

Monaaco

Descripción general del diseño del sistema

Equipo desarrollador del proyecto:

Manuel Báez Sánchez

Alfredo Cerezo Luna

Jorge Cordero Sánchez

Miguel González Pérez

Beatriz Torres Salcedo

Índice

[Monaaco](#)

[Descripción general del diseño del sistema](#)

[Estructura de los paquetes:](#)

[IS2011.Interfaz](#)

[IS2011.FiltrosArchivos](#)

[IS2011.GestorXML](#)

[IS2011.biblioteca](#)

[IS2011.biblioteca.comparators](#)

[IS2011.configuracion](#)

[IS2011.tests](#)

[Formato biblioteca xml](#)

[Carpetas](#)

[images](#)

[sounds](#)

[xml](#)

[doc](#)

[uml](#)

[Estructura de las carpetas en el repositorio:](#)

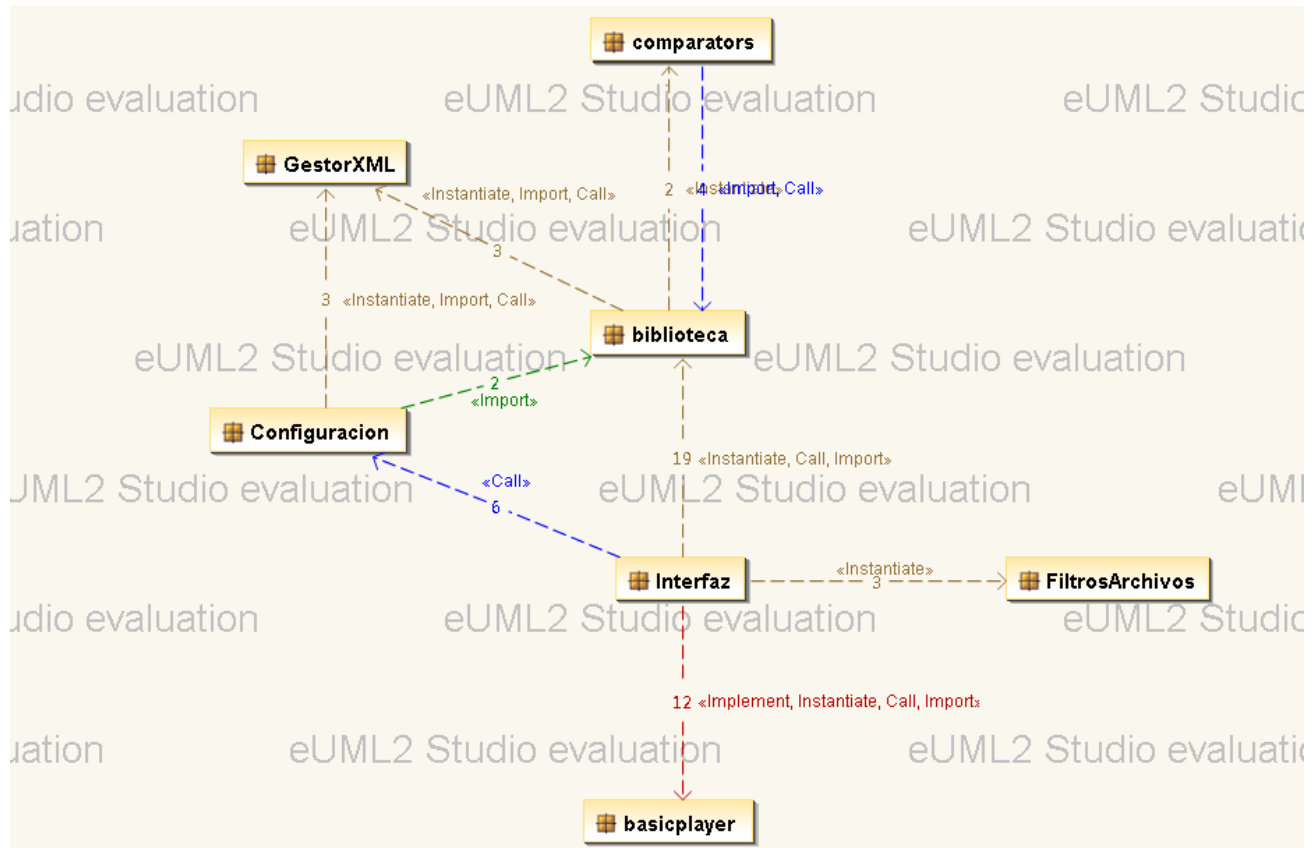
[Instalador](#)

[Manual de usuario](#)

[Proyecto](#)

[Tutoriales](#)

