



PROYECTO VLC

Javier Barrientos Fernandez y Samuel Ferragut Molina

INDICE

1. Explicación del proyecto 2

1.1 Proceso 2

2. Arquitectura de la aplicación y tecnologías utilizadas 4

3. Diagrama de componentes..... 5

4. Diagrama Clokify 6

1. Explicación del proyecto

Construir un programa que genere una lista de reproducción aleatoria (50 canciones como mínimo) de las canciones que tienes en tu biblioteca de canciones de tu máquina. El script llama a VLC una vez que la lista esta creada y la música empieza a sonar en tu pc.

1.1 Proceso

En este proyecto hemos usado la metodología en cascada, decidimos utilizar esta metodología porque nos parecía la más idónea ya que fuimos haciendo modulo por modulo comprobando que la salida que obteníamos era la esperada y como ultimo los juntamos todos en uno.

También en este proyecto decidimos implementar todo lo aprendido hasta ahora de la programación por módulos, creando 4 ficheros.

El main que es el directorio que contiene todos los imports de los demás módulos y donde definimos las variables de la ruta de XML y vlc.

El fichero de importarXml donde parseamos el XML y creamos un diccionario con la ruta de la canción y la id.

El fichero listaDeCanciones donde usamos el diccionario para coger llaves y generar una lista aleatoria con las rutas de esas canciones, y donde convertimos esa lista en un string para enviarlo al VLC.

Y como ultimo fichero accesoVlc en este fichero usamos el resultado del modulo de listaDeCanciones que es un string con la ruta y la variable de la ruta del vlc para mandarlo por consola de windows y que se ejecute.

Nuestro trabajo esta enfocado en usar la metodología en cascada, elegimos esta metodología porque fue la que mas se adaptaba a nuestra forma de trabajar, tambien decidimos implementar todo lo aprendido hasta hoy sobre la programación modulada.

En nuestra programación modulada tenemos el proyecto dividido en 4 módulos, cada uno de ellos lo importamos a un modulo llamado main donde.

El fichero main contiene todos los imports de los demás módulos, eso quiere decir que todos los módulos estan conectados entre si por el modulo main, aqui tambien tenemos indicadas las variables de las rutas del XML y donde esta instalado vlc, de esta forma si el usuario quiere cambiar dependiendo de donde tenga el XML o el fichero XML solo tendra que buscar en el fichero principal.

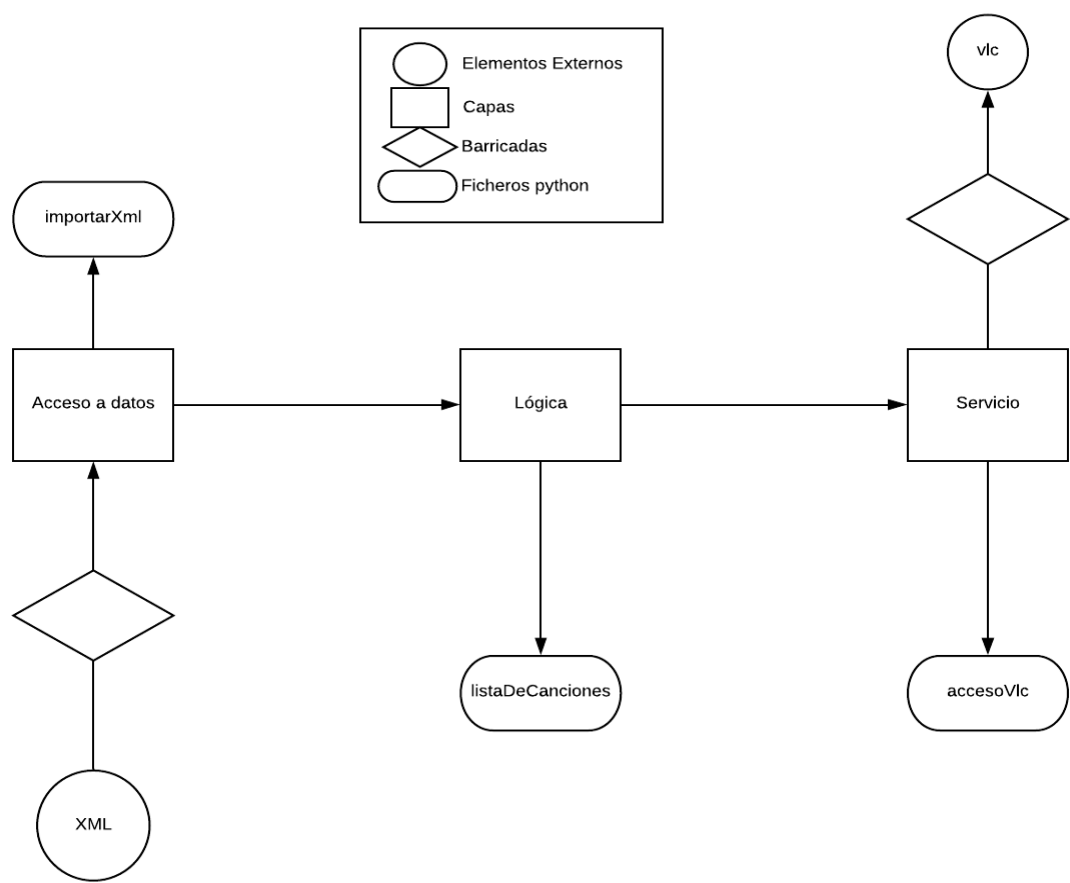
El fichero importarXml esta definido dentro de main como data, esto lo usaremos mas adelante para el modulo de canciones aleatorias. Dentro del modulo de importarXml hemos importado la libreria de xml.etree para poder usar todas sus funciones, antes de parsear el XML en memoria tenemos una barricada que comprueba que ese fichero exista, en el caso de que no exista cortarea el programa y te imprimira un mensaje por terminal. Si el fichero esta bien y no hay ningun problema el fichero XML se parseara a memoria, cogiendo el contenido de la variable rutaXml del fichero main, inicializamos un diccionario vacio, recorremos el primer elemento del fichero XML buscando dentro de tracks y en todo sus hijos el elemento 'track', seguido en la variable de nombreCancion le añadimos como texto lo que esta dentro de la etiqueta 'ruta' dentro del padre track, y como segunda variable que guarde el id o atributo del elemento track, como ultimo añadimos al diccionario cada uno de esos elementos

