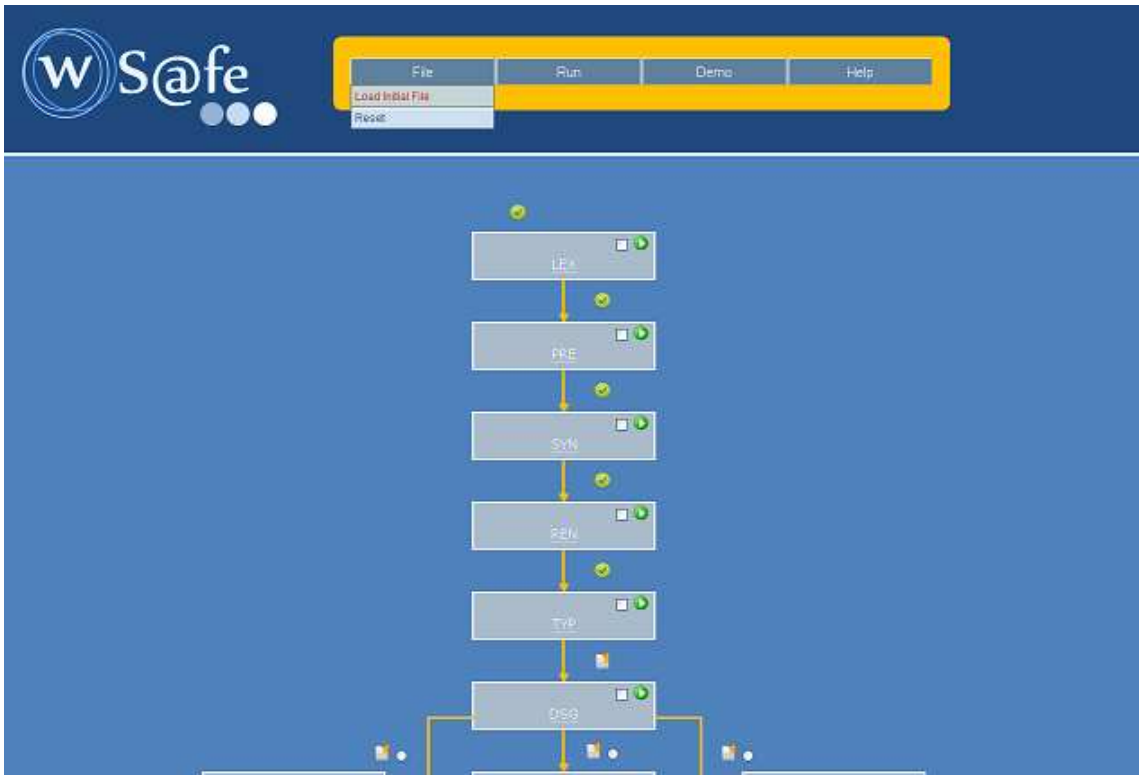
Cuando se carga un fichero y siempre que no se haya producido ningún error, la imagen cambiará a un círculo verde con aspa. Con el icono verde, que se corresponde con la correcta ejecución de una fase, también se podrá cargar un fichero.

Para ello se pulsa en la imagen, y una vez desplegado el menú, hay que hacer clic en *Upload new file*. Saldrá el cuadro de diálogo para la subida del fichero.

Cuando una fase se ha ejecutado y se ha producido un error, el icono cambiará a un círculo de color rojo. Al igual que con el circulo de color verde, pinchando sobre él, y después sobre *Upload new file* se podrá cargar un fichero con el procedimiento habitual.



Hay un caso especial para la carga de un fichero, se trata del fichero inicial con extensión .safe, que además de poder cargarse con los métodos explicados anteriormente, se podrá hacer yendo al menú *File*, seleccionar Load Initial File.



6.3. Ejecución del compilador

6.3.1. Run All.

En esta ejecución, el fichero inicial debe estar cargado. El siguiente paso será seleccionar los *radiobuttons* deseados, según sea la ruta que se quiera realizar.

Es necesario seleccionar un *radiobutton* de todos los posibles caminos que haya, aunque, pueda darse el caso de que un camino marcado impida que el siguiente se pueda realizar.

El siguiente paso será seleccionar la opción Run All del menú superior Run. En ese momento se ejecutarán las fases una a una hasta el final del camino marcado, siempre y cuando, no se produzca un error en la ejecución de alguna fase, en cuyo caso, la aplicación se parará y mostrará el icono rojo de error de ejecución. Un fichero erróneo también podrá visualizarse del mismo modo que uno que se haya generado correctamente, sin embargo, dicho fichero se corresponderá, normalmente, con el mensaje de error generado por el compilador explicando el problema encontrado.

6.3.2. Run Route.

Esta ejecución puede partir de cualquier fase (siempre que no sea terminal). Para ello, habrá que seleccionar la fase de inicio y de fin mediante los ticks que se encuentran en parte superior derecha de cada fase.

Además, como es lógico, el icono previo a la fase de inicio debe estar marcado en color verde, lo que indicará que tiene un archivo de entrada correcto para la posterior ejecución.

Se pulsa en el menú superior Run Route del menú superior Run para que comience.

Como sucedía en la ejecución Run All, si no hay ningún problema los iconos serán verdes y llegarán hasta el final de la ejecución, en caso contrario se detendrá y se pondrá icono rojo en la fase que dio el problema.

6.3.3. Ejecución individual de fase.

En esta fase se ejecutará individualmente una fase. Como requisito, la fase debe tener un fichero cargado correctamente. Como resultado, el icono de resultado siguiente estará en color verde si la ejecución ha sido satisfactoria, y rojo se ha habido algún error en la ejecución de dicha fase.

Para llevar a cabo la ejecución bastara con pulsar el botón play de la fase en cuestión.

6.3.4. Ejecución de la demo.

Para que el usuario pueda observar el funcionamiento de la aplicación, mirar y descargar ficheros de ejemplo que hayan sido ejecutados.

Para iniciar la demo, ir al menú superior Demo, y pulsar Run Demo. El archivo .safe utilizado para la demo puede visualizarse y descargarse a la máquina local pulsando sobre el icono situado sobre la primera fase y eligiendo la opción deseada (visualizar o descargar).

6.4. Ver / Descargar ficheros

Ver y descargar los ficheros son dos de las funcionalidades principales de la aplicación, que permite la interactuar con el usuario.

Tanto ver (View), como descargar (Download), se encuentran disponibles cuando se realiza algún tipo de ejecución, tanto si ha sido producida con error o sin error.

Asi pues habrá que pulsar como el icono verde o rojo y pinchar en la opción preferida.

View permite ver el fichero xml generado en un formato amigable para el usuario.

Download iniciará la descarga del fichero xml, por si el usuario desea editarlo para posteriormente subirlo, o simplemente porque lo quiere tener en su disco duro. La extensión de este fichero, se corresponderá con las tres letras significativas de la fase en la cual haya sido generado, y aunque la extensión no sea xml, el formato del fichero sí que se trata del estándar.

6.5. Reset

Esta función situada en el menú superior File, y a la que se accede pulsando en Reset, sirve para inicializar a cero la aplicación, borrando los archivos subidos o generados anteriormente.

6.6. Un ejemplo de ejecución

A continuación se muestra la ejecución completa de un ejemplo sencillo.

Primeramente se carga el archivo mediante la opción Load Initial File. El archivo utilizado para este ejemplo ha sido el siguiente:

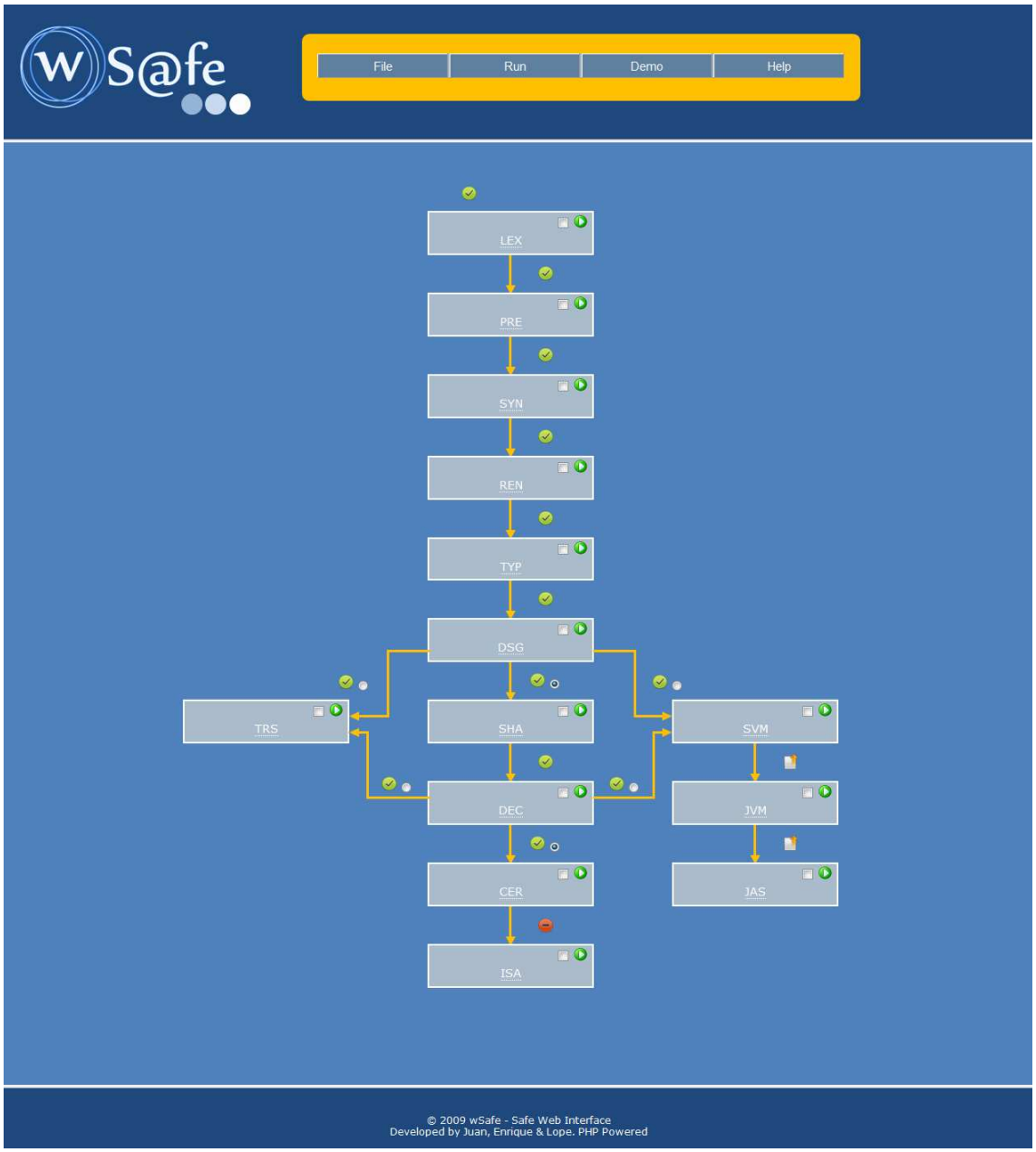
```
{-
   concat.safe
   ===========

   Función de concatenación de listas
-}

concat [] ys = ys
concat (x:xs) ys = x:concat xs ys

2+4
```

Tras ejecutar la opción Run All, el sistema procederá a ejecutar todas las fases del compilador. Al terminar la compilación del archivo, el sistema se encontrará en el siguiente estado:



Haciendo click en los iconos verdes generados por cada fase podremos ver el resultado generado por el compilador en dicha fase.

Si visualizamos el resultado obtenido por la fase LEX (Análisis Léxico), obtendremos la siguiente pantalla:

```
BraIzq (lin:9)(col:1)
Data (lin:9) (col:1)
ConstrTok (Entero) (lin:9) (col:6)
Igual (lin:9) (col:13)
ConstrTok (Ent) (lin:9) (col:15)
ConstrTok (Int) (lin:9) (col:19)
SemiColon (lin:9) (col:19)
Id (fib) (lin:14) (col:1)
ConstNum (0) (lin:14) (col:5)
Igual (lin:14) (col:7)
ConstNum (1) (lin:14) (col:9)
SemiColon (lin:14) (col:9)
Id (fib) (lin:15) (col:1)
ConstNum (1) (lin:15) (col:5)
Igual (lin:15) (col:7)
ConstNum (1) (lin:15) (col:9)
SemiColon (lin:15) (col:9)
Id (fib) (lin:16) (col:1)
Id (n) (lin:16) (col:5)
Igual (lin:16) (col:7)
Id (fib) (lin:16) (col:9)
ParIzq (lin:16) (col:13)
Id (n) (lin:16) (col:14)
Op (-) (lin:16) (col:15)
ConstNum (1) (lin:16) (col:16)
ParDer (lin:16) (col:17)
Op (+) (lin:16) (col:19)
Id (fib) (lin:16) (col:21)
ParIzq (lin:16) (col:25)
Id (n) (lin:16) (col:26)
Op (-) (lin:16) (col:27)
ConstNum (2) (lin:16) (col:28)
```

El resultado es una lista de los tokens producidos por la fase de Análisis Léxico

Si visualizamos el resultado obtenido por la fase SYN (Análisis Sintáctico), obtendremos la siguiente pantalla:

```
{ concat }
concat [] False ys False = ys
concat : x xs False ys False =: x concat xs ys @ _r_
{ concat2 }
concat2 xs False ys False = { }
    case xs of
        [] --> ys
        : z zs -->: z concat2 zs ys @ _r_

{ concatC }
concatC [] False ys False = ys @ _r_
concatC : x xs False ys False =: x concatC xs ys @ _r_
{ insertT }
insertT x False Empty False = Node Empty @ _r_ x Empty @ _r_ @ _r_
insertT x False Node iz y dr False =
    | == x y = Node iz y dr @ _r_
    | < x y = Node insertT x iz y dr @ _r_
    | > x y = Node iz y insertT x dr @ _r_

{ mkTree }
mkTree [] False = Empty @ _r_
mkTree : x xs False = insertT x mkTree xs
{ fall }
fall 0 False = [] @ _r_
fall n False =: Ent n @ _r_ fall - n 1 @ _r_
{ fallE }
fallE Ent 0 False = [] @ _r_
fallE x False =: x fallE Ent - n 1 @ _r_ @ _r_
    where
        Ent n False = x

{ inorder }
inorder Empty False = [] @ _r_
inorder Node iz x dr False = concat inorder iz concat : x [] @ _r_ @ _r_ inorder dr
{ treesort }
treesort xs False = inorder mkTree xs
```

Si visualizamos el resultado obtenido por la fase TYP (Inferencia de Tipos), obtendremos la siguiente pantalla:

```
{ concat :: ListaT a @ rho1 --> ListaT a @ rho2 --> rho2 --> ListaT a @ rho2 }
concat [] ListaT a @ rho1 False 14_ys { ListaT a @ rho2 } False = 1_r 14_ys { ListaT a @ rho2 }
concat : 15_x { a } 16_xs { ListaT a @ rho1 } ListaT a @ rho1 False 17_ys { ListaT a @ rho2 } False = 1_r : 15_x { a } concat 16_xs { ListaT a @ rho1 } 17_ys {
ListaT a @ rho2 } @ 1_r ListaT a @ rho2 @ 1_r
{ concat2 :: ListaT a @ rho1 --> ListaT a @ rho2 --> rho2 --> ListaT a @ rho2 }
concat2 18_xs { ListaT a @ rho1 } False 19_ys { ListaT a @ rho2 } False = 1_r { ListaT a @ rho2 }
    case 18_xs { ListaT a @ rho1 } of
        [] ListaT a @ rho1 --> 19_ys { ListaT a @ rho2 }
        : 20_z { a } 21_zs { ListaT a @ rho1 } ListaT a @ rho1 --> : 20_z { a } concat2 21_zs { ListaT a @ rho1 } 19_ys { ListaT a @ rho2 } @ 1_r ListaT a @
        rho2 @ 1_r

{ concatC :: ListaT a @ rho1 --> ListaT a @ rho2 --> rho3 --> ListaT a @ rho3 }
concatC [] ListaT a @ rho1 False 22_ys { ListaT a @ rho2 } False = 1_r 22_ys @ 1_r

concatC : 23_x { a } 24_xs { ListaT a @ rho1 } ListaT a @ rho1 False 25_ys { ListaT a @ rho2 } False = 1_r : 23_x { a } concatC 24_xs { ListaT a @ rho1 } 25_ys
{ ListaT a @ rho2 } @ 1_r ListaT a @ rho3 @ 1_r
{ insertT :: Int --> BST Int @ rho1 --> rho1 --> BST Int @ rho1 }
insertT 26_x { Int } False Empty BST Int @ rho1 False = 1_r Node Empty @ 1_r 26_x { Int } Empty @ 1_r @ 1_r
insertT 27_x { Int } False Node 28_iz { BST Int @ rho1 } 29_y { Int } 30_dr { BST Int @ rho1 } BST Int @ rho1 False = 1_r
    | == 27_x { Int } 29_y { Int } = Node 28_iz { BST Int @ rho1 } 29_y { Int } 30_dr { BST Int @ rho1 } 1_r BST Int @ rho1
    | < 27_x { Int } 29_y { Int } = Node insertT 27_x { Int } 28_iz { BST Int @ rho1 } @ 1_r BST Int @ rho1 29_y { Int } 30_dr { BST Int @ rho1 } 1_r BST Int
    @ rho1
    | > 27_x { Int } 29_y { Int } = Node 28_iz { BST Int @ rho1 } 29_y { Int } insertT 27_x { Int } 30_dr { BST Int @ rho1 } @ 1_r BST Int @ rho1 1_r BST Int
    @ rho1

{ mkTree :: ListaT Int @ rho1 --> rho2 --> BST Int @ rho2 }
mkTree [] ListaT Int @ rho1 False = 1_r Empty @ 1_r
mkTree : 31_x { Int } 32_xs { ListaT Int @ rho1 } ListaT Int @ rho1 False = 1_r insertT 31_x { Int } mkTree 32_xs { ListaT Int @ rho1 } @ 1_r BST Int @ rho2 @
1_r BST Int @ rho2
{ fall :: Int --> rho1 --> rho2 --> ListaT Entero @ rho1 @ rho2 }
fall 0 False = 1_r 2_r [] @ 2_r
fall 33_n { Int } False = 1_r 2_r : Ent 33_n { Int } @ 1_r fall - 33_n { Int } 1 { Int } Int @ 1_r 2_r ListaT Entero @ rho1 @ rho2 @ 2_r
{ fallE :: Entero @ rho1 --> rho1 --> rho2 --> ListaT Entero @ rho1 @ rho2 }
fallE Ent 0 Entero @ rho1 False = 1_r 2_r [] @ 2_r
fallE 34_x { Entero @ rho1 } False = 1_r 2_r : 34_x { Entero @ rho1 } fallE Ent - 35_n { Int } 1 { Int } Int @ 1_r @ 1_r 2_r ListaT Entero @ rho1 @ rho2 @ 2_r
```

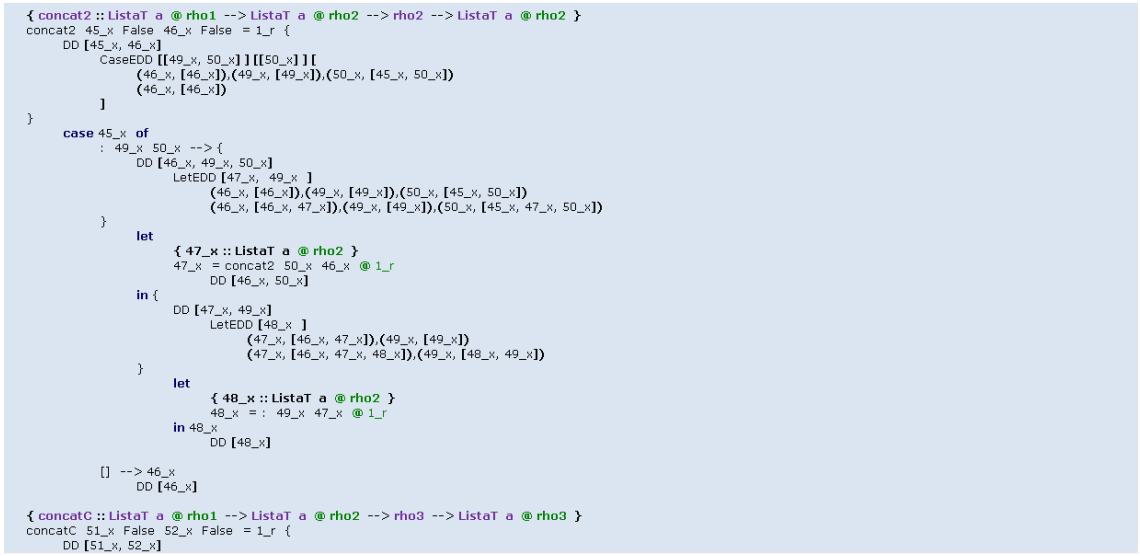
Si visualizamos el resultado obtenido por la fase SHA (Análisis de Compartición), obtendremos la siguiente pantalla:

```
{ concat :: ListaT a @ rho1 --> ListaT a @ rho2 --> rho2 --> ListaT a @ rho2 }
concat 39_x ListaT a @ rho1 False 40_x ListaT a @ rho2 False = 1_r { ListaT a @ rho2 ,
    ShDecCase [
        (40_x, [40_x]),(43_x, [43_x]),(44_x, [39_x, 44_x])
        (40_x, [40_x])
    ]
}
    case 39_x ListaT a @ rho1 of
        : 43_x a 44_x ListaT a @ rho1 ListaT a @ rho1 --> { ListaT a @ rho2 ,
            ShDecLet
                (40_x, [40_x]),(43_x, [43_x]),(44_x, [39_x, 44_x])
                (40_x, [40_x, 41_x]),(43_x, [43_x]),(44_x, [39_x, 41_x, 44_x])
            }
                let
                    { 41_x :: ListaT a @ rho2 }
                    41_x = concat 44_x ListaT a @ rho1 40_x ListaT a @ rho2 @ 1_r ListaT a @ rho2
                        ShDecApp [[39_x] ]
                in { ListaT a @ rho2 ,
                    ShDecLet
                        (41_x, [40_x, 41_x]),(43_x, [43_x])
                        (41_x, [40_x, 41_x, 42_x]),(43_x, [42_x, 43_x])
                    }
                        let
                            { 42_x :: ListaT a @ rho2 }
                            42_x = : 43_x a 41_x ListaT a @ rho2 @ 1_r
                        in 42_x ListaT a @ rho2

        [] ListaT a @ rho1 --> 40_x ListaT a @ rho2

{ concat2 :: ListaT a @ rho1 --> ListaT a @ rho2 --> rho2 --> ListaT a @ rho2 }
concat2 45_x ListaT a @ rho1 False 46_x ListaT a @ rho2 False = 1_r { ListaT a @ rho2 ,
    ShDecCase [
        (46_x, [46_x]),(49_x, [49_x]),(50_x, [45_x, 50_x])
        (46_x, [46_x])
    ]
}
```

Si visualizamos el resultado obtenido por la fase DEC (Análisis de Destrucción), obtendremos la siguiente pantalla:



Además, se observa que la fase CER (Generación de Certificados) ha dado un error y la ejecución se ha detenido en ese punto.