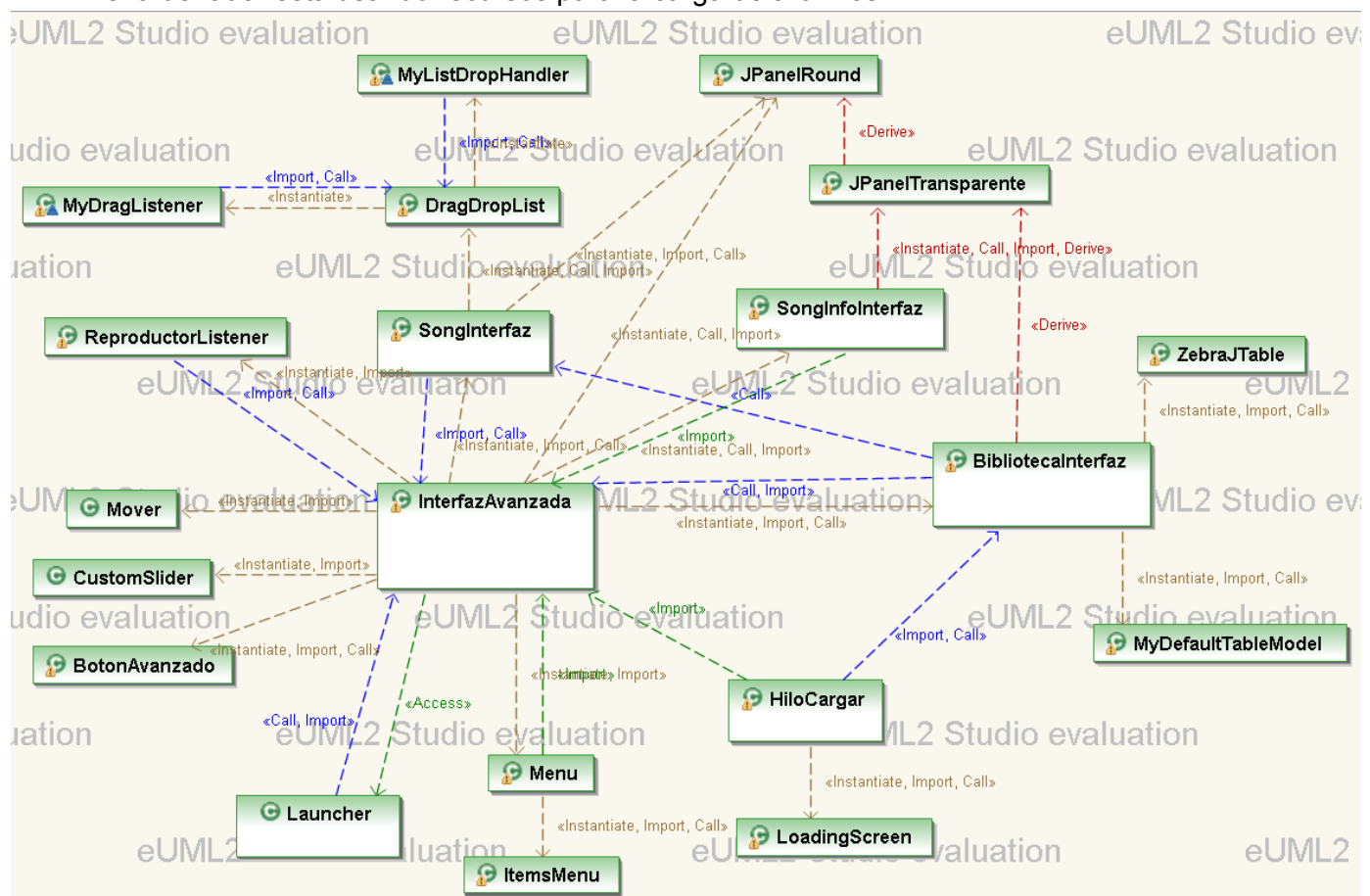
Estructura de los paquetes:



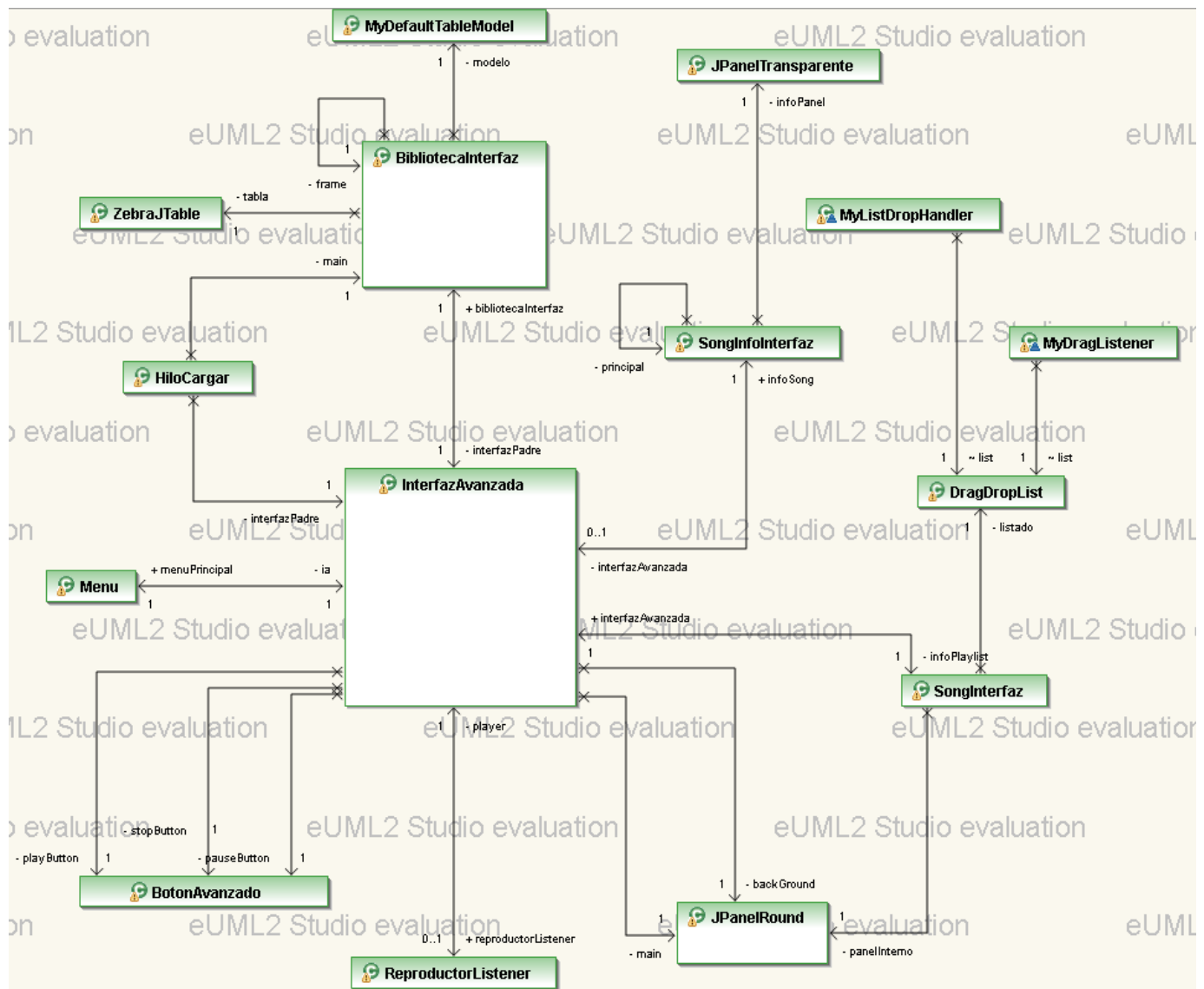
IS2011.Interfaz

Este paquete contiene todas las clases de la interfaz del programa.

- InterfazAvanzada podemos decir que es la clase principal que nos facilitará todas las funcionalidades tanto de reproducción como de la biblioteca.
- La clase SongInterfaz nos proporciona la interfaz de la playList.
- JPanelRound y JPanelTransparente simplemente nos proporcionan los componentes que utilizaremos, en este caso para los Paneles que usaremos en la interfaz.
- La clase Mover implementa el método que utilizaremos para mover la interfaz pichando en cualquier lado, que más tarde agregaremos a un listener de InterfazAvanzada.
- La clase SongInfoInterfaz nos proporciona un JPanelRound con la información del Track en reproducción.
- La clase BibliotecaInterfaz nos proporciona un JPanelTransparente con toda la información de nuestra biblioteca y las funcionalidades de búsqueda y filtrado.
- Hilo cargar nos externaliza la carga de archivos en la biblioteca, así conseguimos mantenerlo en segundo plano y poder hacer cosas mientras carga el archivo, como por ejemplo ejecutar el hilo que implementa la clase LoadingScreen que nos indica que el ordenador está usando recursos para la carga de archivos.

