ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA DE COMPUTADORES

Gestor centralizado de presentaciones para visualización remota a través de redes TCP/IP

Centralized presentations for remote viewing via TCP / IP networks

Realizado por
Francisco Manuel Rico Jurado
Tutorizado por
David Santo Orcero
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, Junio de 2017

Fecha defensa:
El Secretario del Tribunal

Resumen:

El objetivo de este proyecto ha sido obtener un solución que abstraiga el hardware y software subyacente y que, mediante un sistema cliente-servidor permita, a través de una red TCP/IP, mostrar contenido multimedia en formato .odp y .mp4 en cualquier sistema de proyección con entrada HDMI.

Para ello, hemos desarrollado una aplicación de servidor basado en PHP (Gestor de contenidos), el cual se encarga de comunicar y programar los dispositivos físicos para que reproduzcan el contenido multimedia.

Para el dispositivo cliente hemos optado por usar la Raspberry PI con el sistema operativo Ubuntu Mate, hemos establecido un protocolo de comunicación a través de ssh y una configuración específica del sistema operativo.

Una vez configurado el cliente, este puede ser agregado al gestor de contenidos.

Al gestor de contenidos se permite agregar tantos dispositivos clientes como queramos. Una vez añadidos podremos configurar la presentación multimedia que debe reproducir y definir una programación temporal para la reproducción de videos, de forma independiente para cada cliente.

Palabras claves:

Raspberry PI, Ubuntu, Mootools, Geany, PHP, Gestor, Multimedia, Odp, Mp4, Reproductor, Cliente-Servidor, Modelo-Vista-Controlador, MVC, Codeigniter, Cartelería, Difusión, Publicidad.

Abstract:

The aim of this proyect is to obtain a solution that abstracts the hardware and the underlying software and that, by means of a client-server system allows, trough a TCP/IP net, to show multimedia content in .odp and .mp4 format in any projection system with an HDMI interface.

Therefore, we have developed a server application based on PHP (content manager), that is in charge of communicate and programme the physical devices so that they reproduce the multimedia content.

For the client device we have opted to use the Raspberry PI with the Ubuntu Mate operative system, we have established a communication protocol trough ssh and an specific configuration of the operative system.

The content manager is allowed to add as many devices as we want. Once they are added, we would be able to configure the multimedia presentation that have to reproduce and define a temporary programming for the reproduction of videos, in an independently way for each client.

Keywords:

Raspberry PI, Ubuntu, Mootools, Geany, PHP, manager, Multimedia, Odp, Mp4, Player, Client-Server, Model-View-Controller, MVC, Codeigniter, adverstising, Dissemination.

Índice general

Capítulo 1

Introducción

1.1. Cartelería digital

Tradicionalmente hemos entendido la cartelería digital como simples pantallas que reproducen una lista de vídeos e imágenes. Sin embargo, el gran crecimiento de Internet y redes de área local (LANs) ha creado la necesidad de convertir estos dispositivos publicitarios o de difusión en sistemas muchos más dinámicos. Además el incremento de la instalación de dispositivos de este tipo crea la necesidad de que sean fáciles de gestionar.

1.2. Motivaciones

En muchas ocasiones, nos hemos encontrado con el problema de publicar contenidos multimedia para diversas situaciones, como por ejemplo:

- Un ponente desea exponer una presentación cíclica en un proyector y este no dispone de portátil o equipo multimedia para conectar al proyector, o simplemente el formato de presentación no es compatible con el programa del reproducción de presentaciones del ordenador que la organización puede facilitar.

- Exponer una cartelería digital en distintas pantallas, cada una con un sistema de reproducción distinto y local, creando la necesidad de tener que adaptar el contenido a los distintos dispositivos y lo mas importante, tener que desplazarse físicamente a la ubicación concreta de cada dispositivo para establecer la configuración y poner en funcionamiento el sistema.

- En ocasiones, dos interesados tienen la necesidad de publicar contenido de distinta naturaleza al mismo tiempo o mas concretamente, durante el mismo periodo de tiempo, como por ejemplo, un vídeo y una presentación durante la navidad.
  Podemos poner el supuesto práctico en el que el área de publicidad desea difundir sus ofertas especiales y el área administrativa desea divulgar un vídeo felicitando las fiestas.

- Otro escenario común es publicar síncronamente, en todos los dispositivos de proyección publicitarios disponibles, un contenido en concreto durante un corto periodo de tiempo, como podría ser una repudia ante un atentado terrorista, una alerta de cualquier índole o cualquier situación en la que todos los dispositivos deban publicar el mismo contenido.

Todos estos escenarios y muchos más que pueden ocurrir, son muy complejos de administrar mediante sistemas independientes y aislados tanto en la configuración como la gestión.
Al plantearnos la situación de tener que afrontar la gestión y administración de este tipo de dispositivos, iniciamos la necesidad de estudiar, diseñar e implementar una solución que solvente todas las situaciones anteriores de una forma fácil y sencilla, que establezca las bases para dar solución a posibles inconvenientes futuros y sea administrable por un usuario final con una formación básica.

1.3. Objetivos

En la actualidad, existen soluciones especificas de arquitectura y software cerrado y privativo, que dan solución a todos estos problemas. Sin embargo, estas soluciones solo están al alcance de quienes pueden pagarlo, excluyendo a pequeñas empresas y comercios, organizaciones y asociaciones. Además, en muchas ocasiones no permiten reutilizar el material existente, exigiendo la adquisición del material compatible con el producto.
El objetivo global o principal de este proyecto ha sido obtener un solución que abstraiga el hardware y software subyacente y que, mediante un sistema cliente-servidor permita, a través de una red TCP/IP, mostrar contenido multimedia en formato .odp (OpenDocument Presentation File) y .mp4 (Media Player 4) en cualquier sistema de proyección con entrada HDMI.

Los objetivos particulares o singulares para poder realizar este proyecto con éxito han sido los siguientes:

- El servidor gestiona todo el contenido que posteriormente podremos reproducir en los dispositivos clientes. Una vez cargado el contenido multimedia, podrá ser programado de forma independiente en los distintos dispositivos clientes para su reproducción.

- En el caso de contenido .odp, la reproducción es de forma indefinida (bucle infinito) y en el caso de contenido .mp4, puede ser programada para su reproducción cada 'x' minutos.

- Una vez el cliente recibe las instrucciones por parte de servidor, este funcionará independiente y de forma autónoma al servidor, más concretamente se deben dar los siguientes escenarios:

  - Si el cliente sufre un corte de suministro eléctrico, al reanudarse el suministro, el cliente iniciara y comenzará a reproducir la última programación que recibió por parte del servidor.
  - En en el caso de producirse la desconexión de la red TCP/IP, este continuará funcionando tal y como el servidor lo programo.
  - Si se dan ambas situaciones al mismo tiempo, el cliente iniciará y comenzará a reproducir la última programación que recibio por parte del cliente y una vez que se restablezca la comunicación TCP/IP, este podra ser reconfigurado por el administrador o administradores del servidor.

- Lo mas común es que los administradores del servidor se encuentren físicamente alejados de los dispositivos clientes, por esto es necesario poder visualizar remotamente lo que cada cliente proyecta. Para comprobar que la reproducción es acorde a lo programado y que los dispositivos clientes funcionan correctamente, desde el servidor podremos hacer capturas de pantallas de los distintos clientes y visualizarlas.

1.4. Estructura de la memoria

Esta memoria esta estructurada en ocho capítulos, un apéndice y la bibliografía.

- En el primer capítulo comentamos el marco de trabajo, que ha creado la necesidad de realizar este proyecto y los objetivos que se van a cumplir.
- En el segundo capítulo establecemos los requisitos del proyecto.
- En el tercer capítulo realizamos un breve estudio de las tecnologías que hemos usado, exponiendo motivos por lo que hemos elegido usar cada una de ellas.
- En el cuarto capítulo definimos la estructura cliente-servidor realizada, con el fin de que el lector obtenga una visión general de la solución.
- En el quinto capítulo definimos la puerta en marcha la parte servidor.
- En el sexto capítulo realizamos un análisis de la solución y sus posibles casos de uso.
- En el séptimo capítulo explicamos cómo hemos desarrollado la parte cliente.
- En el octavo capítulo expresamos las conclusiones obtenidas.
- En el primer apéndice, explicamos cómo debe ser configurada una presentación para el correcto funcionamiento con el sistema.

Capítulo 2

Estado del Arte

2.1. Introducción

En el mercado podemos encontrar distintas soluciones para cubrir las mismas necesidades de este proyecto. Todas ellas las podemos agrupar en tres tipos: suscripciones, reproductores independientes y soluciones específicas.

A continuación exponemos un ejemplo de cada una de ellas analizando sus ventajas e inconvenientes.

2.2. Suscripciones

Un ejemplo actual de suscripción es "*Smush Digital*", esta empresa ofrece una solución a través de un pago mensual o anual.

Ventajas:

- *Fácil implantación de la solución:* El aplicativo es un servicio web en internet, por lo que no es necesario la implantación en local de ningún tipo de servidor.

Inconvenientes:

- *Conexión permanente a internet:* La solución requiere que los dispositivos reproductores estén conectados de forma permanente a internet, con las desventajas que esto supone.
- *Reproductores específicos:* El software necesita de hardware especifico que no se puede reutilizar en caso de cambiar de solución.

- *Menos seguridad:* El sistema esta expuesto a un ataque en internet hacia el servidor, existiendo la posibilidad de la emisión de contenidos ilícitos por parte de un tercero no autorizado.

La figural 2.1 muestra el esquema de funcionamiento del sistema.

Figura 2.1: Solución Smush Digital

2.3. Reproductores Independientes

Los reproductores independientes no presentan ninguna ventaja, son difíciles de gestionar y cada fabricante dispone de su sistema independiente.

Un ejemplo de reproductor puede ser "*Reproductor Multimedia Sveon Spm820m SPM820M*", ver la figura 2.2.

2.4. Soluciones específicas integrales

Podemos entender el sistema de "*CAYINtech*" (http://www.imaginart.es/cartelera-digital/cayintech/introduccion.htm), como un sistema de solución específica integral, la empresa ofrece una solución completa a la cartelería digital.

Figura 2.2: Reproductor Multimedia Sveon Spm820m SPM820M

Ventajas:

- *Solución robusta:* Al ser una solución desarrollada íntegramente por una empresa, todos sus dispositivos deben ser compatibles entre si, y prácticamente no presentar ningún problema de compatibilidad.

Inconvenientes:

- *Reproductores específicos:* El software necesita de hardware especifico que no se puede reutilizar en caso de cambiar de solución.
- *Coste elevado:* Es necesario adquirir todas las partes (servidor, software y reproductores) al mismo fabricante.

La figural 2.3 muestra el esquema del sistema.

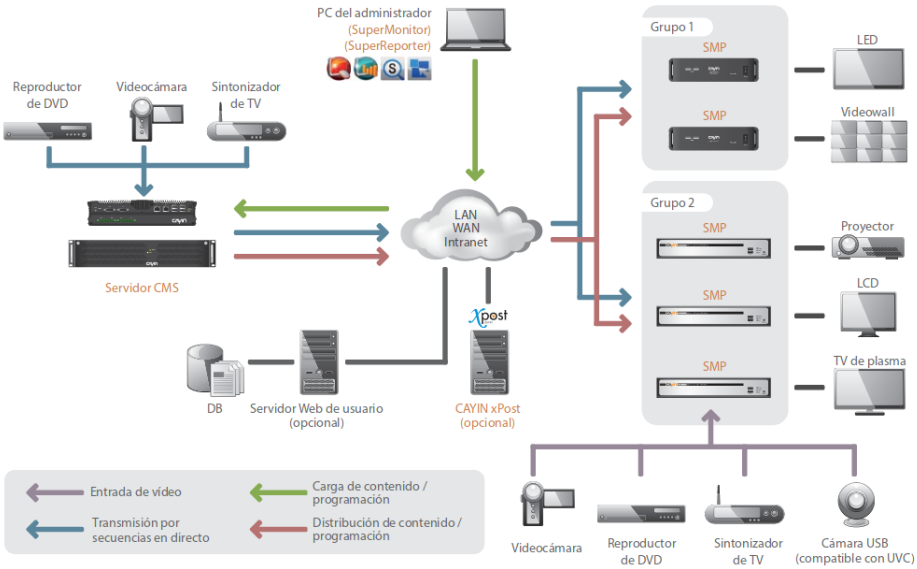


Figura 2.3: Solución CAYINtech

Capítulo 3

Tecnologías Empleadas

3.1. Planteamiento

Raspubli es la denominación que le hemos dado al proyecto. Para empezar a dar forma al proyecto, lo primero que hemos realizado ha sido abstraernos del hardware de proyección de imágenes. En la actualidad, cada dispositivo de proyección dispone de su propio sistema de visualización de contenido multimedia, incluso son capaces de conectarse a servidores multimedia y reproducir el contenido que encuentre en ellos.
Como no podemos saber cómo los fabricantes van a evolucionar y que posibilidades van a disponer los nuevos dispositivos, hemos optado por abstraernos del hardware tomando como cliente un "Ordenador de placa simple (SBC)" con salida multimedia HDMI. De esta forma, queda excluida de nuestro proyecto cómo se van a proyectar el contenido multimedia que nuestros clientes enviarán su contenido a través del HDMI; sin importar si este es posteriormente proyectado en una televisión, proyector, sala de cine o pantalla publicitaria.
Unos de los principales objetivos de nuestro proyecto es obtener un sistema lo más fácil posible de gestionar. El primer paso es plantearnos cómo vamos a gestionar el contenido multimedia, pudiendo abordar esto de diferentes formas:

1º **Mediante un equipo dedicado**, con una aplicación de escritorio instalada con la que podemos configurar los clientes cuando sea necesario. Al tener que otorgar a nuestra solución la tolerancia a fallos en la comunicación entre cliente y servidor, la solución de aplicación de escritorio es perfectamente viable ya que los clientes no necesitan que el servidor este en funcionamiento para operar con normalidad.

2º **Mediante una aplicación web**. Al disponer de tantas herramientas de código abierto, podemos facilmente montar un servidor web en el que hacer funcionar nuestra solución. Mediante esta opción obtenemos grandes ventajas, tales como:

- **Compatibilidad multiplataforma**. Las aplicaciones web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.
- **Actualización**. Las aplicaciones basadas en web se pueden actualizar sin requerir la intervención del usuario, la cual es a veces imposible cuando está trabajando dentro de grandes organizaciones.
- **Inmediatez de acceso**. Las aplicaciones basadas en web no necesitan ser descargadas, instaladas y configuradas. Podemos acceder y trabajar sin importar cuál es su configuración o su hardware.
- **Menos requerimientos de memoria**. Las aplicaciones basadas en web consumen menos memoria RAM de parte del usuario final que los programas instalados en local.
- **Menos Bugs**. Las aplicaciones basadas en web son menos propensas a colgarse y crear problemas técnicos debido a que se ejecutan en el servidor. Con aplicaciones basadas en web los bugs pueden ser corregidos tan pronto como son descubiertos.
- **Precio**. Las aplicaciones basadas en web no requieren la infraestructura de distribución y soporte técnico. Esto permite que las aplicaciones online sean menos costosas que las de escritorio.
- **Los datos son más seguro**. Con el desplazamiento de las aplicaciones locales a sistemas basados en web también los datos que creamos se almacenan en el servidor, con la ventaja de poder acceder a ellos desde distintos dispositivos se mejora considerablemente el tratamiento de la seguridad de los datos.
- **Múltiples usuarios concurrentes**. Las aplicaciones basadas en web pueden realmente ser utilizadas por múltiples usuarios al mismo tiempo.

- **Desarrollar aplicaciones en el lenguaje que deseemos**. Una vez que las aplicaciones han sido separadas de los ordenadores locales y sistemas operativos específicos, podemos desarrollar el software de la manera que deseemos.Ya que las aplicaciones web, muestran un resultado al cliente, y le es totalmente transparente cómo se ha producido dicho resultado.

3.2. Entorno de programación GEANY

Geany es un editor de texto pequeño y ligero basado en *Scintilla* con características básicas de entorno de desarrollo integrado (IDE). Está disponible para distintos sistemas operativos, como GNU/Linux, Mac OS X, BSD, Solaris y Microsoft Windows. Utiliza bibliotecas GTK para su funcionamiento y es distribuido como software libre bajo la Licencia Pública General de GNU.