Capítulo 7

Conclusiones

El objetivo fundamental de este proyecto era desarrollar una herramienta web para manejar el compilador de *SAFE* a través de una interfaz gráfica amigable, intuitiva y sobre todo ampliable a futuras mejoras.

Durante el desarrollo del mismo, se han empelado las siguientes herramientas y tecnologías:

- Las herramientas **WinSCP** y **Tortoise SVN** para el acceso al servidor donde se aloja la aplicación.
- Las tecnologías web **HTML**, **CSS**, **PHP**, **JavaScript**, **XML** y **XLST** (comentadas en capítulos anteriores) para la implementación del código de la aplicación.
- Las herramientas **Microsoft Word 2007** y **PDFCreator** para la realización de la presente memoria.

El resultado de la implementación, que ha sido organizada en diferentes módulos, ha sido el siguiente:

- El módulo **index.php**, que contiene la página principal de la aplicación y que es el encargado de llamar al resto de módulos.
- El módulo **Stages.php**, que se encarga de dibujar en la página principal las fases que componen compilador.
- El módulo **Arcs.php**, cuya función es la de dibujar las aristas que unen las fases del compilador.
- El módulo **execute.php**, que contiene todo lo relacionado con la ejecución del compilador.
- Los módulos **upload_files.php** y **download_file.php**, que permiten la carga y descarga de los archivos que usa el compilador.
- El módulo **reset.php** para reiniciar el sistema.
- El módulo **errors.php** para la gestión de errores.
- El módulo **safe.js**, que comprueba que ha sido seleccionado un *radiobutton* de los posibles grupos existentes

- El módulo **menu.js**, encargado de implementar el script para el menú principal.

- El módulo **wz_jsgraphics.js** que dibuja los iconos de las fases.

Además, el grupo que conforma el presente trabajo ha adquirido conocimientos avanzados de programación web a lo largo del desarrollo del mismo. Nuestros conocimientos previos de tecnologías como PHP, HTML o JavaScript se han visto potencialmente incrementados tras la terminación del proyecto. Un caso muy concreto ha sido el de la tecnología XSL, de la cual apenas habíamos oído hablar al comienzo del proyecto, pero que, tras la finalización del mismo, podríamos decir que dominamos dicha tecnología con cierta soltura.

A parte de los conocimientos adquiridos en las tecnologías web más importantes, también hemos reforzado nuestras ideas iniciales sobre los lenguajes funcionales, que al comienzo del trabajo eran un poco escasas. Finalmente, cabe destacar que hemos aprendido cómo funciona un compilador para un lenguaje funcional concreto y las fases que lo componen, lo cual nos ha servido de ayuda incluso en otras asignaturas.

Como trabajo futuro se propone extender la aplicación web con nuevas fases del compilador o nuevas funciones.

En particular hay que conectar el compilador con otras herramientas, algunas de las cuales fueron mencionadas en la sección 3.1.:

- Isabelle / HOL: Asistente de demostraciones que procesa los certificados emitidos por el compilador

- MU-TERM: Herramienta de reescritura que procesa los sistemas de reescritura emitidos por el compilador y genera demostraciones de terminación.

- Jasmin: Procesador que genera ficheros .class para la Maquina Virtual de Java a partir de ficheros de texto con *bytecodes* emitidos por el compilador.

Bibliografía

Bibliografía relacionada con Safe

M. Montenegro, R. Peña, C. Segura.
Título: “A Type System for Safe Memory Management and its Proof of Correctness”.
Título del libro: “ACM Principles and Practice of Declarative Programming”.
PPDP'08, Valencia, España, Julio 2008.
Páginas: 152 – 162.
Fecha: 2008.

J. de Dios, R. Peña.
Título: “Formal Certification of a Resource-Aware Language Implementation”.
Título del libro: “Int. Conf. on Theorem Proving in Higher Order Logics, TPHOL'09, Munich (Alemania)”.
Publicación: “LNCS 5674, Springer”.
Páginas: 1 – 15.
Fecha: Agosto 2008.

M. Montenegro, R. Peña, C. Segura.
Título: “Experiences in Developing a Compiler for Safe using Haskell”.
Título del libro: “Proceedings of Taller de Programación Funcional, TPF'09, associated to PROLE'09, San Sebastián (España)”.
PPDP'08, Valencia, España, Julio 2008.
Páginas: 1 – 8.
Fecha: Septiembre 2009.

J. Conesa, R. López, A. Lozano.
Título: “Desarrollo de un compilador para un lenguaje funcional con gestión explícita de la memoria”.
Proyecto de Sistemas Informáticos 2005-2006. Facultadad de Informática de la UCM.
Fecha: Julio 2006.

M. Montenegro.
Título: “Inferencias de tipos seguros en un lenguaje funcional con destrucción explícita de memoria”.
Proyecto Fin de Máster. Máster de Investigación en Informática. Universidad Complutense de Madrid.
Fecha: Septiembre 2007.

Bibliografía relacionada con la elaboración de la aplicación web.

