

UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

INGENIERO DE TELECOMUNICACIÓN

LENGUAJES Y SISTEMAS INFORMÁTICOS

PROYECTO FIN DE CARRERA Nº 3133481

**GRABACION Y REPRODUCCION DE AUDIO-VIDEO EN
UN PUESTO MULTIPANTALLA**

DOCUMENTO Nº 3

DISEÑO

YÚSEF HABIB FERNÁNDEZ
NOVIEMBRE 2014

TUTOR: CLAUDIO DE LA RIVA
COTUTOR: MARCELINO AGUINAGA

TABLA DE CONTENIDO

1.	DESCRIPCIÓN GENERAL DE LA ARQUITECTURA.....	5
1.1	Posición de trabajo del controlador.....	7
2.	GRABACIÓN DE LA ESTACIÓN.....	8
2.1	Creación de una nueva sesión de grabación.....	8
2.2	Iniciación de nueva sesión de grabación.....	9
2.2.1	Script videoRecord.sh.....	12
2.2.2	Script audioRecord.sh.....	13
2.2.3	Códec empleados en la grabación.....	14
2.2.3.1	x264.....	14
2.2.3.2	libmp3lame.....	15
3.	GENERACION DE ARCHIVOS REPRODUCIBLES.....	16
3.1	Concatenación de archivos.....	17
3.2	Superposición de los archivos.....	18
4.	REPRODUCCIÓN DE LA ESTACIÓN.....	19
5.	ESTRUCTURA DE DIRECTORIOS.....	20
6.	SOFTWARE EMPLEADO.....	21
6.1	FFmpeg.....	21
6.1.1	ffmpeg.....	21
6.1.1.1	x11grab.....	22
6.1.1.2	concat.....	23
6.1.2	ffserver.....	23
6.1.3	ffplay.....	23
6.1.4	ffprobe.....	23
6.2	Reproducción.....	24
6.2.1	VLC.....	24
6.3	Herramientas adicionales.....	24
6.3.1	Git.....	24
6.3.2	Oracle VM VirtualBox.....	24
6.3.3	CentOs (Community ENTERprise Operating System).....	24
6.3.4	Merlin.....	25
6.3.5	Sublime Text 2.....	25
7.	ANEXO.....	26
7.1	Acrónimos y definiciones.....	26

TABLA DE IMÁGENES

Figura 1: Diagrama del script Menu.sh	6
Figura 2: Arquitectura común de una POS REPRO	7
Figura 3: Diagrama del script saveSession.sh	9
Figura 4: Diagrama del script record.sh	11
Figura 5: Diagrama del script videRecord.sh.....	13
Figura 6: Diagrama del script audioRecord.sh	14
Figura 7: Diagrama del script fileGeneration.sh	16
Figura 8: Diagrama del script concat.sh	17
Figura 9: Superposición de la pantalla principal y la auxiliar	18
Figura 10: Diagrama del script launchVLC.sh.....	19
Figura 11: Árbol de directorios tras la instalación	20
Figura 12: Diagrama del tratamiento de archivos multimedia por parte de FFmpeg	22

IDENTIFICACIÓN DEL DOCUMENTO

Proyecto	Grabación y reproducción de audio-video en un puesto multipantalla
Nº proyecto	3133481
Autor	Yúsef Habib Fernández
Tutor	Claudio de la Riva Álvarez
Cotutor	Marcelino Aguinaga Izquierdo
Documento	Diseño y funcionamiento
Fecha	Noviembre 2014

1. DESCRIPCIÓN GENERAL DE LA ARQUITECTURA

A lo largo de este documento se va a proceder a mostrar el diseño y funcionamiento del producto desarrollado. Este se estructura en un conjunto de scripts que realizan diversas funciones, y que son ofrecidas como funcionalidades del producto al usuario, a través del menú principal del programa.

El producto software desarrollado es implementado a través de una serie de scripts para bash, los cuales cumplen los requisitos del cliente mostrados en el Documento 1. Estos scripts harán uso de herramientas software liberadas bajo licencia GNU GPL, por lo que se garantiza la libertad para modificar y compartir el software cubierto por ella, asegurando la libertad de este software para todos los usuarios, ya sean individuos o empresas

Los scripts que son una solución enfocada a la línea de comandos de Linux, fueron escogidos como solución de diseño ya que permiten minimizar el gasto de recursos en el equipo del controlador, a la par que ofrecen una interfaz básica que no ofrezca distracciones de su trabajo. Otra ventaja asociada a esta elección es que los scripts permiten su fácil ejecución desde un equipo remoto, en caso de que no se busque que sea el controlador el que se encargue de su inicialización y configuración. En el diagrama 1, se puede observar cómo están conectados cada uno de estos scripts entre sí.

Para realizar este análisis se va a proceder con el estudio de cada una de las entradas del menú principal que muestra por pantalla este programa. Es por tanto que el documento se va a centrar en los siguientes en cinco apartados:

1. En la sección 2 se estudia el proceso de grabación de la estación. Esto incluye la creación e inicialización de una nueva sesión de grabación por parte del usuario del producto.
2. En la sección 3 se realiza el estudio de la generación de los archivos finales, es decir los procesos de concatenación de los archivos generados y de superposición de estos.
3. En la sección 4, se muestra cómo se desarrolla el proceso de reproducción de las diferentes sesiones sobre el equipo original.
4. En la sección 5 se expone el informe con todas las herramientas software necesarias para el funcionamiento del producto, y también todas las herramientas empleadas para el desarrollo de este.
5. Por último en la sección 6, se muestra un anexo con información relacionada a los estilos empleados en el código y en el desarrollo de los diagramas. También se incluye un glosario con los acrónimos y definiciones empleadas a lo largo de este tercer documento.

Cada uno de estos procesos irá acompañado de una descripción de su funcionamiento y junto a su diagrama de flujos de estados para facilitar su comprensión.

En la figura 1 se puede observar los diferentes scripts que se lanzan, es decir las diferentes funcionalidades que se activan, en función de la opción elegida por el usuario ya sea en el menú principal o en algún submenú.