Las caracteristicas principales son:

- Resaltado de sintaxis.
- Plegado de código.
- Autocompletado.
- Cierre automático de etiquetas XML y HTML.
- Muchos tipos de archivos soportados tales como C, Java, PHP, Python, Perl, Pascal y más.
- Listas de símbolos.
- Código de navegación.
- Construir un sistema (conjunto de ejecuciones) para compilar y ejecutar el código
- Fácil gestión de proyectos.
- Soporte para *plugins*.

3.3. Servidor virtual basado en KVM

Las siglas KVM provienen de *Kernel-based Virtual Machine*, que en español significan 'Máquina Virtual basada en el Núcleo', es una solución para conseguir virtualización completa bajo *kernel Linux*. Se compone de un módulo del *kernel* (kvm.ko) y herramientas de usuario para gestión y administración.

Esta formada en su totalidad por software libre y esta incluida en el kernel desde la versión 2.6.20. El único requisito para su uso es que el procesador tenga soporte de virtualización hardware. En procesadores Intel y AMD, se denominan VT y SVM respectivamente.
Con el hardware apropiado, KVM permite ejecutar máquinas virtuales completas cuyos sistemas operativos corren como si lo hiciesen en una máquina física.

3.4. Servidor HTTP *Apache*

El servidor HTTP *Apache* es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que alguien quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu *Apache* fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y a la población. Exactamente el mismo paisaje que habían creado los primeros ingenieros de Internet. Además *Apache* consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, *a patchy server* (un servidor parcheado) suena igual que Apache *Server*.

El servidor *Apache* es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la *Apache* Software Foundation dentro del proyecto HTTP Server (httpd).

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, *Apache*, es el servidor HTTP más usado. Jugó un papel fundamental en el desarrollo fundamental de la World Wide Web y alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70 por ciento de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

La arquitectura del servidor *Apache* es muy modular. El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web. Algunos de estos módulos son:

- mod_ssl - Comunicaciones Seguras vía TLS.
- mod_rewrite - Reescritura de direcciones
- mod_dav - Soporte del protocolo WebDAV (RFC 2518).
- mod_deflate - Compresión transparente con el algoritmo *deflate* del contenido enviado al cliente.
- mod_auth_ldap - Permite autentificar usuarios contra un servidor LDAP.
- mod_proxy_ajp - Conector para enlazar con el servidor *Jakarta Tomcat* de páginas dinámicas en Java.
- mod_cfml - Conector CFML usado por Railo.

3.5. Procesador de Hipertexto PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor, originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y

plataformas sin ningún costo.

PHP se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy[cita requerida], lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico, como Facebook, para optar por el mismo como tecnología de servidor.

PHP es un acrónimo recursivo que significa *PHP Pre Hypertext-processor*. Fue creado originalmente por Rasmus Lerdorf; sin embargo, la implementación principal de PHP es producida ahora por *The PHP Group* y sirve como el estándar de facto para PHP, al no haber una especificación formal. Publicado con la *PHP License*, la *Free Software Foundation* considera esta licencia como software libre.

Las características principales son:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos.
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.

- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

3.6. Base de datos MariaDB

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL. Está desarrollado por Michael Widenius (fundador de MySQL) y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria (que reemplaza con ventajas a MyISAM) y otro llamado XtraDB (en sustitución de InnoDB). Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, APIs y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente. Este sistema gestor de bases de datos surge a raíz de la compra de Sun Microsystems (compañía que había comprado previamente MySQL AB) por parte de Oracle.

MariaDB es un fork directo de MySQL que asegura que permanecerá una versión de este producto con licencia GPL. Michael Widenius decidió crear esta variante porque estaba convencido de que el único interés de Oracle en MySQL era reducir la competencia que MySQL daba al mayor vendedor de bases de datos relacionales del mundo que es Oracle.

3.7. Framework de PHP - Codeigniter 1.7.1

CodeIgniter es un conjunto de herramientas para crear aplicaciones web usando PHP. Su objetivo es permitir el desarrollo de proyectos mucho más rápido de lo que sería si tuviéramos que escribir código desde cero. Son un conjunto de librerías que permiten crear las tareas comunes que son necesario para el desarrollo de cualquier proyecto web, así como una interfaz simple y estructura lógica para acceder a estas librerías. *CodeIgniter* permite centrarse en el proyecto, reduciendo al mínimo la cantidad de código necesario para una tarea determinada.
Las características principales son:

- Sistema Basado en Modelo-Vista-Controlador.
- Peso extremadamente ligero.
- Clases completas de bases de datos con soporte para varias plataformas
- Soporte para consultas en base de datos.
- Validación de datos de formulario
- Seguridad y Filtro XSS
- Administración de sesiones
- Envío de correo electrónico. Es compatible con archivos adjuntos de correo electrónico, HTML / Texto, múltiples protocolos y más.
- Manipulación de imágenes Biblioteca (recorte, cambio de tamaño, etc.). Soporta *GD*, *ImageMagick*, y *NetPBM*
- El almacenamiento en caché de página completa
- XML-RPC Biblioteca
- Los motores de búsqueda URLs
- Enrutamiento URI flexibles
- Gran biblioteca de funciones

3.8. Framework de Javascript - Mootools

MooTools (*My object oriented tools*) es un *Framework* web orientado a objetos para *JavaScript*, de código abierto, compacto y modular. El objetivo de *MooTools* es aportar una manera de desarrollar *JavaScript* sin importar en qué navegador se ejecute de una manera elegante. *MooTools* aporta una API documentada más enfocada a la orientación de objetos que la implementación estándar soportada por los navegadores web.

MooTools aporta al usuario muchas ventajas. Alguna de ellas:

- Es un *Framework* modular y extensible, el desarrollador puede elegir (específicamente) que componentes usar y cuales no.
- *MooTools* es orientado a objetos y sigue los principios DRY, que hacen de él un *Framework* rico, potente y eficiente.
- Componente avanzado de efectos, con transiciones, de función parabólica, optimizadas y utilizadas por multitud de desarrolladores *Flash*.
- *Framework* desarrollado por programadores para programadores.

3.9. Bash-Scripting

Bash (*Bourne again shell*) es un intérprete de comandos y un lenguaje de programación de consola (*Scripting*) basado en la *shell* de *Unix*.

Fue escrito por Brian Fox para el proyecto *GNU* y es el intérprete predeterminado en la mayoría de las distribuciones de *GNU* con *Linux*. Es la evolución del *Bourne Shell* que fue uno de los primeros intérpretes importantes de *Unix*.

3.10. Placa Raspberry Pi 3 Modelo B

Rasberry Pi es un ordenador de placa simple (SBC) de bajo coste, desarrollado en Reino Unido por la "*Fundación Raspberry Pi*", con el objetivo de estimular la enseñanza de *ciencias de la computación*.

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, *Raspbian* (derivada de Debian), RISC OS 5, Arch Linux ARM (derivado

de Arch Linux) y Pidora (derivado de Fedora).
En 2006, los primeros diseños de *Raspberry Pi* se basaban en el microcontrolador Atmel ATmega644. En mayo de 2009, la "*Fundación Raspberry Pi*" fue fundada en el Reino Unido como una asociación caritativa que es regulada por la *Comisión de Caridad de Inglaterra y Gales.*

La fundación lanzó dos modelos, el Modelo A y el Modelo B. Las ventas iniciales fueron del Modelo B y posteriormente se pondría en venta el Modelo A.

La última versión del Modelo B es la "Raspberry Pi 3", las especificaciones técnicas son:

- SoC (System on Chip): Broadcom BCM2837 (CPU + GPU + DSP + SDRAM + Puerto USB.
- CPU ARM 1176JZF-S a 1.2GHz 64-bit quad-core ARMv8.
- GPU Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 y VC-1, 1080p30 H.264/MPEG-4 AVC3.
- Memoria SDRAM 1 GB (compartidos con la GPU).
- 4 puertos USB.
- Salida de vídeo mediante conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4) e interfaz DSI para panel LCD.
- Micro SD para almacenamiento.
- Conectividad mediante 10/100 Ethernet (RJ-45), Wifi 802.11n y Bluetooth 4.1.
- 17 x GPIO (*General Purpose Imput Output*) y un bus HAT ID para periféricos de bajo nivel.
- Alimentación de 5V vía Micro USB con un consumo de 800 mA, (4.0 W).

3.11. Ubuntu Mate

Ubuntu MATE es una distribución Linux basada en Ubuntu. Está mantenida por la comunidad y es un derivado de Ubuntu oficialmente reconocido por Canonical, usando el entorno de escritorio MATE.

El proyecto *Ubuntu MATE* fue fundado por Martin Wimpress y Alan Pope, y comenzó como un derivado no oficial de Ubuntu, usando como base a Ubuntu 14.10 para su primer lanzamiento.

La versión 16.04 LTS tiene imágenes creadas específicamente para dispositivos Raspberry Pi 2 y Raspberry Pi 3.

3.12. Decisiones tomadas

Al tener en cuenta todas las ventajas de una aplicación web, optamos por desarrollar nuestro gestor de contenidos multimedia en forma de aplicación web.

Hasta el momento, el modelo de nuestro proyecto está compuesto por una aplicación web (servidor) y un ordenador de placa simple (cliente), quedando pendiente solucionar cómo vamos a realizar la comunicación entre ambos.

Como hemos decidido que nuestra solución servidor va a estar basada en software libre, es evidente que tenemos que usar un sistema operativo en nuestros clientes que sean software libres y cien por cien compatibles. Además deben cumplir el objetivo final de este proyecto, que es proyectar contenido en formato .odp y .mp4.

Tras realizar un breve estudio sobre las distintas "SBC" disponibles en el mercado, elegimos usar la "*Raspberry Pi*", principalmente por los siguientes motivos:

1. Es bastante asequible. Es una placa de bajo coste y fue un producto desarrollado con esa intención.

2. Buena distribución comercial. Al ser pionero en este tipo de productos, su uso esta muy extendido y existe muchisimos puntos de venta donde poder adquirir una.

3. Existe mucha documentación y la comunicad es muy extensa.

De igual forma elegimos que el sistema operativo a usar esa "Ubuntu Mate" por los motivos siguientes:

1. Es una distribución con una larga trayectoria y de cierta reputación.

2. Esta distribución a diferencia de muchas otras disponibles para instalar en un *Raspberry PI*, usa versiones de software muy recientes y actualizadas. Esto es muy importante tenerlo en cuenta por el simple motivo de que los contenidos multimedia que se van a publicar van a estar creados en software reciente y de escritorio.

3. Dispone de todo el software y *codecs* necesarios para la reproducción de vídeos en formato .mp4.

Teniendo definido el servidor y el cliente, estamos en el punto de definir el protocolo de comunicación entre ambos. Para poder cumplir con el objetivo de que nuestro sistema sea lo mas abierto posible en la implantación de la solución, hemos obtado por realizar la comunicación mediante comandos en `bash-script` a través de una consola *ssh*.
Las principales ventajas de hacerlo de esta forma son las siguientes:

1. No es necesario ninguna implementación de un programa especifico de comunicación.

2. El tema de seguridad en la comunicación queda resuelto

3. Obtenemos una nueva capa de abstracción entre cliente y servidor.

Resumiendo podemos definir nuestra solución de la siguiente forma: Sistema basado en un servidor web que gestiona contenidos multimedia y mediante comando *bash* da ordenes a los clientes para que visualicen y contenido multimedia con una programación determinada.

Capítulo 4

Diseño e implementación del protocolo cliente-servidor

4.1. Introducción

Podemos definir la estructura cliente-servidor como una relación entre procesos distintos. Si estos procesos se encargan de proveer servicios, entonces hablamos de servidor, y análogamente si los procesos hacen usos de los servicios, hablamos de cliente.

La estructura completa es formada mediante un mecanismo de paso de mensajes que ambos comprenden, y definen un protocolo de comunicación.

4.2. Definición del protocolo del sistema

Definimos protocolo de sistemas como las comunicaciones básicas que se producen entre las tres partes que intervienen en todo el proceso de funcionamiento. Estas partes son:

- El administrador del servidor: persona encargada de gestionar los contenidos y verificar su correcto funcionamiento.
- El servidor: computadora que sirve para la gestión de contenido multimedia y se comunica con los clientes finales
- Clientes: sistema operativo instalado en la *Raspberry Pi* que se encarga de publicar gráficamente el contenido multimedia.

La figura 4.1 representa la comunicación entre las tres partes:

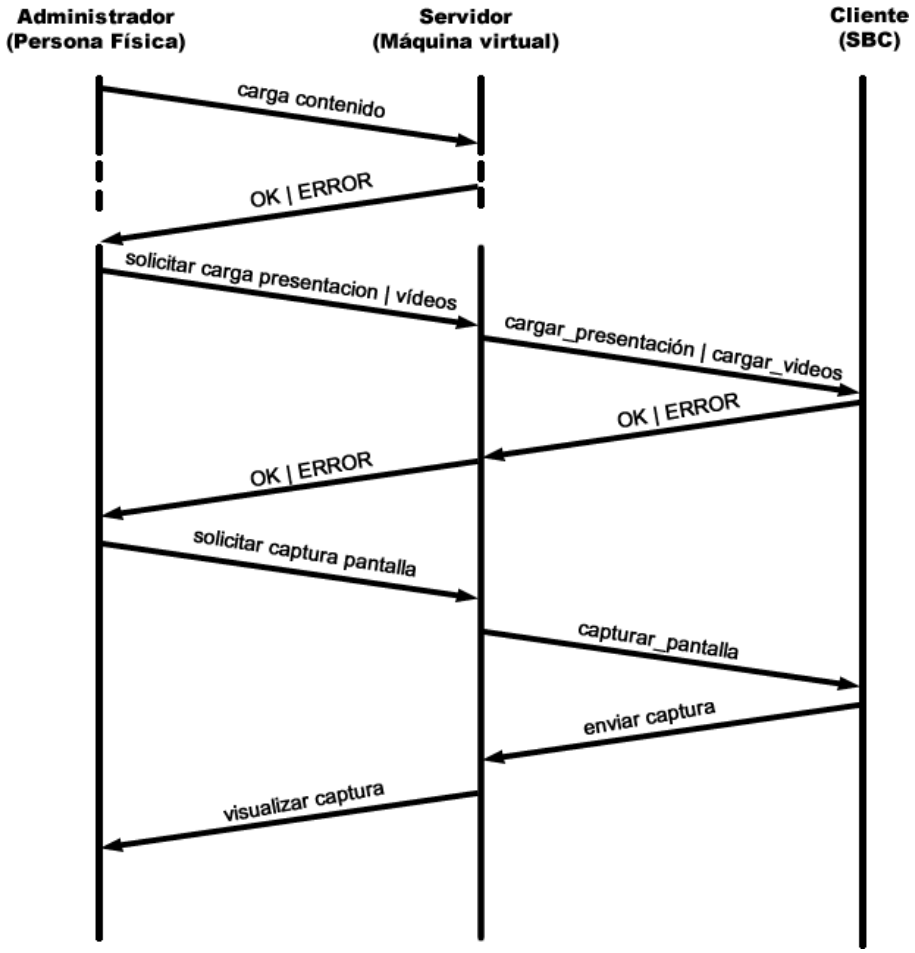


Figura 4.1: Comunicación global.

4.3. Protocolo de comunicación entre gestor y reproductor a través de TCP/IP

La comunicación entre la máquina virtual (gestor de contenidos) y la *Raspberry Pi* (reproductor) la realizamos mediente comandos '*ssh*'. En la apartado anterior hemos simplificado el protocolo de comunicación entre el gestor y el reproductor en tres acciones. Estas acciones definen un protocolo propio cada una de ellas:

4.3.1. Cargar presentación:

Compuesta por seis comandos '*bash*'

1. Apargar la salida gráfica

```
$ ssh <user@ip_server> "export DISPLAY=:0; xset dpms force off"
```

2. Finalizar la presentación actual

```
$ ssh <user@ip_server>  "sudo killall soffice.bin"
```

3. Limpiar archivos temporales

```
$ ssh <user@ip_server>  "rm /home/raspubli/.~lock.pi_presentation.odp"
```

4. Cargar nueva presentación en el reproductor

```
$ scp <path_archivo> <user@ip_server> :/home/raspubli/pi_presentation.odp
```

5. Iniciar nueva presentación

```
$ ssh <user@ip_server> "export DISPLAY=:0; soffice --show /home/raspubli/
    pi_presentation.odp ?norestore"
```

6. Encender salida de vídeo

```
$ ssh <user@ip_server> "export DISPLAY=:0; xset dpms force on; xset -dpms"
```

4.3.2. Cargar vídeo:

Detiene la reproducción actual, elimina la configuración actual y para cada uno de los vídeos configurados, carga el archivo en el disco duro del reproductor, crea un archivo '*bash-script*' para iniciar el vídeo y añade la programación a un archivo de configuración del programa '*cron*'. Por ultimo carga la configuración del *cron* y reinicia el servicio del *cron*.