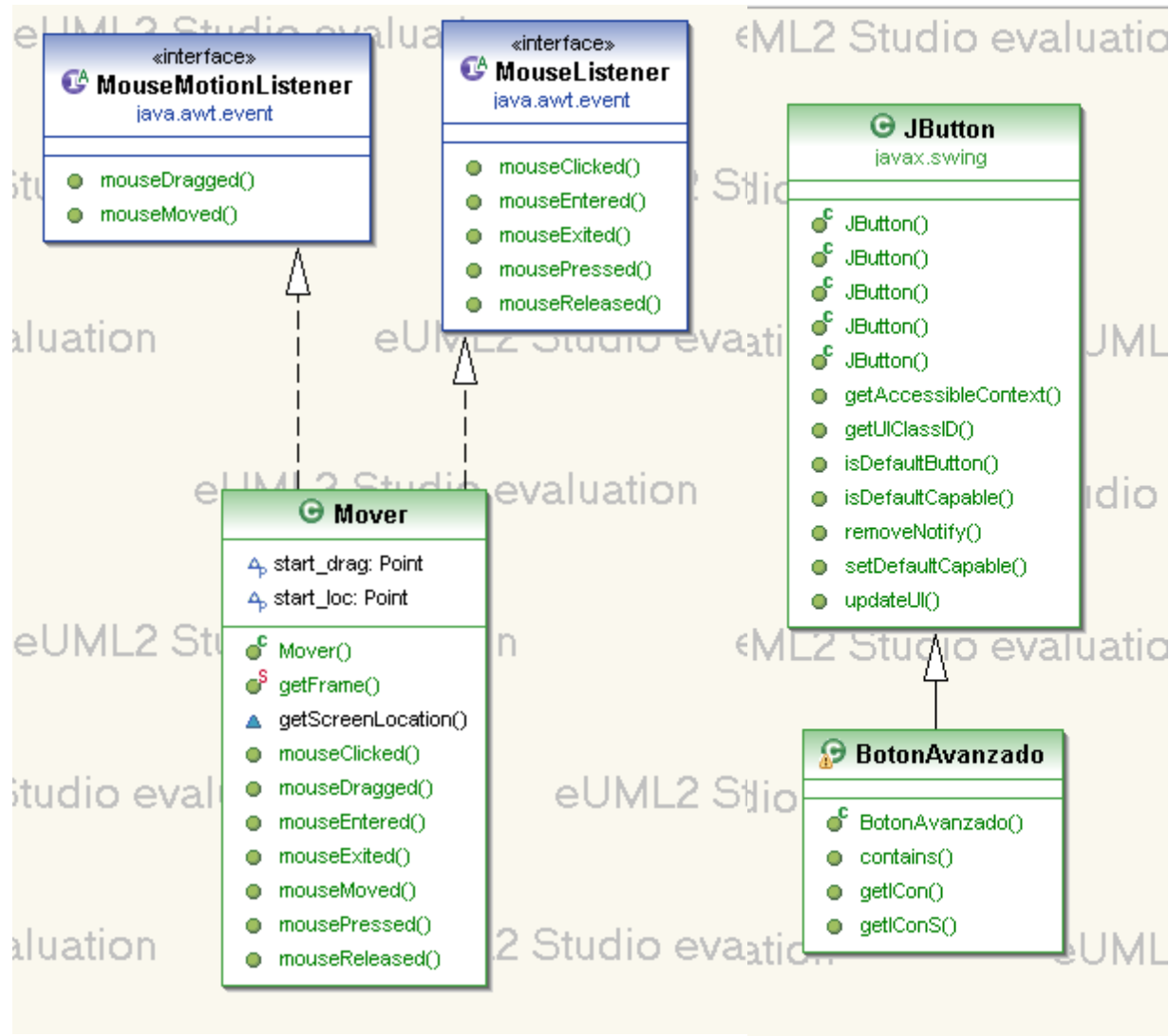


La clase Menu implementa el menu de la aplicación mediante un JPanelTransparente



-En el diagrama de la izquierda podemos ver como se implementa la clase Mover, redefinimos los métodos mouseDragged y mouseMoved de la interface MouseMotionListener , estos métodos guardan la info de la posición de la aplicación en la pantalla y donde se quiere mover, y luego los listener usuales de la interface MouseListener.

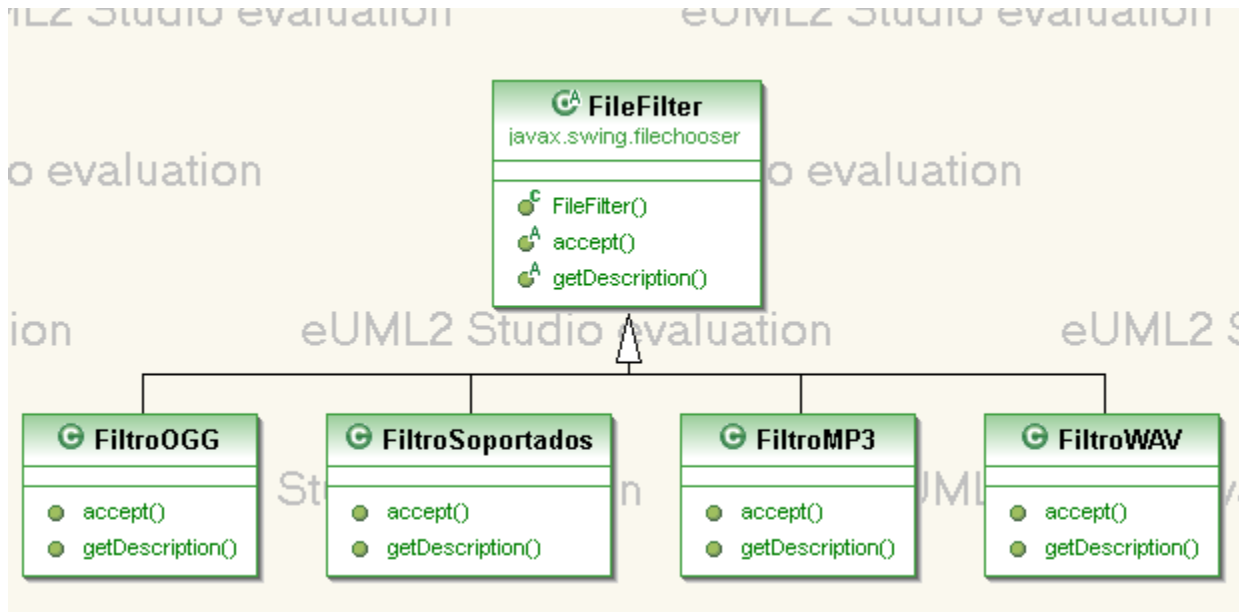
-En el diagrama de la derecha vemos como se implementa la clase BotonAvanzada que implementa botones redondos.