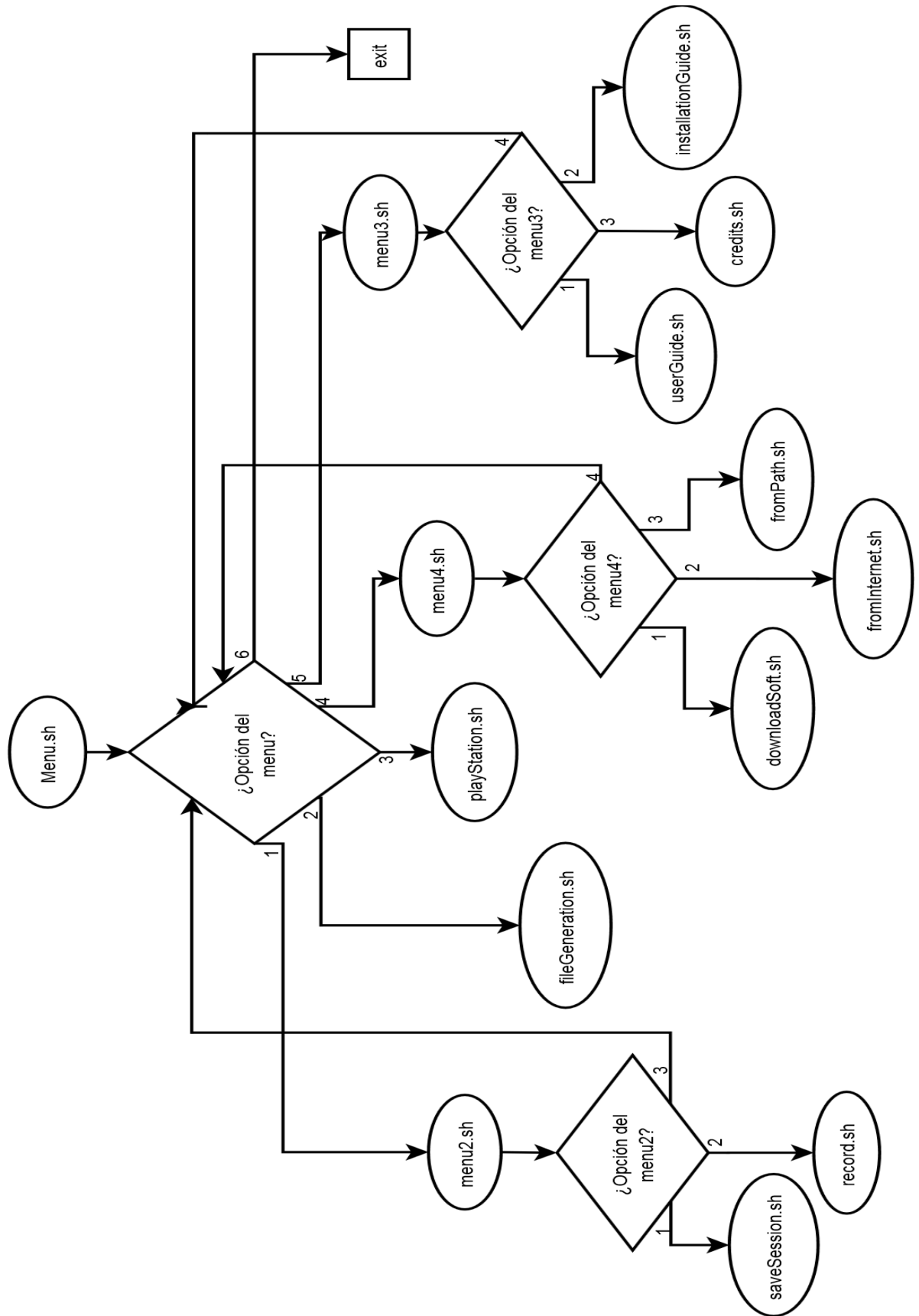


Figura 1: Diagrama del script Menu.sh

1.1 Posición de trabajo del controlador

Este producto software ha sido desarrollado para ser ejecutado en la estación de trabajo del controlador. Esta posición mejor conocida por CWP está formada por los siguientes elementos hardware:

- Un monitor primario 2k: (resolución 2048x2048) para la presentación de la situación aérea.
- Un monitor secundario 1k: (resolución 1280x1024) para la presentación de información auxiliar.
- Conjunto de periféricos formados por teclado, ratón y micrófono de entrada.
- La posición recibe el nombre de Torre, y contiene todo el hardware necesario, es decir el procesador que suele ser SPARC, la tarjeta gráfica, conectores, etc. Actualmente trabajan sobre el sistema operativo Red Hat Enterprise 5.0 aunque algunos equipos ya cuentan con la versión actualizada Red Hat Enterprise 6.0.

A continuación, en la figura 2 se ilustra la arquitectura común de una POS REPRO:

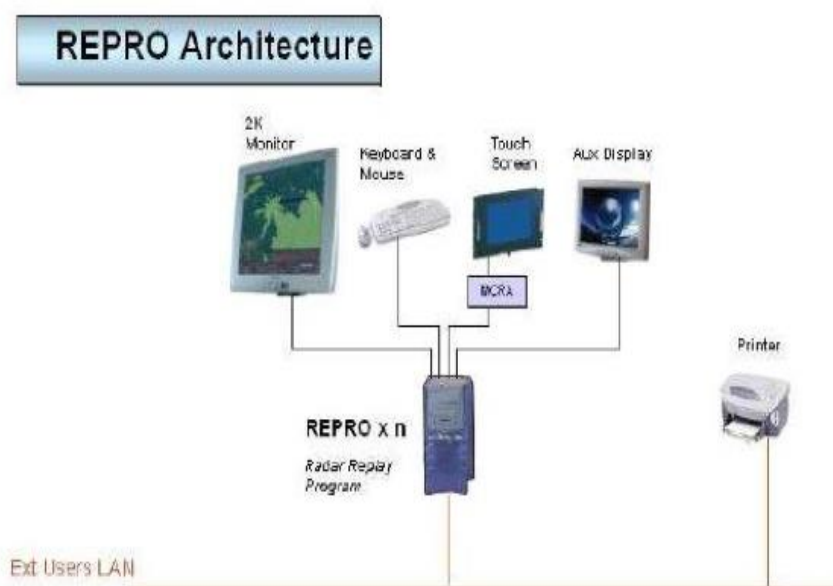


Figura 2: Arquitectura común de una POS REPRO

2. GRABACIÓN DE LA ESTACIÓN

En este segundo apartado se va a proceder con el análisis de diseño y funcionamiento de cada uno de los bloques que componen el la grabación de la estación. Es en esta sección donde se describen los scripts necesarios para lograr la mayoría de las tareas solicitadas a este proyecto:

- Configuración de una sesión de grabación.
- Inicio de una sesión de grabación.
- Selección de una configuración previamente creada como el formato para la nueva sesión.
- Creación de los archivos de video intermedios y finales.

2.1 Creación de una nueva sesión de grabación

Se inicializa el script interactivo “saveSession.sh”. Este a través de un diálogo con el usuario recopilará toda la información necesaria para una nueva sesión de grabación. Esta es:

- Nombre de la nueva sesión que se va a crear.
- Ruta a la biblioteca de FFmpeg, este valor es devuelto tras lanzar el asistente de instalación.
- Ruta al directorio de grabación donde se desea almacenar los archivos generados.
- Resolución de la pantalla principal y secundaria. Primero se auto ejecuta en una nueva ventana el script “resoluciones.sh” que muestra las resoluciones soportadas por cada uno de los monitores al igual que la resolución óptima de estos.
- Conjunto de características asociadas al video: frames¹ por segundo (fps), preset¹ para la codificación H264 y crf¹ para esta codificación.
- Por último información relacionada con la duración la sesión de grabación y la duración de los archivos finales.

En la figura 3, se puede observar el proceso que se lleva a cabo. Una vez lanzado el script, y si es la primera vez que se ejecuta este, se procede a la creación de una carpeta que permita almacenar la configuración que se va a crear y las futuras posibles. A continuación el usuario interactúa con el script mediante una serie de diálogos, en los cuales se le irá preguntando por los valores que quiere introducir para cada uno de los campos que se han de rellenar. Una vez finalizado, se mostrará un resumen con todos los valores seleccionados y se pedirá al usuario que confirme si está de acuerdo con cada uno de estos. Si la respuesta es afirmativa se procederá a la creación de un archivo de tipo “texto plano” con el nombre seleccionado en la carpeta previamente creada.

¹ Todos estos valores son especificados en páginas posteriores de este documento.