nombreCancion como key e idCancion como value, antes de devolver el resultado del diccionario hemos creado un assert que compruebe que el diccionario no esta vacio.

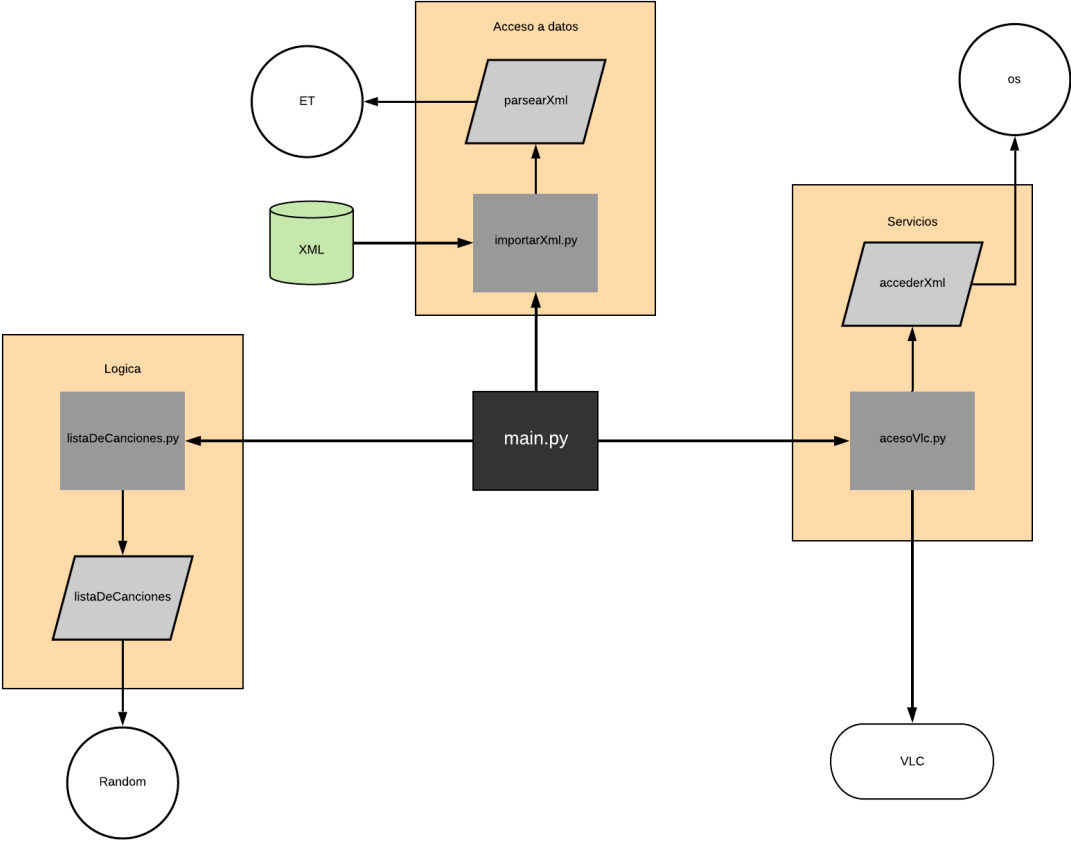
El fichero listaDeCanciones tiene como import la funcion random ya que lo usaremos para coger un elemento aleatorio de la variable diccionario, esta variable esta definida en el fichero main con el nombre de data, antes de nada hemos creado un assert que compruebe que el diccionario que nosotros le pasamos es un diccionario, ahora si una vez comprobado eso inicializamos un contador y una lista vacia, abrimos con un while que mientras el contador sea menor que la longitud del diccionario siga incrementando, iniciamos tres variables, la variable de cancion que nos pasa las keys del diccionario a una lista y elige aleatoriamente una cancion, la variable de comillas que la usaremos para añadir comillas dobles a nuestro string, y como ultimo sera la cancion concatenado con la variable de comillas dobles de esta forma los espacios que tienen las canciones no seran un problema para vlc. Las siguientes lineas son dos asserts que comprueban que la lista no esta vacia, que la lista tenga la misma longitud que el diccionario, despues iniciamos un string que lo usaremos para convertir esa lista en una string y un ultimo assert que compruebe que el string no esta vacio.