Los comandos usados son los siguientes:

1. Finalizar reproducción actual

```
$ ssh <user@ip_server> "sudo killall omxplayer.bin"
```

2. Eliminar los script de inicio de vídeos actuales

```
$ ssh <user@ip_server> "rm /home/raspubli/init_video_*.sh"
```

3. Eliminar los archivos de vídeo

```
$ ssh <user@ip_server> "rm /home/raspubli/raspubli*.mp4"
```

4. Reiniciar el archivo de configuración de *cron*

```
$ ssh <user@ip_server> "> /home/raspubli/cron_video.txt"
```

5. Cargar nuevo archivo de vídeo en el reproductor

```
$ scp <path_archivo> <user@ip_server>:/home/raspubli/<nombre_archivo>.mp4
```

6. Crear *script* de inicio del vídeo ejecutando comandos *echo* a través de *ssh*

```
$ #! /bin/bash
export DISPLAY=:0
sudo killall omxplayer.bin > /dev/null 2>&1
omxplayer -b -o hdmi --no-osd /home/raspubli/
    raspubli82887b7a7f3ede1b2b29e3b23c1391af.mp4 > /dev/null 2>/dev/null &
```

   El *script* de ejecución de vídeo cambia la salida de vídeo a 'DISPLAY=:0' porque va a ser ejecutado por el servicio "*cron*", finaliza la ejecución de una posible reproducción de vídeo e inicia la reproducción del nuevo vídeo mediante el programa *OXMPLAYER*.

7. Cambiar los permisos de ejecución del *script*

```
$ ssh <user@ip_server> "chmod a+x /home/raspubli/init_video_<nombre_archivo
    >_mp4.sh"
```

8. Agregar planificación temporal al archivo de configuración de *cron*

```
$ ssh <user@ip_server> "echo \"<minutos> * * * * /home/raspubli/init_video_
    <nombre_archivo>_mp4.sh\" >> /home/raspubli/cron_video.txt"
```

9. Cargar archivo de configuración de *cron*

```
$ ssh <user@ip_server> "crontab /home/raspubli/cron_video.txt"
```

10. Reiniciar servicio *cron*

```
$ ssh <user@ip_server> "sudo /etc/init.d/cron restart"
```

4.4. Planificación de la ejecución de vídeos

Hemos definido tres posibles tipos de planificación para la reproducción de los vídeos. Estas configuraciones se realizan mediante dos parámetros, el primero indica el tipo de reproducción y el segundo el desplazamiento en minutos de inicio, debiendo ser este segundo distinto de cero. En caso de ser cero, se considera que el administrador quiere que no se reproduzca el vídeo.

- Cada hora (Parámetro 1 = 0): El vídeo con esta configuración es iniciado cada hora y minutos indicados en el Parámetro 2
- Cada hora impar (Parámetro 1 = 1): El vídeo con esta configuración es iniciado cada hora impar y minutos indicados en el Parámetro 2
- Cada hora par (Parámetro 1 = 2): El vídeo con esta configuración es iniciado cada hora par y minutos indicados en el Parámetro 2

El administrador del sistemas puede configurar cada vídeo con una planificación distinta; y es tarea de este que los vídeos no se solapen entre sí. Por ese motivo hemos elegido estos tres tipos de planificaciones repetitivas, para que el administrador disponga de mas posibilidades de configuración.

Capítulo 5

Diseño e implementación del servidor Gestor de Contenidos

5.1. Uso de KVM

La interfaz que *KVM* ofrece al usuario es mediante consola. Existen interfaces gráficas de terceros como *Virt-Manager*, *Qemu Launcher*, que trabajan como *front-end* de *KVM*.

Mediante consola, la sintaxis es simple:

```
virsh [ OPTION ]... [ COMMAND_STRING ]

virsh [ OPTION ]... COMMAND [ ARG ]...
```

- *OPTION* son las posibles opciones de que puede tener *KVM*.
- *COMAND_STRING_STRING* son los posibles comandos que se pueden ejecutar sobre una máquina virtual *KVM*.

Para obtener información sobre las posibles opciones y comandos, ejecutamos:

```
$ virsh --help
```

5.2. Uso de Virt-Manager

Virt-Manager es la interfaz gráfica elegida para *KVM*. Iniciamos la aplicación desde consola ejecutando:

```
$ virt-manager
```

Virt-Manager trabaja con una sola ventana (figura 5.1) excepto en los procedimientos tales como "Crear discos duros virtuales" o crear "Máquinas virtuales", los cuales son realizados mediante ventanas emergentes.
El entorno de trabajo está dividido en cuatro partes.:

1. Barra de menús: Archivo (*File*), Editar (*Edit*), Vista (*View*) y Ayuda (*Help*).
2. Barra de herramientas: Crear máquina virtual, Abrir máquina virtual, Ejecutar máquina virtual, Pausar máquina virtual y Apagar máquina virtual.
3. Área de máquinas virtuales. Formada por un listado de las máquinas virtuales disponibles.
4. Configuración de máquina vitual. Visible seleccionando la máquina en el Área de máquinas virtuales, haciendo clic en "editar" y "detalles de la máquina virtual". Ver figura 5.2.

5.2.1. Crear Máquina Virtual

Seleccionamos dentro del menú *Archivo* la opción *Nueva máquina virtual*. Ver figura 5.3. *Virt-Manager* muestra una nueva ventana con el primer paso del asistente para crear la nueva máquina virtual. El asistente esta compuesto por 5 pasos:

1. Elegir el modo de instalación del sistema operativo. Ver figura 5.4.
2. Seleccionar el origen de la instalación. Ver figura 5.5.
3. Elegir configuración de memoria y CPU. Ver figura 5.6.
4. Crear o seleccionar almacenamiento para la máquina virtual. Ver figura 5.7.
5. Seleccionar configuración de red, establecer nombre de la nueva máquina virtual y finalizar asistente. Ver figura 5.8.

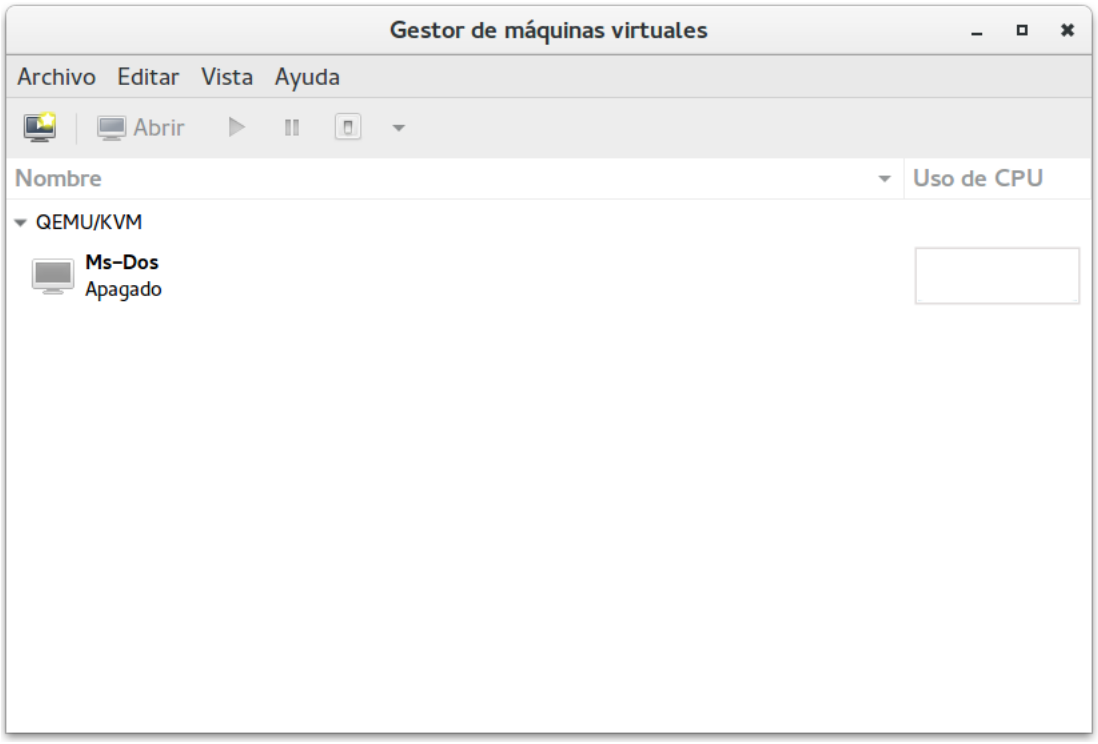


Figura 5.1: Inicio Virt-Manager.

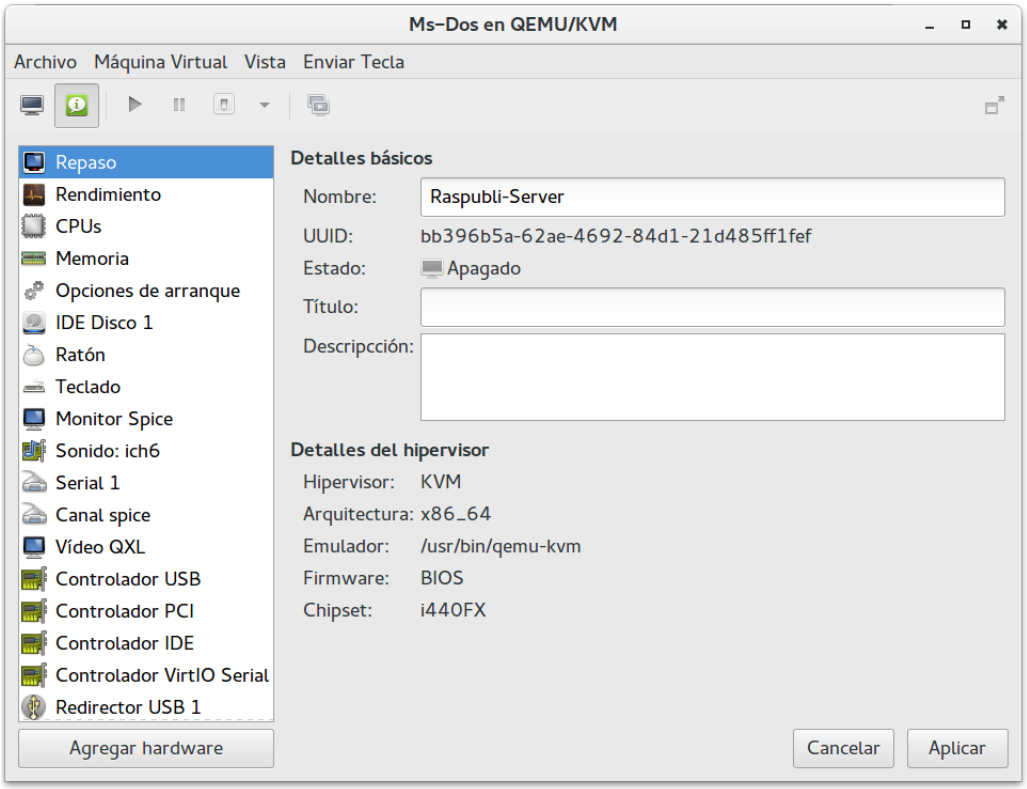


Figura 5.2: Configuración Máquina Virtual.

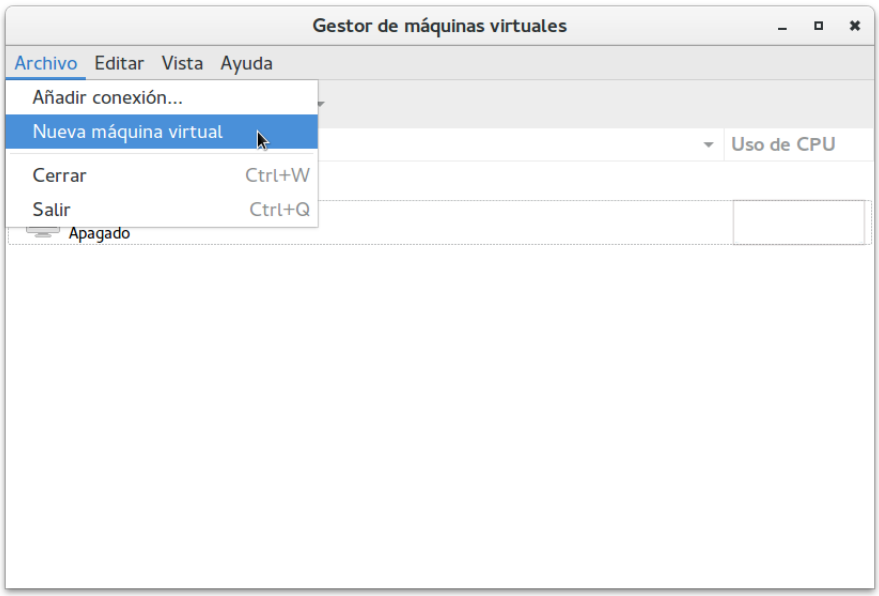


Figura 5.3: Crear Máquina Virtual.