http://www.php.net/

http://www.w3.org/

http://www.w3schools.com/

http://www.w3schools.com/JS/

http://www.w3schools.com/xsl/

http://www.w3schools.com/css/

http://www.w3schools.com/xml/

http://www.w3schools.com/PHP/

http://www.wikipedia.org/

http://www.walterzorn.com/jsgraphics/jsgraphics_e.htm

Anexo A

Archivos php

```php
<?php
include_once('Stage.php');

class Arc{

    private $src;

    private $dest;

    private $state;

    private $has_brother;

    /**
     *
     * @param <type> $src
     * @param <type> $dest
     * @param <type> $state "0" if the stage is waiting for loading a file, "1"
     * generated o loaded file, "2" correct exe, "3" wrong exe
     */
    function __construct($src, $dest, $state, $has_brother){
        $this->src = &$src;
        $this->dest = &$dest;
        $this->state = $state;
        $this->has_brother = $has_brother;
    }

    public function getSrc(){
        return $this->src;
    }

    public function getDest(){
        return $this->dest;
    }

    public function setState($state){
        $this->state = $state;
    }

    public function setStateWaitingFile(){
        $this->state = 0;
    }

    public function setStateHasFile(){
        $this->state = 1;
    }

    public function setStateExeOK(){
        $this->state = 2;
    }

    public function setStateExeError(){
        $this->state = 3;
    }

    public function getState(){
        return $this->state;
    }

    public function render(){?>

<script type="text/javascript">
    var jg = new jsGraphics("stage"+"<?php echo $this->src->getDescrip();?>");
        <?php
        if($this->src->getDescrip() == "safe"){
            $height_difference = $this->src->getPositionY() - $this->src->getPositionY();
            $width_difference = $this->src->getPositionX() - $this->src->getPositionX();
        }
        else{
```

```
        $height_difference = $this->src->getPositionY() - $this->dest->getPositionY();
        $width_difference = $this->src->getPositionX() - $this->dest->getPositionX();
    }
    //Father & son are in the same X axis.
    //A) Father immediately over the son
    if(($width_difference == 0) && ($height_difference == -100)){
        ?>
            jg.setStroke(3);
            jg.setColor("#FFC000");
            var x1 = 100 ;
            var y1 = 85;
            var x2 = 100;
            var y2 = 130;
            jg.drawLine(x1,y1,x2,y2);
            jg.drawImage('./img/arrow_black_down.png', x2-4, y2-8, 12, 11, 2);
            jg.drawExeIcons('<?php echo $this->src->getDescrip(); ?>',
                '<?php echo $this->dest->getDescrip(); ?>',
                '<?php echo $this->has_brother; ?>',
                <?php echo $this->state; ?>, 110, 98);
            jg.paint();
        <?php }
    //B) Father over the son, but not immediately
    elseif(($width_difference == 0) && ($height_difference < 0)){
        ?>
            var height_difference = <?php echo $height_difference; ?>;
            jg.setStroke(3);
            jg.setColor("#FFC000");
            var x1 = 0 ;
            var y1 = 70;
            var x2 = -25;
            var y2 = 70;
            jg.drawLine(x1,y1,x2,y2);
            jg.paint();
            var x1 = -25;
            var y1 = 70;
            var x2 = -25;
            var y2 = 50-height_difference;
            jg.drawLine(x1,y1,x2,y2);
            jg.paint();
            var x1 = -25 ;
            var y1 = 50-height_difference;
            var x2 = 0;
            var y2 = 50-height_difference;
            jg.drawLine(x1,y1,x2,y2);
            jg.drawImage('./img/arrow_black_right.png', x2-10, y2-4, 12, 11, 2);
            jg.drawExeIcons('<?php echo $this->src->getDescrip(); ?>',
                '<?php echo $this->dest->getDescrip(); ?>',
                '<?php echo $this->has_brother; ?>',
                <?php echo $this->state; ?>, -100, -height_difference);
            jg.paint();
        <?php }
    //C) Son over the father, but not immediately
    elseif(($width_difference == 0) && ($height_difference > 0)){
        ?>
            var height_difference = <?php echo $height_difference; ?>;
            jg.setStroke(3);
            jg.setColor("#FFC000");
            var x1 = 203 ;
            var y1 = 50;
            var x2 = 228;
            var y2 = 50;
            jg.drawLine(x1,y1,x2,y2);
            jg.paint();
            var x1 = 228 ;
            var y1 = 50;
            var x2 = 228;
            var y2 = 70-height_difference;
            jg.drawLine(x1,y1,x2,y2);
            jg.paint();
```

```
                var x1 = 228;
                var y1 = 70-height_difference;
                var x2 = 203;
                var y2 = 70-height_difference;
                jg.drawLine(x1,y1,x2,y2);
                jg.drawImage('./img/arrow_black_left.png', x2, y2-4, 12, 11, 2);
                jg.drawExeIcons('<?php echo $this->src->getDescrip(); ?>',
                    '<?php echo $this->dest->getDescrip(); ?>',
                    '<?php echo $this->has_brother; ?>',
                    <?php echo $this->state; ?>, 235, -height_difference+100);
                jg.paint();
            <?php }
        //D) The son immediately on the left of the father
        elseif(($width_difference == 300) && ($height_difference == 0)){
            ?>
                jg.setStroke(3);
                jg.setColor("#FFC000");
                var x1 = 0 ;
                var y1 = 60;
                var x2 = -100;
                var y2 = 60;
                jg.drawLine(x1,y1,x2,y2);
                jg.drawImage('./img/arrow_black_left.png', x2+3, y2-4, 12, 11, 2);
                jg.drawExeIcons('<?php echo $this->src->getDescrip(); ?>',
                    '<?php echo $this->dest->getDescrip(); ?>',
                    '<?php echo $this->has_brother; ?>',
                    <?php echo $this->state; ?>, -75, 35);
                jg.paint();
            <?php }
        //elseif(trim($this->dest->getAlin()) == "left" && trim($this->src->getAlin()) ==
"none"){
        //E) The son on the left and below of the father
        elseif(($width_difference == 300) && ($height_difference < 0)){
            ?>
                var height_difference = <?php echo $height_difference; ?>;
                jg.setStroke(3);
                jg.setColor("#FFC000");
                var x1 = 0 ;
                var y1 = 70;
                var x2 = -50;
                var y2 = 70;
                jg.drawLine(x1,y1,x2,y2);
                jg.paint();
                var x1 = -50 ;
                var y1 = 70;
                var x2 = -50;
                var y2 = 50-height_difference;
                jg.drawLine(x1,y1,x2,y2);
                jg.paint();
                var x1 = -50 ;
                var y1 = 50-height_difference;
                var x2 = -97;
                var y2 = 50-height_difference;
                jg.drawLine(x1,y1,x2,y2);
                jg.drawImage('./img/arrow_black_left.png', x2+1, y2-4, 12, 11, 2);
                jg.drawExeIcons('<?php echo $this->src->getDescrip(); ?>',
                    '<?php echo $this->dest->getDescrip(); ?>',
                    '<?php echo $this->has_brother; ?>',
                    <?php echo $this->state; ?>, -125, -height_difference);
                jg.paint();
            <?php }
        //F) The son on the left and upper of the father
        elseif(($width_difference == 300) && ($height_difference > 0)){
            ?>
                var height_difference = <?php echo $height_difference; ?>;
                jg.setStroke(3);
                jg.setColor("#FFC000");
                var x1 = 0 ;
                var y1 = 50;
```

```
                var x2 = -75;
                var y2 = 50;
                jg.drawLine(x1,y1,x2,y2);
                jg.paint();
                var x1 = -75 ;
                var y1 = 50;
                var x2 = -75;
                var y2 = 70-height_difference;
                jg.drawLine(x1,y1,x2,y2);
                jg.paint();
                var x1 = -75 ;
                var y1 = 70-height_difference;
                var x2 = -97;
                var y2 = 70-height_difference;
                jg.drawLine(x1,y1,x2,y2);
                jg.drawImage('./img/arrow_black_left.png', x2+1, y2-4, 12, 11, 2);
                jg.drawExeIcons('<?php echo $this->src->getDescrip(); ?>',
                    '<?php echo $this->dest->getDescrip(); ?>',
                    '<?php echo $this->has_brother; ?>',
                    <?php echo $this->state; ?>, -72, 25);
                jg.paint();
            <?php }
        //G) The son immediately on the right of the father
        elseif(($width_difference == -300) && ($height_difference == 0)){
            ?>
                jg.setStroke(3);
                jg.setColor("#FFC000");
                var x1 = 203 ;
                var y1 = 60;
                var x2 = 297;
                var y2 = 60;
                jg.drawLine(x1,y1,x2,y2);
                jg.drawImage('./img/arrow_black_right.png', x2-6, y2-4, 12, 11, 2);
                jg.drawExeIcons('<?php echo $this->src->getDescrip(); ?>',
                    '<?php echo $this->dest->getDescrip(); ?>',
                    '<?php echo $this->has_brother; ?>',
                    <?php echo $this->state; ?>, 210, 65);
                jg.paint();
            <?php }
        //elseif(trim($this->dest->getAlin()) == "right" && trim($this->src->getAlin()) ==
"none"){
        //H) The son on the right and below of the father
        elseif(($width_difference == -300) && ($height_difference < 0)){
            ?>
                var height_difference = <?php echo $height_difference; ?>;
                jg.setStroke(3);
                jg.setColor("#FFC000");
                var x1 = 203 ;
                var y1 = 70;
                var x2 = 253;
                var y2 = 70;
                jg.drawLine(x1,y1,x2,y2);
                jg.paint();
                var x1 = 253 ;
                var y1 = 70;
                var x2 = 253;
                var y2 = 50-height_difference;
                jg.drawLine(x1,y1,x2,y2);
                jg.paint();
                var x1 = 253 ;
                var y1 = 50-height_difference;
                var x2 = 297;
                var y2 = 50-height_difference;
                jg.drawLine(x1,y1,x2,y2);
                jg.drawImage('./img/arrow_black_right.png', x2-9, y2-4, 12, 11, 2);
                jg.drawExeIcons('<?php echo $this->src->getDescrip(); ?>',
                    '<?php echo $this->dest->getDescrip(); ?>',
                    '<?php echo $this->has_brother; ?>',
                    <?php echo $this->state; ?>, 260, -height_difference);
```

```
                jg.paint();
            <?php }
        //H) The son on the right and below of the father
        elseif(($width_difference == -300) && ($height_difference > 0)){
            ?>
                var height_difference = <?php echo $height_difference; ?>;
                jg.setStroke(3);
                jg.setColor("#FFC000");
                var x1 = 203 ;
                var y1 = 50;
                var x2 = 276;
                var y2 = 50;
                jg.drawLine(x1,y1,x2,y2);
                jg.paint();
                var x1 = 276 ;
                var y1 = 50;
                var x2 = 276;
                var y2 = 70-height_difference;
                jg.drawLine(x1,y1,x2,y2);
                jg.paint();
                var x1 = 278 ;
                var y1 = 70-height_difference;
                var x2 = 297;
                var y2 = 70-height_difference;
                jg.drawLine(x1,y1,x2,y2);
                jg.drawImage('./img/arrow_black_right.png', x2-9, y2-4, 12, 11, 2);
                jg.drawExeIcons('<?php echo $this->src->getDescrip(); ?>',
                    '<?php echo $this->dest->getDescrip(); ?>',
                    '<?php echo $this->has_brother; ?>',
                    <?php echo $this->state; ?>, 207, 25);
                jg.paint();
            <?php }
        //I) The son and father are the same, so is a end stage
        elseif(($width_difference == 0) && ($height_difference == 0)){
        ?>
                jg.drawExeIcons('<?php echo $this->src->getDescrip(); ?>',
                    '<?php echo $this->dest->getDescrip(); ?>',
                    '<?php echo $this->has_brother; ?>',
                    <?php echo $this->state; ?>, 66, 100);
                jg.paint();
        <?
        }
        ?>
</script>
<?
}
}
?>
```

```php
<?php
include_once('Stage.php');
include_once('Arc.php');

class Arcs{
    private $collectionOfArcs;

    function __construct(){
        $this->collectionOfArcs = Array();
    }

    public function addArc($arc){
        $this->collectionOfArcs[] = $arc;
    }

    public function getArc($src, $dest){

        for($i = 0; $i < count($this->collectionOfArcs); $i++){
            $w = $this->collectionOfArcs[$i];
            if(($w->getSrc()->getId() == $src->getId()) &&
                ($w->getDest()->getId() == $dest->getId())){
                return $w;
            }
        }
        return false;
    }

    public function getArcDesc($src_desc, $dest_desc){
        for($i = 0; $i < count($this->collectionOfArcs); $i++){
            $w = $this->collectionOfArcs[$i];
            if(($w->getSrc()->getDescrip() == $src_desc) &&
                ($w->getDest()->getDescrip() == $dest_desc)){
                return $w;
            }
        }
        return false;
    }

    public function getArcIndex($i){

        if($i < count($this->collectionOfArcs)){
            return $this->collectionOfArcs[$i];
        }
        else{
            return false;
        }
    }

    public function getSize(){
        return count($this->collectionOfArcs);
    }

    /**
     *
     * @param <type> $currentStage
     * @param <type> $stages
     */
    public function render($currentStage, $stages){
        $nextStages = $currentStage->getNextStages();
        //si al menos un hijo dibujar los arcos
        if($nextStages[0] != "-"){
            for($i = 0; $i < count($nextStages);$i++){
                $currentSon = $stages->getStage($nextStages[$i]);
                $arc_aux = &$this->getArc($currentStage, $currentSon);
                $arc_aux->render();
            }
        }
        //for end stage
        else{
```

```
            if($currentStage->getDescrip() == "safe"){
                $arc_aux = &$this->getArc($currentStage, $stages->getStageIndex(0));
                $arc_aux->render();
            }
            else{
                $arc_aux = &$this->getArc($currentStage, $currentStage);
                $arc_aux->render();
            }
        }
    }
    /**
     *
     * @param <type> $desc
     * @param <type> $state
     */
    public function changeArcState($src_desc, $state){
        for($i = 0; $i < count($this->collectionOfArcs); $i++){
            $w = $this->collectionOfArcs[$i];
            if(($w->getSrc()->getDescrip() == $src_desc)){
                $w->setState($state);
            }
        }
    }
}
?>
```

```php
<?php
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 * Description of Constantes
 *
 * @author quique
 */

class Constantes {
    protected $numberOfStages = 15;

    /**
     * Controla el estado en espera de la fase.
     * @var <type>
     */
    protected $ESTADO_EN_ESPERA = 0;

    /**
     * Controla el estado de la fase en la que la ejecucion se realizado correctamente.
     * @var <type>
     */
    protected $ESTADO_EJECUCION_OK = 1;

    /**
     * Controla el estado de la fase en la que la ejecucion ha realizado un error.
     * @var <type>
     */
    protected $ESTADO_EJECUCCION_ERR = 2;

}
?>
```

```php
<?php

    session_start();

    require('errors.php');

    //Cogemos el nombre de la sesion
    $idSession = session_id();

    //Extensiones permitidas
    // Ojo con las mayusculas / minusculas,
    //aunque aqui de igual, en el servidor no es lo mismo...
    $extensions = array("safe","lex", "pre", "syn", "ren", "typ",
        "dsg", "sha", "dec", "trs", "svm", "jvm", "jas", "cer", "isa");
    $id = $_GET["id"];

    //Nombre y extension del archivo:
    $name = $idSession.".".$id;

    //Ruta completa al archivo
    $path = "temp/".$idSession."/".$name;


    if(strpos($id,"/")!==false){
    $err = getErr("You can not browse other directories");
        die($err);
    }

    $ftmp = explode(".",$id);
    $fExt = strtolower($ftmp[count($ftmp)-1]);


    if(!in_array($fExt,$extensions)){
    $err = getErr("Unable to download files with extension: ".$fExt);
    die($err);
    }
    if (!file_exists($path)){
    $err = getErr("No file");
    die($err);
    }

    header("Content-type: application/octet-stream");
    header("Content-Disposition: attachment; filename=\"$path\"\n");
    $fp=fopen("$path", "r");
    fpassthru($fp);
?>
```

```php
<?php
global $basic;
$basic = " <html>
<title>wSafe Project... Developement by Juan, Enrique & Lope</title>
    <head>
        <link type='text/css' href='css/safe.css' rel='stylesheet'>
    </head>
    <h1 id='err'> Error!! </h1>";

function getErr($str){
    $err = "
    <html>
        <title>wSafe Project... Developement by Juan, Enrique & Lope</title>
        <head>
            <link type='text/css' href='css/safe.css' rel='stylesheet'>
        </head>
        <body class='err_body'>
            <h1 class='err_'> Error!! </h1>
            <h2 class='err_str'>".$str."</h2>
            </br>
        <a href='./'> <img src='img/back.png' border='0'/> </a>
        </body>
    </html>";
return $err;
}

?>
```

```php
<?php
/*****************************************************
Recogemos los datos de la accion a realizar
*****************************************************/

    $action = $_POST["execute_action"];
    if ($action == "demo"){
        execDemo();
    }
    else if ($action == "route"){
        execRoute();
    }
    else if ($action == "all"){
        execAll();
    }
    else if ($action == "stage"){
        //Recogemos la fase
        $stage = $_POST["stage"];
        $res = execStage($stage, "stage");
        if($res == 1){
            $stagesInit->getArcs()->changeArcState($stage, 3);
        }
        else{
            $stagesInit->getArcs()->changeArcState($stage, 2);
        }


    }

/*****************************************************
Funciones de ejecución
*****************************************************/


    /**
    ***Funcion que ejecuta una demo
    **/
    function execDemo(){
        global $stagesInit;

        //Recogemos la ruta completa
        $route = array("safe", "LEX", "PRE", "SYN", "REN", "TYP", "DSG", "SHA",
                       "DEC", "CER", "ISA");

        //copiar el archivo de demo al directorio temporal del usuario
        $idSession = session_id();

        $file = "config/demo.safe";
        $newfile = "temp/".$idSession."/$idSession.safe";
        if (!copy($file, $newfile)) {
            echo "failed to copy $file...\n";
        }
        else{
            echo "COPY!!! $file to $newfile\n";
        }
        sleep(1);
        //Nos colocamos al principio del array
        $numStages = count($route);
        for ($i=1;$i<$numStages;$i=$i+1){
            //Ejecutamos la fase
            $res = execStage($route[$i], "demo");
            //change de state of the init file
            $stagesInit->getArcs()->changeArcState($route[0], 2);
            if($res == 1){
                $stagesInit->getArcs()->changeArcState($route[$i], 3);

                //Error!!!!!!!!!!!
                //Salimos:
                $i=$numStages+1;
```

```
        }
        else{
            $stagesInit->getArcs()->changeArcState($route[$i], 2);
            //Ok!!!!
        }
        //Esperamos que se cree correctamente el archivo
        sleep(2);
    }


}

/**
***Funcion que ejecuta la ruta predeterminada por los ticks!
**/
function execRoute(){
    global $stagesInit;

    //Recogemos la ruta completa
    $route = getRadioButtonsWay();

    //Recogemos inicio y fin
    $checkBoxes = getCheckBoxesWay();
    $init = array_search($checkBoxes[0],$route);
    if($init == false){
        $err = getErr("No Stage in this way");
        die($err);
    }
    $end = array_search($checkBoxes[1],$route);
    if($end == false){
        $err = getErr("No Stage in this way");
        die($err);
    }


    //Lanzamos la ejecución:
    for ($i=$init;$i<=$end;$i=$i+1){
        //Ejecutamos la fase
        $res = execStage($route[$i], "route");
        if($res == 1){
            $stagesInit->getArcs()->changeArcState($route[$i], 3);
            //Error!!!!!!!!!!!!
            //Salimos:
            $i=$end+1;
        }
        else{
            $stagesInit->getArcs()->changeArcState($route[$i], 2);
            //Ok!!!!
        }
        //Esperamos que se cree correctamente el archivo
        sleep(2);
    }

}

/**
***Funcion que ejecuta desde la fase uno a la ultima fase!
**/
function execAll(){
    global $stagesInit;

    //Recogemos la ruta completa
    $route = getRadioButtonsWay();

    //Nos colocamos al principio del array