En el ultimo modulo es en el que accedemos a vlc como no es un modulo que recibe informacion nuestra tendremos que hacer una barricada para comprobar que la ruta que tenemos en la variable rutaVlc definida en el fichero main como rutaVlc no este mal escrita o vlc no se encuentre ahi, el try funciona intentando ejecutar la rutaVlc el except es OSError un error del sistema que nos indica que esa ruta no existe mostrando por pantalla el mensaje que querramos, despues de eso queremos comprobar que la informacion que nosotros le mandamos es correcta y creamos dos assert que comprueben que esas dos variables sean string y le mandamos por consola de comandos la ruta del vlc y el string con canciones random concatenado por un espacio en blanco. El modulo que importamos que es os nos sirve para usar popen.

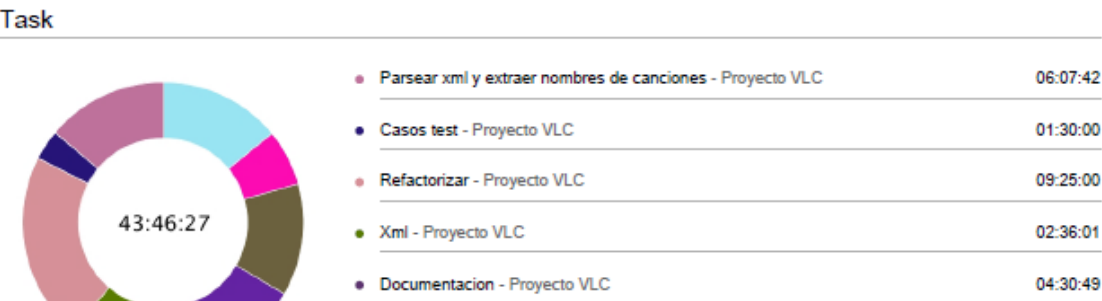
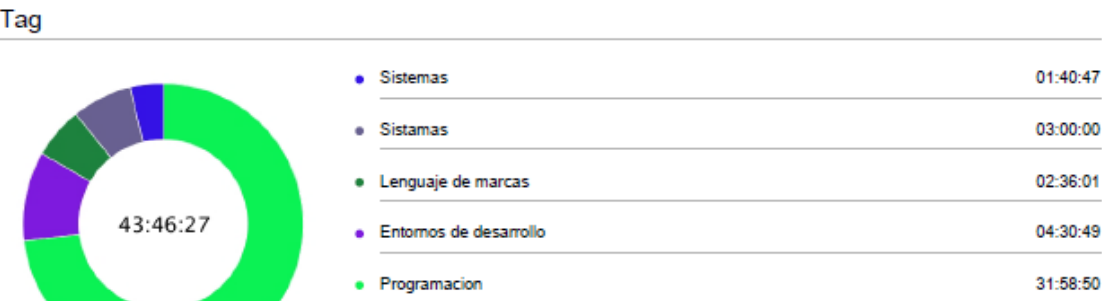
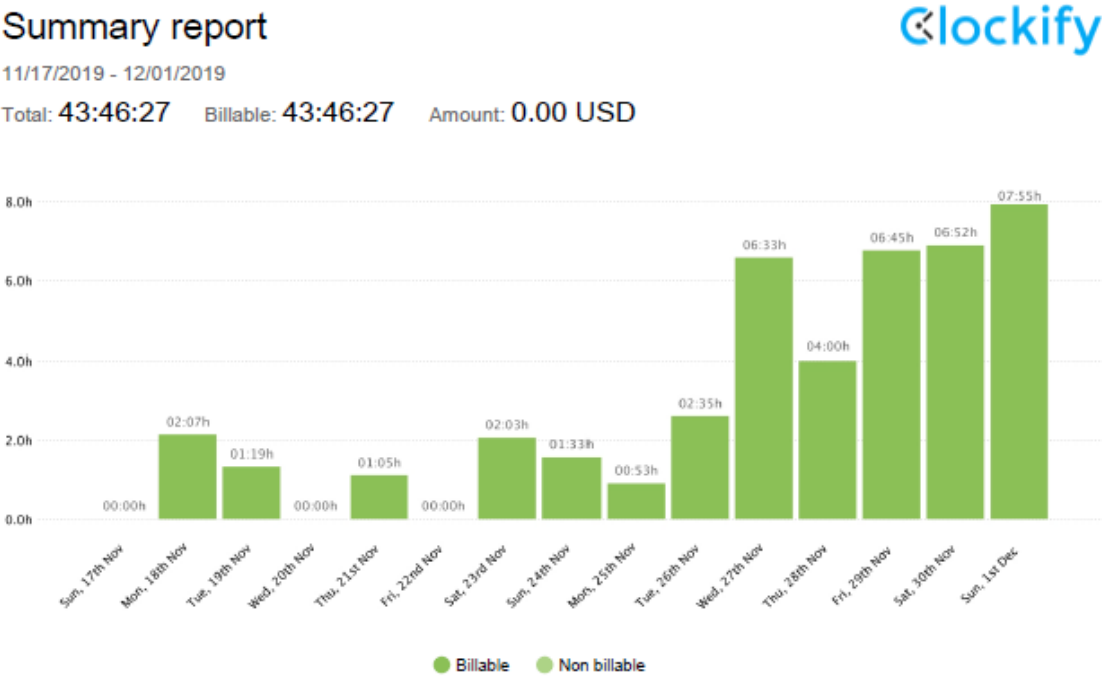
2. Arquitectura de la aplicación y tecnologías utilizadas



3. Diagrama de componentes



4. Diagrama Clotify



2019 - 2020

PROGRAMACION

| | |
|--|----------|
| ● Organización - Proyecto VLC | 00:18:13 |