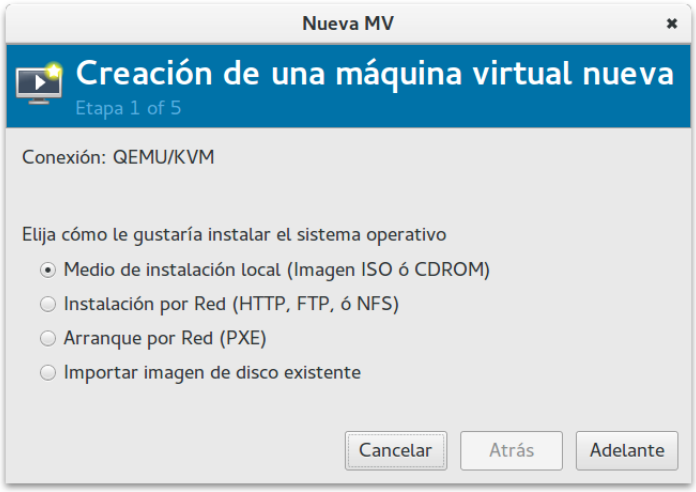


Figura 5.4: Crear Máquina Virtual. Paso 1

Figura 5.5: Crear Máquina Virtual. Paso 2

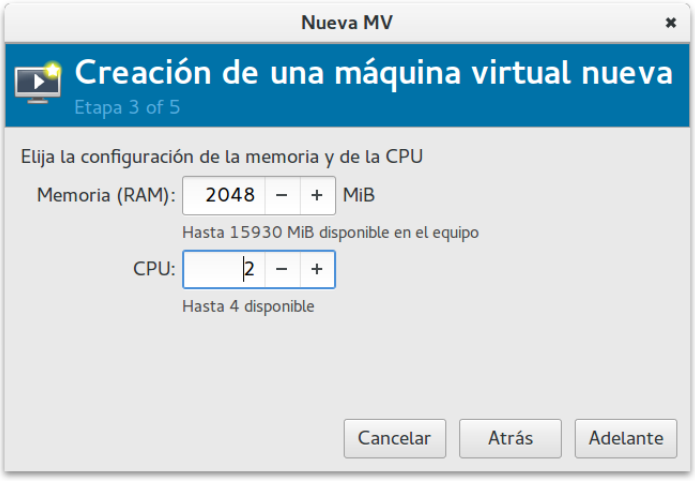


Figura 5.6: Crear Máquina Virtual. Paso 3

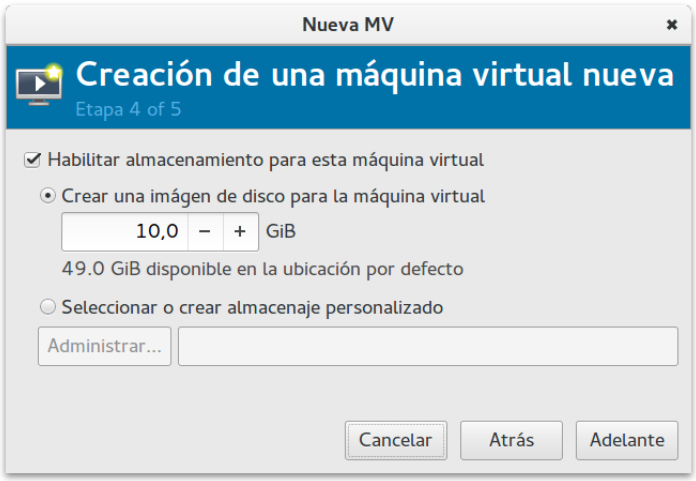


Figura 5.7: Crear Máquina Virtual. Paso 4

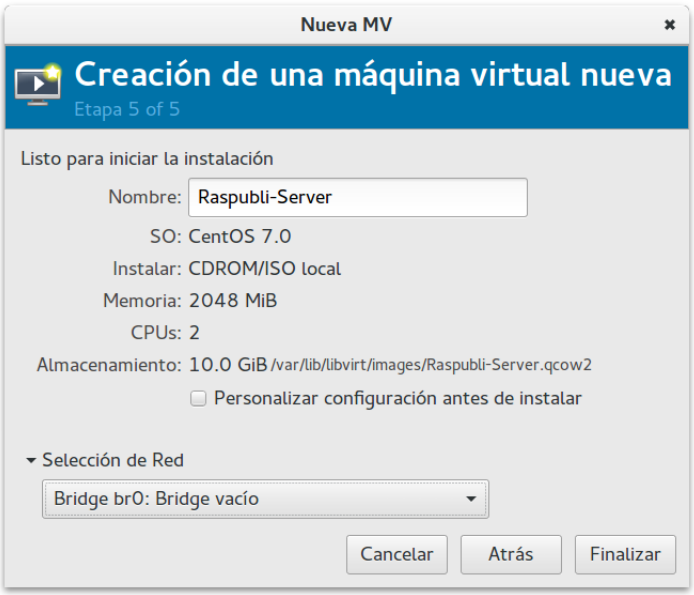


Figura 5.8: Crear Máquina Virtual. Paso 5

5.3. Instalación y configuración del sistema operativo

Realizamos la instalación mínima (*Minimal Install*) de la distribución *CentOS 7*.

5.4. Configuación de Red

Editamos el archivo de configuración de la interfaz de red `eth0`:

```
$ vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

Agregamos la siguiente configuración:

```
DEVICE=eth0
TYPE=Ethernet
UUID=ec611944-aa1b-4196-9d71-824b9ea83244
ONBOOT=yes
BOOTPROTO=none
IPADDR=10.40.0.200
PREFIX=16
GATEWAY=10.40.0.1
DNS1=8.8.8.8
DNS2=8.8.4.4
DEFROUTE=yes
```

Actualizamos el sistema completo:

```
$ yum -y update
```

5.5. Comunicación ssh mediante certificado

En el capítulo 4 establecimos que la comunicación entre el "Gestor de Contenidos" y los reproductores multimedia (clientes) es realizada mediante comandos enviados a través '*ssh*'. Sin embargo, este requiere que una auntenticación mediante contraseña o certificado.

Para que el servidor pueda comunicarse mediante certificado, este debe emitir una clave publica que posteriormente instalaremos en los clientes para que permitan

el acceso sin autenticación. A esto se le llama establecer una relación de confianza.

Al ser una aplicación web quien va a realizar las comunicaciones, el certificado debe ser emitido por el usuario que ejecuta el servicio 'httpd'. Este usuario es creado en la instalación del servicio httpd (ver sección 5.6.1). El usuario creado para la ejecución del servicio web es "*apache*".

Para generar el certificado hacemos los siguiente:

1. Cambiamos la configuración del usuario '*apache*' para que disponga de consola:
   Editamos el archivo `/etc/passwd`:

   ```
   $ vi /etc/passwd
   ```

   Sustituimos la siguiente linea:

   ```
   apache:x:48:48:Apache:/usr/share/httpd:/sbin/lologin
   ```

   por:

   ```
   apache:x:48:48:Apache:/usr/share/httpd:/bin/bash
   ```

2. Creamos el directorio donde vamos a guardar las claves generadas por el usuario '*apache*':

   ```
   $ mkdir /usr/share/httpd/.ssh/
   $ chown apache:apache /usr/share/httpd/.ssh/
   ```

3. Generamos las claves de autentificación, sin introducir una frase de paso (*passphrase*). Es necesario ejecutar el comando como usuario '*apache*':

   ```
   $ su apache -c 'ssh-keygen -t rsa -f /usr/share/httpd/.ssh/id_rsa'
   ```

   ```
   Generating public/private rsa key pair.
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /usr/share/httpd/.ssh/id_rsa.
   Your public key has been saved in /usr/share/httpd/.ssh/id_rsa.pub.
   The key fingerprint is:
   ```

```
42:e2:2f:2f:32:b9:b3:a5:a9:a6:2e:e9:40:ef:77:e2 apache@raspubli.localhost.
    localdomain
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|    . .          |
|   . o           |
| .  . . S        |
|. .  . .         |
|.. oo .          |
|+.*+.= .         |
|O+=BoE+          |
+-----------------+
```

Se generan dos archivos: '`id_rsa`' (clave privada) y '`id_rsa.pub`' (clave pública), este último es el necesario para crear la relación de confianza. En el capítulo 7 mostramos qué hacemos con él.

Verificamos que hemos generado correctamente las claves, visualizando el archivo y observando que al final aparece el nombre del usuario '*apache*' seguido del nombre de la máquina:

```
$ cat /usr/share/httpd/.ssh/id_rsa
```

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC0+GPnZrW9A5/Dom6MB3F2CxEEkup/3
    LMc4hpwSA2y50FdOwB1xaZxSMUADTClkRAODE5uC9mR77lNZOB9XkaAsP461XDiOk3U+
    bRoJYB3D22r3y1PEZJeFGZqcN0GlweQYN93UAT5V7NMyMrAV8X/
    T56h7XdXHflFYLCA8XFmfnepGy6aSQCfI+ezkEc2axYFw6Ouy/nm1ojTs/
    qNlxaF3ayRueKBsnvS/6pDhPsyNfDVWIQHi/prA3F+ZDYlfHOm4l4vn8It4+
    Z9exIjr2Mgd0UWyCqptRf/rB271baebI2Xmm88xdJ+2
    ErbRkl0783BInQZ8bWhYlEYng1qWju9 apache@raspubli.localhost.localdomain
```

5.6. Instalación y configuración de Httpd, MariaDB y PHP

Instalamos fácilmente el software necesario usando el gestor de paquetes de CentOS (*yum*). El gestor de paquetes nos permite instalar la mayoría del software desde un repositorio mantenido por CentOS. Para ver las posibles opciones, ejecutamos:

```
$ yum --help
```

5.6.1. Instalación Httpd

Instalamos el servicio de *http* mediante el siguiente comando:

```
$ yum -y install httpd
```

Una vez finalizada la instalación iniciamos el servicio y habilitamos el inicio automático en el arranque del sistema:

```
$ systemctl start httpd.service
$ systemctl enable httpd.service
```

Podemos hacer un chequeo del correcto funcionamiento accediendo a través de una navegador web:

```
$ http://10.40.0.200/
```

La figura 5.9 corresponde con una correcta respuesta del servicio *httpd* bien instalado.

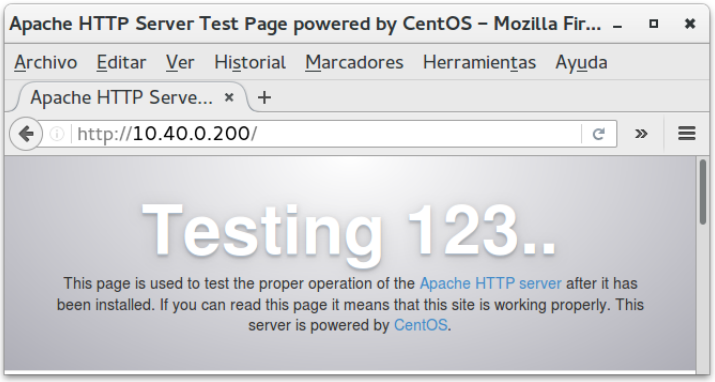


Figura 5.9: Comprobación servicio httpd

5.6.2. Instalación MariaDB

Además de instalar el servidor MariaDB (*mariadb-server*), instalamos un paquete (mariadb) de ayuda para una posterior configuración:

```
$ yum -y install mariadb-server mariadb
```

Una vez finalizada la instalación iniciamos el servicio y habilitados el inicio automático en el arranque del sistema:

```
$ systemctl start mariadb.service
$ systemctl enable mariadb.service
```

Ahora que la base de datos esta funcionando, ejecutamos un "*script*" con el que realizamos una configuración inicial y de seguridad básicos:

```
$ mysql_secure_installation
```

El sistema pide la contraseña del '*root*" actual, puesto que acabamos de instalar MariaDB, no tiene ninguna. Así que pulsamos "*Enter*" para continuar. A continuación, introducimos 'Y' para establecer una contraseña para el "*root*" de MariaDB:

```
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorization.

New password: <password>
Re-enter new password: <password>
Password updated successfully!
Reloading privilege tables..
 ... Success!
```

Para el resto de cuestiones, podemos dejarlas por defecto pulsando "*Enter*".

5.6.3. Instalación PHP

Instalamos el paquete "*php*" y el paquete "*php-mysql*", este último para poder comunicar nuestra aplicación con la base de datos.

```
$ yum -y install php php-mysql
```

Tras finalizar la instalación editamos el archivo de configuración de PHP para que permita la carga de archivos de gran tamaño y permita la ejecución de los *scripts* durante un largo periodo de tiempo. Esta modificación es necesaria por el simple motivo de que nuestro sistema carga archivos de vídeos los cuales son de gran volumen y pueden tardar mas tiempo de los normal en cargarse al servidor.

Editamos el archivo *php.ini*:

```
$ vi /etc/php.ini
```

Modificamos las siguientes directivas:

```
post_max_size = 256M
....
upload_max_filesize = 256M
....
max_execution_time = 60
```

Una vez finalizada la configuración debemos reiniciar el servidor *httpd* para que los cambios surtan efecto:

```
$ systemctl restart httpd.service
```

5.6.4. Configuración del sitio web

Los archivos de *host* virtuales son los que especifican la configuración de nuestros sitios separados y dictan que contenido responderá el servidor web httpd en función del dominio qué se solicita.

Estos archivos de configuración están ubicados en:

```
/etc/httpd/conf/conf.d/
```

Creamos el archivo para el dominio "*raspubli.local*":

```
$ vi /etc/httpd/conf.d/raspubli.local.conf
```

Establecemos la siguiente configuración básica:

```
<VirtualHost *:80>
        ServerAdmin webmaster@raspubli.locals
        ServerName      raspubli.local
        ServerAlias     www.raspubli.local
        DocumentRoot /var/www/raspubli.local/
        <Directory />
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Require all granted
        </Directory>
        ErrorLog /var/log/httpd/error.log
        TransferLog     /var/log/httpd/access.log
        LogLevel warn
        CustomLog /var/log/httpd/access.log combined
        ServerSignature On
```

```
</VirtualHost>
```

Por último, creamos el directorio base donde vamos a alojar nuestra aplicación:

```
$ mkdir /var/www/raspubli.local
```

5.6.4.1. Instalación del Framework Codeigniter

Para la instalación del *framework* necesitamos instalar previamente los paquetes '*wget*' y '*unzip*', mediante '*yum*'.
Descargamos el framework Codeigniter desde la 'web':

```
$ cd /var/www/raspubli.local
$ wget https://codeload.github.com/bcit-ci/CodeIgniter/zip/2.2.6 -O Codeigniter.
    zip
```

Descomprimimos el archivo y movemos todos los ficheros a la ruta base (`'/var/www/raspubli.local'`):

```
$ unzip Codeigniter.zip
$ mv ./CodeIgniter-2.2.6/* ./
```

Accediendo desde un navegador web con el dominio '*raspubli.local*' [1] obtendremos un resultado similar al de la figura 5.10.

5.6.4.2. Configuración de .htaccess y .htpasswd

El archivo `.htaccess` es el archivo de configuración del servidor *httpd* que permite definir directivas a nivel de directorio o de subdirectorios en donde se encuentra, sin necesidad de tener que editar la configuración principal del servidor *httpd*.

Nosotros lo usamos para especificar las restricciones de seguridad y aplicar las reglas de reescritura exigidas por el *framework Codeigniter*. Creamos el archivo en el directorio raíz del sitio web:

```
$ vi /var/www/raspubli.local/
```

[1] Previamente debemos haber resuelto el dominio *raspubli.local* modificando el archivo '*hosts*' o mediante un servidor DNS local

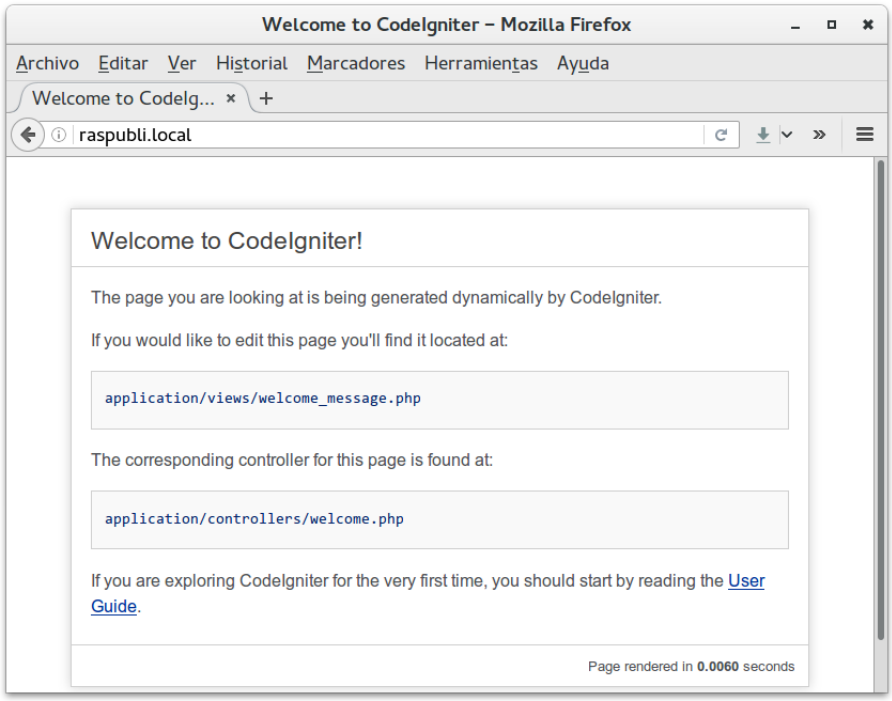


Figura 5.10: Verificación del servicio web y framework Codeigniter

Las directivas exigidas por *Codeigniter* que tenemos que incluir son las siguientes:

```
RewriteEngine on

RewriteCond $1 !^(index\.html|index\.php|resources)
RewriteRule ^(.*)$ /index.php/$1 [L]
```

Para la seguridad añadimos las siguientes directivas:

```
AuthType Basic
AuthName "Área Restringida - Indroduzca usuario y contraseña"
AuthUserFile /var/www/raspubli.local/.htpasswd
Require valid-user
```

La directiva `AuthUserFile` hace referencia al archivo `.htpasswd`; este archivo contiene las credenciales que debe introducir el usuario para poder acceder al contenido web.

El comando `htpasswd` se utiliza para crear y actualizar archivos planos utilizados para almacenar nombres de usuario y contraseña para autenticación de usuarios en servidores *Http*. La sintaxis es:

```
htpasswd [-cimBdpsDv] [-C cost] <archivo_destino> <usuario>
```

Generamos el archivo `.htpasswd` con el siguiente comando:

```
$ htpasswd -c .htpasswd <usuario>
```

```
$ htpasswd -c .htpasswd admin
New password:
Re-type new password:
Adding password for user admin
```

5.7. El gestor multimedia

El gestor multimedia o gestor de contenidos esta completamente desarrollado en PHP, es el elemento principal y mas complejo de todo el proyecto, esta separado en tres partes principales:

1. **Reproductores:** Sección destinada a agregar, editar y eliminar dispositivos reproductores (clientes) del sistema. Ver figura 5.11.