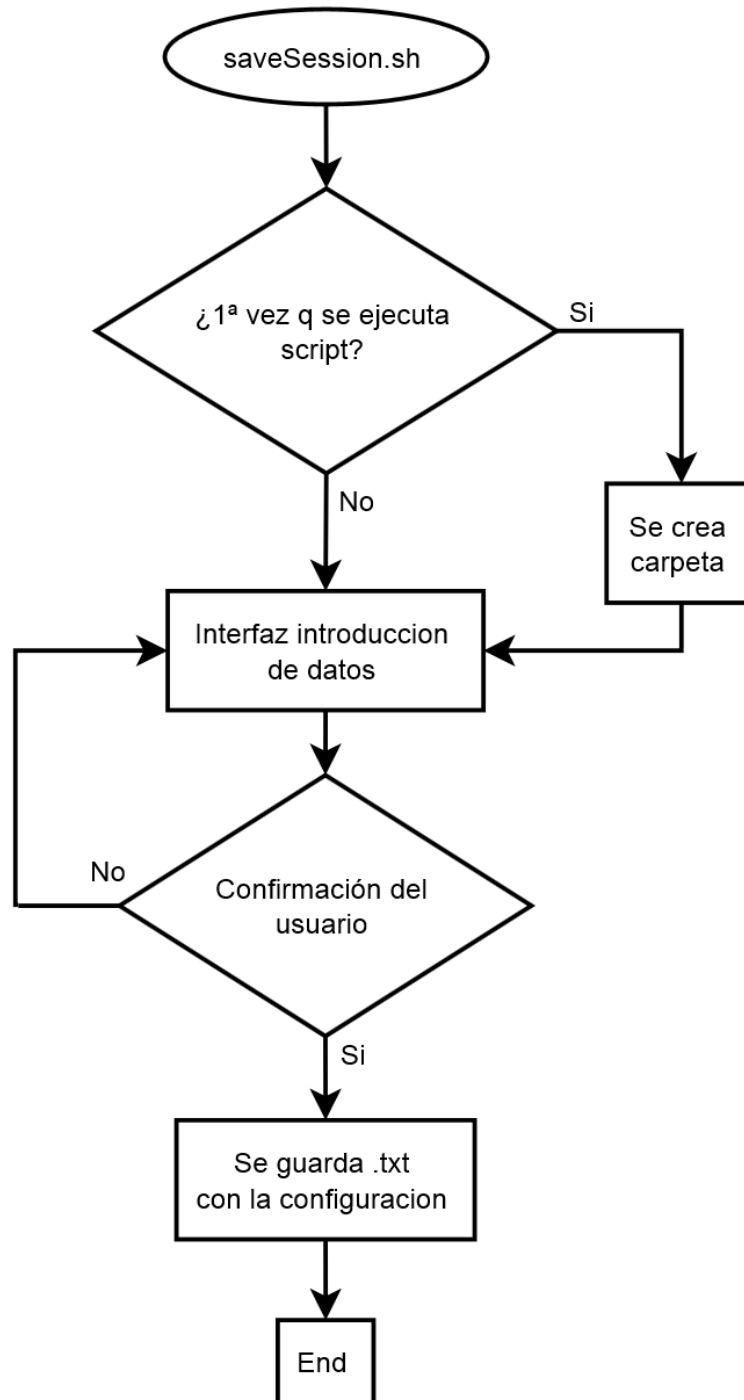


Figura 3: Diagrama del script saveSession.sh

2.2 Iniciación de nueva sesión de grabación

Al seleccionar esta opción del menú se inicializa el script interactivo “record.sh”. Este es el encargado de todo el proceso de grabación de los dos monitores y la entrada de audio a través de un micrófono.

En la figura 4 se puede observar el proceso que es llevado a cabo por el programa. Inicialmente el script le pregunta al usuario por el tipo de configuración que desea emplear para la nueva sesión de grabación. Una vez seleccionado este a partir de la introducción de su nombre, se lanzan tres nuevos scripts encargados de la grabación de cada una de los flujos audiovisuales. Dos instancias del script “videoRecord.sh” con diferentes parámetros de configuración para la grabación de cada una de las pantallas, y el script “audioRecord.sh” encargado de la grabación del flujo de audio que entra a través del micrófono.

El script se encarga de crear una carpeta nueva cada vez que una sesión es lanzada, para poder tener organizados todos los archivos audiovisuales pertenecientes a esta sesión. Además el nombre que le otorga a esta carpeta, es igual a la hora en la que se inició.

El programa recibe como entrada tanto el video como el audio en bruto procedente del equipo, por lo que el primer paso que realiza es crear la sesión a partir de este flujo multimedia. Para ello el script accede al archivo de configuración donde el usuario ya definió la duración tanto de la sesión de grabación como de los archivos finales, a través de las variables “SESSIONTIME” y “RECORDTIME”. Con ello calcula y almacena en una variable el número de interacciones que debe realizar el bucle de trabajo principal para grabar la sesión con la duración deseada a partir de la siguiente expresión:

$$contador = \frac{SESSIONTIME}{10}$$

Donde la variable “SESSIONTIME” indica la duración de la sesión total durante la cual el programa estará activo.

El script se encarga de ir decrementando esta variable con cada iteración del bucle encargado de lanzar las grabaciones de los flujos multimedia, hasta que “contador” llegue a cero, con lo que la sesión habrá finalizado y por ello no se grabará más.

El programa ofrece la posibilidad de finalizar la sesión actual de grabación en cualquier momento a través de la combinación de teclas:

Ctrl + c

Con este proceso el script lo que logra es generar un conjunto de archivos más pequeños. Estas “minis Sesiones” se almacenan en subcarpetas² dentro de la sesión principal con el objetivo de tenerlo todo estructurado

² Ver apartado Arquitectura de directorios donde se encuentra el diagrama con el árbol de directorios.

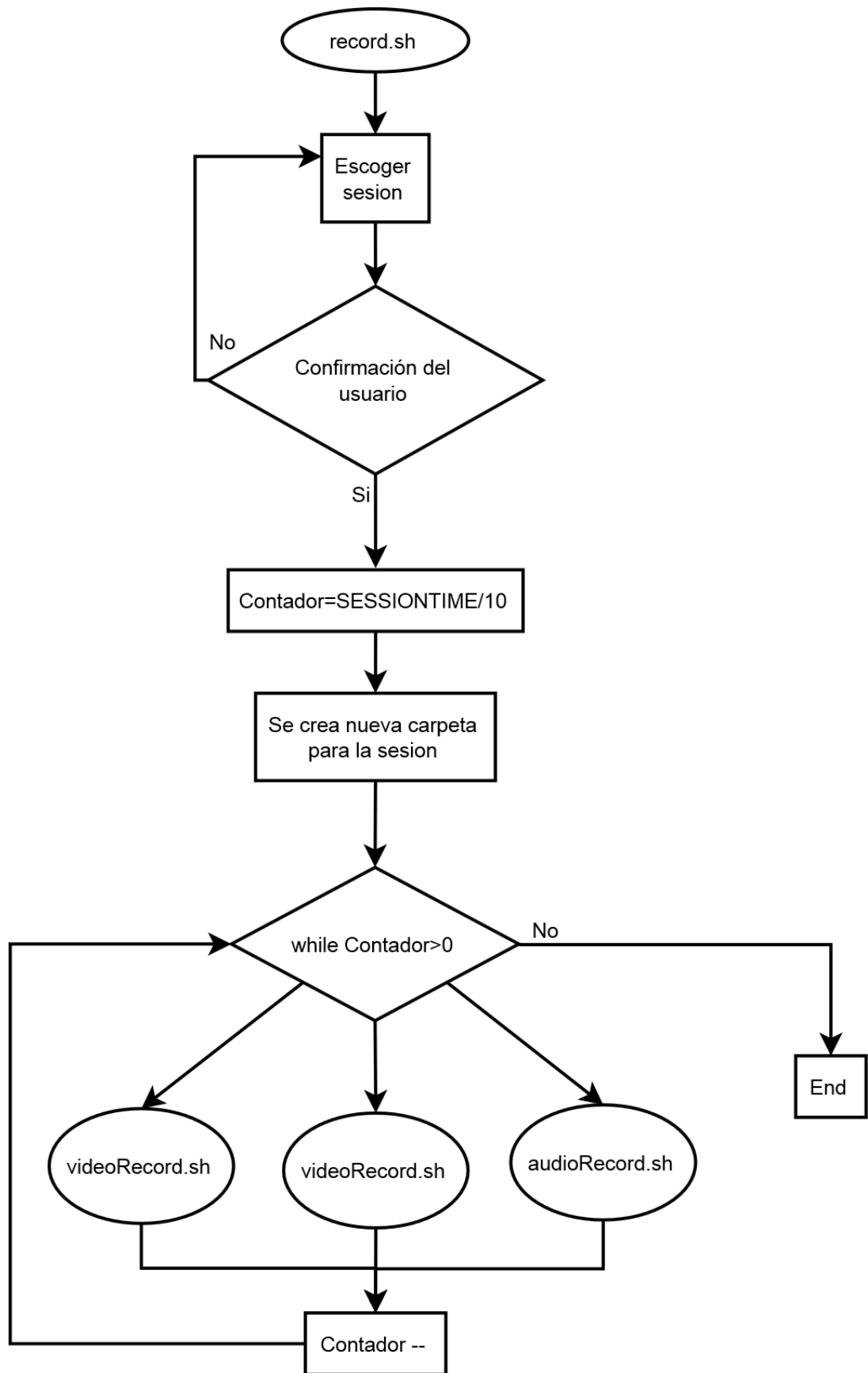


Figura 4: Diagrama del script record.sh

2.2.1 Script videoRecord.sh

Este script es encargado de la grabación de un monitor. Trabaja sobre la tarjeta gráfica para la obtención del video del sistema, grabando todo lo que se genera en esta.