IS2011.FiltrosArchivos

En este paquete están los filtros de archivo utilizados para cargar los archivos de audio en la biblioteca o lista de reproducción.

Hay cuatro filtros, tres para cada archivo soportado y uno para todos a la vez. Los filtros de archivo extienden la clase `FileFilter` y redefinen los métodos `accept()` y `getDescription()`.



IS2011.GestorXML

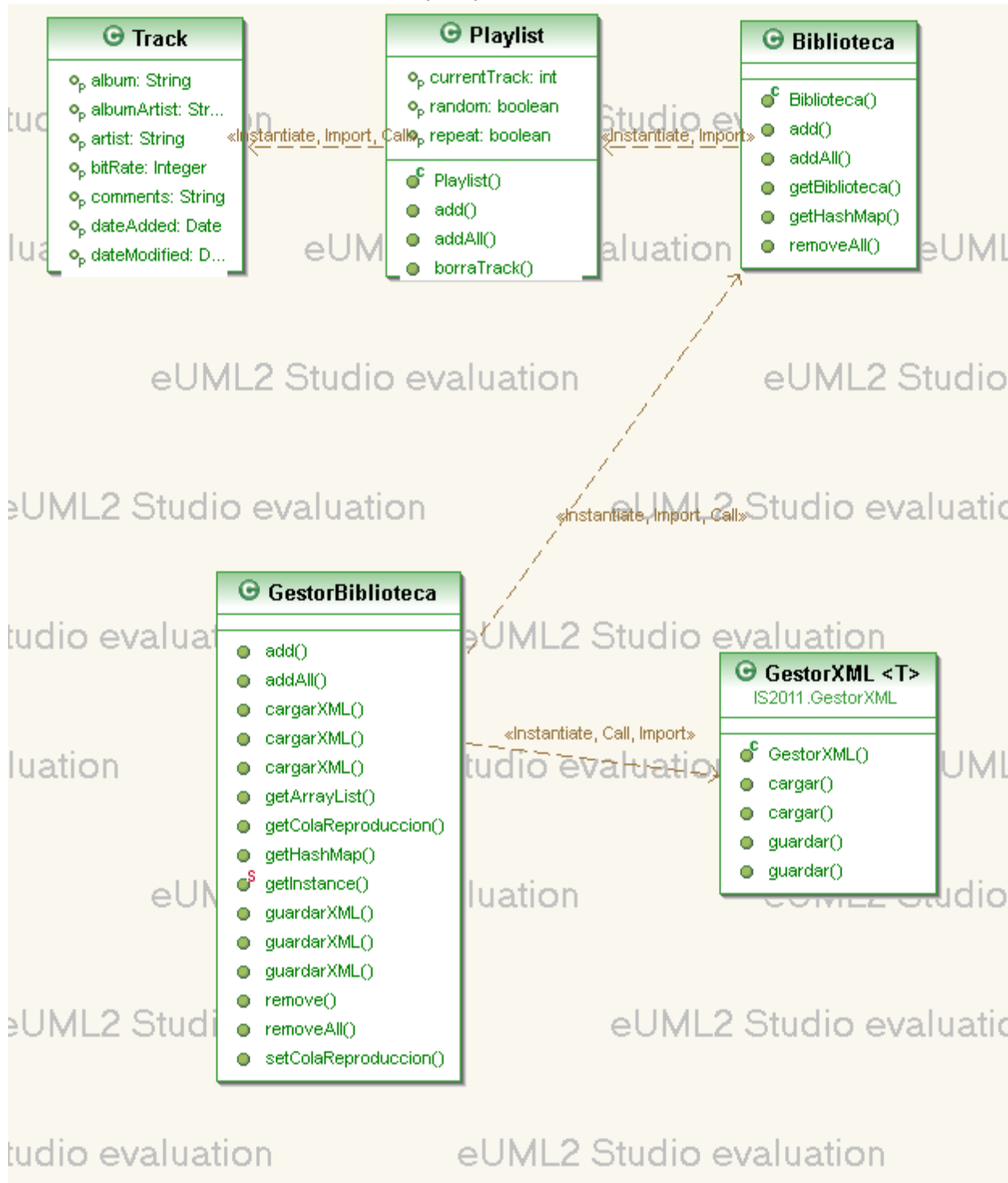
En este paquete contiene la clase GestorXML<T> genérica la cual contiene los métodos necesarios para guardar o cargar un objeto de tipo T en un archivo xml.

Los métodos de carga lanzan excepciones para que se traten en las clases que hagan uso de GestorXML.