    $numStages = count($route);
    for ($i=1;$i<$numStages;$i=$i+1){
        //Ejecutamos la fase
        $res = execStage($route[$i], "all");
```

```
            if($res == 1){
                $stagesInit->getArcs()->changeArcState($route[$i], 3);
                //Error!!!!!!!!!!!
                //Salimos:
                $i=$numStages+1;
            }
            else{
                $stagesInit->getArcs()->changeArcState($route[$i], 2);
                //Ok!!!!
            }
            //Esperamos que se cree correctamente el archivo
            sleep(2);
        }
    }


    /**
    ***Funcion que ejecuta una unica fase!
    **/
    function execStage($stage, $version){

        //Compilador por fases
        $compilador = "../safe/SafeCompXML";

        //Identificador de la sesion
        $idSession = session_id();

        $prevStage = faseAnterior($stage);
        $input = "temp/".$idSession."/$idSession.".$prevStage;

        if (!file_exists($input)){
            $err = getErr("No file");
            die($err);
        }

        //Ruta completa al archivo de salida;

        //Archivo de salida
        $output_file = "$idSession.".$stage;
        $output = "temp/".$idSession."/".$output_file;


        if(!is_dir(dirname($output))){
            mkdir("temp/".$idSession, 0777);
        }


        $comando = $compilador." ".$stage." <".$input." >".$output;
        //Ejecutamos la fase
        $salida = system($comando,$val_retorno);
        return $val_retorno;

    }

/*******************************************************
Funciones auxiliares
*******************************************************/

    //Función que nos devuelve el camino marcado por los radiobutton
    //independientemente del tipo de ejecución
    function getRadioButtonsWay(){

        global $route;
        global $stagesInit;
        $route = Array();
        $route[] = "safe";

        $father = $stagesInit->getStage(1);
        $route[] = $father->getDescrip();
```

```
        $sons = $father->getNextStages();

        //hasta que no haya mas hijos
        while($sons[0] != "-"){
            //si solo tiene un hijo
            if(count($sons) == 1){
                $father = $stagesInit->getStage($sons[0]);
                $route[] = $father->getDescrip();
                $sons = $father->getNextStages();
            }
            //si tiene varios hijos, hallar ruta
            else{
                for($i = 0; $i < count($sons); $i++){
                    $aux = "radioButton".$father->getDescrip();
                    //si esta activado el radiobutton del hijo, ser nuevo padre
                    if($_POST[$aux] ==
"radioButton".$father->getDescrip()."to".$stagesInit->getStage($sons[$i])->getDescrip()){
                        $father = $stagesInit->getStage($sons[$i]);
                        $route[] = $father->getDescrip();
                        $sons = $father->getNextStages();
                    }
                }
            }
        }
        return $route;
    }

    function getCheckBoxesWay(){
        global $route;
        global $stagesInit;
        $route = Array();

        for($i = 1; $i <= $stagesInit->getSize(); $i++){
            $current = $stagesInit->getStage($i);
            $aux = "checkBox".$current->getDescrip();
            echo $aux;
            if($_POST[$aux]){
                $route[] = $current->getDescrip();
            }
        }
        return $route;
    }

    //Funcion que nos devuelve la fase anterior a la pasada por parametro
    function faseAnterior($fase){
        $route = getRadioButtonsWay();
        $id = 0;
        while (strcasecmp($fase,$route[$id])!=0){
            $id += 1;
        }
        return $route[$id-1];
    }

    //Funcion que nos devuelve la fase anterior a la pasada por parametro en la demo
    function faseAnteriorDemo($fase){
        $route = getRadioButtonsWay();

        $id = 0;
        while (strcasecmp($fase,$route[$id])!=0){
            $id += 1;
        }
        return $route[$id-1];
    }



?>
```

```php
<?php session_start(); ?>
<html>
    <title>
        wSafe Project... Developed by Juan, Enrique & Lope
    </title>
    <head>

        <link type="text/css" href="css/safe.css" rel="stylesheet">
        <script src="js/wz_jsgraphics.js"></script>
        <script src="js/safe.js"></script>
        <script src="js/infoboxes.js"></script>
        <script src="js/webtoolkit.aim.js"></script>
        <script language="JavaScript" src="js/menu.js"></script>
        <script type="text/javascript">
            function startCallback() {
                // make something useful before submit (onStart)
                return true;
            }

            function completeCallback(response) {
                // make something useful after (onComplete)
                document.getElementById('nr').innerHTML =
                    parseInt(document.getElementById('nr').innerHTML) + 1;
                document.getElementById('r').innerHTML = response;
            }
        var general_opacity = 1.0;
        </script>
    </head>
    <body onunload="reset.php">
    <div id="container">
        <form id="paths_form" method="post" action="index.php" enctype="multipart/form-data">
            <?php
            //Incluimos el gestor de errores
            require_once('errors.php');
            //inicializar la estructura de fases
            include('init_stages.php');
            //incluir funcionalidad de cargar estado
            include('load_status.php');
            //incluir funcionalidad de cargar fichero
            require_once('upload_form.php');
            require_once('uploads_files.php');
            //incluir la funcionalidad de la ejecucion.
            include('execute.php');
            //incluir la funcionalidad de reset.
            include('reset.php');
            //incluir la funcionalidad de guardar el estado actual de arcos
            include('save_status.php');

            ?>
            <div id="header"></div>
            <div id="logo"><img src="img/logo.png" alt="wSafe"></img></div>
            <div id="app_menu">
                <script type="text/javascript">renderMenu();</script>
            </div>
            <br/><br/><br/><br/>
            <div id="Canvas" class="main" align="center">
                <div id="stagesafe" style="position: absolute;float: none; top:
                    <?php echo $stagesInit->getStageIndex(0)->getPositionY()-100; ?>px;
                    left: 300px;">
                </div>
                <?php
                $stagesInit->render();
                ?>
                <input id="execute_action" type="hidden" name="execute_action" value="">
                <input id="execute_stage" type="hidden" name="stage" value="">
                <input id="initial_file_loaded" type="hidden" name="initial_file_loaded"
                    value="<?php echo $conf_loaded ?>">

            </div>
```

```
        </form>
     <div id="footer">
          <div>&copy; 2009 wSafe - Safe Web Interface</div>
          <div>Developed by Juan, Enrique & Lope. PHP Powered</div><br/>
     </div>
        <script type="text/javascript">
            document.getElementById('Canvas').style.opacity = general_opacity;
            document.getElementById('app_menu').style.opacity = general_opacity;
        </script>
    </div>
    </body>
    <script src="js/load_default_way.js"></script>
</html>
```

```php
<?php
require_once('Stage.php');
require_once('Stages.php');

$info_file = fopen("config/info_fases.txt", "r");
$start = fgets($info_file);
$i = 1;
$conf_loaded = 'false';
$stagesInit = new Stages();

while($start != "END"){
    $id = fgets($info_file);
    $descrip = utf8_decode(fgets($info_file));
    $name = utf8_decode(fgets($info_file));
    $previousStage = fgets($info_file);
    $nextStage = fgets($info_file);
    $positionX = fgets($info_file);
    $positionY = fgets($info_file);
    $alin = fgets($info_file);

    ${"stage".$i} = new Stage($id, $descrip, $name, $previousStage, $nextStage,
                              $positionX, $positionY, $alin);
    $fases[] = ${"stage".$i};
    $stagesInit->addStage(${"stage".$i});

    $start = trim(fgets($info_file));
    $i++;
}$conf_loaded = 'true';
?>
```

```php
<?php
$idSession = session_id();
$file = "temp/".$idSession."/arc_status.txt";


if(file_exists($file)){
    echo "hacer del load <br/>";
    $stagesInit->makeArcs();
    $info_file = fopen($file, "r");

    $i = 0;
    while($src !="END" || $desc !="END" || $desc !="END"){
        $src = trim(fgets($info_file));
        $desc = trim(fgets($info_file));
        $state = trim(fgets($info_file));

        $arc_aux = $stagesInit->getArcs()->getArcIndex($i);
        if($arc_aux){
            $arc_aux->setState($state);
        }
        $i++;
    }
}
else{
    $stagesInit->makeArcs();
}
?>
```

```php
<?php
if (isset($_POST['execute_action']) && $_POST['execute_action'] == "reset"){
    global $conf_loaded;
    unset($conf_loaded);
    $temp_dir = "temp/".session_id();
    removeSession($temp_dir);
    include('init_stages.php');
    include('load_status.php');
}

function removeSession($tmp_dir){

    //Borramos los archivos generados
    $cmd = "rm -R ".$tmp_dir;
    system($cmd, $output);
    //Habra que poner todos los iconos amarillos para la carga!!!
}
?>
```

```php
<?php
require_once('Arc.php');
require_once('Arcs.php');

$idSession = session_id();
$output = "temp/".$idSession."/arc_status.txt";
if(!is_dir(dirname($output))){
    mkdir("temp/".$idSession, 0777);
}
$info_file = fopen("temp/".$idSession."/arc_status.txt", "w");

$arcs = $stagesInit->getArcs();

for($i = 0; $i < $stagesInit->getArcs()->getSize(); $i++){
    $src = $stagesInit->getArcs()->getArcIndex($i)->getSrc()->getDescrip();
    $desc = $stagesInit->getArcs()->getArcIndex($i)->getDest()->getDescrip();
    $state = $stagesInit->getArcs()->getArcIndex($i)->getState();
    fputs($info_file, $src."\n");
    fputs($info_file, $desc."\n");
    fputs($info_file, $state."\n");
}

fputs($info_file, "END"."\n");
fputs($info_file, "END"."\n");
fputs($info_file, "END"."\n");
fclose($info_file);

?>
```

```php
<?php
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
//require_once('Constantes.php');

/**
 * Description of Fase
 *
 * @author quique
 */
class Stage {

    /**
     * Id de la fase
     * @var <type>
     */
    private $id;

    /**
     * Nombre de la fase
     * @var <type>
     */
    private $descrip;

    /**
     * Nombre de la fase
     * @var <type>
     */
    private $name;

    /**
     * Fase que precede a la actual
     * @var <type>
     */
    private $previousStages;

    /**
     * Fase que sigue a la actual
     * @var <type>
     */
    private $nextStages;

    /**
     * Cordenada x de la fase
     * @var <type>
     */
    private $positionX;

    /**
     * Cordenada y de la fase
     * @var <type>
     */
    private $positionY;

    /**
     * Alineación de la fase
     */
    private $alin;

    /**
     * Indica el estado de la fase que puede ser en espera, ejecutada
     * correctamente o ejecutada con error
     * @var <type>
     */
    private $state;

    function __construct($id, $descrip, $name, $previousStages, $nextStages,
```

```
                        $positionX, $positionY, $alin){
        $this->id = rtrim($id);
        $this->descrip = rtrim($descrip);
        $this->name = rtrim($name);
        $this->previousStages = (explode(',',rtrim($previousStages)));
        $this->nextStages = (explode(',',rtrim($nextStages)));
        $this->positionX = rtrim($positionX);
        $this->positionY = rtrim($positionY);
        $this->state = 0;
        $this->alin = rtrim($alin);
    }

    public function setId($id){
        $this->id = $id;
    }

    public function getId(){
        return $this->id;
    }

    public function setName($name){
        $this->name = $name;
    }

    public function getName(){
        return $this->name;
    }

    public function setDescrip($descrip){
        $this->descrip = $descrip;
    }

    public function getDescrip(){
        return $this->descrip;
    }

    public function setPreviousStage($previousStages){
        $this->previousStages = $previousStages;
    }

    public function getPreviousStage(){
        return $this->previousStages;
    }

    public function setNextStages($nextStagess){
        $this->nextStages= $nextStages;
    }

    public function getNextStages(){
        return $this->nextStages;
    }

    public function setPositionX($positionX){
        $this->positionX = $positionX;
    }

    public function getPositionX(){
        return $this->positionX;
    }

    public function setPositionY($positionY){
        $this->positionY = $positionY;
    }

    public function getPositionY(){
        return $this->positionY;
    }

    public function setState($new_state){
```

```
        $this->state = $new_state;
    }

    public function getState(){
        return $this->state;
    }

    public function setAlin($alin){
        $this->alin = $alin;
    }

    public function getAlin(){
        return $this->alin;
    }

    public function render($ste){
        global $initialFileLoaded;
        ?>

<div id="stage<?php echo $this->getDescrip(); ?>" class="stage" style="float:
            <?php echo $this->alin ?>;top:<?php echo $this->getPositionY()."px" ?>;
            left:<?php echo $this->getPositionX()."px" ?>;">
    <div class="stage_int_<?php echo $this->state; ?>">
        <div class="stage_tools" align="center">
            <input id="checkBox<?php echo $this->getDescrip(); ?>" type="checkbox"
                name="checkBox<?php echo $this->getDescrip(); ?>" value="ON" />
            <?php
            if($ste == 2){
            ?>
                <img onclick="if(verifyStage()){
                    document.getElementById('execute_action').value='stage';
                    document.getElementById('execute_stage').value='<?php echo
$this->getDescrip(); ?>';
                    document.getElementById('paths_form').submit()}"
                    src="./img/next_f2.png" type="image" disabled="disabled"
                    value="play" width="16" border="0" style="cursor: pointer;"/>
            <?php
            }else{
            ?>
                <img onclick="alert('A previous stage file it is necessary to execute');"
                    src="./img/next_f2.png" type="image" disabled="disabled"
                    value="play" width="16" border="0" style="cursor: pointer;"/>
            <?php
            }
            ?>
        </div>
        <div class="stage_title" align="center">
            <acronym title="<?php echo $this->name ;?>" xml:lang="en">
                <?php echo $this->descrip; ?>
            </acronym>
        </div>
    </div>
</div>
<?php
    }
}
?>
```

```php
<?php
include_once('Stage.php');
include_once('Arc.php');
include_once('Arcs.php');
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
class Stages{

    private $collectionOfStages;

    private $arcs;

    function __construct(){
        $this->collectionOfStages = Array();
        $this->arcs = new Arcs();
    }

    public function addStage($s){

        $this->collectionOfStages[] = $s;
    }

    public function getStage($id){
        for($i = 0; $i < count($this->collectionOfStages); $i++){
            $s = $this->collectionOfStages[$i];
            if($s->getId() == $id){
                return $s;
            }
        }
        return false;
    }

    public function getStageDesc($desc){
        for($i = 0; $i < count($this->collectionOfStages); $i++){
            $s = $this->collectionOfStages[$i];
            if($s->getDescrip() == $desc){
                return $s;
            }
        }
        return false;
    }

    public function getStageIndex($i){
        if($i < count($this->collectionOfStages)){
            $s = $this->collectionOfStages[$i];
            return $s;
        }
        else{
            return false;
        }
    }

    public function makeArcs(){
        $safe_stage = new Stage(0, "safe", "safe", "-", "-", -100, 300, none);
        $arc = new Arc($safe_stage, $this->getStageIndex(0), 0, 0);
        $this->arcs->addArc($arc);

        for($i = 0; $i < count($this->collectionOfStages); $i++){
            $currentStage = $this->collectionOfStages[$i];
            $nextStages = $currentStage->getNextStages();

            //si al menos un hijo dibujar los arcos
            if($nextStages[0] != "-"){
                for($j = 0; $j < count($nextStages);$j++){
                    $currentSon = $this->getStage($nextStages[$j]);
                    if(count($nextStages) > 1){
                        $arc = new Arc($currentStage, $currentSon, 0, 1);
```

```
                    }
                    else{
                        $arc = new Arc($currentStage, $currentSon, 0, 0);
                    }
                    $this->arcs->addArc($arc);
                }
            }
            else{
                //for not son stages
                $arc = new Arc($currentStage, $currentStage, 1, 0);
                $this->arcs->addArc($arc);
            }
        }
    }

    public function getSize(){
        return count($this->collectionOfStages);
    }

    public function render(){
        //draw the initial "arc", the load initial file
        $this->arcs->render(new Stage(0, "safe", "safe", "-", "-", -100, 300, none), $this);
        for($i = 0; $i < count($this->collectionOfStages); $i++){
            //dibujar el rectangulo de la fase correspondiente
            $s = $this->collectionOfStages[$i];
            foreach($s->getPreviousStage() as $son){
                if(!$this->getStage($son)){
                    $arc = $this->arcs->getArc(
                        new Stage(0, "safe", "safe", "-", "-", -100, 300, none), $s);
                }
                else{
                    $arc = $this->arcs->getArc($this->getStage($son), $s);
                }
            }
            $s->render($arc->getState());
            //private pintar arcos que lo unen a sus hijos
            $this->arcs->render($s, $this);
        }
    }

    public function getArcs(){
        return $this->arcs;
    }
}
?>
```

```php
<?php
if(isset($_POST['execute_action']) && $_POST['execute_action'] == 'uploadrequest'){
    $ext = $_POST['stage'];
    ?>
    <script type="text/javascript">
        general_opacity = 0.2;
    </script>
    <div class="load_file">
        <div id="file">
            <input type="file" name="userfile">
        </div>
        <div id="enviar" style="float: left;">
            <input onclick="document.getElementById('execute_action').value =
            'uploadconfirm';document.getElementById('execute_stage').value=
                '<?php echo $_POST['stage']?>'" value="Send" type="submit">
        </div>
        <div id="back" style="float: right;">
            <input type="button" value="Back" onclick="javascript:alert('\
            It is possible it spends any seconds to get the old status.\nPlease,\n\
            accept the next dialog box.');javascript:history.back()">
        </div>
    </div>
<?php
}
?>
```

```php
<?
if(isset($_POST['execute_action']) && $_POST['execute_action'] == 'uploadconfirm'){

    //Cogemos la extensi?n origen
    $ext_or = $_POST["stage"];

    //Identificador de la sesion
    $idSession = session_id();

    //Creamos el directorio si no existe

    //Trabajamos con el fichero

    //nombre y extensi?n del fichero
    $origen=$_FILES['userfile']['name'];

    // Separamos nombre de extensi?n;
    $tmp = explode(".",$origen);
    $nombre = $tmp[0];
    if($tmp[1]=='safe' || $tmp[1]=='SAFE'){
        $ext = 'safe';
    }else{
        $ext = strtoupper($tmp[1]);
    }
    if ($ext_or != $ext){
        $err = getErr("Incorrect file");
        die($err);
    }

    //Creamos el nombre del fichero: (idSession.safe)
    $fichero = $idSession.".".$ext;

    //ruta origen del archivo
    $path=$_FILES['userfile']['tmp_name'];

    //ruta donde guardaremos el fichero
    $ruta="temp/".$idSession."/".$fichero;
    if(!is_dir(dirname($ruta))){
        mkdir("temp/".$idSession, 0777);
    }

    //tipo del fichero
    $tipo_fichero = $_FILES['userfile']['type'];

    //tama?o del  fichero
    $tam_fichero=$_FILES['userfile']['size'];

    if (is_uploaded_file($path)){
        $upload = move_uploaded_file($path,$ruta);
        if ($upload){
            $stagesInit->getArcs()->changeArcState($ext_or, 2);
        }
        else{
            echo "The file hasn't been uploaded";

        }
    }
    else{
        echo "The file can't be uploaded";
        echo "<br/>";
    }
}
?>
```

```php
<?php

session_start();
$idSession = session_id();

$id = $_GET["id"];

//Fichero para la ejecución
$path = "temp/".$idSession."/".$idSession.".".$id;

//Fichero para mostrar
if($id == "safe"){
    $path_xml = "temp/".$idSession."/".$id.".safe";
}
else{
    $path_xml = "temp/".$idSession."/".$id.".xml";
}

//Creamos el fichero nuevo
copy($path,$path_xml);

//Nos vamos a la pagina xml
header('Location: '.$path_xml);

?>
```