En su llama se le pasan como argumentos los valores que especifican que tipo de grabación va a realizar y de que monitor. Estas variables son:

- La ruta a las librerías FFmpeg y la ruta a la carpeta donde se almacenarán los archivos generados.
- La resolución del monitor a grabar.
- Los valores de frames por segundo, crf, preset.
- Duración de los archivos de grabación.
- Posición del monitor a grabar respecto al principal. Esto se indica mediante coordenadas a las que se le suman los valores de ancho y alto del monitor en función de la configuración del auxiliar.

En la figura 5 se puede observar el proceso que es llevado a cabo por este script. Comienza almacenando en variables los valores pasados como atributos en su llamada. A continuación entre en un bucle en el cual se generarán hasta diez instancias del programa ffmpeg, el cual se encarga del proceso de grabación. Dentro de este bucle el primer paso es generar un nombre para el archivo de video q se va a generar, único. Para ello mediante la función “date”, otorga un nombre al archivo con el valor de la hora con minutos, segundos y cuatro decimales más. A continuación un ejemplo del nombre que recibiría una de es:

15:01:13:0836.mp4

Posteriormente llama a la función con todos los parámetros necesarios, esta que se encarga tanto de la grabación como del procesado del archivo generado en tiempo real. Se puede encontrar en la sección 5 del presente documento y en particular en el apartado FFmpeg, la explicación y ejemplos de uso de este comando:

ffmpeg -f x11grab

Para finalizar escribe el nombre de este archivo sobre un archivo de texto plano, ya que posteriormente será empleado por el script “concat.sh”. Una vez completadas las 10 iteraciones del bucle, saldrá de este para llamar al script anteriormente mencionado “concat.sh” y finalizando así su ejecución. En la sección 3.1 del presente documento se puede observar el funcionamiento de este script.

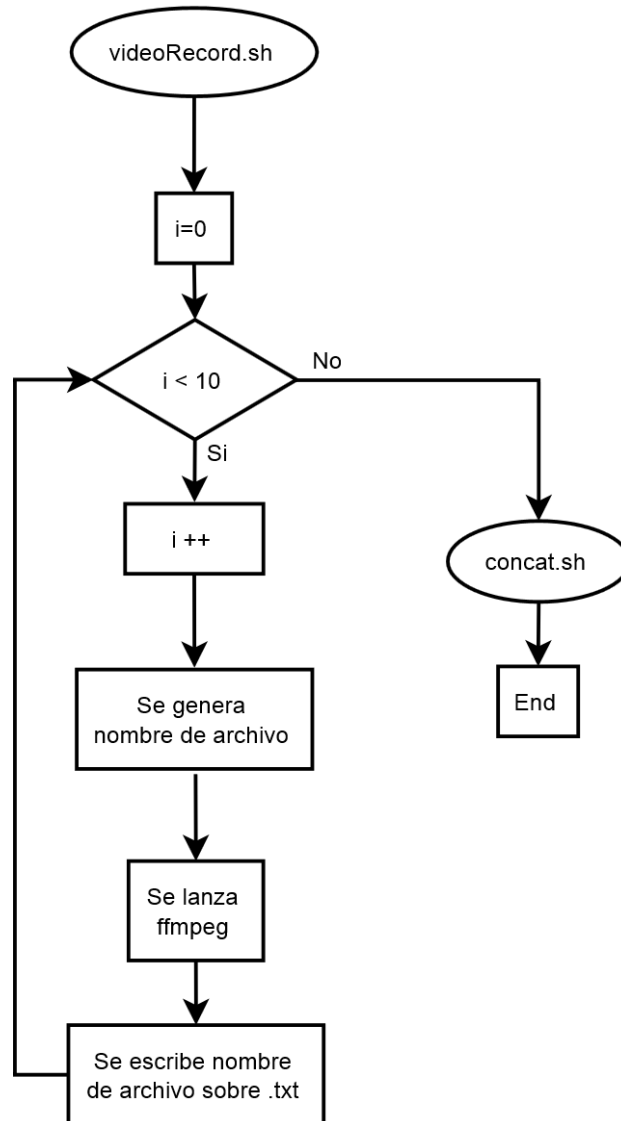


Figura 5: Diagrama del script videRecord.sh

2.2.2 Script audioRecord.sh

Este script es encargado de la grabación del flujo de audio. En su llama se le pasan como argumentos los valores que especifican que tipo de grabación va a realizar y de que monitor. Estas variables son:

- La ruta a las librerías FFmpeg y la ruta a la carpeta donde se almacenarán los archivos generados.
- Duración de los archivos de grabación.

En la figura 6 se puede observar el proceso que es llevado a cabo por este script. Como se puede observar tiene un funcionamiento muy similar al del script anterior “videoRecord.sh”. Las principales diferencias que se pueden encontrar son las relaciones a la codificación del flujo, ya que se emplea un códec diferente y que hay menos valores que especificar para la grabación de audio. Estos parámetros son pasados en la llamada al script. Pero en cuanto al proceso de

grabación, codificación, y generación de archivos finales sigue el mismo funcionamiento, como se puede comprobar en su diagrama.

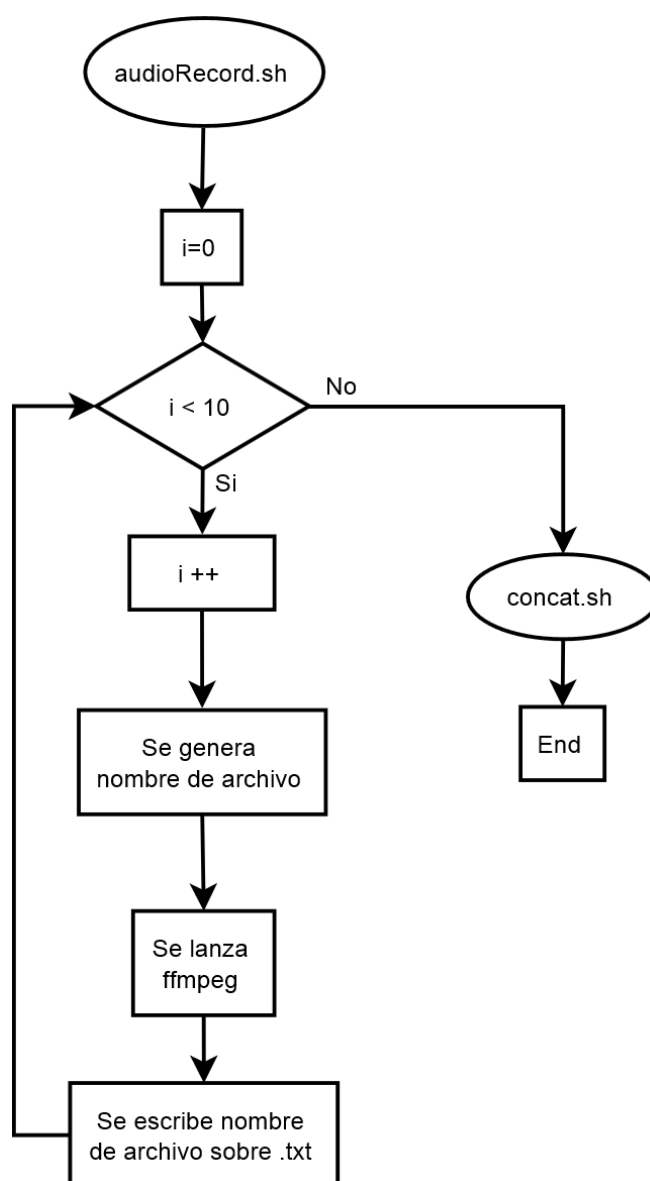


Figura 6: Diagrama del script audioRecord.sh

2.2.3 Códec empleados en la grabación

Son dos los códec empleados para el tratamiento de los archivos generados durante la grabación de la sesión, uno para la grabación del audio y otro para la grabación del video. A continuación un análisis de cada uno de estos.

