UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTRÓNICA

VALPARAÍSO – CHILE

"DESARROLLO DE SISTEMA AUTOMATIZADO DE CAPTURA Y DIFUSIÓN POR INTERNET DE PRESENTACIONES EN LA UTFSM BASADO EN GALICASTER Y OPENCAST"

MAXIMILIANO LIRA DEL CANTO

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE:
"INGENIERO CIVIL TELEMÁTICO"

PROFESOR GUÍA: AGUSTÍN GONZÁLEZ VALENZUELA

PROFESOR CORREFERENTE: RUDY MALONNEK WETZEL

OCTUBRE – 2016

AGRADECIMIENTOS

Este trabajo no habría podido llegar a buen tiempo si no hubiera sido por las personas que me ayudaron en toda esta tarea.

En primer lugar, agradecer a Felipe López y Leo Espina, por haber colaborado en el proyecto que se convirtió en la base de este trabajo.

A mis queridos amigos Tarek Ceriani y Francisca Martínez, su constante preocupación por mi trabajo y ayuda fueron muy valiosos para seguir avanzando en la realización de este este escrito.

Todos los funcionarios del Departamento de Electrónica, María Ibacache, Marisa Plané, Héctor Araya, Caupolicán Acevedo, Pedro Quiroz, Sra. Luisa, Vinka Oliva. Por la ayuda y apoyo constante en las distintas etapas que duró la implementación de la plataforma.

A mi querida Familia, mis padres Octavio y Ana María, mi hermano, Francisco José por todos estos años de apoyo, en loa tiempos difíciles y en los tiempos de alegría siempre han estado allí.

Todos los que ayudaron con la corrección y pruebas del sistema: Niklas Tampier, Walter Grote, Marco Torres, Marco Zamora, Alejandro Suárez, María José Hepp, Alejandro Maass, Cristobal Ganter, Dr Rodrigo Benenson.

A mis Jefes en la dirección de servicios informáticos y comunicaciones PUCV, Alonso Rodriguez y Víctor Villavicencio, por darme las facilidades para trabajar en este escrito.

A los desarrolladores de Opencast y Galicaster, en particular a Lars Kiesow y Alfonso Rodriguez por la ayuda prestada en la resolución de problemas en la implementación del sistema.

Finalmente, a mi Profesor guía y mi profesor Correferente, Agustín Gonzalez y Rudy Malonnek por ser guías en todo este trabajo y darme consejos de cómo afrontarlo y ejecutarlo.

El éxito consiste ir de fracaso en fracaso sin perder el entusiasmo

Winston Churchill

Dedicado a mis padres, Octavio y Ana María

Por su apoyo incondicional para seguir mis sueños

DESARROLLO DE SISTEMA AUTOMATIZADO DE CAPTURA Y DIFUSIÓN POR INTERNET DE PRESENTACIONES EN LA UTFSM BASADO EN GALICASTER Y OPENCAST

Memoria para optar al título de Ingeniero Civil Telemático

Maximiliano Lira Del Canto

Profesor Guía: Agustín González Valenzuela

Octubre 2016

RESUMEN

Los sistemas de grabación de charlas y clases han sido desarrollados en diferentes partes del mundo por más de 20 años. Actualmente la Universidad Técnica Federico Santa María no cuenta con un sistema que permita una fácil grabación y publicación del material para los académicos y tampoco hay una forma de dejar registro de las charlas dictadas, especialmente en el Departamento de Electrónica de ésta casa de estudios. Esto incurre con un desaprovechamiento de la información, por la falta de tiempo de los asistentes o de cupos para estos eventos.

Este trabajo resuelve este problema diseñando e implementando de un sistema de captura de exposiciones y difusión automática bajo demanda. Este sistema permite de una forma muy sencilla que el expositor pueda hacer su charla sin preocuparse de la posterior edición y publicación por Internet de su puesta en escena.

La implementación se basó en tres componentes, un servidor de difusión, encargado de la publicación y distribución de los contenidos grabados, y en dos agentes de captura, un módulo estacionario, instalado en un auditorio, y un módulo móvil, capaz de desplegarse en cualquier lugar en unos pocos minutos.

El resultado obtenido cumple los requerimientos establecidos, logrando un sistema confiable para su uso en producción.

DEVELOPMENT OF AUTOMATIZED CAPTURE AND DIFFUSION OVER INTERNET SYSTEM OF PRESENTATIONS IN THE UTFSM BASED UPON GALICASTER AND OPENCAST

Final Project Report towards the partial fulfillment of the requirements of the Telematics Engineering Degree (6-year Program)

Maximiliano Lira Del Canto

Supervisor Professor: Agustín González Valenzuela

October 2016

ABSTRACT

The recording lecture and talks systems have been developed in many parts of the world for most 20 years. Currently the Santa Maria Technical University don't count with a system who able an easy recording and publishing of the material for the academics and there not a way to keep a record of the talks given, especially in the Electronics Department in this University. What happens is a misuse of information, lack of time attendees or quotas for these events.

This work solves this problem designing and implementing a capture and automatic diffusion system on demand. This system allows in a very simple way in the exponent can make you talk without worrying for the post edition and internet publishing of the performance.

This implementation its based upon in three components, a diffusion server, in charge of the publishing and distribution of the recorded material, and in two capture agents, one stationary module installed in an auditorium, and a mobile module, able to deploy anywhere within a few minutes.

The result meets the requirements established, achieving a reliable system for production use.