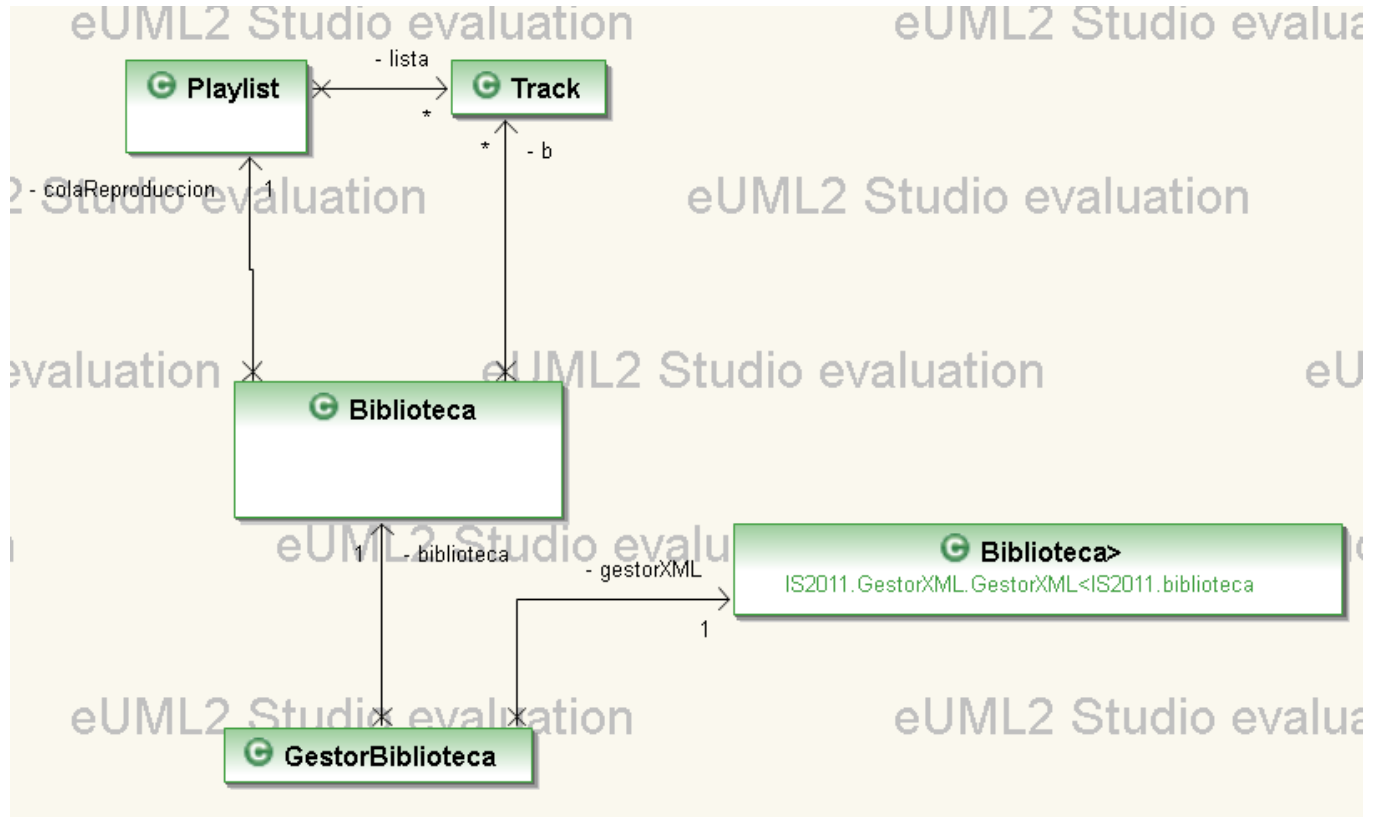


IS2011.biblioteca

Aquí se encuentran las clase Track, Playlist y Biblioteca.