2.2.3.1 x264

Es una librería de software, y de aplicación libre para la codificación de flujos de video al formato de compresión H.264/MPEG-4 AVC, y está liberada bajo los términos de GNU GPL.

Las características más importantes de este codificador son:

- Alta calidad de codificación y decodificación de video para aplicaciones de transmisión de video en tiempo real.
- Tamaño de archivo hasta tres veces menor que con archivos con códec MPEG-2.
- Doble de eficiente que MPEG-2.
- Fácil de integrar y cubre un amplio rango de formato de imágenes.

Las dos opciones de configuración más importantes son las siguientes:

- CFR: El rango de la escala del cuantificador va de 0 a 51. Donde 0 es sin pérdidas, 23 es el valor por defecto y 51 es el peor posible. Normalmente se emplea el valor de CFR más alto que ofrece la calidad suficiente. En este caso se ha optado por emplear un valor menor a la media debido al requisito de la grabación nativa.
- Preset: Determinan la velocidad de codificación para el ratio de compresión. Un valor bajo de preset ofrece mejor compresión pero necesita más tiempo. En cambio un valor alto, realiza la codificación en muy poco tiempo pero ofrece peor calidad.

Es uno de los formatos empleados para la codificación de todo el contenido multimedia en HTML5.

2.2.3.2 *libmp3lame*

Es una librería dentro del proyecto Lame que ofrece codificadores de alta calidad para la capa 3 de MPEG, es decir para MP3. Está liberado bajo licencia LGPL.

Actualmente es considerado el mejor codificador para tasas medio-altas y VBR.

3. GENERACION DE ARCHIVOS REPRODUCIBLES

Este es el proceso encargado de la generación de los archivos finales reproducibles, está formada por un solo script, que realiza dos tareas diferentes: la concatenación de los archivos finales y la generación del archivo superpuesto con los tres flujos audiovisuales.

En la figura 7 se puede observar el proceso que es llevado a cabo por este script. Este tras un breve dialogo con el usuario, con el fin de desplazarse hasta el directorio contenedor de la sesión a tratar, realiza dos tareas. La primera de ellas está relacionada con la concatenación de los archivos del audio y de la pantalla principal, y posteriormente la superposición de todos los flujos multimedia, es decir los dos videos más el audio.

El proceso concatenar el video del monitor principal y el audio se debe a que para poder reproducir la sesión sobre la estación de trabajo, se necesitan dos flujos audiovisuales y no tres, por lo que es necesario que dos de ellos se mezclen. En la sección 4 se explica esto en más detalle.

FFmpeg se encarga de todo este proceso a través del uso de filtros, que vienen a ser una especie de conjunto de funciones especializadas en los flujos multimedia y también de algunas de sus funciones experimentales.

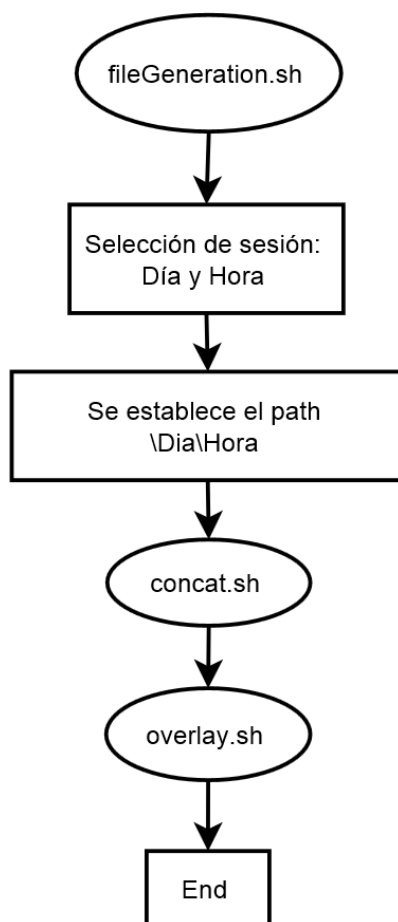


Figura 7: Diagrama del script fileGeneration.sh

3.1 Concatenación de archivos

La concatenación del archivo de video de la pantalla principal con el del audio con el fin de que una de las instancias de reproducción se encargue de este archivo se realiza a partir de una un filtro de FFmpeg que realice una copia del video e incrusta el flujo de audio, manteniendo la sincronización de ambos archivos a partir de un reloj interno.

El producto desarrollado hace uso de este script en dos momentos diferentes a lo largo de todo el proceso. La primera vez que se usa es tras generar los 10 archivos multimedia por parte de los scripts “videoRecord.sh” y “audioRecord.sh”, ya que necesita concatenar estas diez instancias que han creado cada uno de estos scripts tras la grabación de uno de los flujos audiovisuales.

En la figura 8, se puede observar el proceso que es llevado a cabo por este script. El cual vaya leyendo de un archivo de texto plano cada una de las entradas (líneas), que tiene este y las va concatenando una tras otra, sobre el mismo archivo final. En la sección Anexo, se puede encontrar un ejemplo de archivo del que leería este script.

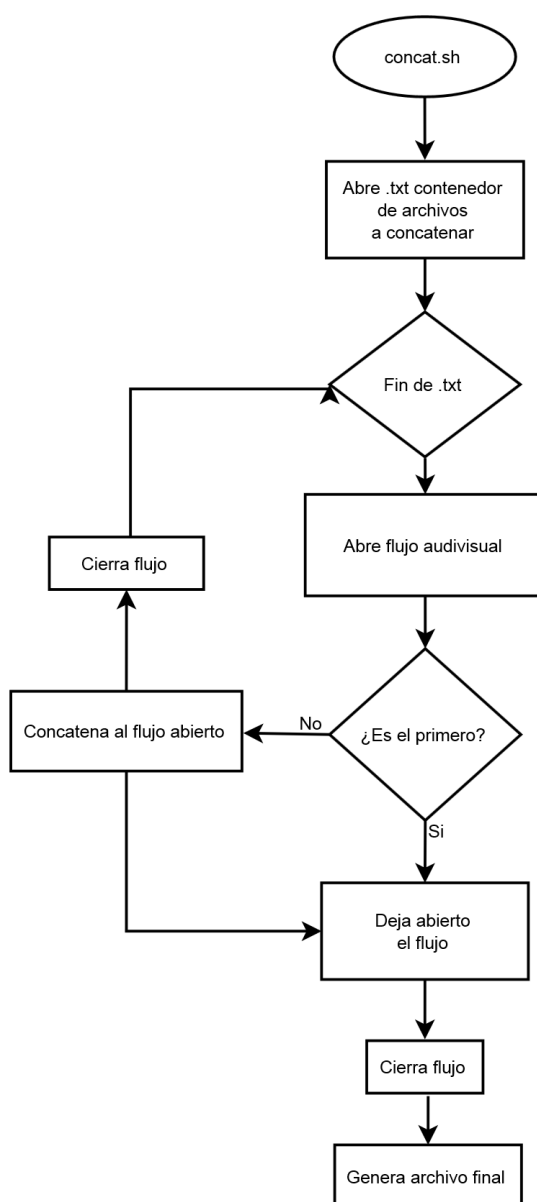


Figura 8: Diagrama del script `concat.sh`

El código encargado de este proceso se muestra a continuación:

```
ffmpeg -i $video1 -i $audio -c:v copy -c:a aac
      -strict experimental $video
```