2. **Contenido M:** En esta sección añadimos al servidor tantos vídeos y presentaciones como deseemos para su posterior uso en los clientes. Ver figura 5.12.

3. **Configuración R:** Sección que combina los datos de la primera y segunda sección para poder programar las reproducciones. Ver figura 5.13.

En el capítulo 6 realizamos el análisis detallado del "Gestor multimedia".

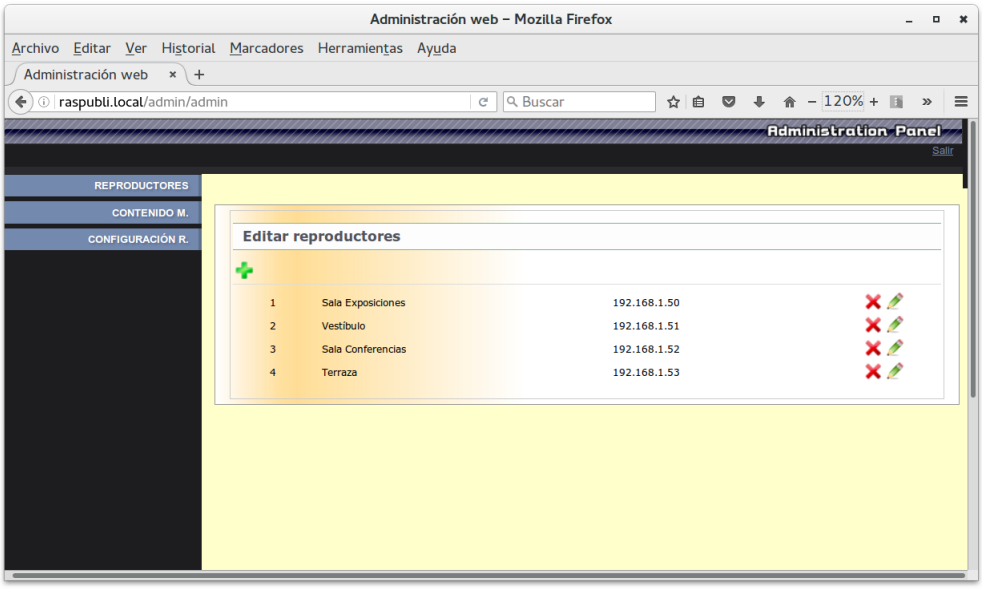


Figura 5.11: Gestor multimedia - Editar reproductores

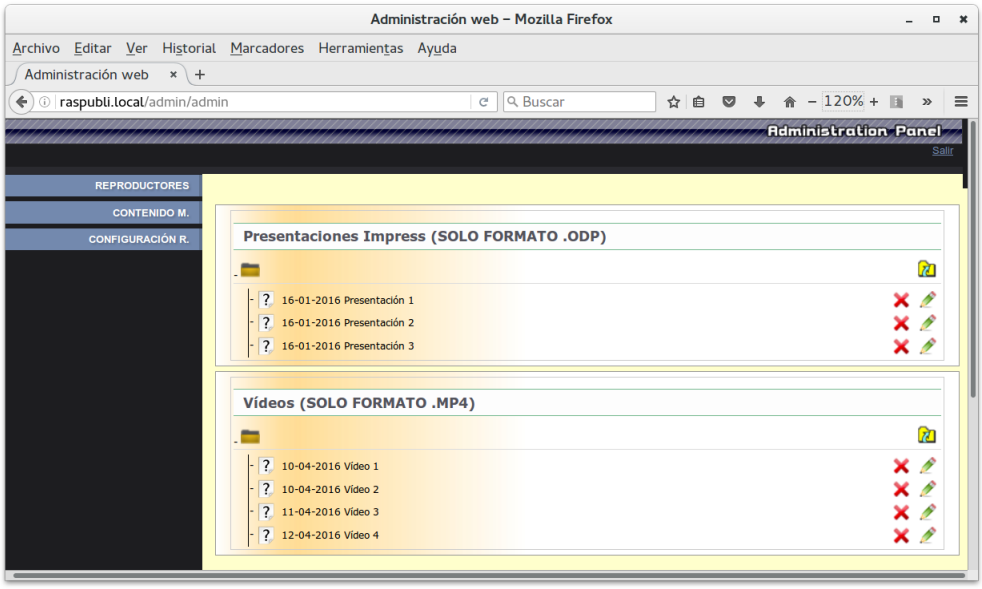


Figura 5.12: Gestor multimedia - Contenido Multimedia

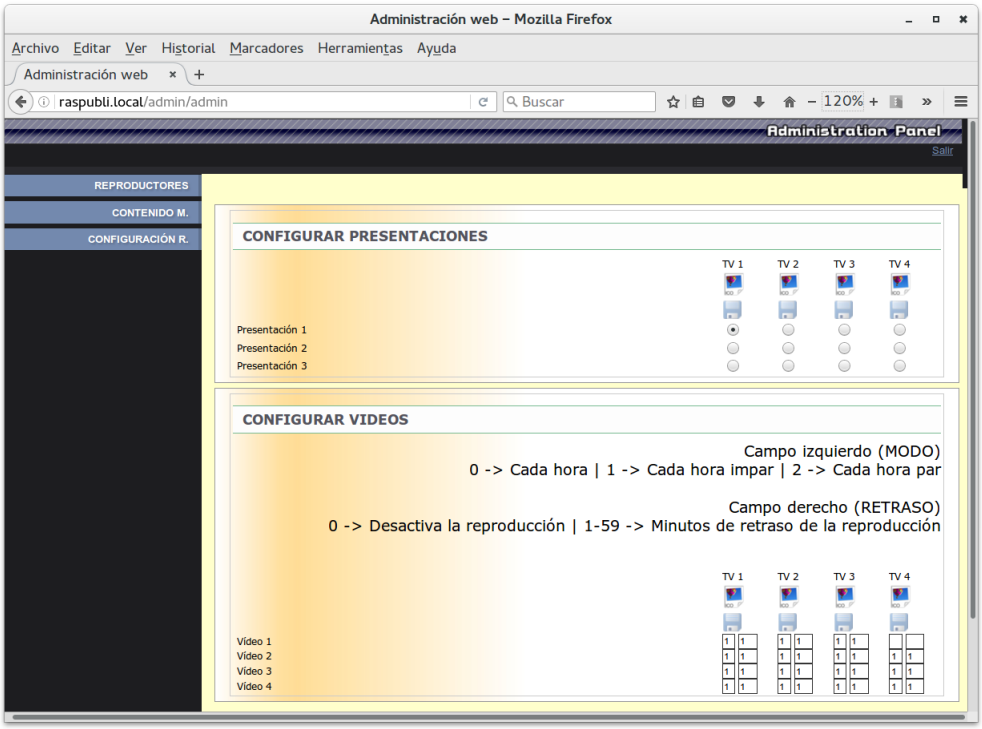


Figura 5.13: Gestor multimedia - Configuración Reproductores

Capítulo 6

Análisis

6.1. Introducción

Todo el sistema esta enfocado en una **programación orientada a objetos (POO)**, partiendo de la clase principal *raspubli o gestor* compuesta por tres métodos principales que corresponden con las tres partes antes descritas: Reproductores, Contenido y Configuración.

En todas las clases, los atributos se definen en base de datos, mediante una tabla (asociada a cada clase) en la que cada columna representa a un atributo y cada fila un objeto instanciado de dicha clase.

Los métodos definidos en cada clase operan sobre los atributos almacenados en la base de datos. Estos atributos pueden ser de tipo básico de la propia base de datos o de otra *clase* definida.

6.2. La Clase Raspubli

Denominamos a la clase principal como "`Raspubli`", una instancia de esta clase crea un objeto "*Servidor Gestor de Contenidos*" (ver sección 5).

La clase esta implementada como "*Controlador*" en el esquema de trabajo del *FrameWork Codeigniter*, y hace uso de las demás clases (librerías), dando como resultado las "*vistas*". Respetando así el modelo de trabajo: modelo-vista-controlador.

La clase **"raspubli"** contiene tres atributos:

- `Directorio Presentaciones`: Objeto de la clase "`Directorios`" que gestiona un conjunto de presentaciones.
- `Directorio Vídeos`: Objeto de la clase "`Directorios`" que gestiona un conjunto de vídeos.
- `Conjunto de Reproductores`: Objeto de la clase "`Reproductores`" que gestiona un conjunto de reproductores.

Esta clase puede ser interpretada como librería y ser incorporada a otro proyecto de mayor alcance que incorpore una sección que tenga que resolver la misma problemática de este proyecto.

6.3. Clases y diagrama de clases

Las clases que hemos definido para realizar el gestor de contenidos son:

La clase **"Lista"**: Gestionada de forma nativa a través de la base de datos. Por ejemplo, un objeto '`Lista`', lo tratamos como una tabla que almacena tantas filas como objetos contiene la lista. Todas la operaciones que podemos hacer sobre una tabla en la base de datos son los métodos que tiene la clase '`Lista`' y las columnas, con sus diferentes tipos, son los atributos.

La clase **"Reproductores"** contiene un atributo:

- `Lista de reproductores`: Lista que contiene objetos de clase "`Reproductor`"

La clase **"Reproductor"** contiene cuatro atributos:

- `Nombre`: Objeto de la clase "`varchars`" que contiene el nombre del reproductor.
- `IP`: Objeto de la clase "`varchars`" que contiene la dirección IP del reproductor.

- `Presentación`: Objeto de la clase "`archivos`" que contiene el archivo .odt que tiene que reproducir.
- `Lista de vídeos`: Lista de la clase "`configuración de vídeo`" que contiene los vídeos que tiene que reproducir y su configuración horaria.

La clase **"Configuración de vídeo"** contiene tres atributos:

- `Archivo`: Objeto de la clase "`archivos`" que contiene el archivo correspondiente a un vídeo.
- `tiempo`: Atributo básico de tipo "`entero`" para guardar la configuración de la reproducción del vídeo.
- `modo`: Atributo básico de tipo "`entero`" para guardar la configuración de la reproducción del vídeo.

La clase **"Directorios"** contiene un atributo:

- `Lista de archivos`: Lista que contiene objetos de la clase "`archivos`".

La clase **"Archivos"** contiene seis atributos:

- `Descripción`: Objeto de la clase "`varchars`" para guardar la descripción del archivo.
- `Fecha`: Atributo básico de tipo `Fecha` para guardar un valor de fecha.
- `Nombre`: Atributo básico de tipo `Cadena de Caracteres` para guardar el nombre del archivo almacenado en el servidor.
- `Formato`: Atributo básico de tipo `Cadena de Caracteres` para guardar el formato (extensión) del archivo almacenado en el servidor.
- `Path`: Atributo básico de tipo `Cadena de Caracteres` para guardar la ruta en donde esta almacenado el archivo en el servidor.

La clase **"Varchars"** contiene un atributo:

- `Nombre`: Atributo básico de tipo `Cadena de Caracteres` para guardar el nombre del archivo almacenado en el servidor.

Todas las clases tiene como mínimo dos métodos: "`crear_objeto`" y "`eliminar_objeto`".

- `crear_objeto`: Es el constructor de la clase y devuelve el 'id' (identificador) del objeto que crea. Este método crea una fila en la tabla que corresponde con la clase en la base de datos.
- `eliminar_objeto`: Destructor de la clase. A este método hay que pasarle un 'id' de un objeto previamente creado, y lo elimina. Este método eliminar una fila de la tabla que corresponde con la clase en la base de datos.

La figura 6.1 representa el diagrama de clases implementadas.

6.4. Modelo de datos

En la figura 6.2 podemos ver el modelo de datos, podemos observar una correspondencia con el diagrama de clases al ser este, una implementación de los atributos de las clases.

Tal como hemos definido el *diagrama de clases* y *modelo de datos*, la aplicación permite una gran escalabilidad de la solución.