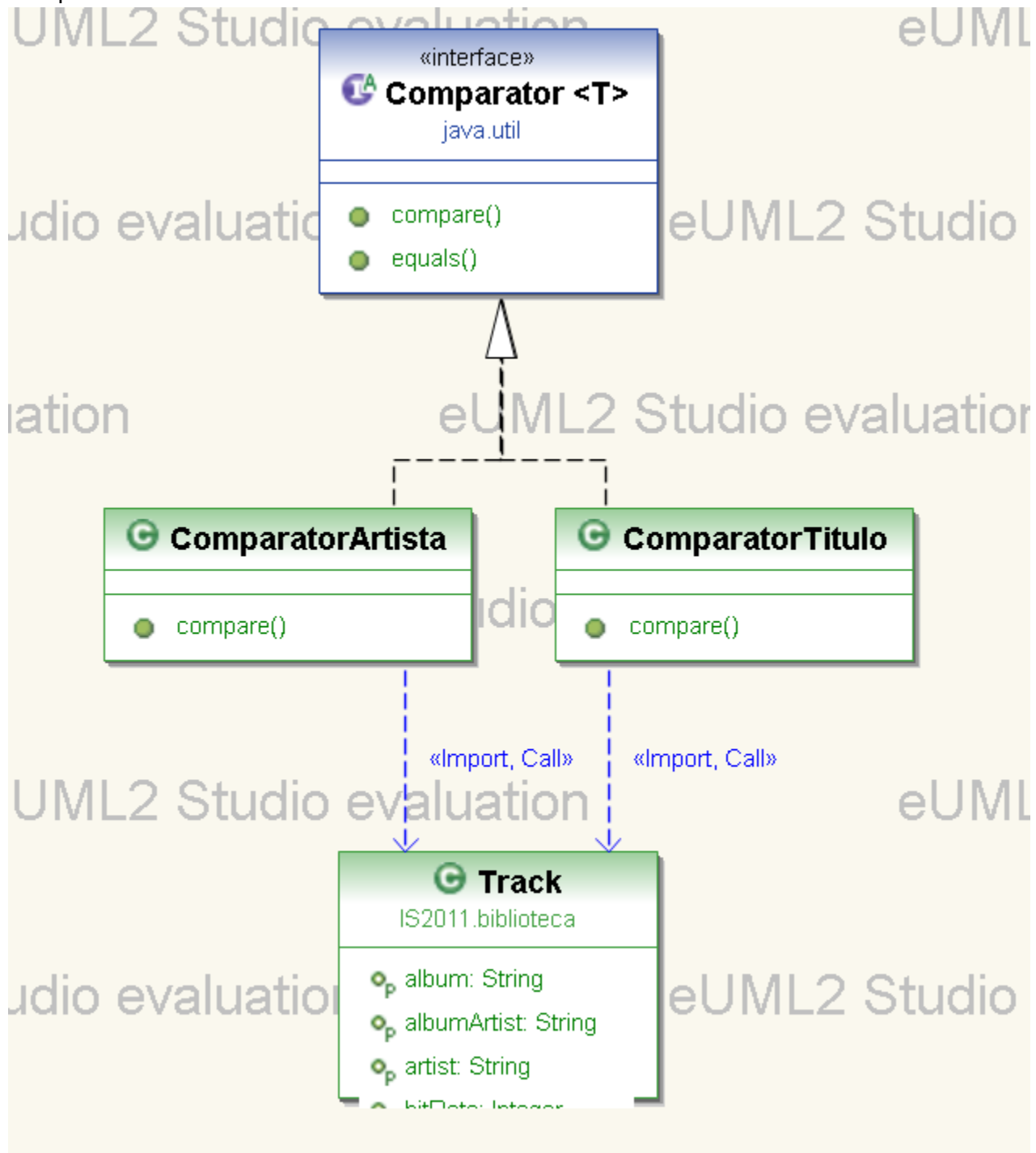


Dentro de este paquete se encuentra la clase GestorBiblioteca, la cual tiene aplicado el patrón singleton y adapta la biblioteca y el GestorXML a la interfáz, de forma que si desde el paquete IS2011.Interfaz se quiere acceder a la biblioteca, o guardarla o cargarla en XML, esto se hace a través de la instancia de Gestor biblioteca. Por ejemplo para guardar la biblioteca en el fichero por defecto xml desde la interfaz se ejecutara “GestorBiblioteca.getInstance().guardarXML();”.