3.2 Superposición de los archivos

Por otra par el proceso de superposición, que precede al de concatenación. Lo que hace es superponer los flujos de los tres archivos que se le especifican sobre un archivo de salida. Con esto se obtiene un archivo multimedia en el que encontramos los videos de la pantalla principal y de la auxiliar así como el audio de la grabación. El código encargado de este proceso se muestra a continuación:

```
ffmpeg -i $video1 -i $video2 -i $audio -filter_complex "[0:v]setpts
      = PTS - STARTPTS,pad = iw * 2:ih[bg];
      [1:v]setpts = PTS - STARTPTS[fg]; [bg][fg]overlay = w" $output
```

Se muestra en la figura 9 una imagen con el resultado de la operación de superposición de todos los archivos. En ella se pueden observar las dos ventanas asociadas cada una de ellas a un monitor diferente:

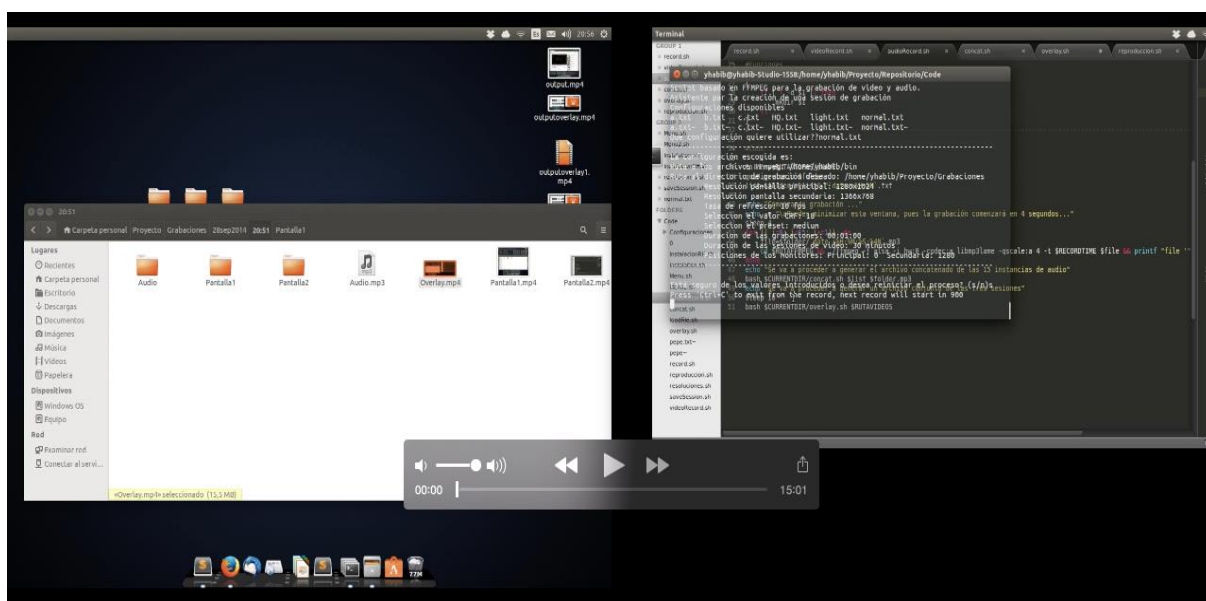


Figura 9: Superposición de la pantalla principal y la auxiliar

4. REPRODUCCIÓN DE LA ESTACIÓN

El reproductor VLC es usado en este proyecto con el fin de poder visualizar las sesiones grabadas en un equipo formado por una doble pantalla. Para lograr esto se requiere la reproducción de dos instancias de video más una de audio simultáneamente y sincronizadas.

VLC es un reproductor y framework multimedia libre y de código abierto desarrollado por el proyecto VideoLAN y hace uso de las herramientas FFmpeg. Es un programa multiplataforma con versiones para Windows, OS X y GNU/Linux, y que es capaz de reproducir casi cualquier formato de video sin necesidad de instalar ningún códec externo adicional, y con sus resoluciones nativas hasta inclusive 4K.

Es software libre distribuido bajo la licencia GPL, en concreto bajo la licencia LGPL v2.1 +.

Este proyecto hace uso de su capacidad para reproducir dos instancias de video, definiendo a una de ellas como la maestra y a la otra como la esclava. La maestra será la que contenga el video del monitor principal y el audio grabado, mientras que la esclava será la que reproduzca el video asociado a la pantalla secundaria. Es por esto por lo que se necesita concatenar el archivo de video de la pantalla principal con el archivo de audio como ya se comentó en la sección 3.1.

En la figura 10, se puede observar el proceso que es llevado a cabo por este script. En él se puede comprobar como el funcionamiento encargado de la reproducción es bastante sencillo, ya que su tarea se reduce básicamente a llamar a VLC para que se cree una instancia de su reproductor en la sesión actual de trabajo. Una vez creada, se configura el reproductor como se muestra en el manual de usuario que se puede encontrar en el documento N° 4.

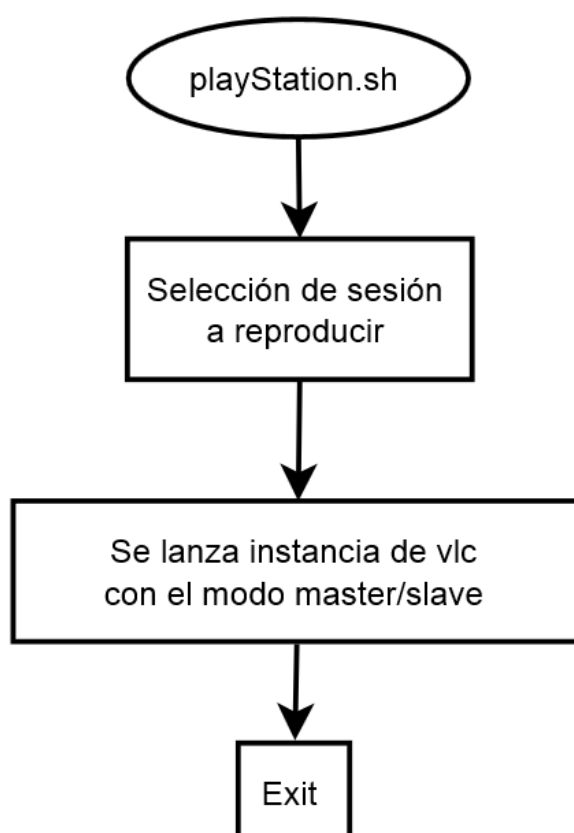


Figura 10: Diagrama del script `launchVLC.sh`

5. ESTRUCTURA DE DIRECTORIOS

Esta sección del presente documento tiene como intención, ofrecer una visión de la estructura de directorios que se va a obtener como resultado a emplear este producto.

En la figura 11, se puede observar el árbol de directorios que se genera. La carpeta importante es la que recibe el nombre de Proyecto, en ella se almacenarán a su vez en nuevas subcarpetas los archivos de grabación, merece especial atención los nombres que reciben cada una de las subcarpetas.