Anexo B

Archivos javaScript

```
/**
 * Hace visible la caja de informacion pasada por parametro
 */
function displayInfoBox(box){
    box.style.display = "block";
}

/**
 * Hace no visible la caja de informacion pasada por parametro
 */
function hideInfoBox(box){
    box.style.display = "none";
}

/**
 * Cambia la visibilada de la caja pasada por parametro
 */
function displayInfoBox_edit(box){
    if(box.style.display == "block"){
        box.style.display = "none";
    }
    else{
        box.style.display = "block";
    }

}
```

```
/**
 * Script que establece el valor por defecto de los radiobuttons cuando hay
 * bifurcaciones
 */

bifurcation1 = "radioButtonDSGtoSHA";
bifurcation2 = "radioButtonDECtoCER";
inputs = document.getElementsByTagName('input');
//take all the inputs objects
for(i = 0; i<inputs.length; i++){
    //tall only the radiobuttons objects
    if(inputs[i].type == "radio"){
        if(inputs[i].value == bifurcation1 || inputs[i].value == bifurcation2){
            inputs[i].checked = 'checked';
        }
    }
}
```

```
/*
 +-------------------------------------------------------------------+
 |                     J S - M E N U   (v1.8)                        |
 |                                                                   |
 | Copyright Gerd Tentler               www.gerd-tentler.de/tools    |
 | Created: Jul. 19, 2002               Last modified: May 1, 2007   |
 +-------------------------------------------------------------------+
 | This program may be used and hosted free of charge by anyone for  |
 | personal purpose as long as this copyright notice remains intact. |
 |                                                                   |
 | Obtain permission before selling the code for this program or     |
 | hosting this software on a commercial website or redistributing   |
 | this software over the Internet or in any other medium. In all    |
 | cases copyright must remain intact.                               |
 +-------------------------------------------------------------------+

=============================================================================================
=========
 This script was tested with the following systems and browsers:

 - Windows XP: IE 6, NN 7, Opera 7 + 9, Firefox 2
 - Mac OS X:   IE 5, Safari 1

 If you use another browser or system, this script may not work for you - sorry.
=============================================================================================
=========
*/

var DOM = document.getElementById;
var IE4 = document.all;
var NN4 = document.layers;

var mObj = new Array();

function MENU(style) {
    //-------------------------------------------------------------------------------------
-------------
    // Configuration
    //-------------------------------------------------------------------------------------
-------------

    this.style = style ? style : "left";      // menu style (top or left)
    this.floatMenu = true;                    // floating menu (true or false)*

    this.fadeInSpeed = 20;                    // fade-in speed (0 - 30; 0 = no fading)
    this.fadeOutSpeed = 10;                   // fade-out speed (0 - 30; 0 = no fading)
    this.fadeOutDelay = 500;                  // fade-out delay (1/1000 seconds)

    this.imgBlank = "img/blank.gif";              // path to blank image
    this.imgArrow = "img/arrow.gif";              // path to arrow image
    this.imgArrowWidth = 7;                   // arrow image width (pixels)

    // main menu configuration:

    this.mainTop = 0;                         // top position (pixels)
    this.mainLeft = 0;                        // left position (pixels)
    this.mainBGColor = "#406080";             // background color OR background image
    this.mainBorderWidth = 10;                // border width (pixels)
    this.mainArrows = true;                   // show sub menu arrows (true or false)
    this.mainOpacity = 100;                   // opacity (0 - 100)**

    this.mainItemWidth = 100;                 // item width (pixels)
    this.mainItemColor = "#6080A0";           // item color
    this.mainItemHilight = "#80A0C0";         // item hilight color
    this.mainItemFont = "Arial, Helvetica";   // font family (CSS-spec)
    this.mainItemFontColor = "white";         // font color
    this.mainItemFontHilight = "#FFFF00";     // font hilight color; NN4: not supported
    this.mainItemFontSize = 15;               // font size (pixels)
    this.mainItemAlign = "center";            // text align (left / center / right)
```

```
    this.mainItemPadding = 4;                   // text padding (pixels)
    this.mainItemSpacer = 10;                   // space between items (pixels)
    this.mainItem3D = 2;                        // 3D border (pixels); NN4: not supported

    // sub menu(s) configuration:

    this.subBGColor = "#406080";                // background color OR background image
    this.subBorderWidth = 1;                    // border width (pixels)
    this.subArrows = true;                      // show sub menu arrows (true or false)
    this.subOpacity = 90;                       // opacity (0 - 100)**
    this.subOffsetLeft = 10;                    // left offset (pixels)

    this.subItemWidth = 100;                    // item width (pixels)
    this.subItemColor = "#E0F0FF";              // item color
    this.subItemHilight = "#C0D8F0";            // item hilight color
    this.subItemFont = "Arial, Helvetica";      // font family (CSS-spec)
    this.subItemFontColor = "#204060";          // font color
    this.subItemFontHilight = "#D00000";        // font hilight color; NN4: not supported
    this.subItemFontSize = 12;                  // font size (pixels)
    this.subItemAlign = "left";                 // text align (left / center / right)
    this.subItemPadding = 4;                    // text padding (pixels)
    this.subItemSpacer = 1;                     // space between items (pixels)
    this.subItem3D = 0;                         // 3D border (pixels); NN4: not supported

    // *  With Safari 1 (Mac OS X) performance was poor, i.e. floating speed was slow. With
IE 5 (Mac OS X)
    //    floating didn't work properly.
    //
    // ** Opacity was successfully tested only on Windows XP with IE 6, NN 7, Opera 9 and
Firefox 1 + 2;
    //    NN 7 showed different results, though. It should also work with Safari, version 1.2
or higher.

    //--------------------------------------------------------------------------------------
-------------
    // Functions
    //--------------------------------------------------------------------------------------
-------------

    this.mOver = false;
    this.mNr = this.iv = this.timer = 0;
    this.sections = new Array();
    this.items = new Array();
    this.targetWindow = 0;

    this.entry = function(level, height, text, url, target, onClick) {
        var nr = this.items.length;
        var parent = nr - 1;
        this.items[nr] = new makeItem(level, height, text, url, target, onClick);
        if(parent >= 0) {
            if(level == this.items[parent].level + 1) this.items[nr].parent = parent;
            else if(level == this.items[parent].level) this.items[nr].parent =
this.items[parent].parent;
            else if(level < this.items[parent].level) {
                for(var i = parent; i >= 0; i--) {
                    if(this.items[i].level == level) {
                        this.items[nr].parent = this.items[i].parent;
                        break;
                    }
                }
            }
        }
    }

    this.getObj = function(id, cont) {
        var obj;
        if(DOM) obj = document.getElementById(id).style;
        else if(IE4) obj = document.all[id].style;
        else if(NN4) obj = cont ? document.layers[cont].document.layers[id] :
```

```
document.layers[id];
        return obj;
    }

    this.getScrTop = function() {
        var scrTop = 0;
        if(document.documentElement && document.documentElement.scrollTop)
            scrTop = document.documentElement.scrollTop;
        else if(document.body && document.body.scrollTop)
            scrTop = document.body.scrollTop;
        else if(window.pageYOffset) scrTop = window.pageYOffset;
        return scrTop;
    }

    this.setOpacity = function(nr, opacity) {
        if(!NN4) {
            var obj = this.getObj('section' + this.mNr + '_' + nr);
            if(obj) {
                obj.opacity = opacity / 100;
                obj.MozOpacity = opacity / 100;
                obj.KhtmlOpacity = opacity / 100;
                obj.filter = 'alpha(opacity=' + opacity + ')';
            }
        }
    }

    this.fadeIn = function(nr) {
        if(this.sections[nr].active) {
            var maxOp = (this.items[this.sections[nr].nr].level < 2) ? this.mainOpacity :
this.subOpacity;
            if(this.fadeInSpeed && this.sections[nr].opacity < maxOp) {
                this.sections[nr].opacity += this.fadeInSpeed;
                if(this.sections[nr].opacity > maxOp) this.sections[nr].opacity = maxOp;
                this.setOpacity(nr, this.sections[nr].opacity);
                if(this.sections[nr].timer) clearTimeout(this.sections[nr].timer);
                this.sections[nr].timer = setTimeout('mObj[' + this.mNr + '].fadeIn(' + nr +
')', 1);
            }
            else {
                this.sections[nr].opacity = maxOp;
                this.setOpacity(nr, maxOp);
            }
        }
    }

    this.fadeOut = function(nr) {
        if(this.fadeOutSpeed && this.sections[nr].opacity > 0) {
            this.sections[nr].opacity -= this.fadeOutSpeed;
            if(this.sections[nr].opacity < 0) this.sections[nr].opacity = 0;
            this.setOpacity(nr, this.sections[nr].opacity);
            if(this.sections[nr].timer) clearTimeout(this.sections[nr].timer);
            this.sections[nr].timer = setTimeout('mObj[' + this.mNr + '].fadeOut(' + nr +
')', 1);
        }
        else {
            var obj = this.getObj('section' + this.mNr + '_' + nr);
            obj.visibility = NN4 ? 'hide' : 'hidden';
            this.sections[nr].opacity = 0;
        }
    }

    this.showMenu = function(nr) {
        var obj = this.getObj('section' + this.mNr + '_' + nr);
        if(!this.sections[nr].active) {
            if(this.floatMenu) this.sections[nr].y = this.getScrTop() +
this.sections[nr].topY;
            if(IE4) obj.pixelTop = this.sections[nr].y;
            else obj.top = this.sections[nr].y + (DOM ? 'px' : '');
            obj.visibility = NN4 ? 'show' : 'visible';
```

```
            this.sections[nr].active = true;
            this.fadeIn(nr);
        }
    }

    this.hideMenu = function(nr) {
        var obj = this.getObj('section' + this.mNr + '_' + nr);
        if(this.sections[nr].active) {
            this.sections[nr].active = false;
            this.fadeOut(nr);
        }
    }

    this.hilight = function(section, item, bgColor, fontColor) {
        var obj = this.getObj('item' + this.mNr + '_' + item, 'section' + this.mNr + '_' +
section);
        if(NN4) obj.bgColor = bgColor;
        else obj.backgroundColor = bgColor;
        if(!NN4) {
            obj = this.getObj('text' + this.mNr + '_' + item);
            obj.color = fontColor;
        }
        if(bgColor == this.subItemColor || bgColor == this.mainItemColor) this.mOver = false;
        else this.mOver = true;
    }

    this.getMenu = function(item, section) {
        if(this.sections[section].active) {
            for(var i = section+1; i < this.sections.length; i++) {
                if(this.sections[i].nr == item) {
                    if(!this.sections[i].active) this.showMenu(i);
                }
                else if(this.sections[i].active) this.hideMenu(i);
            }
        }
    }

    this.hideSubs = function(start) {
        if(!this.mOver) for(var i = start; i < this.sections.length; i++) {
            if(this.sections[i].active) this.hideMenu(i);
        }
        this.timer = 0;
    }

    this.jumpURL = function(item) {
        if(this.items[item].url) {
            if(this.items[item].target) {
                if(this.items[item].target.indexOf('parent.') == -1 &&
this.items[item].target.indexOf('top.') == -1) {
                    if(this.targetWindow && !this.targetWindow.closed)
this.targetWindow.location.href =
                        this.items[item].url;
                    else this.targetWindow = window.open(this.items[item].url,
'targetWindow');
                    this.targetWindow.focus();
                }
                else eval(this.items[item].target + '.location.href = "' +
this.items[item].url + '"');
            }
            else document.location.href = this.items[item].url;
        }
    }

    this.checkIt = function() {
        var active = false;
        var i;
        if(this.floatMenu) for(i = 0; i < this.sections.length; i++) {
            if(this.sections[i].active) this.floatIt(i, this.sections[i].topY);
        }
```

```
        if(!this.mOver && !this.timer) {
            for(i = 0; i < this.sections.length && !active; i++) {
                if(this.sections[i].active) active = true;
            }
            if(active) {
                if(this.fadeOutDelay) this.timer = setTimeout('mObj[' + this.mNr +
']. hideSubs(1)', this.fadeOutDelay);
                else this.hideSubs(1);
            }
        }
    }

    this.floatIt = function(nr, topY) {
        var obj = this.getObj('section' + this.mNr + '_' + nr);
        if(obj.visibility == 'visible' || obj.visibility == 'show') {
            var scrTop = this.getScrTop() + topY;
            var elmTop = IE4 ? obj.pixelTop : parseInt(obj.top);
            if(elmTop != scrTop) this.smoothIt(obj, nr, scrTop);
        }
    }

    this.smoothIt = function(obj, nr, scrTop) {
        if(scrTop != this.sections[nr].y) {
            var percent = .1 * (scrTop - this.sections[nr].y);
            if(percent > 0) percent = Math.ceil(percent);
            else percent = Math.floor(percent);
            this.sections[nr].y += percent;
            if(IE4) obj.pixelTop = this.sections[nr].y;
            else if(NN4 || DOM) obj.top = this.sections[nr].y + (DOM ? 'px' : '');
        }
    }

    this.buildItems = function(parent, section) {
        var arrows, link, img, color, itemColor, itemHilight, itemAlign, left;
        var width, height, border, spacer, itemPadding, item3D, topY, clsItem, j;

        for(var i = cnt = 0; i < this.items.length; i++) {
            if(this.items[i].parent == parent) {
                if(this.items[i].level < 2) {
                    border = this.mainBorderWidth;
                    color = this.mainBGColor;
                    spacer = this.mainItemSpacer;
                    arrows = this.mainArrows;
                    itemColor = this.mainItemColor;
                    itemFontColor = this.mainItemFontColor;
                    itemHilight = this.mainItemHilight;
                    itemFontHilight = this.mainItemFontHilight;
                    itemAlign = this.mainItemAlign;
                    itemPadding = this.mainItemPadding;
                    item3D = this.mainItem3D;
                    clsItem = 'clsMainItem' + this.mNr;
                    width = this.mainItemWidth + itemPadding*2;
                    if(this.style == 'top') left = border + (width + item3D*2 + spacer) *
cnt++;
                    else left = border;
                }
                else {
                    border = this.subBorderWidth;
                    color = this.subBGColor;
                    spacer = this.subItemSpacer;
                    arrows = this.subArrows;
                    itemColor = this.subItemColor;
                    itemFontColor = this.subItemFontColor;
                    itemHilight = this.subItemHilight;
                    itemFontHilight = this.subItemFontHilight;
                    itemAlign = this.subItemAlign;
                    itemPadding = this.subItemPadding;
                    item3D = this.subItem3D;
                    clsItem = 'clsSubItem' + this.mNr;
```

```
                    width = this.subItemWidth + itemPadding*2;
                    left = border;
                }
                height = this.items[i].height + itemPadding*2;

                if(!topY) topY = border;
                link = 'javascript:mObj[' + this.mNr + '].jumpURL(' + i + ')';
                if(arrows && i+1 < this.items.length && this.items[i+1].level >
this.items[i].level) img = this.imgArrow;
                else img = '';

                for(j = 0; j < this.sections.length; j++) {
                    if(this.sections[j].nr == i+1) {
                        this.sections[j].topY = this.sections[section].topY + topY;
                        this.sections[j].y = this.sections[j].topY;
                    }
                }

                if(IE4 || DOM) document.write('<div id="item' + this.mNr + '_' + i + '"
style="position:absolute' +
                    '; top:' + topY + 'px' +
                    '; left:' + left + 'px' +
                    '; width:' + width + 'px' +
                    '; height:' + height + 'px' +
                    (itemColor ? '; background-color:' + itemColor : '') +
                    (item3D ? '; border:' + item3D + 'px outset white' : '') +
                    '; z-index:1">');
                else if(NN4) document.write('<layer id="item' + this.mNr + '_' + i + '"' +
                    ' top=' + topY +
                    ' left=' + left +
                    ' width=' + width +
                    ' height=' + height +
                    (itemColor ? ' bgcolor=' + itemColor : '') +
                    ' z-index=1>');
                document.write('<table border=0 cellspacing=0 cellpadding=' + itemPadding +
                    ' width=' + width + ' height=' + height + '><tr>' +
                    '<td id="text' + this.mNr + '_' + i + '" class="' + clsItem + '" align="'
+ itemAlign + '">' +
                    this.items[i].text + '</td>' +
                    (img ? '<td width=' + this.imgArrowWidth + ' align=right><img src="' +
img + '" width=' +
                        this.imgArrowWidth + '></td>' : '') +
                    '</tr></table>');
                if(IE4 || DOM) document.write('</div>');
                else if(NN4) document.write('</layer>');

                if(IE4 || DOM) document.write('<div style="position:absolute' +
                    '; top:' + topY + 'px' +
                    '; left:' + left + 'px' +
                    '; z-index:2">');
                else if(NN4) document.write('<layer' +
                    ' top=' + topY +
                    ' left=' + left +
                    ' z-index=2>');
                document.write('<a href="' + link + '" ' +
                    'onMouseOver="mObj[' + this.mNr + '].hilight(' + section + ', ' + i + ',
\'' + itemHilight +
                    '\', \'' + itemFontHilight + '\'); ' +
                    'mObj[' + this.mNr + '].getMenu(' + (i+1) + ', ' + section + '); ' +
                    'window.status=\'' + this.items[i].url + '\'; return true" ' +
                    'onMouseOut="mObj[' + this.mNr + '].hilight(' + section + ', ' + i + ',
\'' + itemColor + '\', \'' +
                    itemFontColor + '\'); ' +
                    'window.status=\'\'" ' + (this.items[i].onClick ? 'onClick="' +
this.items[i].onClick + '" ' : '') +
                    'onFocus="if(this.blur) this.blur()">' +
                    '<img src="' + this.imgBlank + '" border=0 width=' + (width + item3D*2) +
' height=' +
                        (height + item3D*2) + '></a>');
```

```
                if(IE4 || DOM) document.write('</div>');
                else if(NN4) document.write('</layer>');

                if(this.items[i].level > 1 || this.style != 'top') topY += height + item3D*2
+ spacer;
            }
        }
    }

    this.buildSections = function() {
        document.write('<style> ' +
            '.clsMainItem' + this.mNr + ' { color:' + this.mainItemFontColor +
            '; font-family:' + this.mainItemFont +
            '; font-size:' + this.mainItemFontSize + 'px; } ' +
            '.clsSubItem' + this.mNr + ' { color:' + this.subItemFontColor +
            '; font-family:' + this.subItemFont +
            '; font-size:' + this.subItemFontSize + 'px; } ' +
            '</style>');

        var width, border, color, itemPadding, item3D, spacer, left, height, level, section,
cnt, j;
        var mainHeight = 0;
        var bgImg = '';

        for(var i = 0; i < this.items.length; i++) {
            if(!i || this.items[i].level > this.items[i-1].level) {
                this.sections[this.sections.length] = new makeSection(i, this.items[i].level,
this.mainTop);
            }
        }

        for(i = cnt = 0; i < this.sections.length; i++) {
            section = this.sections[i].nr;
            level = this.sections[i].level;
            if(level < 2) {
                border = this.mainBorderWidth;
                color = this.mainBGColor;
                itemPadding = this.mainItemPadding;
                item3D = this.mainItem3D;
                spacer = this.mainItemSpacer;
                width = (this.style == 'top') ? 0 : this.mainItemWidth + border*2 +
itemPadding*2 + item3D*2;
                left = this.mainLeft;
            }
            else {
                border = this.subBorderWidth;
                color = this.subBGColor;
                itemPadding = this.subItemPadding;
                item3D = this.subItem3D;
                spacer = this.subItemSpacer;
                width = this.subItemWidth + border*2 + itemPadding*2 + item3D*2;
                if(this.style == 'top') {
                    if(level == 2) for(j = cnt = 0; j < section; j++) {
                        if(this.items[j].level == 1) cnt++;
                    }
                    left = this.mainLeft + this.mainBorderWidth + (this.mainItemWidth +
this.mainItemPadding*2 +
                        this.mainItem3D*2 + this.mainItemSpacer) * (cnt-1);
                    left += (level-2) * (width - this.subOffsetLeft);
                }
                else left = this.mainLeft + this.mainItemWidth - this.subOffsetLeft +
((this.mainBorderWidth+
                    this.mainItemPadding+this.mainItem3D)*2) + (level-2) *
(width-this.subOffsetLeft);
            }
            height = 0;

            for(j = section; j < this.items.length; j++) {
                if(this.items[j].parent == this.items[section].parent) {
```

```
                    if(level < 2 && this.style == 'top') {
                        width += this.mainItemWidth + itemPadding*2 + item3D*2 + spacer;
                        if(this.items[j].height > height) height = mainHeight =
this.items[j].height;
                    }
                    else height += this.items[j].height + itemPadding*2 + item3D*2 + spacer;
                }
            }
            if(this.style == 'top') {
                if(level < 2) {
                    width += border*2 - spacer;
                    height += itemPadding*2 + item3D*2 + spacer;
                    mainHeight += border + itemPadding*2 + item3D*2;
                }
                else if(level == 2) {
                    this.sections[i].topY += mainHeight;
                    this.sections[i].y = this.sections[i].topY;
                }
            }
            height += border*2 - spacer;

            if(color.search(/\.(jpg|jpeg|jpe|gif|png)$/i) != -1) {
                bgImg = color;
                color = '';
            }
            else bgImg = this.imgBlank;

            this.sections[i].width = width;
            this.sections[i].height = height;

            if(IE4 || DOM) document.write('<div id="section' + this.mNr + '_' + i + '"
style="position:absolute' +
                '; top:' + this.sections[i].topY + 'px' +
                '; left:' + left + 'px' +
                '; width:' + width + 'px' +
                '; height:' + height + 'px' +
                (color ? '; background-color:' + color : '') +
                '; z-index:' + i +
                '; visibility:hidden">');
            else if(NN4) document.write('<layer id="section' + this.mNr + '_' + i + '"' +
                ' top=' + this.sections[i].topY +
                ' left=' + left +
                ' width=' + width +
                ' height=' + height +
                (color ? ' bgcolor=' + color : '') +
                ' z-index=' + i +
                ' visibility="hide">');

            document.write('<a href="#" onMouseOver="mObj[' + this.mNr + '].mOver=true"
onMouseOut="mObj[' +
                this.mNr + '].mOver=false">' +
                '<img src="' + bgImg + '" border=0 width=' + this.sections[i].width +
                ' height=' + this.sections[i].height + '></a>');

            this.buildItems(section-1, i);

            if(IE4 || DOM) document.write('</div>');
            else if(NN4) document.write('</layer>');

            if(this.mainOpacity && level < 2) this.setOpacity(i, this.mainOpacity);
            else if(this.subOpacity && level > 1) this.setOpacity(i, this.subOpacity);
        }
    }

    this.create = function() {
        this.mNr = mObj.length;
        if(mObj[this.mNr] = this) {
            this.buildSections();
            this.showMenu(0);
```

```
            if(this.iv) clearInterval(this.iv);
            this.iv = setInterval('mObj[' + this.mNr + '].checkIt()', 1);
        }
        else alert("Could not create menu!");
    }

    //--------------------------------------------------------------------------------
    // Arguments: position level 1, [position level 2], ... [position level n]
    // Example:   jumpTo(1, 3, 2, 1) ==> this jumps to menu item 1.3.2.1
    // Note:      does not work with NN 4 (who knows why?)
    //
    this.jumpTo = function() {
        var pos, aktPos;
        var item = 0;
        var level = 1;
        if(!arguments) var arguments = this.jumpTo.arguments;
        for(i = 0; i < arguments.length; i++, level++) {
            pos = arguments[i];
            for(aktPos = 0; item < this.items.length && aktPos < pos; item++) {
                if(this.items[item].level == level) aktPos++;
            }
        }
        if(item) {
            item--;
            if(this.items[item].onClick) eval(this.items[item].onClick);
            this.jumpURL(item);
        }
    }
//--------------------------------------------------------------------------------
}

function makeItem(level, height, text, url, target, onClick) {
    this.level = level;
    this.height = height;
    this.text = text;
    this.url = url;
    this.target = target;
    this.onClick = onClick;
    this.parent = -1;
}

function makeSection(nr, level, topY) {
    this.nr = nr;
    this.level = level;
    this.topY = topY;
    this.active = false;
    this.y = topY;
    this.width = 0;
    this.height = 0;
    this.timer = 0;
    this.opacity = 0;
}

function renderMenu(){
    var mPrincipal  = new MENU("top");

    mPrincipal.entry(1,20,"File");
    //mPrincipal.entry(2,15,"Load Initial File", "upload_form.php?stage=safe","blank_");
    mPrincipal.entry(2,15,
        "Load Initial File",0,0,
        "document.getElementById('execute_action').value='uploadrequest';\n\
    document.getElementById('execute_stage').value='safe';document.getElementById('paths_for
m').submit()");
    mPrincipal.entry(2,15,"Reset",0,0,
"document.getElementById('execute_action').value='reset';\n\
        document.getElementById('paths_form').submit()");
    //mPrincipal.entry(2,15,"Reset", "index.php?action=reset");

    mPrincipal.entry(1,20,"Run");
```

```
    mPrincipal.entry(2,15,"Run All",0,0,"if(verifyRadioButtons()){\n\
        document.getElementById('execute_action').value='all';document.getElementById('paths_
form').submit()}");
    mPrincipal.entry(2,15,"Run Route",0,0,"if(verifyRadioButtons() && verifyCheckBoxes()){\n\
        document.getElementById('execute_action').value='route';document.getElementById('path
s_form').submit()}");

    mPrincipal.entry(1,20,"Demo");
    mPrincipal.entry(2,15,"Run Demo",0,0,"if(setRadioButtons()){\n\
        document.getElementById('execute_action').value='demo';document.getElementById('paths
_form').submit()}");


    mPrincipal.entry(1,20,"Help");
    mPrincipal.entry(2,15,"About wSafe");
    mPrincipal.entry(2,15,"Help","http://google.es","blank_");


    mPrincipal.mainTop = 60;                // menu top position in pixels
    mPrincipal.mainLeft = 385;              // menu left position in pixels
    mPrincipal.mainBGColor = "#A0B8C7";     // Color de fondo
    mPrincipal.mainArrows = false;          // show sub menu arrows (true or false)
    mPrincipal.mainItem3D = 1;              // 3D border (pixels); NN4: not supported
    mPrincipal.mainItemWidth = 150;         // item width (pixels)
    mPrincipal.subItemWidth = 150;          // item width (pixels)
    mPrincipal.mainItemSpacer = 2;         // space between items (pixels)
    mPrincipal.fadeOutDelay = 50;           // Retraso en el que desaparece el submenú

    mPrincipal.mainBorderWidth = 0;         // border width (pixels)
    mPrincipal.floatMenu = false;           // we don't want a floating menu

    mPrincipal.create();

}
//--------------------------------------------------------------------------------------------
---------
```

```
/**
 * Comprueba que ha sido seleccionado un radiobutton de los posibles
 * grupos existentes
 */
function verifyRadioButtons(){
    inputs = document.getElementsByTagName('input');
    any_checked = true;
    group_studied = "----"
    //take all the inputs objects
    for(i = 0; i<inputs.length; i++){
        //tall only the radiobuttons objects
        if(inputs[i].type == "radio"){
            //if all the radiobuttons in a same group name are false,
            //there is an error
            if(group_studied != inputs[i].name){
                //if all the radiobuttons belong a group are not checked, error
                if(!any_checked){
                    alert('Please, choose any way of each bifurcation');
                    return false;
                }
                group_studied = inputs[i].name;
                any_checked = false;
            }
            if(inputs[i].checked){
                any_checked = true;
            }
        }
    }
    //verifying the last group
    if(any_checked){
        return true;
    }
    else{
        alert('Please, choose any way of each bifurcation');
        return false;
    }
}

/**
 * Verifica que de todos los checkbox presentes en la aplicacion,
 *  se han seleccionado 2 y solo 2
 */
function verifyCheckBoxes(){
    inputs = document.getElementsByTagName('input');
    number_of_checks = 0;
    //take all the inputs objects
    for(i = 0; i<inputs.length; i++){
        //tall only the checkboxes objects
        if(inputs[i].type == "checkbox"){
            //count all the checkbox, it must be 2
            if(inputs[i].checked)
                number_of_checks++;

        }
    }
    if(number_of_checks == 2){
        return true;
    }
    else{
        alert('Please, select only 2 stages, start and end');
        return false;
    }
}

/**
 * Comprueba que se puede seguir adelante con la ejecucion de la fase
 */
function verifyStage(){
    if(verifyRadioButtons())
```

```
        return true;
    else
        return false;
}

/**
 * Establece el camino por defecto que seguira la ruta teniendo en cuenta los
 * radiobuttons
 */
function setRadioButtons(){
    inputs = document.getElementsByTagName('input');
    //take all the inputs objects
    for(i = 0; i<inputs.length; i++){
        //tall only the radiobuttons objects
        if(inputs[i].type == "radio"){
            //if all the radiobuttons in a same group name are false,
            // there is an error
            if(inputs[i].value == 'radioButtonDSGtoSHA' || inputs[i].value ==
                'radioButtonDECtoCER'){
                inputs[i].checked = 'checked';
            }
        }
    }
    return true;
}
```

```
/**
     * Dibuja los iconos de la fase correspondiente
     */
    this.drawExeIcons = function(id_father, id_son, hasBrother, state, x, y)
    {
        //alert(id_father+'to'+id_son);
        //alert(desc_father);
        imgSrc1 = 'img\/icon_ok.png';
        imgSrc2 = 'img\/icon_err.png';
        imgSrc3 = 'img\/icon_up.png';
        imgSrc4 = 'img\/trans.gif';

        this.htm += '<div id="exe_info_'+id_father+'_to_'+id_son+
            '" class="infobox" style="position: absolute;'+
        'left:' + (x+100) + 'px;'+
        'top:' + y + 'px;'+
        'display: none;'+
        '">Menu buttons of '+
        id_father+' to '+id_son+
        '<\/div>' +
        '<div id="edit_see_'+id_father+'_to_'+id_son+
            '" class="infobox_edit" style="position: absolute;'+
        'left:' + (x+0) + 'px;'+
        'top:' + (y-50) + 'px;'+
        'display: none;'+
        '"><a href="view.php?id='+id_father+
        '" target="_blank">View</a><br/><a href="download_file.php?id='+
        id_father+'">Download</a>'+
        '<br/><a style="text-decoration: underline;cursor: pointer;" \n\
        onclick="document.getElementById(\'execute_action\').value=\'uploadrequest\';\n\
        document.getElementById(\'execute_stage\').value=\''+id_father+'\';\n\
        document.getElementById(\'paths_form\').submit()">Upload new file</a>'+
        '<\/div>' +
        '<div id="exe_' + id_father + '_to_' + id_son + '" style="position:absolute;'+
        'left:' + x + 'px;'+
        'top:' + y + 'px;'+
        'width: 70px;' +
        'z-index: 1;" onmouseover="javascript:displayInfoBox(exe_info_'+
        id_father+
        '_to_'+
        id_son+
        ')" onmouseout="javascript:hideInfoBox(exe_info_'+
        id_father+
        '_to_'+
        id_son+
        ')">';
        if(state == 0){
            this.htm +='<a style="cursor: pointer;"\n\
         onclick="document.getElementById(\'execute_action\').value=\'uploadrequest\';\n\
        document.getElementById(\'execute_stage\').value=\''+id_father+'\';\n\
        document.getElementById(\'paths_form\').submit()"><img id="arcImg'+
                id_father+'_to_'+id_son+'" src="' + imgSrc3 +'" border="0"/></a>'
        }
        else if(state == 1){
            this.htm +='<a><img src="' + imgSrc4 +'" border="0"/></a>'
        }
        else if(state == 2){
            this.htm += '<img style="cursor: pointer;"
onclick="javascript:displayInfoBox_edit(edit_see_'+
            id_father+
            '_to_'+
            id_son+
            ')" src="' + imgSrc1 +'"/>';
        }
        else if(state == 3){
            this.htm += '<img style="cursor: pointer;"
onclick="javascript:displayInfoBox_edit(edit_see_'+
            id_father+
            '_to_'+
```

```
            id_son+
            ')" src="' + imgSrc2 +'"/>';
        }
        if(hasBrother == '1'){
            this.htm += '<input value="radioButton'+id_father+'to'+id_son+
                '" type="radio" name="radioButton'+id_father+'"/>';
        }

        this.htm+='<\/div>';
```