| ● Acceder a vlc - Proyecto VLC | 04:40:47 |
| ● Lista random - Proyecto VLC | 05:38:35 |
| ● Barricadas acceso a datos - Proyecto VLC | 02:50:04 |
| ● Archivo main - Proyecto VLC | 06:09:16 |

| Tag/Task | Duration | Amount |
|---|-----------------|-----------------|
| Sistemas | 01:40:47 | 0.00 USD |
| Acceder a vlc - Proyecto VLC | 01:40:47 | 0.00 USD |
| Sistamas | 03:00:00 | 0.00 USD |
| Acceder a vlc - Proyecto VLC | 03:00:00 | 0.00 USD |
| Lenguaje de marcas | 02:36:01 | 0.00 USD |
| Xml - Proyecto VLC | 02:36:01 | 0.00 USD |
| Entornos de desarrollo | 04:30:49 | 0.00 USD |
| Documentacion - Proyecto VLC | 04:30:49 | 0.00 USD |
| Programacion | 31:58:50 | 0.00 USD |
| Parsear xml y extraer nombres de canciones - Proyecto VLC | 06:07:42 | 0.00 USD |
| Casos test - Proyecto VLC | 01:30:00 | 0.00 USD |
| Refactorizar - Proyecto VLC | 09:25:00 | 0.00 USD |
| Organización - Proyecto VLC | 00:18:13 | 0.00 USD |
| Lista random - Proyecto VLC | 05:38:35 | 0.00 USD |
| Barricadas acceso a datos - Proyecto VLC | 02:50:04 | 0.00 USD |
| Archivo main - Proyecto VLC | 06:09:16 | 0.00 USD |