
INFORME DE PRÁCTICAS

Repositorio de proxecto: <https://github.com/Darktuse/VVS>

Participantes no proxecto: Alberto Sueiro Posada

Validación e Verificación de Software

1. Descripción do proxecto

Este proyecto emula el comportamiento de una aplicación de reproducción de música.

Los contenidos deben tener siempre una duración, un nombre y una lista de reproducción (que explicaremos en detalle más adelante). Los tipos de contenido son:

- Canciones, el contenido simple en el cual se basa nuestro proyecto.
- Anuncio, este contenido representa la publicidad. Su duración siempre es de 5 segundos.
- Emisora, contenido compuesto por otros tipos de contenido. No tienen duración propia, ya que su duración es la suma de la duración de todos sus contenidos.

Todos los contenidos tienen una lista de reproducción, independientemente de que sean simples (anuncio, canción) o compuestos (emisora).

Si se trata de devolver la lista de reproducción de un contenido simple se devolverá una lista con solo dicho contenido, mientras que si se trata de una emisora, devolverá la lista de los contenidos que la componen (en este caso NO devolverá la propia emisora).

Ahora seguimos con las especificaciones de los servidores

En el proyecto tenemos 2 tipos de servidores: simple y con respaldo. Su principal diferencia radica en el comportamiento ante una búsqueda vacía. Si un servidor simple realiza una búsqueda que como resultado devuelve una lista vacía, no hace nada más. En cambio, un servidor con respaldo, realizará la misma búsqueda en su servidor asociado.

Ambos servidores pueden introducir anuncios en una búsqueda, siguiendo una serie de especificaciones comentadas más adelante.

Cuando se da de alta en un servidor, se devuelve un token con 10 usos.

2. Estado actual

- **Alta en un servidor.** Esta funcionalidad representa el login en el servidor. Los usuarios con token válido tendrán ventajas (visualizar contenidos sin anuncios) respecto a los que no estén logueados. El token caduca después de devolver 10 contenidos.
- **Baja en un servidor.** Esta representaría el logout en el servidor. Es necesario que se le pase el token que se quiere dar de baja. Si se tratase de usar un token eliminado, lanzaría un error o se trataría como un usuario sin loguearse, según proceda.
- **Agregar contenido en un servidor.** Añade contenido nuevo (de cualquier tipo) al servidor. Sólo un token especial, el administrador, puede agregar contenidos.
- **Eliminar contenido en un servidor.** Busca el contenido y lo elimina del servidor. Solo un token especial, el administrador, puede eliminar contenido de un servidor. En caso de no encontrar el contenido, NO lanza error.

- **Buscar contenido en un servidor.** A partir de una cadena y un token dados, busca en los contenidos del servidor todas las coincidencias.

Los servidores pueden devolver anuncios, pero siguen una serie de reglas:

- Una búsqueda con un token válido nunca introducirá anuncios entre los contenidos.
- Si no se le pasa un token introducirá un anuncio al inicio de la lista y otro por cada 3 contenidos.

En el siguiente apartado se presentan las pruebas realizadas.

2.1. Componentes avaluados

En el proyecto hay dos componentes clave

- Contenido, que puede ser Emisora, Anuncio, Cancion. Estas no tienen lógica del sistema por lo que no se han hecho pruebas sobre estas.
- Servidor, que puede ser con respaldo o simple. Para esto se han creado pruebas para comprobar que todas las funcionalidades funcionan correctamente y se han hecho pruebas negativas para la funcionalidad **buscar contenido en un servidor** y añadir anuncios en las búsquedas sin tokens

Después de ejecutarlas se han corregido los errores que se detectaron.

3. Especificación de pruebas

En todas las pruebas se ha usado un servidor (simple o con respaldo) con 11 canciones y una emisora con 4 canciones y 1 anuncio.

- Agregar contenido correctamente al servidor. Para esto es necesario usar el token especial, ya que solo con ese es posible modificar el contenido de los servidores. Como entrada se ha utilizado el token especial y se agregó de nuevo la canción 4. No tiene salidas, simplemente añade la canción al servidor.
- Eliminar contenido de un servidor. Igual que en el anterior, es necesario usar el token especial. Se busca y borra la canción 4 y se buscó después. No encuentro la canción, por el borrado funciona correctamente. Se ha utilizado el token especial. No tiene salidas
- Agregar contenido a un servidor incorrectamente. En este se ha usado un token normal, para comprobar que no se puede usar un token no especial. Se da de alta un token y se intenta utilizar el token creado. Da error de token inválido, por lo que se comprueba que no se puede eliminar con un token no especial.
- Eliminar contenido de un servidor. De forma análoga al anterior, se ha usado un token normal, produciéndose un error. Se da de alta un token y se usa para borrar un contenido del servidor. Da un error de token inválido, por lo que se comprueba que no se puede borrar sin el token especial.
- Buscar contenido con un token no válido. Con esto se comprueba que en caso de no tener un token válido, ni siquiera se podrá realizar una búsqueda satisfactoria no vacía. Se usa un token no válido para realizar la búsqueda, la salida es una lista vacía.