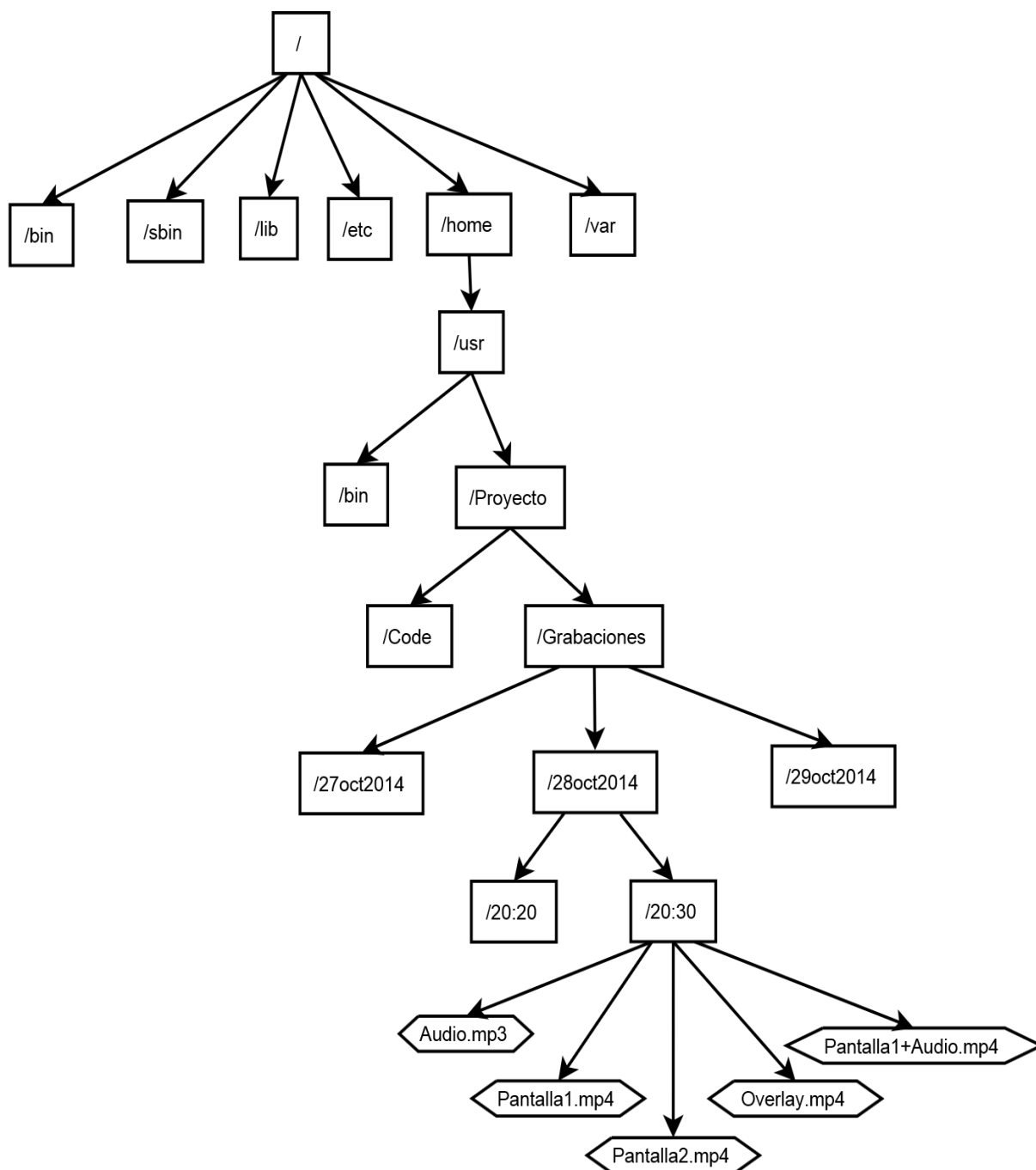


Figura 11: Árbol de directorios tras la instalación

6. SOFTWARE EMPLEADO

A continuación se describirá el software de terceros empleado en la realización de este proyecto. Las herramientas empleadas se pueden dividir en tres grupos:

- Grabación: la herramienta empleada para todo el proceso de captura de video y audio, grabación y procesamiento de este fue realizado a través del conjunto de librerías y programas FFmpeg.
- Reproducción: la herramienta empleada para la reproducción multipantalla sincronizada y con opciones en el equipo original es VLC.
- Software adicional: conjunto de herramientas software independiente del proceso de grabación y reproducción que fueron empleadas en la realización del proyecto.

A continuación se muestra un breve análisis de las herramientas empleadas, dando especial interés a las funcionalidades que permitieron la realización del proyecto.

6.1 FFmpeg

Es un proyecto de software libre que ofrece un conjunto de librerías y programas para el manejo de datos multimedia. Está liberado bajo una licencia de software libre GNU Lesser General Public License 2.1+ o GNU General Public License 2+ (dependiendo de las bibliotecas que estén incluidas).

El proyecto está compuesto por un conjunto de herramientas que pueden funcionar tanto de forma independiente como en conjunto para llegar a ofrecer una solución total. Está formado por ffmpeg, ffmpegserver, ffmpegplay y ffmpegprobe.

6.1.1 ffmpeg

Es una herramienta software de “línea de comandos” para convertir audio o video de un formato a otro, que también es capaz de grabar de una fuente en directo audio/video. También puede convertir entre tasas arbitrarias de muestreo y modificar el tamaño sobre la marcha con filtros de alta calidad..

ffmpeg es capaz de leer de un conjunto de entradas arbitrarias, a los que considera “archivos”, los cuales pueden ser archivos comunes, dispositivos grabadores, ‘pipes’, flujos audiovisuales online, etc., especificados por la opción “-i”, y también es capaz de escribir a un número arbitrario de salidas a las que también define como “archivo”. Todo lo que encuentra en la línea de comandos que no puede interpretar como una opción de funcionamiento lo considera como nombres para los diferentes archivos de salida.

El proceso de transcodificación empleado por ffmpeg para cada archivo de salida puede ser descrito por el diagrama de la figura 12:

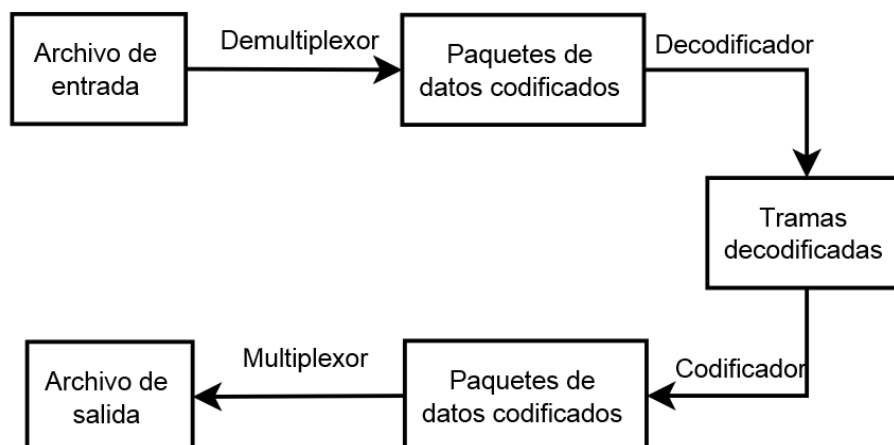


Figura 12: Diagrama del tratamiento de archivos multimedia por parte de FFmpeg

Ffmpeg llama a la librería “libavformat” (contiene los demultiplexadores) para que lean los archivos de entrada y genere los paquetes contenedores de los datos codificados. Cuando se encuentra con múltiples archivos de entrada, ffmpeg intenta mantenerlos sincronizados mediante el seguimiento de la muestra de tiempo más baja de cualquier flujo de entrada activo. Los paquetes codificados son posteriormente pasados al decodificador (a menos que una opción haya sido escogida previamente, para el uso de los paquetes originales). El decodificador produce tramas descomprimidas (video y audio puro) las cuales pueden ser procesadas posteriormente mediante filtros. A continuación, las tramas son pasadas al codificador, el cual genera paquetes con la codificación deseada. Finalmente estos son pasados al multiplexador, el cual tiene por última tarea la de escribir los paquetes codificados al archivo de salida.

El proceso de filtrado es el encargado de tratar con las tramas de video y audio puro mediante un conjunto muy variado de filtros que permiten realizar diferentes tareas. Estos filtros se encuentran en la librería libavfilter. FFmpeg distingue entre dos tipos de filtros gráficos, los simples y los compuestos.