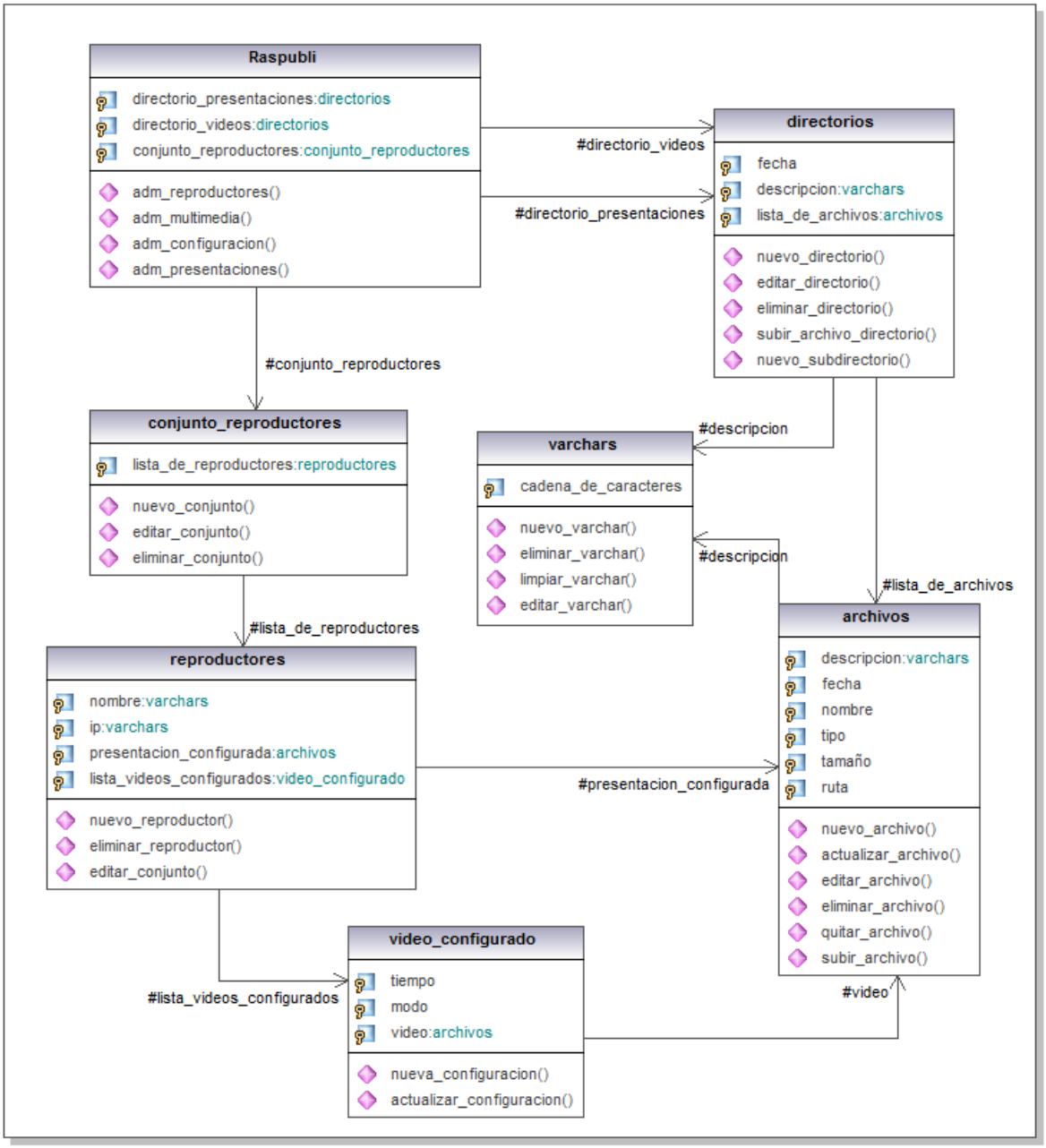


Figura 6.1: Diagram de clases

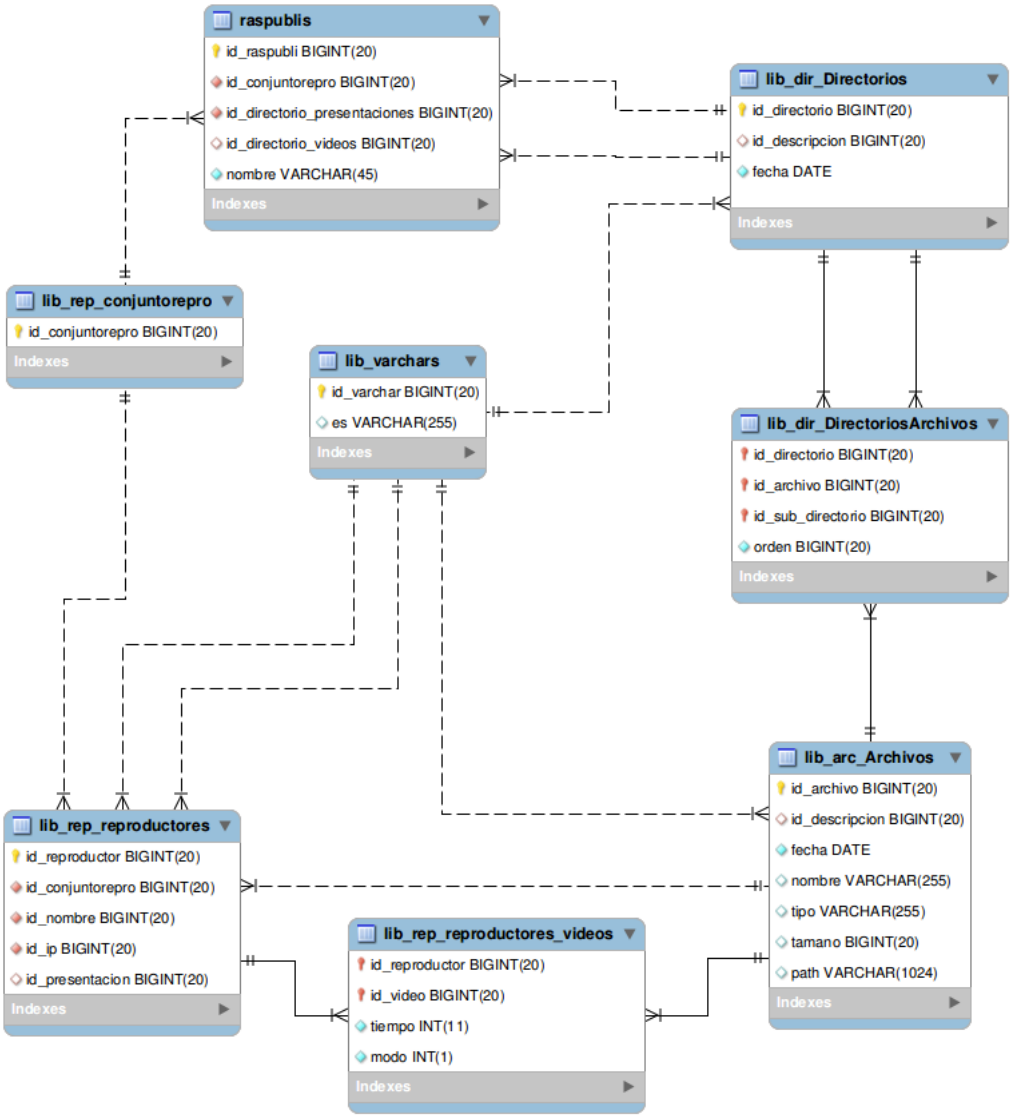


Figura 6.2: Modelo de datos

6.5. Casos de uso

6.5.1. Crear nuevo reproductor

Caso de uso	Crear nuevo reproductor
Resumen	Antes de configurar un reproductor, debe ser registrado en el sistema
Actores	Administrador
Precondición	Ninguna
Poscondición	El nuevo reproductor debe ser configurado
Curso Normal	1. El Administrador accede al gestor 2. El Administrador entra en "Reproductores" 3. El Administrador agrega un nuevo reproductor haciendo clic en el botón "Nuevo Reproductor" () 4. El Administrador comprueba que se ha creado una nueva fila en la lista de reproductores y da por creado el nuevo reproductor
Curso Alternativo	No existe
Observaciones	Al crear un reproductor, el sistema no devuelve aviso de que se ha creado

6.5.2. Configurar reproductor

Caso de uso	Configurar reproductor
Resumen	Es necesario configurar la dirección IP y asignarle un nombre a los reproductores
Actores	Administrador
Precondición	El reproductor no debe estar configurado o debe estar mal configurado
Poscondición	El reproductor puede ser cargado con presentaciones y vídeos programados
Curso Normal	1. El Administrador accede al gestor 2. El Administrador entra en "Reproductores" 3. Aparece una lista de reproductores creados en el sistema. Cada fila corresponde a un reproductor y muestra los datos configurados y dos botones: "Eliminar reproductor" () y "Editar reproductor" (). El Administrador activa la edición haciendo clic en el botón "Editar Reproductor" 4. El Administrador introduce los datos del reproductor, asignando un nombre y una dirección ip 5. El Administrador guarda los datos haciendo clic en los botones "Guardar" ()
Curso Alternativo	No existe
Observaciones	El campo IP también puede ser configurado con un nombre de dominio que pueda ser resuelto por el servidor

6.5.3. Eliminar reproductor

Caso de uso	Eliminar reproductor
Resumen	En caso de no necesitar o no existir un reproductor, este puede ser eliminar del sistema
Actores	Administrador
Precondición	El reproductor debe estar creado en el sistema
Poscondición	El reproductor no podrá ser gestionado
Curso Normal	1. El Administrador accede al gestor 2. El Administrador entra en "Reproductores" 3. Aparece una lista de reproductores creados en el sistema. Cada fila corresponde a un reproductor y muestra los datos configurados y dos botones: "Eliminar reproductor" () y "Editar reproductor"() 4. El Administrador introduce los datos del reproductor, asignando un nombre y una dirección ip 5. El Administrador elimina el reproductor haciendo clic en el botón "Eliminar" () 6. El Administrador debe confirmar que desea eliminar el reproductor en un cuadro de diálogo emergente
Curso Alternativo	El Administrador no confirma la eliminación del reproductor y el procedimiento se aborta
Observaciones	El reproductor queda eliminado del sistema, pero no se elimina la configuración que el dispositivo tiene cargada, este continuará funcionando de forma independiente con la última configuración

6.5.4. Agregar nueva presentación o vídeo

Caso de uso	Agregar nueva presentación o vídeo
Resumen	Para poder reproducir una presentación o vídeo en un reproductor, previamente debe ser cargada en el gestor
Actores	Administrador
Precondición	El archivo debe ser de tipo '.otp' o '.mp4'
Poscondición	Si el archivo cargado es de tipo '.otp', este aparecerá en el listado de presentaciones disponibles para cargar en los reproductores
Curso Normal	1. El Administrador accede al gestor 2. El Admin. entra en "Contenido Multimedia" 3. Aparece dos secciones claramente diferenciadas, el Administrador hace clic en el botón "Cargar archivo" () de la sección correspondiente (vídeo o presentación) 4. El Administrador selecciona el archivo en el navegador de archivos emergente del propio sistema operativo 5. El sistema valida el tipo de archivo y el tamaño y añade el archivo al sistema
Curso Alternativo	1. El archivo es de tamaño superior al admitido por el servidor 2. El archivo no es de tipo '.otp' ni '.mp4'
Observaciones	Podemos cargar en el gestor tantas presentaciones como deseemos, posteriormente podremos elegir que presentación deseamos cargar en cada reproductor

6.5.5. Eliminar vídeo o presentación

Caso de uso	Eliminar vídeo o presentación
Resumen	Cuando un vídeo o presentación ya no se va a volver a usar, este puede ser eliminado
Actores	Administrador
Precondición	El vídeo o presentación debe estar cargado en el sistema
Poscondición	El vídeo o presentación no volverá a parecer en el listado de "Configuración Reproductores"
Curso Normal	1. El Administrador accede al gestor 2. El Administrador entra en "Contenido Multimedia" 3. El Administrador hace clic en el botón "Eliminar archivo" (❌) del vídeo o presentación que desea eliminar 4. El Administrador debe confirmar que desea eliminar el archivo en un cuadro de diálogo emergente
Curso Alternativo	No existe
Observaciones	El vídeo o presentación queda eliminado del sistema, pero si un reproductor esta usando el contenido eliminado, este lo sigue usando hasta que se vuelva a configurar el reproductor con otro contenido

6.5.6. Editar vídeo o presentación

Caso de uso	Editar vídeo o presentación
Resumen	Cada vídeo o configuración puede ser nominado mediante un nombre y/o fecha
Actores	Administrador
Precondición	Ninguna
Poscondición	Ninguna
Curso Normal	1. El Administrador accede al gestor 2. El Administrador entra en "Contenido Multimedia" 3. El Administrador hace clic en el botón "Editar archivo" () del vídeo o presentación que desea eliminar 4. Los datos 'Fecha' y 'Nombre' cambian a ser campos editables y el Administrador introduce los nuevos datos. 5. El Administrador guarda los datos haciendo clic en los botones "Guardar" ()
Curso Alternativo	No existe
Observaciones	Los datos son meramente informativos, cuya función es organizativa hacia el Administrador

6.5.7. Cargar presentación en reproductor

Caso de uso	Cargar presentación en reproductor
Resumen	Cargar una presentación en un reproductor para que la reproduzca
Actores	Administrador
Precondición	La presentación debe estar cargada en el sistema
Poscondición	La presentación actual del reproductor sera sustituida por la nueva
Curso Normal	1. El Administrador accede al gestor 2. El Administrador entra en "Configurar Reproductor" 3. Aparecen dos secciones claramente diferenciadas, "Configurar Presentaciones" y "Configurar Vídeos". Cada una de ellas esta estructurada en filas y columnas. Las filas indican los posibles contenidos multimedia a cargar, y las columnas los reproductores disponibles. El administrador selecciona la presentación que desea cargar en la columna del reproductor a configurar 4. El Administrador hace clic en el "Guardar" () 5. El sistema informa mediante una ventana emergente que el proceso puede tardar más de un minuto
Curso Alternativo	1. El sistema devuelve 'error' si no puede conectar con el reproductor
Observaciones	Este proceso puede realizarse para todos los reproductores en paralelo

6.5.8. Cargar vídeos reproductor

Caso de uso	Cargar vídeos en reproductor
Resumen	Cargar una programación de videos en un reproductor para que la reproduzcan
Actores	Administrador
Precondición	La vídeos deben estar cargados en el sistema
Poscondición	La presentación actual del reproductor sera sustituida por la nueva
Curso Normal	1. El Administrador accede al gestor 2. El Administrador entra en "Configurar Reproductor" 3. Aparecen dos secciones claramente diferenciadas, "Configurar Presentaciones" y "Configurar Vídeos". Cada una de ellas esta estructurada en filas y columnas. Las filas indican los posibles contenidos multimedia a cargar, y las columnas los reproductores disponibles. El administrador configura los parámetros de la columna correspondiente al reproductor que desea configurar 4. El Administrador hace clic en el "Guardar" (💾) 5. El sistema informa mediante una ventana emergente que el proceso puede tardar más de un minuto
Curso Alternativo	1. El sistema devuelve 'error' si no puede conectar con el reproductor
Observaciones	Este proceso puede realizarse para todos los reproductores en paralelo

6.5.9. Obtener captura de un reproductor

Caso de uso	Obtener captura de un reproductor
Resumen	Obtener un captura de pantalla de lo que un reproductor esta emitiendo en un momento concreto
Actores	Administrador
Precondición	La vídeos deben estar cargados en el sistema
Poscondición	La presentación actual del reproductor sera sustituida por la nueva
Curso Normal	1. El Administrador accede al gestor 2. El Administrador entra en "Contenido Multimedia" 3. El Administrador hace clic en el botón "Obtener captura" () de la columna correspondiente al reproductor del que desea realizar la captura de pantalla 4. El Administrador visualiza la captura de pantalla en una nueva ventana del navegador
Curso Alternativo	1. El sistema devuelve 'ERROR DE CONEXION' si no puede conectar con el reproductor
Observaciones	Para comprobar el correcto funcionamiento del sistema, es posible realizar capturas.

Capítulo 7

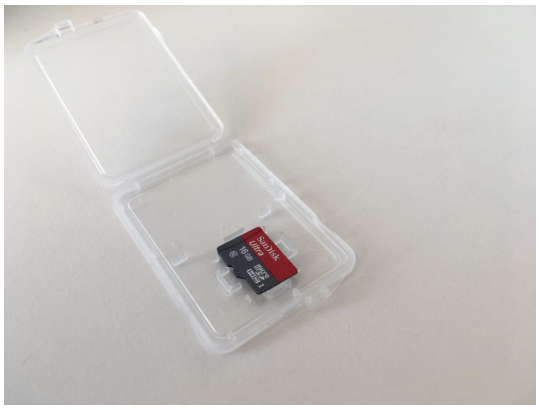
Diseño e implementación del cliente Reproductor Multimedia

7.1. Hardware - Raspberry Pi

En la sección 3.10 describimos el ordenador de placa simple (SBC) que vamos a utilizar en la implementación del Reproductor Multimedia. Todos los materiales necesarios para una completa implementación son los siguientes:

1. Placa Rasberry Pi 3 Model B

2. Tarjeta de memoria SD Sandisk 16 GB microSDHC UHS-I



3. Caja para RRasberry Pi Model B

4. Cable HDMI

5. Cable de alimentación microusb

6. Teclado USB

7. Monitor con entrada HDMI

7.2. Instalación del sistema operativo

Descargamos la imagen de la web oficial de ubuntu-mate: **(https://ubuntu-mate.org/raspberry-pi/)**:

```
$ wget https://ubuntu-mate.org/raspberry-pi/ubuntu-mate-16.04.2-desktop-armhf-
    raspberry-pi.img.xz
```

Descomprimimos con el programa 'unxz':

```
$ unxz ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img.xz
```

La imagen puede ser grabada en la tarjeta SD Sandisk 16GB con el programa 'dd' o 'ddrescue', para ello necesitamos saber cómo está la tarjeta SD presente

en el sistema, con los siguientes comandos obtenemos información que nos ayuda a saber que dispositivo de bloques corresponde con la tarjeta SD:

```
$ lsblk
....
$ ls -l /dev/disk/by-id/
```

Una vez que sabemos que dispositivo de bloques, procedemos a grabar la imagen en la tarjeta SD:

```
$ ddrescue -D --force ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img /dev/
    mmcblk0
```

Una vez finalizado el proceso de grabación, introducimos la SD en la *Raspberry Pi*, y la encendemos a través del cable de alimentación USB.
En el primer inicio del sistema comienza un asistente de instalación del *Ubuntu Mate*. La figura 7.1 muestra el comienzo del asistente, que corresponde con la selección del idioma:

Figura 7.1: Asistente de instalación - Selección de idioma

Paso dos (figura 7.2) - Ubicación:

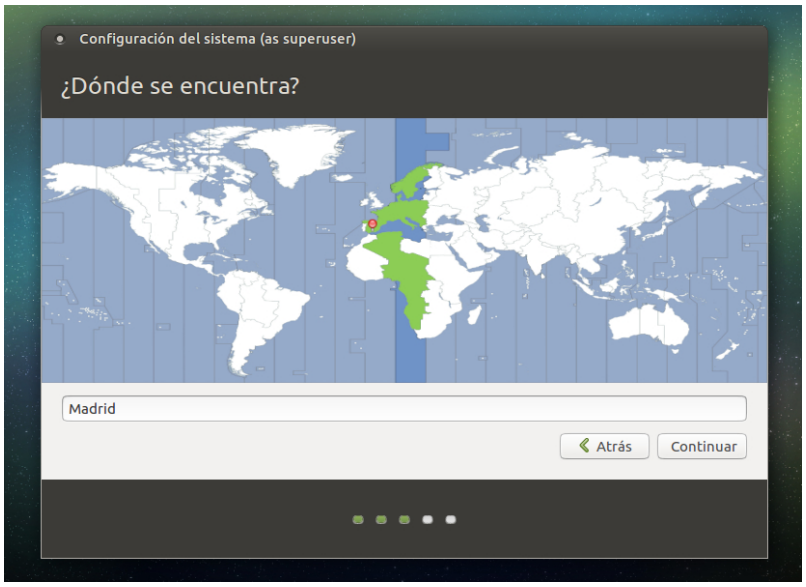


Figura 7.2: Asistente de instalación - ¿Dónde se encuentra?

