MONITORIZACIÓN DE PRUEBAS

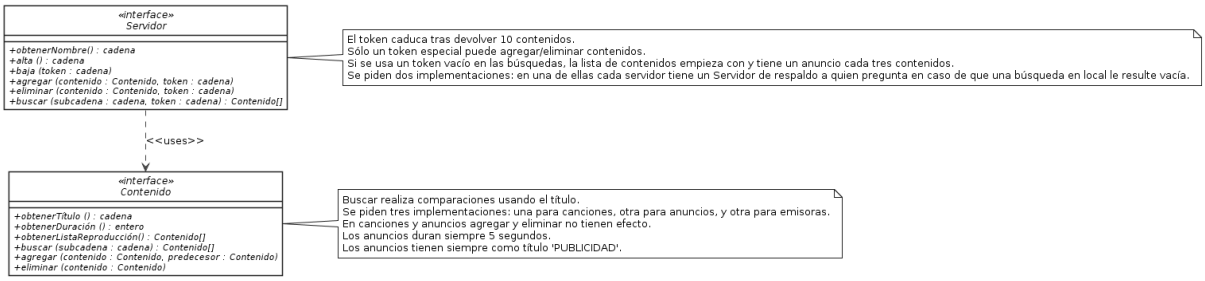
Título proxecto: VVS Music
Ref. proyecto: MIP015
Autores: José Miguel del Río, Faustino Castro, Víctor Blanco
Nota: se adjunta en el repositorio la presentación, que es un complemento necesario para el presente informe.

Validación y Verificación de Software

Fecha de aprobación	Control de versiones	Observaciones
07-12-15	1.0	

1. Contexto

Este informe hace referencia al conjunto de pruebas realizadas para validar el código del proyecto “VVS-Music”.



2. Estado actual

En el diagrama anterior se puede observar como el proyecto tiene dos módulos bien definidos que son por una parte el módulo del Servidor que va alogar la información y el módulo de Contenido que es el tipo de información que puede tener el servidor.

El listado de funcionalidades de las que se disponen son:

- **Servidor**
  - *Obtener nombre*: Devuelve el nombre del servidor.
  - *Alta de un servidor*: Permite dar de alta un servidor.
  - *Baja de un servidor*: Permite dar de baja un servidor.
  - *Agregar contenido*: Permite añadir contenido a un servidor
  - *Eliminar contenido*: Permite quitar contenido a un servidor.
  - *Buscar contenido*: Permite realizar búsquedas por contenido en el servidor. La búsqueda tiene una restricción de que no se pueden buscar contenidos de tipo Anuncio.
- **Contenido**
  - *Obtener titulo*: Devuelve el titulo de un tipo de contenido(canciones,anuncios o emisoras).
  - *Obtener la duración*: Devuelve la duración de un contenido.
  - *Obtener la lista de reproducción*: Devuelve el contenido de un tipo concreto, como pueden ser Canciones,Anuncios o Emisoras.

- *Buscar*: Permite realizar búsquedas de contenido.
- *Agregar contenido*: Permite añadir contenido.
- *Eliminar contenido*: Permite quitar contenido.

El desarrollo de dicho proyecto se planificado de la siguiente manera.

- Primero se ha realizado un análisis rápido del proyecto, ya que se disponía de unos requisitos y un diagrama inicial.
- Después se ha realizado un diagrama de clases completo del proyecto, como se puede observar en la figura 1
- Finalmente se ha implementados los dos módulos en paralelo. El responsable de realizar el módulo del servidor ha sido el desarrollador José del Río mientras que la parte del Contenido ha sido a cargo del desarrollador Faustino Castro. Las pruebas se ha encargado de su realización Víctor Blanco

El plan de pruebas que se ha realizado para este proyecto se ha estructurado en tres paquetes que son :

- Contenido
- Servidor.test
- Servidor.utils

Se han preparado un total de 40 pruebas para los tres paquetes anteriormente mencionado. Los resultados actualmente son :