Anexo C

Archivos XSLT

```
<?xml version='1.0' encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:template match='/'>
  <!-- Esta es la plantilla para el elemento raiz -->
  <!-- Se pone la estructura de un documento html y dentro del cuerpo llamamos
  a apply-templates para que aplique las plantillas a los hijos de la raiz -->
    <html>
    <style type="text/css">
      body {
        padding-left: 1em;
        color: black;
        background-color: #DBE5F1;
      }
      #green {
        color: green;
      }
      #blue {
        color: #000066;
      }
      #red {
        color: #C00000;
      }
      #yellow {
        color: #FF9900;
      }
      #purple {
        color: #7030A0;
      }
      #black {
        color: #000000;
      }
    </style>
    <body>
    <font face="Verdana" size="2">
        <xsl:apply-templates /> <!-- esto aplicaria a todos los hijos de la raiz
        su correspondiente plantilla -->
    </font>
    </body>
    </html>
</xsl:template>


<!-- Plantilla para 9.DEC -->
<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato">
    <xsl:apply-templates/>
</xsl:template>

<!-- Strings -->
<xsl:template match="list-string">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="string">
    <!-- Para los nodos string, simplemente escribimos su valor, y metemos un espacio -->
    <xsl:if test=". != 'PAT'">
    <xsl:value-of select="."/> <!-- espacio en blanco -->
    </xsl:if>
</xsl:template>

<!-- Lit -->
<xsl:template match="int">
    <!-- Para los nodos int, simplemente escribimos su valor, y metemos un espacio -->
    <xsl:value-of select="@value"/> <!-- la arroba es para indicar que es un atributo
-->
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="bool">
    <!-- Para los nodos bool, simplemente escribimos su valor, y metemos un espacio -->
```

```
    <xsl:value-of select="@value"/> 
    <xsl:apply-templates/>
</xsl:template>

<!-- DestDec -->
<xsl:template match="ConstE-DestDec/ConstrT/string">
    { <xsl:value-of select="."/> }
</xsl:template>

<xsl:template match="ConstrT">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-ExpTipo"/>
    <xsl:if test="count(list-string/string)>0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span></xsl:if>
</xsl:template>

<xsl:template match="list-Exp-DestDec">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="ConstrT/bool">
    <!-- si el valor del atributo value es true se escribe un ! si no no se hace nada -->
    <xsl:if test="@value='True'">
        <strong>! </strong>
    </xsl:if>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Flecha">
    <span id="purple"><xsl:apply-templates select="node()[position()=1]"/></span>
    <span id="black"><strong>--> </strong></span>
    <span id="purple"><xsl:apply-templates select="node()[position()=2]"/></span>
</xsl:template>

<xsl:template match="Rec">
    <span id="red"><strong>REC </strong></span><!-- Simplemente escribimos la palabra REC en
negrita -->
    <xsl:apply-templates/><!-- Aplicamos plantillas al resto, esto hay que ponerlo siempre
-->
</xsl:template>

<!-- DATA -->
<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato">
    <xsl:apply-templates/>
    <br/>
</xsl:template>

<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato/string">
    <span id="blue"><strong>data </strong></span> <!-- la palabra data delante, -->
    <xsl:value-of select="string"/> <!-- despues el valor del string -->
    <xsl:apply-templates/> <!-- aplicamos plantillas al resto y escribimos un espacio al
final-->
</xsl:template>

<xsl:template
match="list-tuple4-string-list-string-list-string-list-AltDato/list-string[position() mod 2
!= 0]">
<!-- aqui la plantilla se aplica para los hijos list-string cuya posicion sea impar -->
    <xsl:apply-templates/> <!-- se llama al apply-templates antes, para que aplique las
plantillas a los hijos y que la arroba la ponga al final-->
    <xsl:if test="node()=text()"><span id="green"><strong>@ </strong></span>
    </xsl:if>
</xsl:template>

<xsl:template
match="list-tuple4-string-list-string-list-string-list-AltDato/list-string[position() mod 2 =
0]">
<!-- aqui la plantilla se aplica para los hijos list-string cuya posicion sea par -->
    <xsl:apply-templates/>
```

```
    <strong>= </strong><br/>
</xsl:template>

<!-- AltDato -->
<xsl:template
match="list-tuple4-string-list-string-list-string-list-AltDato/list-AltDato/ConstrA">
         <strong>| </strong> <!-- para los ConstrA escribimos
primero un | y delante espacios para tabular-->
    <xsl:apply-templates select="string[1]"/> <!-- aplicamos la plantilla para el primer hijo
string (:) -->
    <xsl:apply-templates select="list-ExpTipo"/> <!-- aplicamos la plantilla para el hijo
list-ExpTipo (a REC) -->
    <span id="green"><strong>@ </strong> <!-- metemos un simbolo arroba -->
    <xsl:apply-templates select="string[2]"/></span> <!-- aplicamos la plantilla para el
segundo hijo string (rho1) -->
    <br/><!-- salto de linea -->
</xsl:template>

<!-- DestDec -->
<xsl:template match="ConstrE-DestDec">
    <xsl:apply-templates select="string[1]"/>
    <xsl:apply-templates select="list-Exp-DestDec"/>
    <span id="green"><strong>@ </strong>
    <xsl:apply-templates select="string[2]"/></span>
</xsl:template>

<xsl:template match="VarE-DestDec">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="node()[position()=2]"/>
</xsl:template>

<xsl:template match="CopyE-DestDec">
    <xsl:apply-templates select="string[1]"/> <!-- Aplicamos la plantilla para el primer
string de CopyE -->
    <span id="green"><strong>@ </strong> <!-- Despues escribimos una arroba -->
    <xsl:apply-templates select="string[2]"/></span><br/><!-- Aplicamos la plantilla para el
segundo string de CopyE -->
</xsl:template>

<xsl:template match="ReuseE-DestDec">
    <xsl:apply-templates select="string"/>
    <strong>! </strong>
    <xsl:apply-templates select="node()[position()=2]"/>
</xsl:template>

<xsl:template match="AppE-DestDec">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-Exp-DestDec"/>
    <xsl:if test="count(list-string/string)>0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span></xsl:if>
    <xsl:apply-templates select="node()[position()=4]"/>
</xsl:template>

<xsl:template match="list-tuple2-Patron-DestDec-bool">
    <xsl:apply-templates/><strong>= </strong>
</xsl:template>

<xsl:template match="VarP-DestDec">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="node()[position()=2]"/>
</xsl:template>

<xsl:template
match="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-DestDec-bool-list-string-Der
-DestDec">
    <!--<xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-tuple2-Patron-DestDec-bool"/><strong>=</strong><br/>
    <xsl:apply-templates select="list-string"/>
    <xsl:apply-templates select="Simple-DestDec"/>-->
```

```
    <xsl:apply-templates/>
</xsl:template>

<xsl:template
match="DestInferenceProg/list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-DestDec-bo
ol-list-string-Der-DestDec/list-ExpTipo">
    <xsl:if test="following-sibling::string">
    <br/>
    <xsl:if test="count(node())>0">
    <strong>{ <span id="purple">
    <xsl:value-of select="following-sibling::string"/></span> <xsl:choose><xsl:when test=".
!= '1Sintipo'"> ::
<xsl:apply-templates/></xsl:when><xsl:otherwise> </xsl:otherwise></xsl:choose>}</strong>
<br/>
    </xsl:if>
    </xsl:if>
</xsl:template>

<xsl:template
match="LetE-DestDec/list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-DestDec-bool-li
st-string-Der-DestDec/list-ExpTipo">
    <xsl:if test="following-sibling::string">
    <xsl:if test="count(node())>0">
    <strong>{
    <xsl:value-of select="following-sibling::string"/><xsl:choose><xsl:when test=". !=
'1Sintipo'"> ::
<xsl:apply-templates/></xsl:when><xsl:otherwise> </xsl:otherwise></xsl:choose>}</strong>
<br/>
    </xsl:if>
    </xsl:if>
</xsl:template>

<xsl:template match="CaseE-DestDec">
    { <xsl:apply-templates select="node()[position()=3]"/>}<br/>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>case </strong></span>
    <xsl:apply-templates select="node()[position()=1]"/>
    <span id="blue"><strong>of</strong></span>
    <ul style="list-style-type:node;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple2-Patron-DestDec-Exp-DestDec"/></ul>
    </ul>
</xsl:template>

<xsl:template match="CaseE-DestDec/list-tuple2-Patron-DestDec-Exp-DestDec">
    <xsl:for-each select="child::node()">
        <xsl:if test="position() mod 2 != 0"><xsl:apply-templates select="."/><strong>-->
</strong></xsl:if>
        <xsl:if test="position() mod 2 = 0"><xsl:apply-templates select="."/>
        <xsl:if test="position() != last()"><br/></xsl:if></xsl:if>
    </xsl:for-each>
</xsl:template>

<xsl:template match="list-tuple2-Patron-DestDec-Exp-DestDec">
    <xsl:apply-templates/>
</xsl:template>

<!--<xsl:template
match="CaseE-DestDec/list-tuple2-Patron-DestDec-Exp-DestDec/node()[position() mod 2 =
0]/node()[position()=last()]">
    <xsl:apply-templates/><br/>
</xsl:template>

<xsl:template match="CaseE-DestDec/list-tuple2-Patron-DestDec-Exp-DestDec/node()[position()
mod 2 != 0]/node()[position()=last()]">
    <xsl:apply-templates/><strong>-> </strong>
</xsl:template>-->

<xsl:template match="CaseDE-DestDec">
    { <xsl:apply-templates select="node()[position()=3]"/>}<br/>
```

```
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>case! </strong></span>
    <xsl:apply-templates select="node()[position()=1]"/>
    <span id="blue"><strong>of</strong></span>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple2-Patron-DestDec-Exp-DestDec"/></ul>
    </ul>
</xsl:template>

<xsl:template match="CaseDE-DestDec/list-tuple2-Patron-DestDec-Exp-DestDec">
    <xsl:for-each select="child::node()">
        <xsl:if test="position() mod 2 != 0"><xsl:apply-templates select="."/><strong>-->
</strong></xsl:if>
        <xsl:if test="position() mod 2 = 0"><xsl:apply-templates select="."/>
        <xsl:if test="position() != last()"><br/></xsl:if></xsl:if>
    </xsl:for-each>
</xsl:template>

<!--<xsl:template
match="CaseDE-DestDec/list-tuple2-Patron-DestDec-Exp-DestDec/node()[position() mod 2 =
0]/node()[position()=last()]">
    <xsl:apply-templates/><br/>
</xsl:template>

<xsl:template match="CaseDE-DestDec/list-tuple2-Patron-DestDec-Exp-DestDec/node()[position()
mod 2 != 0]/node()[position()=last()]">
    <xsl:apply-templates/><strong>-> </strong>
</xsl:template>-->

<xsl:template match="LetE-DestDec">
    { <xsl:apply-templates select="node()[position()=3]"/>}<br/>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>let </strong></span>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="node()[position()=1]"/></ul>
    <span id="blue"><strong>in </strong></span>
    <xsl:apply-templates select="node()[position()=2]"/>
    </ul>
</xsl:template>

<!-- Der-Decor -->
<xsl:template match="Simple-DestDec">
    <xsl:apply-templates select="node()[position()=1]"/>
    <xsl:if
test="count(list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-DestDec-bool-list-strin
g-Der-DestDec/string)!=0"><br/>
    <ul style="margin-bottom:0;margin-top:0"><span id="blue"><strong>where</strong></span>
    <ul style="margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-DestDec-bool-list-string-De
r-DestDec"/></ul></ul>
    </xsl:if>
</xsl:template>

<xsl:template match="Simple-DestDec/ConstE-DestDec/ConstrT">
    <xsl:apply-templates/> <!--<br/><br/>-->
</xsl:template>

<xsl:template match="Guardado-DestDec">
    <ul style="margin-bottom:0;margin-top:0">
    <xsl:apply-templates select="node()[position()=1]"/></ul>
    <xsl:if
test="count(list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-DestDec-bool-list-strin
g-Der-DestDec/string)!=0">
    <ul style="margin-bottom:0;margin-top:0"><span id="blue"><strong>where</strong></span>
    <ul style="margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-DestDec-bool-list-string-De
r-DestDec"/></ul></ul>
    </xsl:if>
</xsl:template>
```

```
<xsl:template match="list-tuple2-Exp-DestDec-Exp-DestDec">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Guardado-DestDec/list-tuple2-Exp-DestDec-Exp-DestDec/AppE-DestDec">
    <xsl:if test="position()!=1"><br/></xsl:if>
    <strong>| </strong>
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-Exp-DestDec"/>
    <xsl:if test="count(list-string/string)>0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span></xsl:if>
    <strong>= </strong>
</xsl:template>

<xsl:template match="list-tuple2-Exp-DestDec-Exp-DestDec/node()[position() mod 2 !=
0]/node()[position()=last()]">
    <strong>| </strong><xsl:apply-templates/><strong>= </strong>
</xsl:template>

<xsl:template match="list-tuple2-Exp-DestDec-Exp-DestDec/node()[position() mod 2 =
0]/node()[position()=last()]">
    <xsl:apply-templates/><br/>
</xsl:template>

<!-- Variables y SharingMaps -->
<!-- ********************** -->
<xsl:template match="DD">
    <xsl:if test="count(list-string/string)>0">
        <ul style="margin-bottom:0;margin-top:0">DD <xsl:apply-templates/></ul>
    </xsl:if>
</xsl:template>

<xsl:template match="LetEDD">
    <xsl:if test="count(list-string/string)>0">
        <ul style="margin-bottom:0;margin-top:0">LetEDD <xsl:apply-templates/></ul>
    </xsl:if>
</xsl:template>

<xsl:template match="CaseEDD">
    <xsl:if test="count(list-list-string/list-string/string)>0">
        <ul style="margin-bottom:0;margin-top:0">CaseEDD <xsl:apply-templates/></ul>
    </xsl:if>
</xsl:template>

<xsl:template match="CaseDEDD">
    <xsl:if test="count(list-list-string/list-string/string)>0">
        <ul style="margin-bottom:0;margin-top:0">CaseDEDD <xsl:apply-templates/></ul>
    </xsl:if>
</xsl:template>

<!-- LetDD -->
<xsl:template match="LetEDD/list-string">
    <xsl:if test="count(string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>] </strong>
    </xsl:if>
</xsl:template>

<xsl:template match="LetEDD/list-string/string">
    <xsl:value-of select="."/>
    <xsl:if test="position() != last()">,</xsl:if> <!-- espacio en blanco -->
</xsl:template>

<!-- DD -->
<xsl:template match="DD/list-string">
    <xsl:if test="count(string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>] </strong>
    </xsl:if>
</xsl:template>
```

```
<xsl:template match="DD/list-string/string">
    <xsl:value-of select="."/>
    <xsl:if test="position() != last()">, </xsl:if>
</xsl:template>

<!-- CaseDEDD -->
<xsl:template match="CaseDEDD/list-list-string">
    <xsl:if test="count(list-string/string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>] </strong>
    </xsl:if>
</xsl:template>

<xsl:template match="CaseDEDD/list-list-string/list-string">
    <xsl:if test="count(string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>] </strong>
    </xsl:if>
</xsl:template>

<xsl:template match="CaseDEDD/list-list-string/list-string/string">
    <xsl:value-of select="."/>
    <xsl:if test="position() != last()">, </xsl:if>
</xsl:template>

<!-- CaseEDD -->
<xsl:template match="CaseEDD/list-list-string">
    <xsl:if test="count(list-string/string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>] </strong>
    </xsl:if>
</xsl:template>

<xsl:template match="CaseEDD/list-list-string/list-string">
    <xsl:if test="count(string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>] </strong>
    </xsl:if>
</xsl:template>

<xsl:template match="CaseEDD/list-list-string/list-string/string">
    <xsl:value-of select="."/>
    <xsl:if test="position() != last()">, </xsl:if>
</xsl:template>

<!-- SharingMaps -->
<xsl:template match="list-tuple2-string-list-string">
    <!--<strong>(</strong><xsl:apply-templates/><strong>)</strong>-->
    <ul style="margin-bottom:0;margin-top:0"><xsl:apply-templates/></ul>
</xsl:template>

<xsl:template match="list-tuple2-string-list-string/string">
    <xsl:if test="count(.)>0"><strong>(</strong><xsl:value-of select="."/>, </xsl:if>
</xsl:template>

<xsl:template match="list-tuple2-string-list-string/list-string">
    <xsl:if test="count(string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>])</strong><xsl:if test="position()
!= last()">,</xsl:if>
    </xsl:if>
</xsl:template>

<xsl:template match="list-tuple2-string-list-string/list-string/string">
    <xsl:value-of select="."/>
    <xsl:if test="position() != last()">, </xsl:if>
</xsl:template>

<xsl:template match="list-list-tuple2-string-list-string">
    <xsl:if test="count(list-tuple2-string-list-string/string)>0">
        <strong>[</strong><xsl:apply-templates
select="list-tuple2-string-list-string"/><strong>] </strong>
    </xsl:if>
```

```
</xsl:template>

</xsl:stylesheet>
```

```
<?xml version='1.0' encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:template match='/'>
  <!-- Esta es la plantilla para el elemento raiz -->
  <!-- Se pone la estructura de un documento html y dentro del cuerpo llamamos a
apply-templates para que aplique las plantillas a los hijos de la raiz -->
    <html>
    <style type="text/css">
      body {
        padding-left: 1em;
        color: black;
        background-color: #DBE5F1;
      }
      #green {
        color: green;
      }
      #blue {
        color: #000066;
      }
      #red {
        color: #C00000;
      }
      #yellow {
        color: #FF9900;
      }
      #purple {
        color: #7030A0;
      }
      #black {
        color: #000000;
      }
    </style>
    <body>
    <font face="Verdana" size="2">
        <xsl:apply-templates /> <!-- esto aplicaria a todos los hijos de la raiz su
correspondiente plantilla -->
    </font>
    </body>
    </html>
</xsl:template>


<!-- Plantilla para 4.TYP y 5.DSG -->
<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato">
    <xsl:apply-templates/>
</xsl:template>

<!-- Strings -->
<xsl:template match="list-string">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="string">
    <!-- Para los nodos string, simplemente escribimos su valor, y metemos un espacio -->
    <xsl:if test=". != 'PAT'">
    <xsl:value-of select="."/> <!-- espacio en blanco -->
    </xsl:if>
</xsl:template>

<!-- Lit -->
<xsl:template match="int">
    <!-- Para los nodos int, simplemente escribimos su valor, y metemos un espacio -->
    <xsl:value-of select="@value"/> <!-- la arroba es para indicar que es un atributo -->
    <xsl:apply-templates/> 
</xsl:template>

<xsl:template match="bool">
    <!-- Para los nodos bool, simplemente escribimos su valor, y metemos un espacio -->
    <xsl:value-of select="@value"/>
```

```
    <xsl:apply-templates/> 
</xsl:template>

<!-- ExpTipo -->
<xsl:template match="ConstrT">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-ExpTipo"/>
    <xsl:if test="count(list-string/string)!=0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span></xsl:if>
</xsl:template>

<xsl:template match="list-Exp-ExpTipo">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="ConstrT/bool">
    <!-- si el valor del atributo value es true se escribe un ! si no no se hace nada -->
    <xsl:if test="@value='True'">
        <strong>! </strong>
    </xsl:if>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Flecha">
    <span id="red"><xsl:apply-templates select="node()[position()=1]"/></span>
    <span id="black"><strong>--> </strong></span>
    <span id="red"><xsl:apply-templates select="node()[position()=2]"/></span>
</xsl:template>

<xsl:template match="Rec">
    <span id="red"><strong>REC </strong></span><!-- Simplemente escribimos la palabra REC en
negrita -->
    <xsl:apply-templates/><!-- Aplicamos plantillas al resto, esto hay que ponerlo siempre
-->
</xsl:template>

<!-- DATA -->
<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato">
    <xsl:apply-templates/>
    <br/>
</xsl:template>

<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato/string">
    <span id="blue"><strong>data </strong></span> <!-- la palabra data delante, -->
    <xsl:value-of select="string"/> <!-- despues el valor del string -->
    <xsl:apply-templates/> <!-- aplicamos plantillas al resto y escribimos un espacio al
final-->
</xsl:template>

<xsl:template
match="list-tuple4-string-list-string-list-string-list-AltDato/list-string[position() mod 2
!= 0]">
<!-- aqui la plantilla se aplica para los hijos list-string cuya posicion sea impar -->
    <xsl:apply-templates/> <!-- se llama al apply-templates antes, para que aplique las
plantillas a los hijos y que la arroba la ponga al final-->
    <xsl:if test="node()=text()"><strong>@ </strong>
    </xsl:if>
</xsl:template>

<xsl:template
match="list-tuple4-string-list-string-list-string-list-AltDato/list-string[position() mod 2 =
0]">
<!-- aqui la plantilla se aplica para los hijos list-string cuya posicion sea par -->
    <xsl:apply-templates/>
    <strong>= </strong><br/>
</xsl:template>

<!-- AltDato -->
<xsl:template
```

```
match="list-tuple4-string-list-string-list-string-list-AltDato/list-AltDato/ConstrA">
        <strong>| </strong> <!-- para los ConstrA escribimos
primero un | y delante espacios para tabular-->
    <xsl:apply-templates select="string[1]"/> <!-- aplicamos la plantilla para el primer hijo
string (:) -->
    <xsl:apply-templates select="list-ExpTipo"/> <!-- aplicamos la plantilla para el hijo
list-ExpTipo (a REC) -->
    <span id="green"><strong>@ </strong> <!-- metemos un simbolo arroba -->
    <xsl:apply-templates select="string[2]"/></span> <!-- aplicamos la plantilla para el
segundo hijo string (rho1) -->
    <br/><!-- salto de linea -->
</xsl:template>

<!-- Exp-ExpTipo -->
<xsl:template match="ConstE-ExpTipo/ConstrT/string">
    { <xsl:value-of select="."/> }
</xsl:template>

<xsl:template match="ConstrE-ExpTipo">
    <xsl:apply-templates select="string[1]"/>
    <xsl:apply-templates select="list-Exp-ExpTipo"/>
    <span id="green"><strong>@ </strong>
    <xsl:apply-templates select="string[2]"/></span>
</xsl:template>

<xsl:template match="VarE-ExpTipo">
    <xsl:apply-templates select="string"/>
    { <xsl:apply-templates select="node()[position()=2]"/>}
</xsl:template>

<xsl:template match="CopyE-ExpTipo">
    <xsl:apply-templates select="string[1]"/> <!-- Aplicamos la plantilla para el primer
string de CopyE -->
    <span id="green"><strong>@ </strong> <!-- Despues escribimos una arroba -->
    <xsl:apply-templates select="string[2]"/></span><br/> <!-- Aplicamos la plantilla para el
segundo string de CopyE -->
</xsl:template>

<xsl:template match="ReuseE-ExpTipo">
    <xsl:apply-templates select="string"/>
    <strong>! </strong>
    <xsl:apply-templates select="node()[position()=2]"/>
</xsl:template>

<xsl:template match="AppE-ExpTipo">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-Exp-ExpTipo"/>
    <xsl:if test="count(list-string/string)>0"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></xsl:if>
    <xsl:apply-templates select="node()[position()=4]"/>
</xsl:template>

<xsl:template match="list-tuple2-Patron-ExpTipo-bool">
    <xsl:apply-templates/><strong>= </strong>
</xsl:template>

<xsl:template match="VarP-ExpTipo">
    <xsl:apply-templates select="string"/>
    { <xsl:apply-templates select="node()[position()=2]"/>}
</xsl:template>

<xsl:template
match="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-ExpTipo-bool-list-string-Der
-ExpTipo">
    <!--<xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-tuple2-Patron-ExpTipo-bool"/><strong>=</strong><br/>
    <xsl:apply-templates select="list-string"/>