Paso tres (figura 7.3) - Disposición del teclado:

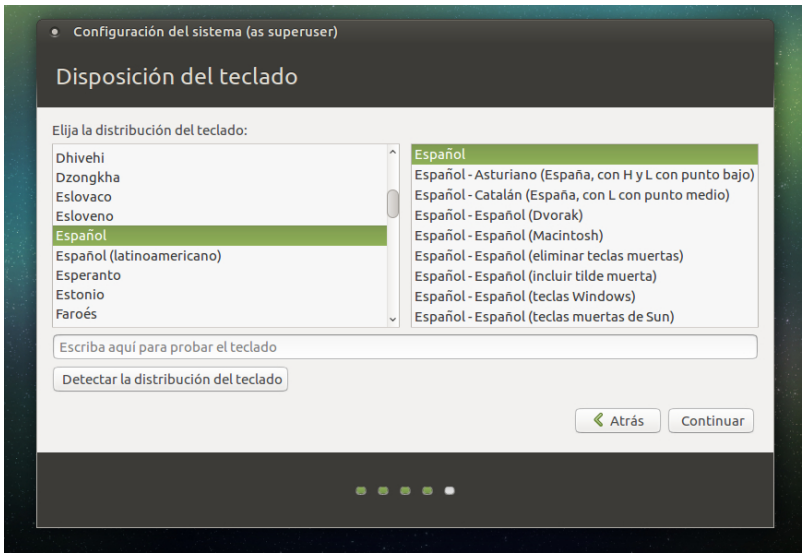


Figura 7.3: Asistente de instalación - Disposición del teclado

Paso cuatro (figura 7.4) - Datos de usuario: En este paso es importante marcar la opción **"Inicar sesión automáticamente"** y el nombre de usuario debe ser **"raspubli"**.

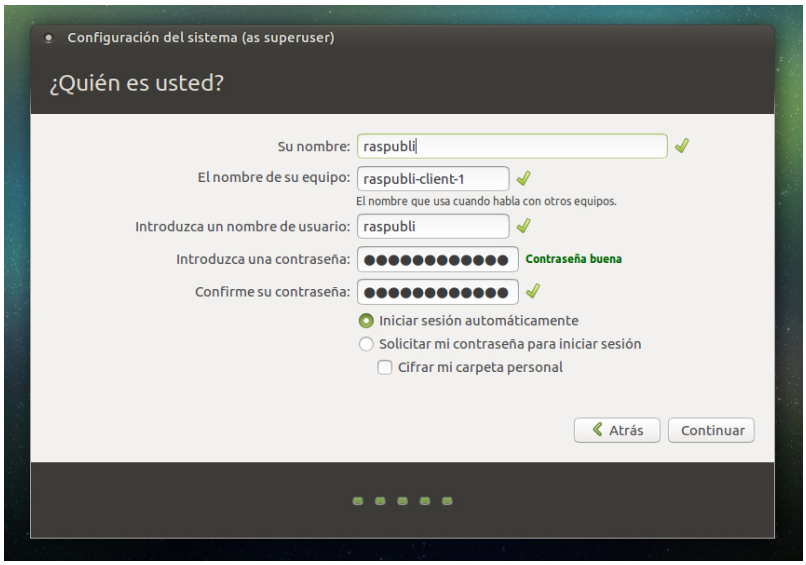


Figura 7.4: Asistente de instalación - ¿Quién es usted?

Despues del cuarto paso comienza la instalación: ver figura 7.5.

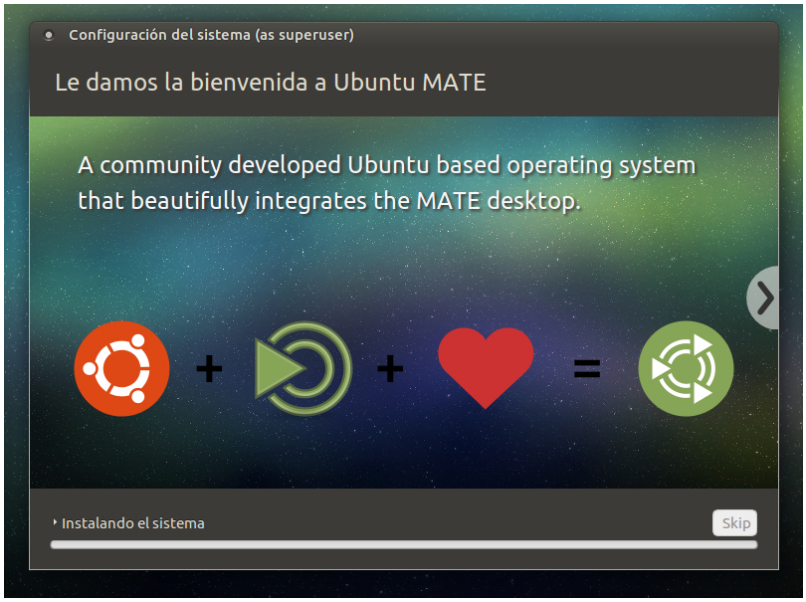


Figura 7.5: Instalación del sistema

Llegados a este punto, la instalación se da por finalizada, el sistema se reinicia y podemos proceder a la configuración. (Ver sección 7.3).

7.3. Configuración

7.3.1. Configuración de la red

Paso uno (figura 7.6) - Inicio de la configuración: La configuración de la red, la realizamos mediante el programa "Network Manager", para iniciar el programa, seleccionamos el menú `'Sistema' ->'Preferencias' ->'Internet y red' ->'Network Connections'`.

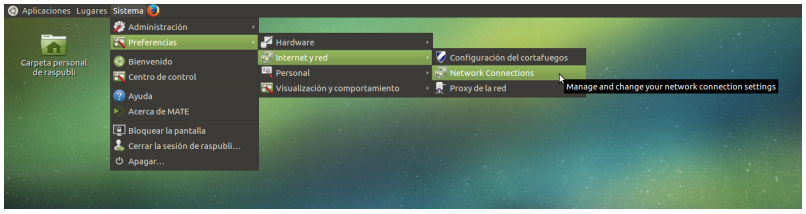


Figura 7.6: Configuración de la red - Inicio de la configuración

Paso dos (figura 7.7) - Agregar conexión: Hacemos clic en 'Add' para agregar una nueva conexión de red:

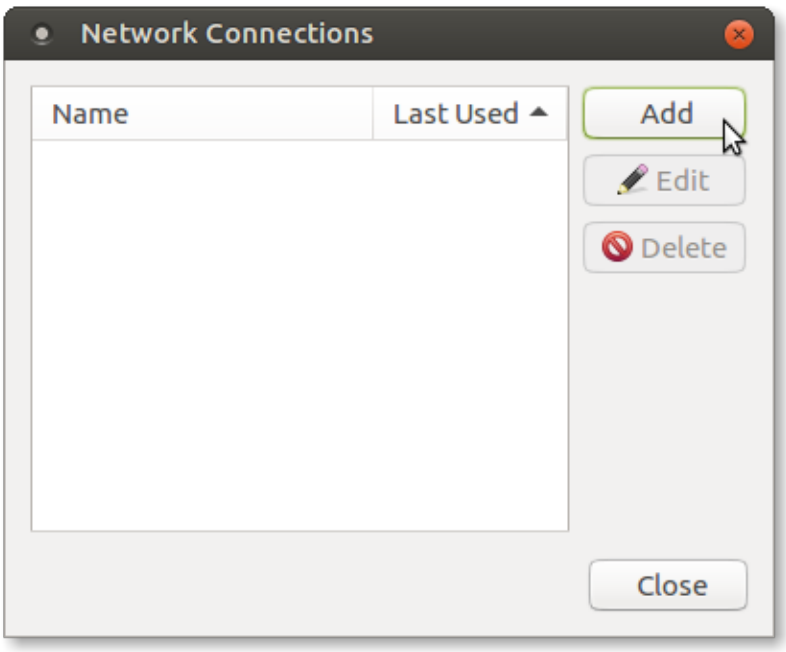


Figura 7.7: Configuración de la red - Agregar conexión de red

Paso tres (figura 7.8) - Seleccionar tipo de conexión: El primer paso para agregar un conexión, es seleccionar el tipo de conexión, en este caso, seleccionamos 'Ethernet'.

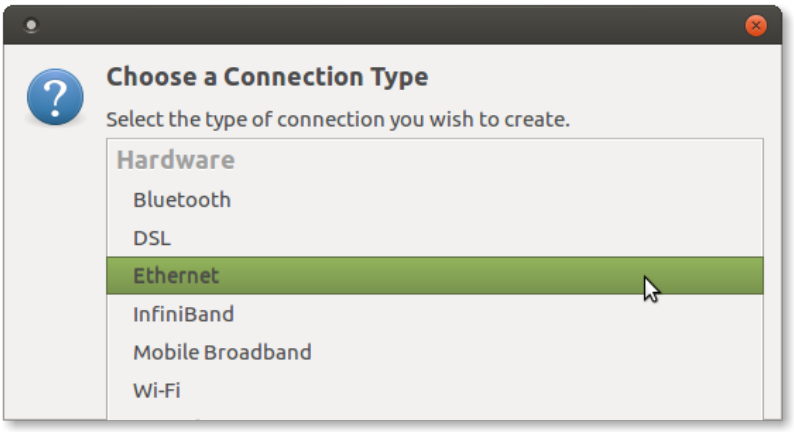


Figura 7.8: Configuración de la red - Seleccionar tipo de conexión

Paso cuatro - Configuración de la conexión: En la pestaña '*Ethernet*' seleccionamos el dispositivo que va a hacer uso la nueva conexión (Ver figura 7.9) y en la pestaña '*IPv4 Settings*' configuramos el protocolo IP. Es estrictamente necesario que el 'Método' sea 'Manual' para poder establecer una IP fija, la cual es usada por el "Gestor de contenidos" para realizar la conexión a través de la red.

Una vez seleccionado el método 'Manual', agregamos la configuración IP haciendo clic en 'Add' (Ver figura 7.10), una vez establecida la configuración hacemos clic en 'Save' para guardar la configuración.

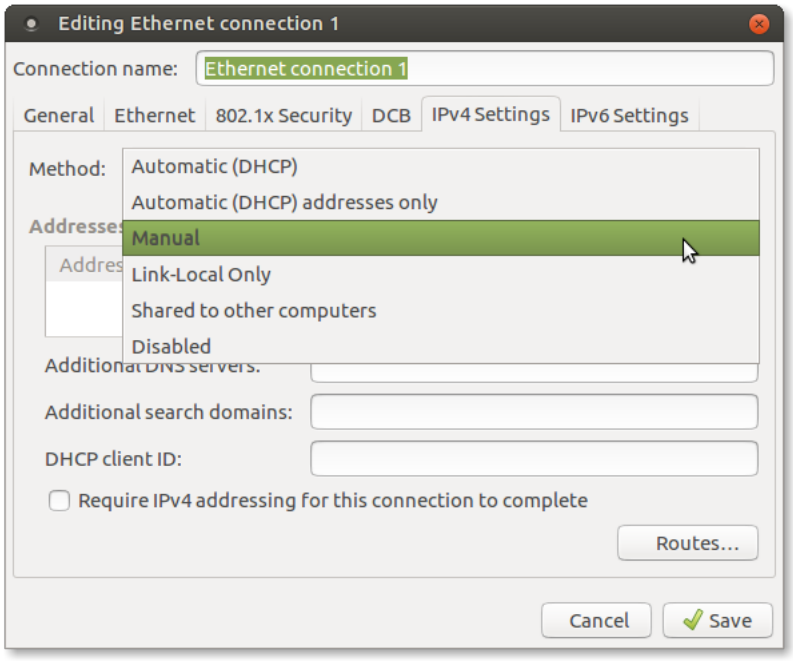


Figura 7.9: Configuración de la red - Seleccionar tipo de conexión

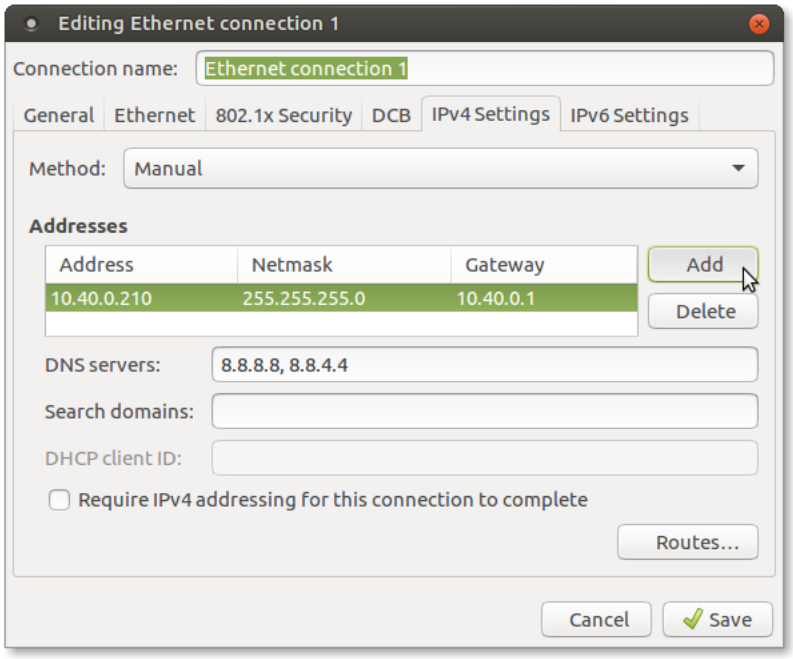


Figura 7.10: Configuración de la red - Configuración IP

Reiniciamos el sistema y la configuración de red queda completa. Una vez reiniciado el sistema, probamos la conexión mediante el comando '`ping`' desde el servidor:

```
$ ping 10.40.0.210
```

```
PING 10.40.0.210 (10.40.0.210) 56(84) bytes of data.
64 bytes from 10.40.0.210: icmp_seq=1 ttl=64 time=1.13 ms
64 bytes from 10.40.0.210: icmp_seq=2 ttl=64 time=0.178 ms
64 bytes from 10.40.0.210: icmp_seq=3 ttl=64 time=0.174 ms
...
```

7.3.2. Configuración del sistema

Para un correcto funcionamiento del sistema configuramos la salida HDMI y gestión de energía.