- Dar de baja un token. Una vez dado de baja el token se comprueba si se puede realizar una búsqueda con el token, dando un error por ser un token inválido. Se da de baja un token y se intenta realizar una búsqueda con el mismo. Se produce un error en la búsqueda.
- Buscar 3 veces contenido sin token y recibir un anuncio. Cada 3 búsquedas sin token se debe obtener un anuncio. Se crea un token y se busca varias veces un contenido, cada 3 veces se comprueba que devuelve una lista con el contenido y un anuncio. La salida en una lista de contenidos con un anuncio.
- Comprobar que los tokens válidos caducan después de 10 usos. Se realiza 11 búsquedas, en la búsqueda número 11 ya no tiene un token válido, por lo que se produce un error de token inválido.

Para el servidor con respaldo se probó también una búsqueda que no devolviese ningún contenido para que lo buscara en el servidor de respaldo asociado. Se busca lacancion5, que solo está en el servidor del respaldo, y de salida tenemos una lista con dicho contenido.

4. Rexisto de probas

4.1. Pruebas dinámicas de caja blanca - Junit

Con este framework se nos ha permitido realizar una ejecución de las clases del proyecto de una manera controlada, de manera que introduciendo una serie de datos de entrada, evaluábamos el valor de salida esperado o el comportamiento correcto.

Los casos de prueba de los contenidos eran bastante sencillos, dado que abarcaban solo las funcionalidades de devolver nombre, duración o lista de reproducción. Y, en el caso de la emisora, agregar o eliminar alguno de sus contenidos.

Los casos de pruebas creados son los mismo que los explicados en el apartado anterior:

- Agregar contenido correctamente al servidor. Para esto es necesario usar el token especial, ya que solo con ese es posible modificar el contenido de los servidores
- Eliminar contenido de un servidor. Igual que en el anterior, es necesario usar el token especial.
- Agregar contenido a un servidor incorrectamente. En este se ha usado un token normal, para comprobar que no se puede usar un token no especial.
- eliminar contenido de un servidor. De forma análoga al anterior, se ha usado un token normal, produciéndose un error.
- Buscar contenido con un token no válido. Con esto se comprueba que en caso de no tener un token válido, ni siquiera se podrá realizar una búsqueda satisfactoria no vacía.
- Dar de baja un token. Una vez dado de baja el token se comprueba si se puede realizar una búsqueda con el token, dando un error por ser un token inválido.
- Buscar 3 veces contenido sin token y recibir un anuncio. Cada 3 búsquedas sin token se debe obtener un anuncio.

- Comprobar que los tokens válidos caducan después de 10 usos. Una vez caducado, el token es simi

Dichos test se han ejecutado para los servidores sin respaldo como para los respaldados. A mayores para los servidores con respaldo se ha probado que en caso de no encontrar el contenido pedido en el servidor principal, se busque en el servidor de respaldo con éxito.

4.2. Pruebas estáticas de caja blanca - Findbugs

FindBugs es una herramienta de análisis estático que permite la búsqueda de posibles errores durante el desarrollo de software.

Se ha ejecutado *mvn findbugs:gui*. Saldrá una ventana en la que se pueden ver los errores encontrados por la herramienta y pistas sobre cómo corregirlas.

4.3. Mutación de código - Pitest

La mutación de código consiste en introducir pequeños cambios en el código de manera que se pueda observar cuál es la reacción y comprobar la consistencia del código. En caso de fallar la ejecución, significa que ha sido satisfactoria y el "mutante" muere. En caso contrario el "mutante" vive, lo que nos dice que existe una mala especificación en nuestro código.

Para ejecutarlo usar *mvn org.pitest:pitest-maven:mutationCoverage*. Esto generará un html con la cobertura de mutación y cobertura de líneas.

No se ha conseguido matar a todos los mutantes generados.

4.4. Cobertura de líneas

Para esto se ha aprovechado la herramienta anterior, que con la ejecución también se obtiene la cobertura de líneas.

También se usó la herramienta cobertura, ejecutada con *mvn cobertura:cobertura* para tener así también la cobertura de ramas.

4.5. Estrés

Se han creado test usando la herramienta Jetm pero no se han conseguido ejecutar satisfactoriamente.

4.6. Selección de datos aleatorios

Al igual que en el anterior, se ha intentado crear test usando correctamente la herramienta jcheck, pero no se ha logrado.

5. Rexistro de erros

5.1. Pruebas dinámicas

Con la ejección de las pruebas se ha detectado errores con la inserción de anuncios cada 3 búsquedas. Esto se solucionó en lo servidores simples en la primera semana de prácticas.

La siguiente semana se probó de forma similar los servidores con respaldo y se descubrió el mismo error. Este también se solucionó en dicha semana de prácticas

6. Estadísticas

No se llevo un seguimiento regular durante la práctica, por lo que es difícil cumplimentar este apartado