    <xsl:apply-templates select="Simple-ExpTipo"/>-->
    <xsl:apply-templates/>
```

```
</xsl:template>

<xsl:template
match="TypedProg/list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-ExpTipo-bool-list-
string-Der-ExpTipo/list-ExpTipo">
    <xsl:if test="following-sibling::string">
    <br/>
    <xsl:if test="count(node())>0">
    <strong>{ <span id="red"><xsl:value-of select="following-sibling::string"/></span> ::
<xsl:apply-templates/>}</strong><br/>
    </xsl:if>
    </xsl:if>
</xsl:template>

<xsl:template
match="LetE-ExpTipo/list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-ExpTipo-bool-li
st-string-Der-ExpTipo/list-ExpTipo">
    <xsl:if test="following-sibling::string">
    <xsl:if test="count(node())>0">
    <strong>{ <xsl:value-of select="following-sibling::string"/> ::
<xsl:apply-templates/>}</strong><br/>
    </xsl:if>
    </xsl:if>
</xsl:template>

<xsl:template match="CaseE-ExpTipo">
    { <xsl:apply-templates select="node()[position()=3]"/>}
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>case </strong></span>
    <xsl:apply-templates select="node()[position()=1]"/>
    <span id="blue"><strong>of</strong></span>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple2-Patron-ExpTipo-Exp-ExpTipo"/></ul>
    </ul>
</xsl:template>

<!--<xsl:template match="list-tuple2-Patron-ExpTipo-Exp-ExpTipo">
    <xsl:for-each select="node()[position() mod 2 != 0]">
        <xsl:apply-templates select="."/><strong>-></strong>
    </xsl:for-each>
    <xsl:for-each select="node()[position() mod 2 = 0]">
        <xsl:apply-templates select="."/><br/>
    </xsl:for-each>
</xsl:template>-->

<xsl:template match="CaseE-ExpTipo/list-tuple2-Patron-ExpTipo-Exp-ExpTipo">
    <xsl:for-each select="child::node()">
        <xsl:if test="position() mod 2 != 0"><xsl:apply-templates select="."/><strong>-->
</strong></xsl:if>
        <xsl:if test="position() mod 2 = 0"><xsl:apply-templates select="."/>
        <xsl:if test="position() != last()"><br/></xsl:if></xsl:if>
    </xsl:for-each>
    <!--<xsl:apply-templates select="node()[1]"/><strong>-> </strong><xsl:apply-templates
select="node()[2]"/><br/>
    <xsl:apply-templates select="node()[3]"/><strong>-> </strong><xsl:apply-templates
select="node()[4]"/><br/>
    <xsl:apply-templates select="node()[5]"/><strong>-> </strong><xsl:apply-templates
select="node()[6]"/>-->
</xsl:template>

<!--<xsl:template
match="CaseE-ExpTipo/list-tuple2-Patron-ExpTipo-Exp-ExpTipo/ConstE-ExpTipo">
    <xsl:apply-templates/><br/>
</xsl:template>

<xsl:template match="CaseE-ExpTipo/list-tuple2-Patron-ExpTipo-Exp-ExpTipo/node()[position()
mod 2 != 0]">
    <xsl:apply-templates/><strong>-> </strong>
</xsl:template>-->
```

```
<xsl:template match="CaseDE-ExpTipo">
    { <xsl:apply-templates select="node()[position()=3]"/>}<br/>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>case! </strong></span>
    <xsl:apply-templates select="node()[position()=1]"/>
    <span id="blue"><strong>of</strong></span>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple2-Patron-ExpTipo-Exp-ExpTipo"/></ul>
    </ul>
</xsl:template>

<xsl:template match="CaseDE-ExpTipo/list-tuple2-Patron-ExpTipo-Exp-ExpTipo">
    <xsl:for-each select="child::node()">
        <xsl:if test="position() mod 2 != 0"><xsl:apply-templates select="."/><strong>-->
</strong></xsl:if>
        <xsl:if test="position() mod 2 = 0"><xsl:apply-templates select="."/>
        <xsl:if test="position() != last()"><br/></xsl:if></xsl:if>
    </xsl:for-each>
</xsl:template>

<!--<xsl:template
match="CaseDE-ExpTipo/list-tuple2-Patron-ExpTipo-Exp-ExpTipo/ConstE-ExpTipo">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="CaseDE-ExpTipo/list-tuple2-Patron-ExpTipo-Exp-ExpTipo/node()[position()
mod 2 != 0]/node()[position()=last()]">
    <xsl:apply-templates/><strong>-> </strong>
</xsl:template>-->

<xsl:template match="LetE-ExpTipo">
    { <xsl:apply-templates select="node()[position()=3]"/>}<br/>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>let </strong></span>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="node()[position()=1]"/></ul>
    <span id="blue"><strong>in </strong></span>
    <xsl:apply-templates select="node()[position()=2]"/>
    </ul>
</xsl:template>

<!-- Der-Decor -->
<xsl:template match="Simple-ExpTipo">
    <xsl:apply-templates select="node()[position()=1]"/>
    <xsl:if
test="count(list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-ExpTipo-bool-list-strin
g-Der-ExpTipo/string)!=0"><br/>
    <ul style="margin-bottom:0;margin-top:0"><span id="blue"><strong>where</strong></span>
    <ul style="margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-ExpTipo-bool-list-string-De
r-ExpTipo"/></ul></ul>
    </xsl:if>
</xsl:template>

<xsl:template match="Simple-ExpTipo/ConstE-ExpTipo/ConstrT">
    <xsl:apply-templates/> <!--<br/><br/>-->
</xsl:template>

<xsl:template match="Guardado-ExpTipo">
    <ul style="margin-bottom:0;margin-top:0">
    <xsl:apply-templates select="node()[position()=1]"/></ul>
    <xsl:if
test="count(list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-ExpTipo-bool-list-strin
g-Der-ExpTipo/string)!=0">
    <ul style="margin-bottom:0;margin-top:0"><span id="blue"><strong>where</strong></span>
    <ul style="margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-ExpTipo-bool-list-string-De
r-ExpTipo"/></ul></ul>
```

```
    </xsl:if>
</xsl:template>

<!--<xsl:template match="list-tuple2-Exp-ExpTipo-Exp-ExpTipo">
    <xsl:apply-templates/>
</xsl:template>-->

<xsl:template match="Guardado-ExpTipo/list-tuple2-Exp-ExpTipo-Exp-ExpTipo/AppE-ExpTipo">
    <strong>| </strong>
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-Exp-ExpTipo"/>
    <xsl:if test="count(list-string/string)>0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span></xsl:if>
    <strong>= </strong>
</xsl:template>

<!--<xsl:template match="list-tuple2-Exp-ExpTipo-Exp-ExpTipo/node()[position() mod 2 != 0]">
    <br/><strong>| </strong><xsl:apply-templates/><strong>= </strong>
</xsl:template>-->

<xsl:template match="list-tuple2-Exp-ExpTipo-Exp-ExpTipo/node()[position() mod 2 = 0]">
    <xsl:apply-templates/><br/>
</xsl:template>

</xsl:stylesheet>
```

```
<?xml version='1.0' encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:template match='/'>
  <!-- Esta es la plantilla para el elemento raiz -->
  <!-- Se pone la estructura de un documento html y dentro del cuerpo llamamos a
apply-templates para que aplique las plantillas a los hijos de la raiz -->
    <html>
    <style type="text/css">
      body {
        padding-left: 1em;
        color: black;
        background-color: #DBE5F1;
      }
      #green {
        color: green;
      }
      #blue {
        color: #000066;
      }
      #red {
        color: #C00000;
      }
      #yellow {
        color: #FF9900;
      }
      #purple {
        color: #7030A0;
      }
      #black {
        color: #000000;
      }
    </style>
    <body>
    <font face="Verdana" size="2">
        <xsl:apply-templates /> <!-- esto aplicaria a todos los hijos de la raiz su
correspondiente plantilla -->
    </font>
    </body>
    </html>
</xsl:template>


<!-- Plantilla para 8.SHA -->
<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato">
    <xsl:apply-templates/>
</xsl:template>

<!-- Strings -->
<xsl:template match="list-string">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="string">
    <!-- Para los nodos string, simplemente escribimos su valor, y metemos un espacio -->
    <xsl:if test=". != 'PAT'">
    <xsl:value-of select="."/> <!-- espacio en blanco -->
    </xsl:if>
</xsl:template>

<!-- Lit -->
<xsl:template match="int">
    <!-- Para los nodos int, simplemente escribimos su valor, y metemos un espacio -->
    <xsl:value-of select="@value"/> <!-- la arroba es para indicar que es un atributo
-->
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="bool">
    <!-- Para los nodos bool, simplemente escribimos su valor, y metemos un espacio -->
```

```
    <xsl:value-of select="@value"/> 
    <xsl:apply-templates/>
</xsl:template>

<!-- tuple2-ExpTipo-SharingDec -->
<xsl:template match="ConstE-tuple2-ExpTipo-SharingDec/ConstrT/string">
    { <xsl:value-of select="."/> }
</xsl:template>

<xsl:template match="ConstrT">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-ExpTipo"/>
    <xsl:if test="count(list-string/string)>0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span></xsl:if>
</xsl:template>

<xsl:template match="list-Exp-tuple2-ExpTipo-SharingDec">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="ConstrT/bool">
    <!-- si el valor del atributo value es true se escribe un ! si no no se hace nada -->
    <xsl:if test="@value='True'">
        <strong>! </strong>
    </xsl:if>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Flecha">
    <span id="yellow"><xsl:apply-templates select="node()[position()=1]"/></span>
    <span id="black"><strong>--> </strong></span>
    <span id="yellow"><xsl:apply-templates select="node()[position()=2]"/></span>
</xsl:template>

<xsl:template match="Rec">
    <span id="red"><strong>REC </strong></span><!-- Simplemente escribimos la palabra REC en
negrita -->
    <xsl:apply-templates/><!-- Aplicamos plantillas al resto, esto hay que ponerlo siempre
-->
</xsl:template>

<!-- DATA -->
<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato">
    <xsl:apply-templates/>
    <br/>
</xsl:template>

<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato/string">
    <span id="blue"><strong>data </strong></span> <!-- la palabra data delante, -->
    <xsl:value-of select="string"/> <!-- despues el valor del string -->
    <xsl:apply-templates/> <!-- aplicamos plantillas al resto y escribimos un espacio al
final-->
</xsl:template>

<xsl:template
match="list-tuple4-string-list-string-list-string-list-AltDato/list-string[position() mod 2
!= 0]">
<!-- aqui la plantilla se aplica para los hijos list-string cuya posicion sea impar -->
    <xsl:apply-templates/> <!-- se llama al apply-templates antes, para que aplique las
plantillas a los hijos y que la arroba la ponga al final-->
    <xsl:if test="node()=text()"><span id="green"><strong>@ </strong></span>
    </xsl:if>
</xsl:template>

<xsl:template
match="list-tuple4-string-list-string-list-string-list-AltDato/list-string[position() mod 2 =
0]">
<!-- aqui la plantilla se aplica para los hijos list-string cuya posicion sea par -->
    <xsl:apply-templates/>
```

```
    <strong>= </strong><br/>
</xsl:template>

<!-- AltDato -->
<xsl:template
match="list-tuple4-string-list-string-list-string-list-AltDato/list-AltDato/ConstrA">
         <strong>| </strong> <!-- para los ConstrA escribimos
primero un | y delante espacios para tabular-->
    <xsl:apply-templates select="string[1]"/> <!-- aplicamos la plantilla para el primer hijo
string (:) -->
    <xsl:apply-templates select="list-ExpTipo"/> <!-- aplicamos la plantilla para el hijo
list-ExpTipo (a REC) -->
    <span id="green"><strong>@ </strong> <!-- metemos un simbolo arroba -->
    <xsl:apply-templates select="string[2]"/></span> <!-- aplicamos la plantilla para el
segundo hijo string (rho1) -->
    <br/><!-- salto de linea -->
</xsl:template>

<!-- Exp-tuple2-ExpTipo-SharingDec -->
<xsl:template match="ConstrE-tuple2-ExpTipo-SharingDec">
    <xsl:apply-templates select="string[1]"/>
    <xsl:apply-templates select="list-Exp-tuple2-ExpTipo-SharingDec"/>
    <span id="green"><strong>@ </strong>
    <xsl:apply-templates select="string[2]"/></span>
</xsl:template>

<xsl:template match="VarE-tuple2-ExpTipo-SharingDec">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="node()[position()=2]"/>
    <xsl:apply-templates select="node()[position()=3]"/>
</xsl:template>

<xsl:template match="CopyE-tuple2-ExpTipo-SharingDec">
    <xsl:apply-templates select="string[1]"/> <!-- Aplicamos la plantilla para el primer
string de CopyE -->
    <span id="green"><strong>@ </strong> <!-- Despues escribimos una arroba -->
    <xsl:apply-templates select="string[2]"/></span>
    <xsl:apply-templates select="node()[position()=3]"/>
    <xsl:apply-templates select="node()[position()=4]"/><br/>
</xsl:template>

<xsl:template match="ReuseE-tuple2-ExpTipo-SharingDec">
    <xsl:apply-templates select="string"/>
    <strong>! </strong>
    <xsl:apply-templates select="node()[position()=2]"/>
    <xsl:apply-templates select="node()[position()=3]"/>
</xsl:template>

<xsl:template match="AppE-tuple2-ExpTipo-SharingDec">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-Exp-tuple2-ExpTipo-SharingDec"/>
    <xsl:if test="count(list-string/string)>0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span></xsl:if>
    <xsl:apply-templates select="node()[position()=4]"/>
    <xsl:apply-templates select="node()[position()=5]"/>
</xsl:template>

<xsl:template match="list-tuple2-Patron-tuple2-ExpTipo-SharingDec-bool">
    <xsl:apply-templates/><strong>= </strong>
</xsl:template>

<xsl:template match="VarP-tuple2-ExpTipo-SharingDec">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="node()[position()=2]"/>
    <xsl:apply-templates select="node()[position()=3]"/>
</xsl:template>

<xsl:template
match="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-tuple2-ExpTipo-SharingDec-bo
```

```
ol-list-string-Der-tuple2-ExpTipo-SharingDec">
    <!--<xsl:apply-templates select="string"/>
    <xsl:apply-templates
select="list-tuple2-Patron-tuple2-ExpTipo-SharingDec-bool"/><strong>=</strong><br/>
    <xsl:apply-templates select="list-string"/>
    <xsl:apply-templates select="Simple-tuple2-ExpTipo-SharingDec"/>-->
    <xsl:apply-templates/>
</xsl:template>

<xsl:template
match="SharingProg/list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-tuple2-ExpTipo-S
haringDec-bool-list-string-Der-tuple2-ExpTipo-SharingDec/list-ExpTipo">
    <xsl:if test="following-sibling::string">
    <br/>
    <xsl:if test="count(node())>0">
    <strong>{ <span id="yellow">
    <xsl:value-of select="following-sibling::string"/></span> <xsl:choose><xsl:when test=".
!= '1Sintipo'"> ::
<xsl:apply-templates/></xsl:when><xsl:otherwise> </xsl:otherwise></xsl:choose>}</strong>
<br/>
    </xsl:if>
    </xsl:if>
</xsl:template>

<xsl:template
match="LetE-tuple2-ExpTipo-SharingDec/list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patr
on-tuple2-ExpTipo-SharingDec-bool-list-string-Der-tuple2-ExpTipo-SharingDec/list-ExpTipo">
    <xsl:if test="following-sibling::string">
    <xsl:if test="count(node())>0">
    <strong>{
    <xsl:value-of select="following-sibling::string"/> <xsl:choose><xsl:when test=". !=
'1Sintipo'"> ::
<xsl:apply-templates/></xsl:when><xsl:otherwise> </xsl:otherwise></xsl:choose>}</strong>
<br/>
    </xsl:if>
    </xsl:if>
</xsl:template>

<xsl:template match="CaseE-tuple2-ExpTipo-SharingDec">
    { <xsl:apply-templates select="node()[position()=3]"/><xsl:if
test="count(node()[position()=4]/node()/node()/node())!=0">, <xsl:apply-templates
select="node()[position()=4]"/></xsl:if>}<br/>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>case </strong></span>
    <xsl:apply-templates select="node()[position()=1]"/>
    <span id="blue"><strong>of</strong></span>
    <ul style="list-style-type:node;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple2-Patron-tuple2-ExpTipo-SharingDec-Exp-tuple2-ExpTipo-SharingDec"/></ul>
    </ul>
</xsl:template>

<xsl:template
match="CaseE-tuple2-ExpTipo-SharingDec/list-tuple2-Patron-tuple2-ExpTipo-SharingDec-Exp-tuple
2-ExpTipo-SharingDec">
    <xsl:for-each select="child::node()">
        <xsl:if test="position() mod 2 != 0"><xsl:apply-templates select="."/><strong>-->
</strong></xsl:if>
        <xsl:if test="position() mod 2 = 0"><xsl:apply-templates select="."/>
        <xsl:if test="position() != last()"><br/></xsl:if></xsl:if>
    </xsl:for-each>
</xsl:template>

<xsl:template
match="list-tuple2-Patron-tuple2-ExpTipo-SharingDec-Exp-tuple2-ExpTipo-SharingDec">
    <xsl:apply-templates/>
</xsl:template>

<!--<xsl:template
match="CaseE-tuple2-ExpTipo-SharingDec/list-tuple2-Patron-tuple2-ExpTipo-SharingDec-Exp-tuple
```

```
2-ExpTipo-SharingDec/node()[position() mod 2 = 0]/node()[position()=last()]">
    <xsl:apply-templates/><br/>
</xsl:template>

<xsl:template
match="CaseE-tuple2-ExpTipo-SharingDec/list-tuple2-Patron-tuple2-ExpTipo-SharingDec-Exp-tuple
2-ExpTipo-SharingDec/node()[position() mod 2 != 0]/node()[position()=last()]">
    <xsl:apply-templates/><strong>-> </strong>
</xsl:template>-->

<xsl:template match="CaseDE-tuple2-ExpTipo-SharingDec">
    { <xsl:apply-templates select="node()[position()=3]"/>, <xsl:apply-templates
select="node()[position()=4]"/>}<br/>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>case! </strong></span>
    <xsl:apply-templates select="node()[position()=1]"/>
    <span id="blue"><strong>of</strong></span>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple2-Patron-tuple2-ExpTipo-SharingDec-Exp-tuple2-ExpTipo-SharingDec"/></ul>
    </ul>
</xsl:template>

<xsl:template
match="CaseDE-tuple2-ExpTipo-SharingDec/list-tuple2-Patron-tuple2-ExpTipo-SharingDec-Exp-tupl
e2-ExpTipo-SharingDec">
    <xsl:for-each select="child::node()">
        <xsl:if test="position() mod 2 != 0"><xsl:apply-templates select="."/><strong>-->
</strong></xsl:if>
        <xsl:if test="position() mod 2 = 0"><xsl:apply-templates select="."/>
        <xsl:if test="position() != last()"><br/></xsl:if></xsl:if>
    </xsl:for-each>
</xsl:template>

<!--<xsl:template
match="CaseDE-tuple2-ExpTipo-SharingDec/list-tuple2-Patron-tuple2-ExpTipo-SharingDec-Exp-tupl
e2-ExpTipo-SharingDec/node()[position() mod 2 = 0]/node()[position()=last()]">
    <xsl:apply-templates/><br/>
</xsl:template>

<xsl:template
match="CaseDE-tuple2-ExpTipo-SharingDec/list-tuple2-Patron-tuple2-ExpTipo-SharingDec-Exp-tupl
e2-ExpTipo-SharingDec/node()[position() mod 2 != 0]/node()[position()=last()]">
    <xsl:apply-templates/><strong>-> </strong>
</xsl:template>-->

<xsl:template match="LetE-tuple2-ExpTipo-SharingDec">
    { <xsl:apply-templates select="node()[position()=3]"/>, <xsl:apply-templates
select="node()[position()=4]"/>}<br/>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>let </strong></span>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="node()[position()=1]"/></ul>
    <span id="blue"><strong>in </strong></span>
    <xsl:apply-templates select="node()[position()=2]"/>
    </ul>
</xsl:template>

<!-- Der-Decor -->
<xsl:template match="Simple-tuple2-ExpTipo-SharingDec">
    <xsl:apply-templates select="node()[position()=1]"/>
    <xsl:if
test="count(list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-tuple2-ExpTipo-SharingD
ec-bool-list-string-Der-tuple2-ExpTipo-SharingDec/string)!=0"><br/>
    <ul style="margin-bottom:0;margin-top:0"><span id="blue"><strong>where</strong></span>
    <ul style="margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-tuple2-ExpTipo-SharingDec-b
ool-list-string-Der-tuple2-ExpTipo-SharingDec"/></ul></ul>
    </xsl:if>
</xsl:template>
```

```
<xsl:template
match="Simple-tuple2-ExpTipo-SharingDec/ConstE-tuple2-ExpTipo-SharingDec/ConstrT">
    <xsl:apply-templates/> <!--<br/><br/>-->
</xsl:template>

<xsl:template match="Guardado-tuple2-ExpTipo-SharingDec">
    <ul style="margin-bottom:0;margin-top:0">
    <xsl:apply-templates select="node()[position()=1]"/></ul>
    <xsl:if
test="count(list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-tuple2-ExpTipo-SharingD
ec-bool-list-string-Der-tuple2-ExpTipo-SharingDec/string)!=0">
    <ul style="margin-bottom:0;margin-top:0"><span id="blue"><strong>where</strong></span>
    <ul style="margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-tuple2-ExpTipo-SharingDec-b
ool-list-string-Der-tuple2-ExpTipo-SharingDec"/></ul></ul>
    </xsl:if>
</xsl:template>

<xsl:template
match="list-tuple2-Exp-tuple2-ExpTipo-SharingDec-Exp-tuple2-ExpTipo-SharingDec">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template
match="Guardado-tuple2-ExpTipo-SharingDec/list-tuple2-Exp-tuple2-ExpTipo-SharingDec-Exp-tuple
2-ExpTipo-SharingDec/AppE-tuple2-ExpTipo-SharingDec">
    <xsl:if test="position()!=1"><br/></xsl:if>
    <strong>| </strong>
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-Exp-tuple2-ExpTipo-SharingDec"/>
    <xsl:if test="count(list-string/string)>0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span></xsl:if>
    <strong>= </strong>
</xsl:template>

<xsl:template
match="list-tuple2-Exp-tuple2-ExpTipo-SharingDec-Exp-tuple2-ExpTipo-SharingDec/node()[positio
n() mod 2 != 0]/node()[position()=last()]">
    <strong>| </strong><xsl:apply-templates/><strong>= </strong>
</xsl:template>

<xsl:template
match="list-tuple2-Exp-tuple2-ExpTipo-SharingDec-Exp-tuple2-ExpTipo-SharingDec/node()[positio
n() mod 2 = 0]/node()[position()=last()]">
    <xsl:apply-templates/><br/>
</xsl:template>

<!-- Variables y SharingMaps -->
<!-- ********************** -->
<xsl:template match="ShDecReuse">
    <xsl:if test="count(list-string/string)>0">
        <ul style="margin-bottom:0;margin-top:0">ShDecReuse <xsl:apply-templates/></ul>
    </xsl:if>
</xsl:template>

<xsl:template match="ShDecApp">
    <xsl:if test="count(list-list-string/list-string/string)>0">
        <ul style="margin-bottom:0;margin-top:0">ShDecApp <xsl:apply-templates/></ul>
    </xsl:if>
</xsl:template>

<xsl:template match="ShDecLet">
    <xsl:if test="count(list-tuple2-string-list-string/string)>0">
        <ul style="margin-bottom:0;margin-top:0">ShDecLet <xsl:apply-templates/></ul>
    </xsl:if>
</xsl:template>

<xsl:template match="ShDecCase">
```

```
    <xsl:if
test="count(list-list-tuple2-string-list-string/list-tuple2-string-list-string/string)>0">
        <ul style="margin-bottom:0;margin-top:0">ShDecCase <xsl:apply-templates/></ul>
    </xsl:if>
</xsl:template>

<xsl:template match="ShDecCaseD">
    <xsl:if
test="count(list-list-tuple2-string-list-string/list-tuple2-string-list-string/string)>0">
        <ul style="margin-bottom:0;margin-top:0">ShDecCaseD <xsl:apply-templates/></ul>
    </xsl:if>
</xsl:template>

<!-- ShDecReuse -->
<xsl:template match="ShDecReuse/list-string">
    <xsl:if test="count(string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>] </strong>
    </xsl:if>
</xsl:template>

<xsl:template match="ShDecReuse/list-string/string">
    <xsl:value-of select="."/>
    <xsl:if test="position() != last()">,</xsl:if> <!-- espacio en blanco -->
</xsl:template>

<!-- ShDecApp -->
<xsl:template match="ShDecApp/list-list-string">
    <xsl:if test="count(list-string/string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>] </strong>
    </xsl:if>
</xsl:template>

<xsl:template match="ShDecApp/list-list-string/list-string">
    <xsl:if test="count(string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>] </strong>
    </xsl:if>
</xsl:template>

<xsl:template match="ShDecApp/list-list-string/list-string/string">
    <xsl:value-of select="."/>
    <xsl:if test="position() != last()">, </xsl:if>
</xsl:template>

<!-- ShDecCaseD -->
<xsl:template match="ShDecCaseD/list-string">
    <xsl:if test="count(string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>] </strong>
    </xsl:if>
</xsl:template>

<xsl:template match="ShDecCaseD/list-list-string/list-string/string">
    <xsl:value-of select="."/>