Dentro de este conjunto de comandos se encuentra los dos más importantes para la captura y grabación del video y del audio de entrada:

6.1.1.1 *x11grab*

Esta funcionalidad del comando ffmpeg permite la grabación de lo que sucede en la pantalla o en las X como es definido en Linux. A continuación se muestra un ejemplo del funcionamiento de esta poderosa herramienta y de sus opciones:

```
ffmpeg -t $RECORDTIME -f x11grab -video_size $RES -framerate $FPS -i :0.0 + $POS -
-vcodec libx264 -preset $PRESET -crf $CRF -y $file
```

Cada una de las opciones que se le introducen al comando ffmpeg viene precedida por un guión y sucedida por el valor deseado:

- -t : indica la duración de la grabación en el formato hh:mm:ss.
- -f: indica la función de ffmpeg que se pretende emplear, en este caso “x11grab” aunque si se quisiese grabar el audio habría que emplear en su lugar la instancia “alsa”.

- `-vide_size`: indica la resolución del archivo a generar.
- `-framerate`: indica la tasa de frames por segundo con la que se desea grabar.
- `-i`: indica la posición del monitor que se desea grabar respecto al principal.
- `-vcodec`: indica el códec que se ha de emplear en la codificación de los paquetes de video obtenidos.
- `-preset`: opción asociada al códec H.264 e indica la velocidad de codificación para el ratio de compresión. Esto permite grabar con más calidad a cambio de escoger un preset menor. Relación entre la calidad de la grabación y la sincronización de esta.
- `-crf`: esta opción también está asociada al códec H.264, e indica la calidad de la grabación. Esta opción relaciona la calidad de cada una de las tramas grabadas con el peso de estas.

6.1.1.2 concat

Es otra funcionalidad dentro del comando ffmpeg y es la encargada de concatenar diferentes archivos que contengan paquetes de datos de las mismas características en uno solo.

6.1.2 ffserver

Conjunto de herramientas que tienen como función la de ofrecer un servidor de streaming para video y audio. Soporta múltiples alimentadores en directo, streaming desde archivos, etc.

Se configura a partir de un archivo de configuración, que es leído cuando se ejecuta por primera vez la instancia, aunque si este no es especificado tiene un funcionamiento básico. Cada instancia ffserver escuchará en el puerto especificado en el archivo de configuración.

Esta herramienta permitiría el envío de la sesión grabada en la POS a un punto de almacenamiento o seguimiento, para evitar problemas de espacio en el equipo REPRO.

6.1.3 ffplay

Es un reproductor audiovisual muy simple y portable, que fue construido a partir de las librerías FFmpeg y SDL. Puede ser empleado como herramienta de testeo de las instancias grabadas, pero no permite la reproducción múltiple sincronizada con opciones, por lo que no fue la opción escogida para la reproducción de la sesión.

6.1.4 ffprobe

Herramienta muy útil que permite recopilar información de un flujo multimedia y presentarla de una manera entendible tanto para la máquina como para el humano. Por ejemplo puede ser usada para comprobar el tipo de contenedor usado por un flujo multimedia y el formato de cada uno de los flujos contenidos (video, audio y metadatos).

6.2 Reproducción

6.2.1 VLC

Es un reproductor multimedia y framework multimedia libre y de código abierto desarrollado por el proyecto VideoLAN. Hace uso de las herramientas FFmpeg. Es un programa multiplataforma con versiones para Windows, OS X y GNU/Linux, y que es capaz de reproducir casi cualquier formato de video sin necesidad de instalar ningún códec externo adicional, y con sus resoluciones nativas inclusive 4K.

Es software libre distribuido bajo la licencia GPL, en concreto bajo la licencia LGPL v2.1 +.

Este proyecto hace uso de su capacidad para reproducir dos instancias de video, definiendo a una de ellas como la maestra y a la otra como la esclava. La maestra será la que contenga el video del monitor principal y el audio grabado, mientras que la esclava será la que reproduzca el video asociado a la pantalla secundaria.

6.3 Herramientas adicionales

6.3.1 Git

Herramienta multiplataforma para el control de versiones liberada bajo licencia GNU GPLv2. Se empleó la plataforma de desarrollo colaborativo de software GitHub, para el alojamiento online del proyecto.

Fue empleada en este proyecto para el desarrollo tanto del código del proyecto como de los documentos asociados a este, y su disponibilidad desde diferentes equipos para pruebas, diseño, etc.

6.3.2 Oracle VM VirtualBox

Es un software de virtualización para arquitecturas x86/amd64. Permite instalar sistemas operativos adicionales, estos son considerados como “sistemas invitados”, dentro del sistema operativo que corre el programa el cuál es llamado “anfitrión”. Soporta la virtualización de una gran cantidad de sistemas operativos pertenecientes a las tres familias más importantes GNU/Linux, Windows y OS X. Está liberado bajo licencia privativa pero también tiene una versión la OSE (Open Source Edition) bajo licencia GPL 2.

Fue la herramienta empleada para la virtualización del sistema operativo CentOS, a la hora de realizar las pruebas de funcionamiento.

6.3.3 CentOS (Community ENTERprise Operating System)

Es una bifurcación a nivel binario de la distribución Linux Red Hat Enterprise Linux RHEL, compilada a partir del código fuente libereado por Red Hat. Este programa de fuente abierta, es el encargado de ofrecer una solución gratuita de “clase empresarial”. Tiene el mismo aspecto y funcionamiento de que RHEL.

Emplea el mismo gestor de paquetes YUM y los repositorios disponibles son prácticamente los mismos, dando lugar a un sistema operativo prácticamente igual a RHEL.

Fue el sistema operativo empleado en este proyecto para la realización de las pruebas, en el intento de buscar un sistema que se comportase de la forma más parecida al de las POS Repro, Es por por ello, que la guía de usuario está diseñada para este S.O.

6.3.4 Merlin

Software para el manejo de proyectos sobre sistemas operativos OS X. Se centra en la creación de planes, el seguimiento del estado de las actividades, la asignación de recursos, y el manejo de presupuestos. Contiene un sistema de reporte que incluye visualización temporal de la actividad, mediante un diagrama de Gantt, y la carga de trabajo de los recursos.

Se empleó la versión Student, la cuál ofrece una licencia para estudiantes universitarios sin coste para este.

6.3.5 Sublime Text 2

Editor de texto y editor de código fuente que está escrito principalmente en C++ y sus plugins en Python. Desarrollado orifinalmente como una extensión de Vim, con el tiempo fue creó una identidad propia, por esto aún conserva un modo de edición tipo vi llamado *Vintage mode*.

Se distribuye de forma gratuita, sin embargo no es software libre o de código abierto, se puede obtener una licencia para su uso ilimitado, pero el no disponer de esta no genera ninguna limitación más allá de una alerta cada cierto tiempo.

7. ANEXO

7.1 Acrónimos y definiciones

A continuación se muestran los acrónimos presentes en este o en alguno de los otros documentos que conforman este proyecto:

- **Bash** (Bourne Again Shell): Software intérprete de órdenes o comandos. Desarrollado por el Proyecto GNU, e intérprete de comandos por defecto para la mayoría de las distribuciones Linux.
- **CWP** (Control Working Position): Posición de Trabajo del Controlador.
- **Fork** (bifuración): En el ámbito del desarrollo de software, es la creación de un proyecto en una dirección distinta de la principal u oficial tomando el código fuente del proyecto ya existente.
- **GNU** (GNU is Not Unix): Sistema operativo del tipo Unix desarrollado por el Proyecto GNU, y formado en su totalidad por software libre.
- **GPL** (General Public License): Licencia usada en el mundo del software que garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.
- **RPM** (RedHat Package Manager): Herramienta de administración de paquetes para Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas.
- **Script**: También conocido como archivo de procesamiento por lotes. Es un programa normalmente simple que se almacena en un archivo de texto plano.
- **VIM** (Vi IMproved): Versión mejorada del editor de texto VI, presente en todos los sistemas Unix.
- **YUM** (Yellow dog Updater, Modified): Gestor de paquetes para sistemas Linux basados en RPM.