- Pruebas preparadas: 40
- Pruebas ejecutadas: 100 %
- Pruebas superadas: 100 %

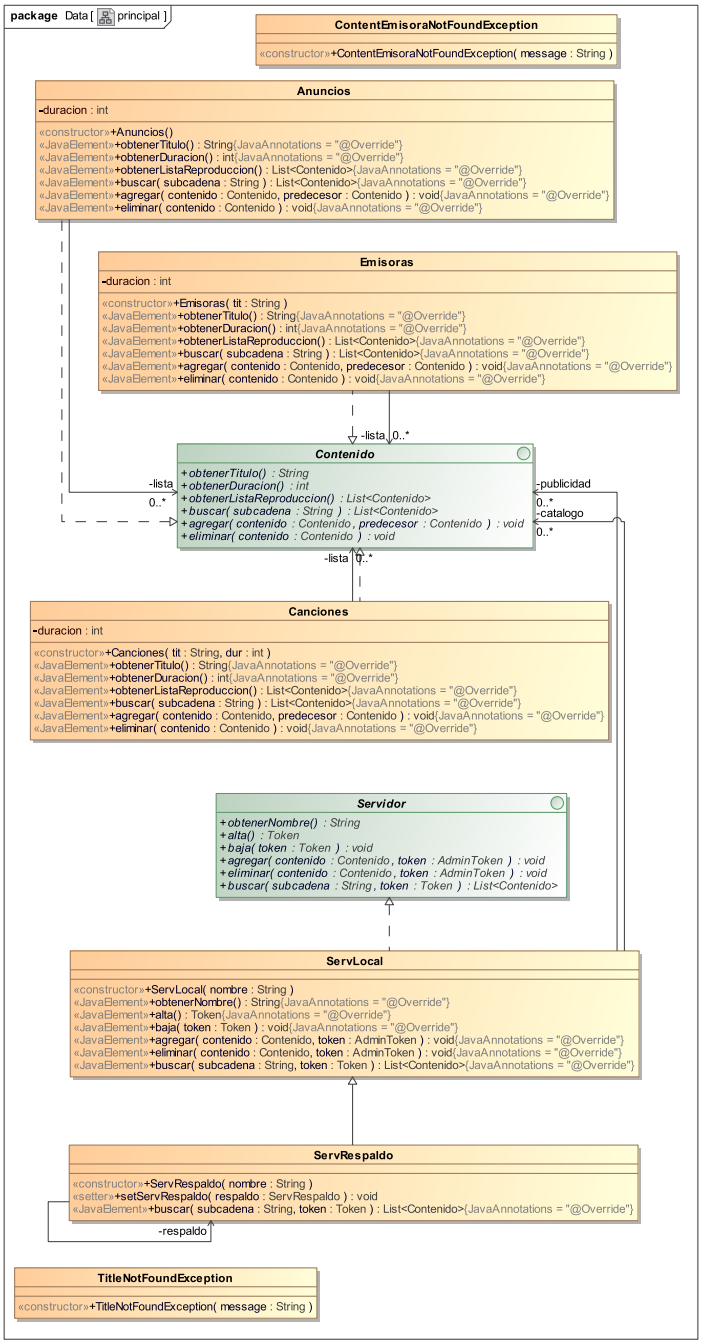


Figura 1: *Diagrama de Clases de VVS Music*

3. Registro de pruebas

A la hora de realizar las pruebas nos hemos encontrado con diferentes problemas de las herramientas utilizadas para llevar a cabo dichas pruebas.

Para validar la calidad de las pruebas se ha utilizado Cobertura(de rama) con la cual tuvimos problemas de integración con GitHub, ya que cada vez que se fusionaba la cobertura con el proyecto producía que no se superasen las pruebas y que tuviésemos que probar con la versión anterior a su fusión.

Además de la Cobertura hemos utilizado herramientas como JUnit,CheckStyle,FindBugs y Graphwalker.

Esta última herramienta la utilizamos para realizar pruebas dinámicas.

3.1. GraphWalker

El primer paso para la aplicación de Graphwalker es la creacion del diagrama con un programa de gráficos, en este caso el yEd, dando como resultado el siguiente grafo de ejecución.