    <xsl:if test="position() != last()">, </xsl:if>
</xsl:template>

<!-- SharingMaps -->
<xsl:template match="list-tuple2-string-list-string">
    <!--<strong>(</strong><xsl:apply-templates/><strong>)</strong>-->
    <ul style="margin-bottom:0;margin-top:0"><xsl:apply-templates/></ul>
</xsl:template>

<xsl:template match="list-tuple2-string-list-string/string">
    <xsl:if test="count(.)>0"><strong>(</strong><xsl:value-of select="."/>, </xsl:if>
</xsl:template>

<xsl:template match="list-tuple2-string-list-string/list-string">
    <xsl:if test="count(string)>0">
        <strong>[</strong><xsl:apply-templates/><strong>])</strong><xsl:if test="position()
!= last()">,</xsl:if>
```

```
    </xsl:if>
</xsl:template>

<xsl:template match="list-tuple2-string-list-string/list-string/string">
    <xsl:value-of select="."/>
    <xsl:if test="position() != last()">, </xsl:if>
</xsl:template>

<xsl:template match="list-list-tuple2-string-list-string">
    <xsl:if test="count(list-tuple2-string-list-string/string)>0">
        <strong>[</strong><xsl:apply-templates
select="list-tuple2-string-list-string"/><strong>] </strong>
    </xsl:if>
</xsl:template>

</xsl:stylesheet>
```

```
<?xml version='1.0' encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:template match='/'>
  <!-- Esta es la plantilla para el elemento raiz -->
  <!-- Se pone la estructura de un documento html y dentro del cuerpo llamamos a
apply-templates para que aplique las plantillas a los hijos de la raiz -->
    <html>
    <style type="text/css">
      body {
        padding-left: 1em;
        color: black;
        background-color: #DBE5F1;
      }
    </style>
    <body>
    <font face="Verdana" size="2">
        <xsl:apply-templates /> <!-- esto aplicaria a todos los hijos de la raiz su
correspondiente plantilla -->
    </font>
    </body>
    </html>
</xsl:template>

<!-- LEX -->
<xsl:template match="AlexPn">
    (lin:<xsl:apply-templates select="int[2]"/>) (col:<xsl:apply-templates
select="int[3]"/>)<br/>
</xsl:template>

<xsl:template match="AlexPn/int">
    <xsl:value-of select="@value"/>
</xsl:template>

<xsl:template match="string">
    (<xsl:value-of select="."/>)
</xsl:template>

<xsl:template match="int">
    (<xsl:value-of select="@value"/>)
</xsl:template>

<xsl:template match="BraIzq/AlexPn">
    (lin:<xsl:apply-templates select="int[2]"/>)(col:<xsl:apply-templates
select="int[3]"/>)<br/>
</xsl:template>

<xsl:template match="BraDer/AlexPn">
    (lin:<xsl:apply-templates select="int[2]"/>)(col:<xsl:apply-templates
select="int[3]"/>)<br/>
</xsl:template>

<xsl:template match="BraIzq">
    <strong>BraIzq</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="BraDer">
    <strong>BraDer</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="ConstBool">
    <strong>ConstBool</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Self">
    <strong>Self</strong>
    <xsl:apply-templates/>
```

```
</xsl:template>

<xsl:template match="Subrayado">
    <strong>Subrayado</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="ListaVacia">
    <strong>ListaVacia</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="CorDer">
    <strong>CorDer</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="CorIzq">
    <strong>CorIzq</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Coma">
    <strong>Coma</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="ParDer">
    <strong>ParDer</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="ParIzq">
    <strong>ParIzq</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="SemiColon">
    <strong>SemiColon</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Def">
    <strong>Def</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="At">
    <strong>At</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Desigual">
    <strong>Desigual</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Igual">
    <strong>Igual</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="FlechaDoble">
    <strong>FlechaDoble</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="list-Token/Flecha">
    <strong>Flecha</strong>
```

```
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Excl">
    <strong>Excl</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Guarda">
    <strong>Guarda</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="In">
    <strong>In</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Of">
    <strong>Of</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Data">
    <strong>Data</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Case">
    <strong>Case</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Where">
    <strong>Where</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Let">
    <strong>Let</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Include">
    <strong>Include</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Cadena">
    <strong>Cadena</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Reservada4">
    <strong>Reservada4</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="ConstNum">
    <strong>ConstNum</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="ConstrTok">
    <strong>ConstrTok</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="CInfija">
```

```
    <strong>CInfija</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Id">
    <strong>Id</strong>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Op">
    <strong>Op</strong>
    <xsl:apply-templates/>
</xsl:template>

</xsl:stylesheet>
```

```
<?xml version='1.0' encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:template match='/'>
  <!-- Esta es la plantilla para el elemento raiz -->
  <!-- Se pone la estructura de un documento html y dentro del cuerpo llamamos a
apply-templates para que aplique las plantillas a los hijos de la raiz -->
    <html>
    <style type="text/css">
      body {
        padding-left: 1em;
        color: black;
        background-color: #DBE5F1;
      }
      #green {
        color: green;
      }
      #blue {
        color: #000066;
      }
      #red {
        color: #C00000;
      }
      #yellow {
        color: #FF9900;
      }
      #purple {
        color: #7030A0;
      }
      #black {
        color: #000000;
      }
    </style>
    <body>
    <font face="Verdana" size="2">
        <xsl:apply-templates /> <!-- esto aplicaria a todos los hijos de la raiz su
correspondiente plantilla -->
    </font>
    </body>
    </html>
</xsl:template>


<!-- Plantilla para 2.SYN y 3.REN -->
<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato">
    <xsl:apply-templates/>
</xsl:template>

<!-- Strings -->
<xsl:template match="list-string">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="string">
    <!-- Para los nodos string, simplemente escribimos su valor, y metemos un espacio -->
    <xsl:if test=". != 'PAT'">
    <xsl:value-of select="."/> <!-- espacio en blanco -->
    </xsl:if>
</xsl:template>

<!-- Lit -->
<xsl:template match="int">
    <!-- Para los nodos int, simplemente escribimos su valor, y metemos un espacio -->
    <xsl:value-of select="@value"/> <!-- la arroba es para indicar que es un atributo
-->
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="bool">
    <!-- Para los nodos bool, simplemente escribimos su valor, y metemos un espacio -->
```

```
    <xsl:value-of select="@value"/> 
    <xsl:apply-templates/>
</xsl:template>

<!-- unit -->
<xsl:template match="ConstE-unit/ConstrT/string">
    { <xsl:value-of select="."/> }
</xsl:template>

<!--<xsl:template match="unit">
    {()}
</xsl:template>-->

<xsl:template match="ConstrT">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-ExpTipo"/>
    <xsl:if test="count(list-string/string)>0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span> </xsl:if>
</xsl:template>

<xsl:template match="list-Exp-unit">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="ConstrT/bool">
    <!-- si el valor del atributo value es true se escribe un ! si no no se hace nada -->
    <xsl:if test="@value='True'">
        <strong>! </strong>
    </xsl:if>
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Flecha">
    <span id="red"><xsl:apply-templates select="node()[position()=1]"/></span>
    <span id="black"><strong>--> </strong></span>
    <span id="red"><xsl:apply-templates select="node()[position()=2]"/></span>
</xsl:template>

<xsl:template match="Rec">
    <span id="red"><strong>REC </strong></span><!-- Simplemente escribimos la palabra REC en
negrita -->
    <xsl:apply-templates/><!-- Aplicamos plantillas al resto, esto hay que ponerlo siempre
-->
</xsl:template>

<!-- DATA -->
<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato">
    <xsl:apply-templates/>
    <br/>
</xsl:template>

<xsl:template match="list-tuple4-string-list-string-list-string-list-AltDato/string">
    <span id="blue"><strong>data </strong></span> <!-- la palabra data delante, -->
    <xsl:value-of select="string"/> <!-- despues el valor del string -->
    <xsl:apply-templates/> <!-- aplicamos plantillas al resto y escribimos un espacio al
final-->
</xsl:template>

<xsl:template
match="list-tuple4-string-list-string-list-string-list-AltDato/list-string[position() mod 2
!= 0]">
<!-- aqui la plantilla se aplica para los hijos list-string cuya posicion sea impar -->
    <xsl:apply-templates/> <!-- se llama al apply-templates antes, para que aplique las
plantillas a los hijos y que la arroba la ponga al final-->
    <xsl:if test="node()=text()"><span id="green"><strong>@ </strong></span>
    </xsl:if>
</xsl:template>

<xsl:template
```

```
match="list-tuple4-string-list-string-list-string-list-AltDato/list-string[position() mod 2 =
0]">
<!-- aqui la plantilla se aplica para los hijos list-string cuya posicion sea par -->
    <xsl:apply-templates/>
    <strong>= </strong><br/>
</xsl:template>

<!-- AltDato -->
<xsl:template
match="list-tuple4-string-list-string-list-string-list-AltDato/list-AltDato/ConstrA">
         <strong>| </strong> <!-- para los ConstrA escribimos
primero un | y delante espacios para tabular-->
    <xsl:apply-templates select="string[1]"/> <!-- aplicamos la plantilla para el primer hijo
string (:) -->
    <xsl:apply-templates select="list-ExpTipo"/> <!-- aplicamos la plantilla para el hijo
list-ExpTipo (a REC) -->
    <span id="green"><strong>@ </strong> <!-- metemos un simbolo arroba -->
    <xsl:apply-templates select="string[2]"/></span> <!-- aplicamos la plantilla para el
segundo hijo string (rho1) -->
    <br/><!-- salto de linea -->
</xsl:template>

<!-- unit -->
<xsl:template match="ConstrE-unit">
    <xsl:apply-templates select="string[1]"/>
    <xsl:apply-templates select="list-Exp-unit"/>
    <span id="green"><strong>@ </strong>
    <xsl:apply-templates select="string[2]"/></span>
</xsl:template>

<xsl:template match="VarE-unit">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="node()[position()=2]"/>
</xsl:template>

<xsl:template match="CopyE-unit">
    <xsl:apply-templates select="string[1]"/> <!-- Aplicamos la plantilla para el primer
string de CopyE -->
    <span id="green"><strong>@ </strong> <!-- Despues escribimos una arroba -->
    <xsl:apply-templates select="string[2]"/></span><!-- Aplicamos la plantilla para el
segundo string de CopyE -->
</xsl:template>

<xsl:template match="ReuseE-unit">
    <xsl:apply-templates select="string"/>
    <strong>! </strong>
    <xsl:apply-templates select="node()[position()=2]"/>
</xsl:template>

<xsl:template match="AppE-unit">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-Exp-unit"/>
    <xsl:if test="count(list-string/string)>0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span></xsl:if>
    <xsl:apply-templates select="node()[position()=4]"/>
</xsl:template>

<xsl:template match="list-tuple2-Patron-unit-bool">
    <xsl:apply-templates/><strong>= </strong>
</xsl:template>

<xsl:template match="VarP-unit">
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="node()[position()=2]"/>
</xsl:template>

<xsl:template
match="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-unit-bool-list-string-Der-un
it">
```

```
    <!--<xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-tuple2-Patron-unit-bool"/><strong>=</strong><br/>
    <xsl:apply-templates select="list-string"/>
    <xsl:apply-templates select="Simple-unit"/>-->
    <xsl:apply-templates/>
</xsl:template>

<xsl:template
match="ParsedProg/list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-unit-bool-list-st
ring-Der-unit/list-ExpTipo">
    <xsl:if test="following-sibling::string">
    <br/>
    <xsl:if test="count(node())>0">
    <strong>{ <span id="red">
    <xsl:value-of select="following-sibling::string"/></span> <xsl:choose><xsl:when test=".
!= '1Sintipo'"> ::
<xsl:apply-templates/></xsl:when><xsl:otherwise> </xsl:otherwise></xsl:choose>}</strong>
<br/>
    </xsl:if>
    </xsl:if>
</xsl:template>

<xsl:template
match="LetE-unit/list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-unit-bool-list-str
ing-Der-unit/list-ExpTipo">
    <xsl:if test="following-sibling::string">
    <xsl:if test="count(node())>0">
    <strong>{
    <xsl:value-of select="following-sibling::string"/><xsl:choose><xsl:when test=". !=
'1Sintipo'"> ::
<xsl:apply-templates/></xsl:when><xsl:otherwise> </xsl:otherwise></xsl:choose>}</strong>
<br/>
    </xsl:if>
    </xsl:if>
</xsl:template>

<xsl:template match="CaseE-unit">
    { <xsl:apply-templates select="node()[position()=3]"/>}<br/>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>case </strong></span>
    <xsl:apply-templates select="node()[position()=1]"/>
    <span id="blue"><strong>of</strong></span>
    <ul style="list-style-type:node;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple2-Patron-unit-Exp-unit"/></ul>
    </ul>
</xsl:template>

<xsl:template match="CaseE-unit/list-tuple2-Patron-unit-Exp-unit">
    <xsl:for-each select="child::node()">
        <xsl:if test="position() mod 2 != 0"><xsl:apply-templates select="."/><strong>-->
</strong></xsl:if>
        <xsl:if test="position() mod 2 = 0"><xsl:apply-templates select="."/>
        <xsl:if test="position() != last()"><br/></xsl:if></xsl:if>
    </xsl:for-each>
</xsl:template>

<xsl:template match="list-tuple2-Patron-unit-Exp-unit">
    <xsl:apply-templates/>
</xsl:template>

<!--<xsl:template match="CaseE-unit/list-tuple2-Patron-unit-Exp-unit/node()[position() mod 2
= 0]/node()[position()=last()]">
    <xsl:apply-templates/><br/>
</xsl:template>

<xsl:template match="CaseE-unit/list-tuple2-Patron-unit-Exp-unit/node()[position() mod 2 !=
0]/node()[position()=last()]">
    <xsl:apply-templates/><strong>-> </strong>
</xsl:template>-->
```

```
<xsl:template match="CaseDE-unit">
    { <xsl:apply-templates select="node()[position()=3]"/>}<br/>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>case! </strong></span>
    <xsl:apply-templates select="node()[position()=1]"/>
    <span id="blue"><strong>of</strong></span>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple2-Patron-unit-Exp-unit"/></ul>
    </ul>
</xsl:template>

<xsl:template match="CaseDE-unit/list-tuple2-Patron-unit-Exp-unit">
    <xsl:for-each select="child::node()">
        <xsl:if test="position() mod 2 != 0"><xsl:apply-templates select="."/><strong>-->
</strong></xsl:if>
        <xsl:if test="position() mod 2 = 0"><xsl:apply-templates select="."/>
        <xsl:if test="position() != last()"><br/></xsl:if></xsl:if>
    </xsl:for-each>
</xsl:template>

<!--<xsl:template match="CaseDE-unit/list-tuple2-Patron-unit-Exp-unit/node()[position() mod 2
= 0]/node()[position()=last()]">
    <xsl:apply-templates/><br/>
</xsl:template>

<xsl:template match="CaseDE-unit/list-tuple2-Patron-unit-Exp-unit/node()[position() mod 2 !=
0]/node()[position()=last()]">
    <xsl:apply-templates/><strong>-> </strong>
</xsl:template>-->

<xsl:template match="LetE-unit">
    { <xsl:apply-templates select="node()[position()=3]"/>}<br/>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0">
    <span id="blue"><strong>let </strong></span>
    <ul style="list-style-type:none;margin-bottom:0;margin-top:0"><xsl:apply-templates
select="node()[position()=1]"/></ul>
    <span id="blue"><strong>in </strong></span>
    <xsl:apply-templates select="node()[position()=2]"/>
    </ul>
</xsl:template>

<!-- Der-Decor -->
<xsl:template match="Simple-unit">
    <xsl:apply-templates select="node()[position()=1]"/>
    <xsl:if
test="count(list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-unit-bool-list-string-D
er-unit/string)!=0"><br/>
    <ul style="margin-bottom:0;margin-top:0"><span id="blue"><strong>where</strong></span>
    <ul style="margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-unit-bool-list-string-Der-u
nit"/></ul></ul>
    </xsl:if>
</xsl:template>

<xsl:template match="Simple-unit/ConstE-unit/ConstrT">
    <xsl:apply-templates/> <!--<br/><br/>-->
</xsl:template>

<xsl:template match="Guardado-unit">
    <ul style="margin-bottom:0;margin-top:0">
    <xsl:apply-templates select="node()[position()=1]"/></ul>
    <xsl:if
test="count(list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-unit-bool-list-string-D
er-unit/string)!=0">
    <ul style="margin-bottom:0;margin-top:0"><span id="blue"><strong>where</strong></span>
    <ul style="margin-bottom:0;margin-top:0"><xsl:apply-templates
select="list-tuple3-list-ExpTipo-tuple3-string-list-tuple2-Patron-unit-bool-list-string-Der-u
nit"/></ul></ul>
```

```
    </xsl:if>
</xsl:template>

<xsl:template match="list-tuple2-Exp-unit-Exp-unit">
    <xsl:apply-templates/>
</xsl:template>

<xsl:template match="Guardado-unit/list-tuple2-Exp-unit-Exp-unit/AppE-unit">
    <xsl:if test="position()!=1"><br/></xsl:if>
    <strong>| </strong>
    <xsl:apply-templates select="string"/>
    <xsl:apply-templates select="list-Exp-unit"/>
    <xsl:if test="count(list-string/string)>0"><span id="green"><strong>@ </strong>
    <xsl:apply-templates select="list-string"/></span></xsl:if>
    <strong>= </strong>
</xsl:template>

<xsl:template match="list-tuple2-Exp-unit-Exp-unit/node()[position() mod 2 !=
0]/node()[position()=last()]">
    <strong>| </strong><xsl:apply-templates/><strong>= </strong>
</xsl:template>

<xsl:template match="list-tuple2-Exp-unit-Exp-unit/node()[position() mod 2 =
0]/node()[position()=last()]">
    <xsl:apply-templates/><br/>
</xsl:template>

</xsl:stylesheet>
```

```
/* basic elements */
html {
    top: 0;
    margin: 0;
    padding: 0;
    }
body {

    background:#4F81BD;
    color: white;
       font-family: Verdana;
    font-size: 12;
    margin: 0;
    padding: 0;
    top: 0;
    }
p {
    margin-top: 0;
    text-align: justify;
    }


#logo {
    top: 0px;
    left: 0px;
    float:left;
    position:absolute;
}


#header {
    background: url(../img/header.png) repeat-x;
    margin:0;
    padding:0;
    width: 100%;
    height:185px;
    top: 0px;
    left: 0px;
    position: absolute;
}

#Canvas{
        /*background: url(../img/body1.jpg) repeat-y;*/
    top:80px;
    left:20px;
    right:20px;
        position:relative;
}


#container{
        background: url(../img/body1.jpg) repeat-y;
    margin:0;
    top:0px;
    left:0px;
        position:relative;
}


root {
    display: block;
}

.main{
    width: 800px;
    position: absolute;
    margin-top: 30px;
    margin-left: 150px;
    margin-bottom: 150px;
```

```
}

.fase_main_0{
    position:absolute;
    width: 200px;
    height: 50px;
    background-color: #0000ff;
    border-top-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-right-style: solid;
    border-top-color: black;
    border-bottom-color: black;
    border-left-color: black;
    border-right-color: black;
}

.fase_main_1{
    position:absolute;
    width: 200px;
    height: 50px;
    background-color: #00ff00;
    border-top-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-right-style: solid;
    border-top-color: black;
    border-bottom-color: black;
    border-left-color: black;
    border-right-color: black;
}

.fase_main_2{
    position:absolute;
    width: 200px;
    height: 50px;
    background-color: #ff0000;
    border-top-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-right-style: solid;
    border-top-color: black;
    border-bottom-color: black;
    border-left-color: black;
    border-right-color: black;
}

.stage_int_0{
    width: 200px;
    height: 50px;
    background-color: #AABBCC;
    border-style: solid;
    border-color: white;
    border-width: 2px;
    clear: both;
}

.stage_int_1{
    width: 200px;
    height: 50px;
    background-color: #00ff00;
    border-top-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-right-style: solid;
    border-top-color: black;
    border-bottom-color: black;
    border-left-color: black;
    border-right-color: black;
```

```
    clear: both;
}

.stage_int_2{
    width: 200px;
    height: 50px;
    background-color: #ff0000;
    border-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-right-style: solid;
    border-top-color: black;
    border-bottom-color: black;
    border-color: black;
    border-right-color: black;
    clear: both;
}
.stage{
    position:absolute;
    width: 200px;
    height: 50px;
    padding-top: 31px;
}

.stage_ext{
    /*margin-top: -37px;
    margin-left: 110px;*/
    width: 100px;
    height: 33px;
    float: right;
    margin-right: -20px;
}
.stage_tools {
    float:right;
    height:16px;
    margin-top:3px;
    width:53px;
}
.stage_title{
    width: 200px;
    height: 15px;
    margin-top: 25px;
    font-size: 14px;
    color: #FFFFFF;
    font-family: verdana;
}

#menuhere{
    margin-bottom: 50px;
}

.infobox{
    color: #507090;
    background: #FDECA0;
    border: solid 1px;
    font-family: verdana;
    font-size: 10px;
    z-index: 99;
    width: 150px;
    height: 30px;
}

.infobox_edit{
    color: #507090;
    background: #C6DCF1;
    border: solid 1px;
    font-family: Arial, Helvetica;
    font-size: 10px;
    font-color: #204060;
```

```
    z-index: 99;
    width: 150px;
    height: 45px;
}

.upload_icon{
    background-image: url(../img/icon_up.png);
}

label.upload_file {
    background:transparent url(../img/icon_up.png) no-repeat scroll 0 0;
    cursor:pointer;
    display:block;
    height:23px;
    overflow:hidden;
    width:23px;
}
label.upload_file input.file {
    height:100%;
    opacity:0;
    position:relative;
    width:auto;
}



.err_{
    font-family: Arial, Helvetica;
    color: red;
    font-size: 50px;

}
.err_str{
    color: black;

}
.err_body {
    background:yellow repeat scroll 0 0;
    border:solid;
    text-align:center;
}
.load_file{
    background-color: #6080A0;
    margin-top: 180px;
    margin-left: 215px;
    padding: 15px;
    width: 237px;
    border-style: outset;
    border-color: gray;
    position: absolute;
}

.load_file div{
    padding: 3px;
}

#footer{
    background: transparent url(../img/footer.png) repeat-x bottom;
    bottom:0px;
    text-align: center;
    padding-top:1100px;
    padding-bottom:0px;
}
```

Palabras clave

Compilador

Safe

Html

Php

Javascript

Xml

Xlst

Css

Autorización

Lope Javier Hidalgo Garrido-Lestache, Enrique Omar Huidobro Moradillo y Juan Antonio Masa González autorizan a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Lope Javier Hidalgo Garrido-Lestache

Enrique Omar Huidobro Moradillo

Juan Antonio Masa González