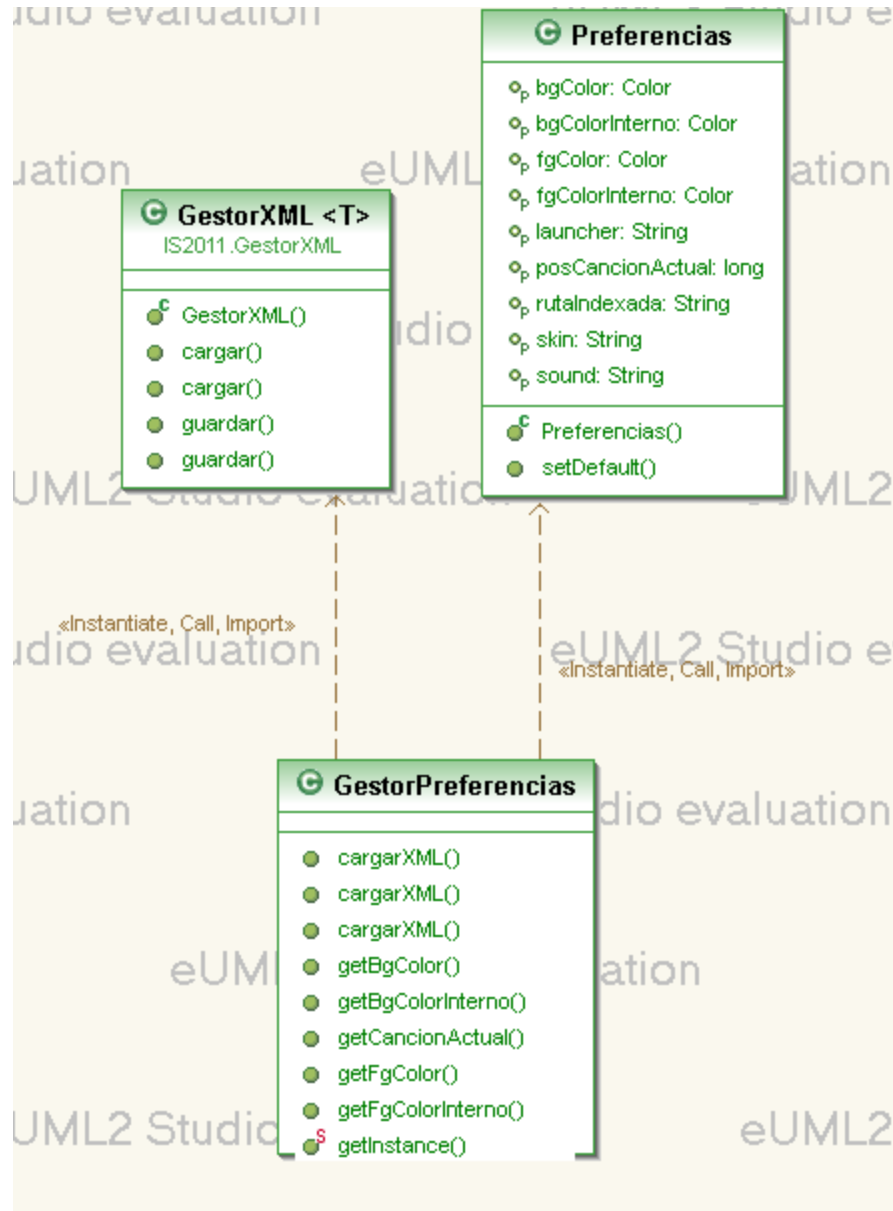


IS2011.biblioteca.comparators

En este paquete están definidos los comparadores de la clase Track, ComparadorArtista y ComparadorTítulo.



IS2011.configuracion



Igual que ocurre con la Biblioteca la configuración se guarda en la clase **Preferencias** pero se utiliza el **GestorPreferencias** para adaptarla a la Interfaz, haciendo uso de la clase **GestorXML** para guardar la clase **Preferencias** en xml.

GestorPreferencias tiene aplicado el patrón singleton.

IS2011.tests

Los tests, ya sean automatizados (JUnit) o manuales, estan dentro de este paquete.

Formato biblioteca xml

El archivo xml de la biblioteca contiene la información de la la clase “Biblioteca”:

- La lista de canciones de la biblioteca.
- Cola de reproducción. (Toda la información de clase Playlist)

```
<biblioteca>
  -- Canciones de la biblioteca
  <tracks>
  </tracks>
  -- Cola de reproducción
  <trackQueue>
    -- Lista de canciones
    <trackList/>
    -- Canción actual
    <currentTrack>-1</currentTrack>
    -- Atributos de la cola de reproducción
    <repeat>true</repeat>
    <random>false</random>
  </trackQueue>
</biblioteca>
```

Carpetas

Los archivos utilizados por el programa se encuentran fuera de los paquetes organizados por carpetas.

images

Contiene las imágenes del programa. Dentro de esta carpeta se encuentran las carpetas de skin, cada una con los botones necesarios, imágenes del launcher, iconos...

sounds

Contiene el mp3 que se reproduce durante la carga del programa.

xml

Contiene dos archivos xml:

- biblioteca.xml
Información de la biblioteca.
- preferencias.xml
Configuración del programa; skin, colores, carpeta de la biblioteca...

doc

Contiene la documentación del proyecto:

- Javadoc
- Manual de usuario (pdf)
- Documentación de las librerías externas utilizadas

uml

En esta carpeta se encuentran los diagramas uml del proyecto.

Estructura de las carpetas en el repositorio:

Instalador

Contiene:

- monaaco- 1.0.243.exe primer ejecutable del programa .
- monaaco- 1.0.282.exe segundo ejecutable del programa.

- nsis-2.46-setup.exe: Programa para hacer instaladores
- Material: Carpeta con todos los archivos necesarios para hacer el instalador de Monaaco.

Manual de usuario

Contiene:

- ManualWord.docx: Manual de usuario en formato .docx, para que se pueda modificar fácil y así pueda estar siempre actualizado.
- ManualdeusuarioMonaaco.pdf: Manual adjuntado con el proyecto, en versión .pdf, se modificaría mediante el .docx de la misma carpeta.
- Imagenes: Carpeta con las imágenes utilizadas en el manual.

Proyecto

Contiene:

- src: Clases de java del programa(en .java) y diagramas(en .ucd)
- images: Imágenes del programa y posibles skins
- lib: Librerías usadas por el programa
- sounds: Sonidos de prueba del programa
- uml: png's de los diagramas uml
- xml: biblioteca y preferencias del programa(ambas en .xml)

Tutoriales

Contiene:

- Tutorial JLayer y JAudioTagger: Carpeta que contiene las librerías JLayer y LAudioTagger.
- Tutorial XML: Carpeta que contiene la librería XStream que usamos para leer y escribir en XML