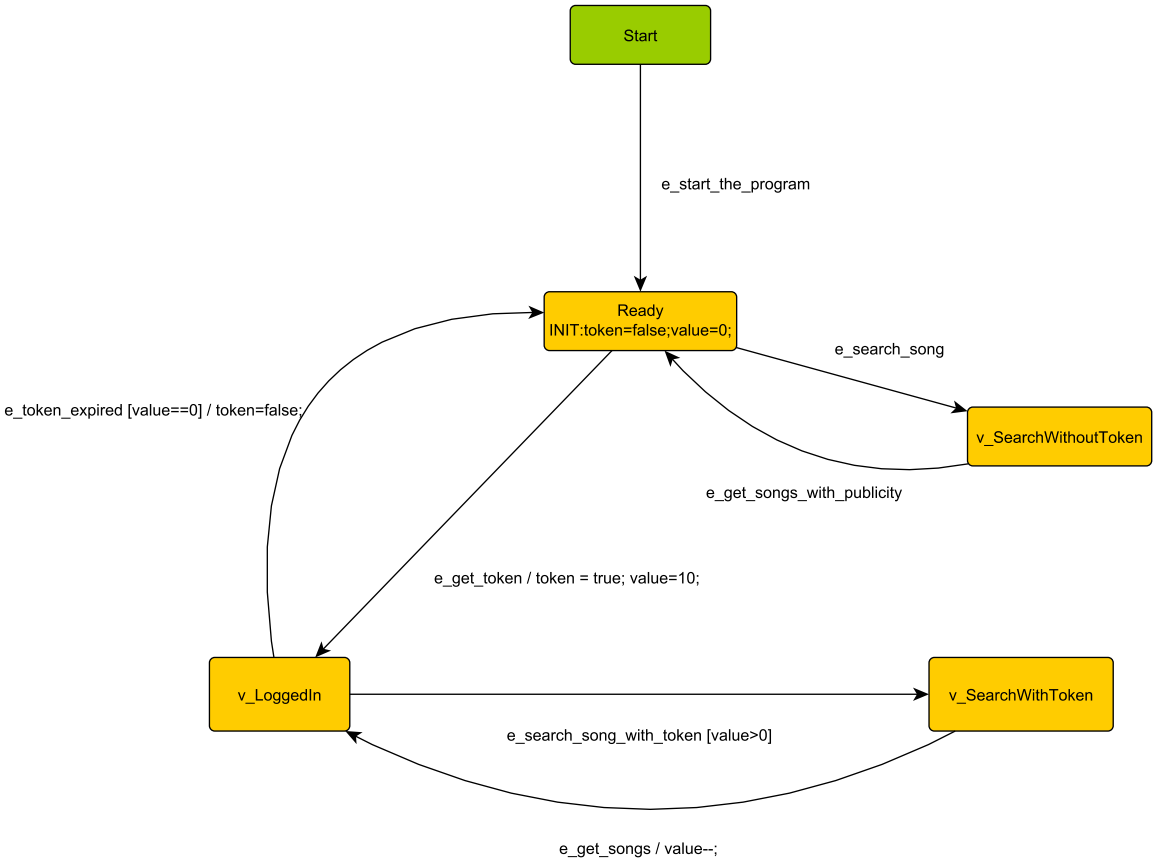


Figura 2: *Grafo GraphWalker*

Una vez realizado el Grafo se comprueba con el standalone client de graphwalker para comprobar su correcta implementación.

Al copiarlo a la carpeta /src/main/resources y modificar el pom con las dependencias adecuadas genera un interfaz al ejecutar el mvn build, que es la que debes implementar en los test. Una vez implementada, se pueden usar varios tipos de prueba, en nuestro caso usamos SmokeTest(Test de prueba básico, se ejecuta hasta llegar a un estado en particular), FunctionalTest(Test que recorre todas las ramas de la ejecución) y StabilityTest(Test que se ejecuta durante x tiempo en forma de bucle).

4. Registro de errores

En las pruebas realizadas no se han detectado errores de incumplimiento de requisitos pero si errores de la metodología aplicada como son realizar comprobaciones que no se van a dar nunca, falta de documentación, etc.