Para cambiar la resolución de salida, editamos el archivo '`/boot/config.txt`':

```
$ sudo vim /boot/config.txt
```

Descomentamos '`hdmi_goup`' y '`hdmi_mode`':

```
hdmi_group = 1
hdmi_mode = 16
```

Establecemos el valor de '`hdmi_mode`' según los valores que se muestran a continuación:

```
Value    hdmi_group=CEA                  hdmi_group=DMT
-----------------------------------------------------------------------------
1        VGA                             640x350   85Hz
2        480p  60Hz                      640x400   85Hz
3        480p  60Hz  H                   720x400   85Hz
4        720p  60Hz                      640x480   60Hz
5        1080i 60Hz                      640x480   72Hz
6        480i  60Hz                      640x480   75Hz
7        480i  60Hz  H                   640x480   85Hz
8        240p  60Hz                      800x600   56Hz
9        240p  60Hz  H                   800x600   60Hz
10       480i  60Hz  4x                  800x600   72Hz
11       480i  60Hz  4x H                800x600   75Hz
12       240p  60Hz  4x                  800x600   85Hz
13       240p  60Hz  4x H                800x600   120Hz
14       480p  60Hz  2x                  848x480   60Hz
15       480p  60Hz  2x H                1024x768  43Hz  DO NOT USE
16       1080p 60Hz                      1024x768  60Hz
17       576p  50Hz                      1024x768  70Hz
18       576p  50Hz  H                   1024x768  75Hz
```

```
      19       720p  50Hz                      1024x768  85Hz
      20       1080i 50Hz                      1024x768  120Hz
      21       576i  50Hz                      1152x864  75Hz
      22       576i  50Hz  H                   1280x768        reduced blanking
      23       288p  50Hz                      1280x768  60Hz
      24       288p  50Hz  H                   1280x768  75Hz
      25       576i  50Hz  4x                  1280x768  85Hz
      26       576i  50Hz  4x H                1280x768  120Hz reduced blanking
      27       288p  50Hz  4x                  1280x800        reduced blanking
      28       288p  50Hz  4x H                1280x800  60Hz
      29       576p  50Hz  2x                  1280x800  75Hz
      30       576p  50Hz  2x H                1280x800  85Hz
      31       1080p 50Hz                      1280x800  120Hz reduced blanking
      32       1080p 24Hz                      1280x960  60Hz
      33       1080p 25Hz                      1280x960  85Hz
      34       1080p 30Hz                      1280x960  120Hz reduced blanking
      35       480p  60Hz  4x                  1280x1024 60Hz
      36       480p  60Hz  4x H                1280x1024 75Hz
      37       576p  50Hz  4x                  1280x1024 85Hz
      38       576p  50Hz  4x H                1280x1024 120Hz reduced blanking
      39       1080i 50Hz  reduced blanking    1360x768  60Hz
      40       1080i 100Hz                     1360x768  120Hz reduced blanking
      41       720p  100Hz                     1400x1050       reduced blanking
      42       576p  100Hz                     1400x1050 60Hz
      43       576p  100Hz H                   1400x1050 75Hz
      44       576i  100Hz                     1400x1050 85Hz
      45       576i  100Hz H                   1400x1050 120Hz reduced blanking
      46       1080i 120Hz                     1440x900        reduced blanking
      47       720p  120Hz                     1440x900  60Hz
      48       480p  120Hz                     1440x900  75Hz
      49       480p  120Hz H                   1440x900  85Hz
      50       480i  120Hz                     1440x900  120Hz reduced blanking
      51       480i  120Hz H                   1600x1200 60Hz
      52       576p  200Hz                     1600x1200 65Hz
      53       576p  200Hz H                   1600x1200 70Hz
      54       576i  200Hz                     1600x1200 75Hz
      55       576i  200Hz H                   1600x1200 85Hz
      56       480p  240Hz                     1600x1200 120Hz reduced blanking
      57       480p  240Hz H                   1680x1050       reduced blanking
      58       480i  240Hz                     1680x1050 60Hz
      59       480i  240Hz H                   1680x1050 75Hz
      60                                       1680x1050 85Hz
      61                                       1680x1050 120Hz reduced blanking
      62                                       1792x1344 60Hz
      63                                       1792x1344 75Hz
      64                                       1792x1344 120Hz reduced blanking
      65                                       1856x1392 60Hz
      66                                       1856x1392 75Hz
      67                                       1856x1392 120Hz reduced blanking
      68                                       1920x1200       reduced blanking
      69                                       1920x1200 60Hz
      70                                       1920x1200 75Hz
      71                                       1920x1200 85Hz
      72                                       1920x1200 120Hz reduced blanking
```

```
      73                                        1920x1440 60Hz
      74                                        1920x1440 75Hz
      75                                        1920x1440 120Hz reduced blanking
      76                                        2560x1600       reduced blanking
      77                                        2560x1600 60Hz
      78                                        2560x1600 75Hz
      79                                        2560x1600 85Hz
      80                                        2560x1600 120Hz reduced blanking
      81                                        1366x768  60Hz
      82                                        1080p     60Hz
      83                                        1600x900        reduced blanking
      84                                        2048x1152       reduced blanking
      85                                        720p      60Hz
      86                                        1366x768        reduced blanking
```

Descomentamos 'disable_overscan' para desactivar los bordes negros en la pantalla:

```
disable_overscan=1
```

Descomentamos 'hdmi_drive' para habilitar el sonido a través de la salida HDMI:

```
hdmi_drive=2
```

Descomentamos 'hdmi_force_hotplug' para habilitar el sonido a través de la salida HDMI:

```
hdmi_force_hotplug=1
```

Por defecto, el sistema viene configurado para establecer la pantalla en reposo si está el sistema inactivo durante 5 minutos, para desactivar esta propiedad, debemos ir a: "Sistema->Preferencias->Hardware->Gestor de energía" y establecer el tiempo a 'Nunca'.

También debemos desactivar el salvapantallas, para ello vamos a: "Sistemas->Preferencias->Visu. y Compor.->Salvapantallas" y desactivamos la casilla de "*Activar el salvapantallas cuando el eqipo este inactivo*".

7.3.3. Configuración del cliente

7.3.3.1. Comunicación directa entre servidor y cliente

Tal como hemos visto en la sección 4, el servidor se comunica con los clientes a través de usuario '*raspubli*' y el protocolo de comunicación, en ocasiones, hace uso del comando 'sudo', por tanto es necesario que el usuario '*raspubli*' pueda hacer '`sudo`' sin contraseña, para ello creamos el siguiente archivo de configuración:

```
$ sudo vim /etc/sudoers.d/raspubli
```

y añadimos la siguiente linea:

```
raspubli ALL=NOPASSWD: ALL
```

Además, la conexión `ssh` debe realizarse sin contraseña, por ello usamos la parte pública del certificado generado en la sección 5.5, y lo guardamos dentro del archivo "`/home/raspubli/.ssh/authorized_keys`":

```
$ mkdir /home/raspubli/.ssh
$ scp root@10.40.0.200:/usr/share/httpd/.ssh/id_rsa.pub /home/raspubli/.ssh/
    authorized_keys
```

Para completar la configuración, desde el servidor, como usuario '*apache*', nos conectamos al cliente aceptando el certificado y verificando el correcto funcionamiento:

```
$ ssh raspubli@10.40.0.210
```

```
The authenticity of host '10.40.0.210 (10.40.0.210)' can't be established.
ECDSA key fingerprint is 44:21:bb:fb:cc:8f:4c:84:8b:97:79:f3:76:c4:f4:18.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.40.0.210' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.38-v7+ armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Pueden actualizarse 0 paquetes.
0 actualizaciones son de seguridad.

Last login: Sun Apr  9 21:18:00 2017 from 10.40.0.200
raspubli@raspubli-client-1:~$
```

7.3.3.2. Programa de autoinicio

Nada mas iniciarse el cliente, este debe reproducir la ultima presentación configurada y reproducir los vídeos según la última programación establecida por el servidor. Los vídeos son programados a través del programa '`cron`', el cual es un servicio que se inicia automáticamente y para la presentación programamos un *script*, este *script* lo llamamos `init_presentation.sh`, lo creamos mediante:

```
$ vim /home/raspubli/init_presentation.sh
```

con el siguiente contenido:

```
#! /bin/bash
export DISPLAY=:0
xset dpms force off
sudo killall soffice.bin > /dev/null 2>&1 &
soffice --show /home/raspubli/pi_presentation.odp --norestore > /dev/null 2>&1 &
xset dpms force on
xset -dpms
```

Los pasos que realiza el '*script*' son los siguientes:

1. Establece que toda salida gráfica ejecutada por el *script* sea por el HDMI.
2. Apaga la pantalla para no mostrar nada, mientras se carga el contenido.
3. Por seguridad, eliminar cualquier proceso '`soffice.bin`'.
4. Carga la ultima presentación configurada por el servidor.
5. Enciende la pantalla.

Para que pueda ser ejecutado, habilitados los permisos correspondientes:

```
$ chmod a+x /home/raspubli/init_presentation.sh
```

Y garantizamos la ejecución al inicio de la sesión de usuario creando el siguiente archivo:

```
$ vim /home/raspubli/.config/autostart/init_presentation.sh.desktop
```

con el siguiente contenido:

```
[Desktop Entry]
Type=Application
Exec=/home/raspubli/init_presentation.sh
```

```
Hidden=false
X-MATE-Autostart-enabled=true
Name[es_ES]=Raspubli
Name=Raspubli
Comment[es_ES]=
Comment=
```

7.3.3.3. El programa de captura de pantalla

Unos de los programas mas comunes para hacer capturas de pantallas es el "`scrot`", sin embargo, cuando un usuario solicita una captura de pantalla, y el cliente se encuentra reproduciendo un vídeo, el usuario obtiene un captura de la presentación que se está reproduciendo en segundo plano.

Para hacer capturas de lo que realmente está mostrando el cliente por pantalla, usamos la utilidad "`raspi2png`". Esta utilidad la podemos descargar de la web del proyecto: `https://github.com/AndrewFromMelbourne/raspi2png`.

```
$ wget https://raw.githubusercontent.com/AndrewFromMelbourne/raspi2png/master/
    raspi2png -O /home/raspubli/raspi2png
$ chmod a+x /home/raspubli/raspi2png
```

Capítulo 8

Conclusiones

En el desarrollo de este proyecto, no hemos encontrado grandes dificultades que impiden un desarrollo, fluido, coordinado y sin altibajos. Todo lo contrario, hemos aprendido que nos encontramos en una época en la que podemos dar solución a muchos problemas que hasta hace pocos años sólo algunas empresas disponían de la capacidad de ofrecer una solución válida a un precio elevado. Hoy en día, disponemos de muchos recursos a precios muy asequibles, permitiendo que cualquier persona o equipo de trabajo pueda realizar lo que se propongan.

Con el lenguaje de programación PHP junto al patrón de programación Modelo-Vista-Controlador (Model-View-Controller) hemos obtenido un entorno de trabajo en el que eliminamos las desventajas de PHP con respecto a otros lenguajes de programación como JAVA, facilitando la programación tal como la Ingeniería del Software recomienda.

Al finalizar el trabajo, hemos concluido que el sistema depende ligeramente del sistema operativo del cliente, por lo que existen un dependencia del cliente con el sistema global. Sin embargo, esto puede verse como una línea de trabajo futura, en la que aislamos la configuración del cliente del sistema operativo del mismo.

Las líneas de trabajo futuras que se plantean a partir de este trabajo fin de grado son:

- Gestor multiusuario
- Incorporar mas formatos de reproducción

- Añadir la posibilidad de reproducir mas de una presentación
- Programación horaria de contenidos

Bibliografía

[APA] *Apache: HTTP Server Project.*
https://httpd.apache.org/.

[ARD] *Arduino - Tutorials.*
https://www.arduino.cc/en/Tutorial/HomePage.

[COD] *CodeIgniter User Guide.*
https://codeigniter.com/userguide2/.

[GEA] *Geany: A fast, light, GTK+ IDE.*
http://www.geany.org/manual/current/index.html.

[KVM] *Kernel Virtual Machine.*
https://www.linux-kvm.org/page/Main_Page.

[MAR] *MariaDB: Getting Started.*
https://mariadb.org/learn/.

[MOO] *Mootools.*
https://mootools.net/.

[PHP] *PHP: Documentation.*
http://php.net/docs.php.

[RP] *Raspberry Pi - Teach, Learn, and Make with Raspberry Pi.*
https://www.raspberrypi.org/.

[UBU] *Ubuntu MATE | For a retrospective future.*
https://ubuntu-mate.org/.

Apéndice A

Configuración de un presentación .odp

Para que una presentación se reproduzca indefinidamente de forma cíclica, debemos configurar la presentación en el momento de confección de la misma. La figura A.1 muestra la interfaz típica de *LibreOffice Impress*.

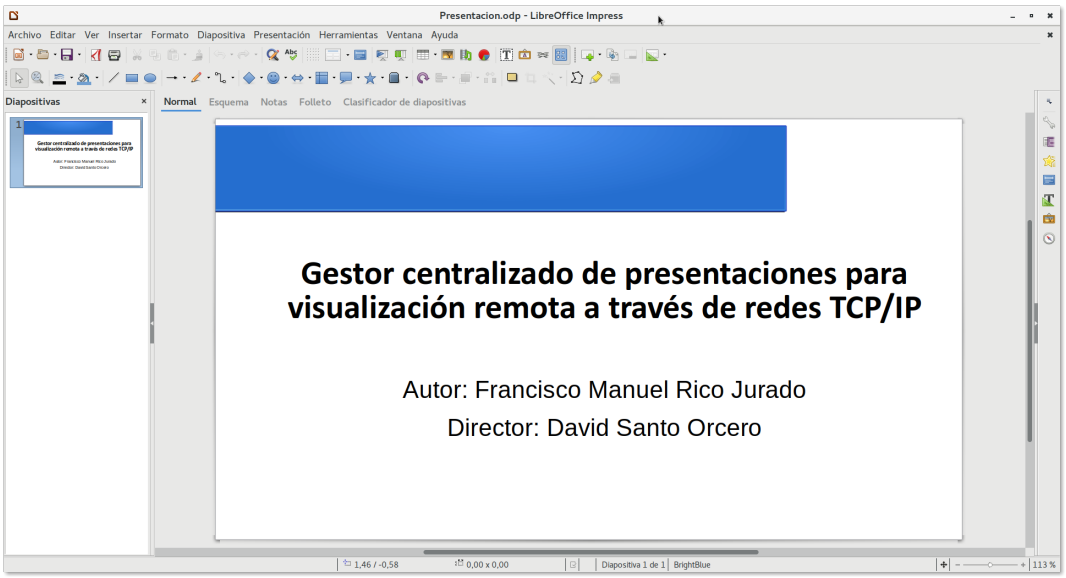


Figura A.1: Configuración de una presentación - Paso 1

Una vez abierta la presentación, hacemos clic en el menú 'Presentación' ->'Configurar presentación...'. Ver figura A.2.

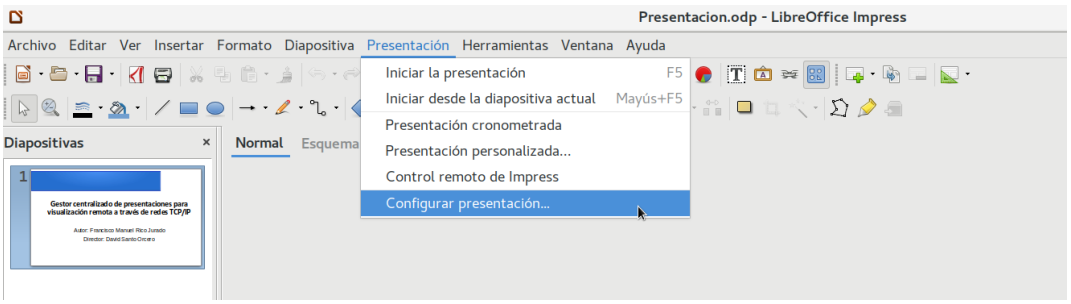


Figura A.2: Configuración de una presentación - Paso 2

En la ventana de "Configuración de la presentación" debemos selección el modo de presentación **'Bucle y repetición tras:'** y establecer el tiempo a cero. La figura A.3 muestra un ejemplo.

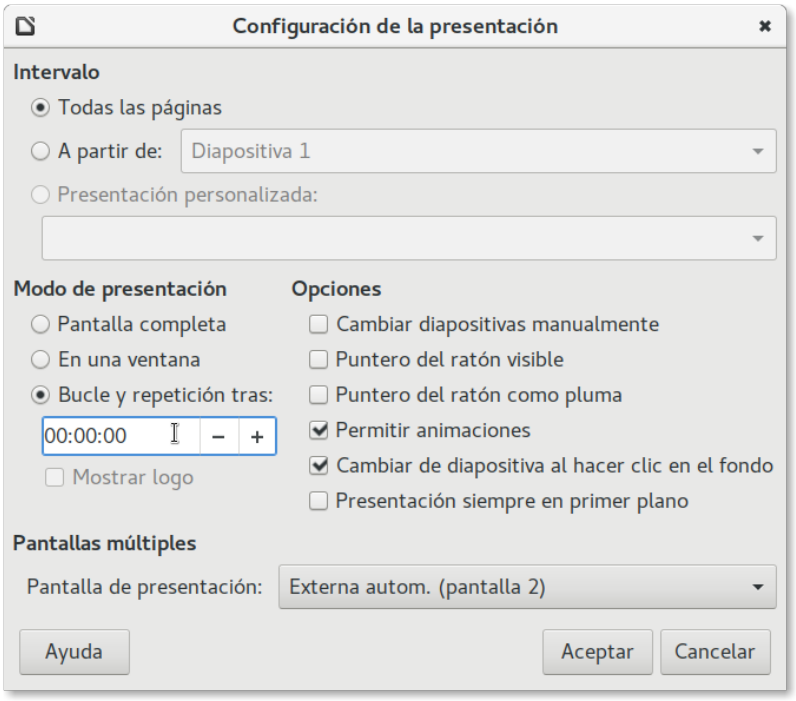


Figura A.3: Configuración de una presentación - Paso 3