Package	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	13	88% 162/184	77% 62/80	2,033
contenido	4	93% 60/64	100% 20/20	1,704
servidor	3	85% 57/67	66% 28/42	2,444
servidor.Tokens	2	87% 34/39	77% 14/18	2,7
utils.exceptions.contenido	3	100% 6/6	N/A N/A	1
utils.exceptions.tokens	1	50% 3/6	N/A N/A	1

Figura 3

Como se ha mencionado anteriormente se ha utilizado Cobertura para validar la calidad de las pruebas y se han obtenido los siguientes resultados.

El porcentaje total de cobertura conseguida ha sido del 77 % de cobertura de decisiones que es la importante y recorrería todos los casos posibles.

Se han detectado casos que nunca se darán como por ejemplo:
Servidor Local

- El método agregar y eliminar contenido nunca pasará por la decisión que se le pase un token de administrador incorrecto ya que al ser singleton siempre se va obtener la misma instancia y no sería posible crear dos tokens de administrador diferentes.
- Otro error que se encontró en esta clase ha sido en el método buscar que cuando se busca una canción se añade una publicidad cada tres canciones y se detectó en las pruebas que se estaba añadiendo en una lista interna que no se correspondía con la lista que luego se mostraba.

5. Estadísticas

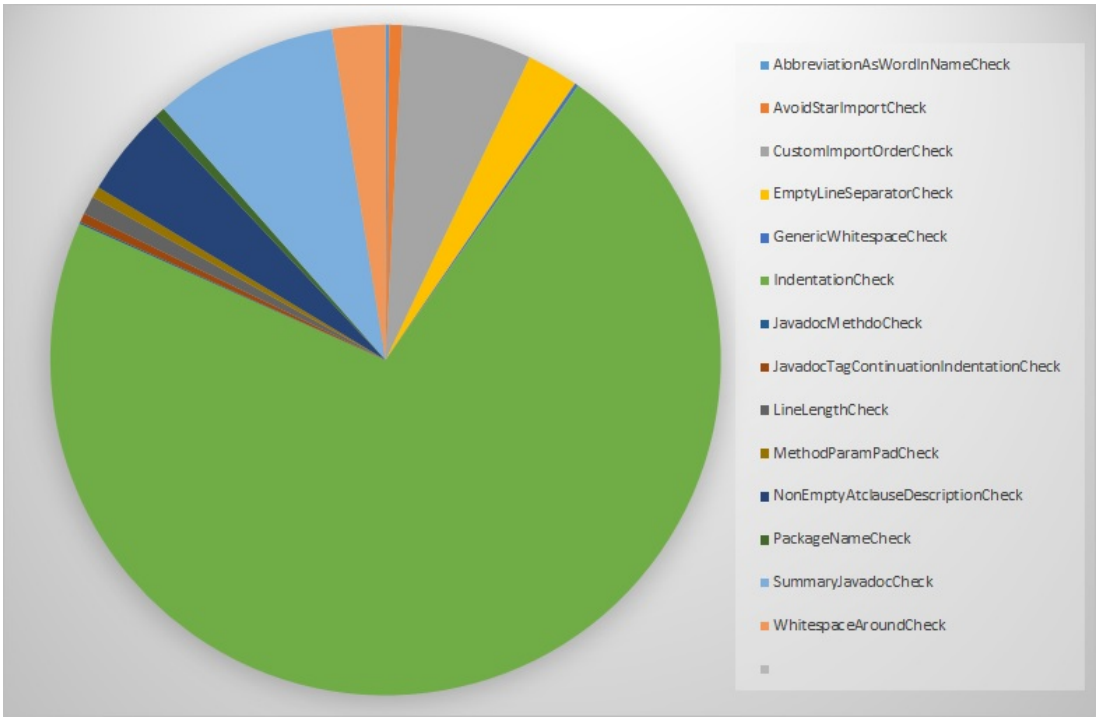


Figura 4: *Resultados CheckStyle*

Los resultados de las pruebas encontrados son:

- Bugs encontrados: 3
- Enhancement propuestos: 7

Se ha utilizado herramientas como *FindBugs* y CheckStyle para realizar pruebas estructurales. El resultado ha sido 1129 errores encontrados de los cuales 812 han sido problemas de indentación.(véase figura 4)

6. Otros aspectos de interés

No aplica