GLOSARIO

CDN	Content Delivery Network. Es una red superpuesta de computadoras que contienen copias de datos, colocados en varios puntos de una red con el fin de maximizar el ancho de banda para el acceso a los datos de clientes por la red.
CMS	Un sistema de gestión de contenidos (en inglés: Content Management System, más conocido por sus siglas **CMS**) es un programa informático que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web
Código Abierto	El **código abierto** es el software distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al **código** fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.
CSS	Hoja de estilo en cascada o **CSS** (siglas en inglés de cascading style sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).
EPEL	EPEL (paquetes adicionales para Enterprise Linux) es un esfuerzo de la comunidad de voluntarios del proyecto Fedora para crear un repositorio de paquetes de add-on de alta calidad que complementan el de Red Hat Enterprise Linux basada en Fedora (RHEL) y sus spin-offs compatibles, tales como CentOS y Scientific Linux.
GStreamer	Es un framework multimedia libre multiplataforma escrito en el lenguaje de programación C, usando la biblioteca GObject.
GTK	GTK+ o The GIMP Toolkit es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario (GUI)
HSQLDB	**HSQLDB** (**H**yperthreaded **S**tructured **Q**uery **L**anguage **D**atabase) es un sistema gestor de bases de datos libre escrito en Java.
HTML5	HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: una «clásica», HTML (text/html), conocida como HTML5, y una variante XHTML conocida como sintaxis XHTML5 que deberá servirse con sintaxis XML (application/xhtml+xml).
HTTP	Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force
IP	Una dirección IP es un número que identifica, de manera lógica y jerárquica, a una Interfaz en red (elemento de comunicación/conexión) de un dispositivo
LMS	Del inglés Learning Management System, un sistema de gestión de aprendizaje es un software instalado en un servidor web que se emplea para administrar, distribuir y controlar las actividades de formación no presencial

Metadatos	(del griego μετα, meta, 'después de, más allá de' y latín datum, 'lo que se da', «dato»), literalmente «sobre datos», son datos que describen otros datos. En general, un grupo de metadatos se refiere a un grupo de datos que describen el contenido informativo de un objeto al que se denomina recurso
NFS	El Network File System (Sistema de archivos de red), o NFS, es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local.
Nube	La computación en la nube, conocida también como servicios en la nube, informática en la nube, nube de cómputo o nube de conceptos (del inglés cloud computing), es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es Internet.
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.
Python	Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.
REST	La Transferencia de Estado Representacional (Representational State Transfer) o REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.
RSS	RSS son las siglas de Really Simple Syndication, un formato XML para sindicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos.
Screencast	Es una grabación digital de la salida por pantalla de la computadora, a veces conteniendo narración de audio.
Streaming	La retransmisión (en inglés streaming, también denominado transmisión, transmisión por secuencias, lectura en continuo, difusión en continuo o descarga continua) es la distribución digital de multimedia a través de una red de computadoras, de manera que el usuario consume el producto (generalmente archivo de vídeo o audio) en paralelo mientras que se lo descarga.
URL	Un localizador de recursos uniforme LRU (más conocido por la sigla URL, del inglés Uniform Resource Locator)
VLAN	Una VLAN, acrónimo de virtual LAN (Red de área local virtual), es un método para crear redes lógicas independientes dentro de una misma red física.
XML	XML, siglas en inglés de eXtensible Markup Language ("lenguaje de marcas Extensible"), es un meta-lenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

I CONTENIDO

Agradecimientos v
RESUMEN XI
ABSTRACT XIII
Glosario XV
I Contenido XVII
II Índice de Tablas XX
III Índice de Figuras XX
1 Introducción 1
1.1 Problemática actual 1
1.2 La efectividad de la grabación de clases 2
1.3 ¿Qué es un sistema de captura y distribución de videos? 2
1.3.1 Sistema de Captura 3
1.3.2 Sistema de difusión 4
1.4 Objetivos del trabajo 5
2 Tecnologías y casos de uso actuales 5
2.1 Sistemas de captura de video y distribución en el mundo 5
2.1.1 Webcast de la universidad de California Berkeley 5
2.1.2 Replay, universidad de Oxford. 9
2.1.3 Sistema de captura de clases de la universidad de Leeds 12
2.2 Aplicaciones disponibles para la captura audiovisual y difusión por Internet 14
2.2.1 Kaltura 14
2.2.2 Panopto 16
2.2.3 TechSmith Relay 17
2.2.4 Cisco WebEx 19
2.2.5 Opencast + Galicaster 20
2.3 En síntesis 21
3 Desarrollo de la solución en Opencast y Galicaster 22
3.1 Levantamiento de requerimientos 22
3.1.1 Requerimientos funcionales 22

3.1.2 Requerimientos no funcionales 23
3.2 Alcance del proyecto 25
3.3 Diseño 25
3.3.1 Actividades del diseño de módulos 25
3.4 Implementación 37
3.4.1 Servidor de difusión 37
3.4.2 Agente de captura estacionario 55
3.4.3 Agente de captura móvil 66
4 Pruebas del sistema 76
4.1 Descripción del ambiente de pruebas 76
4.2 Pruebas para usuario 76
4.2.1 Grabación utilizando agente de captura estático 76
4.2.2 Grabación utilizando agente de captura móvil 81
4.3 Pruebas para administrador 83
4.3.1 Ingreso de evento 84
4.3.2 Edición de evento ya ingresado 90
4.3.3 Publicación de evento a sistemas de video bajo demanda (YouTube) 97
4.4 Pruebas de desempeño 98
4.4.1 Tiempo y recursos utilizados en el procesamiento de grabación y codificación 98
5 Conclusiones y trabajo futuro 108
6 Referencias 110
7 Anexos 114
7.1 Instalación de proxy reverso utilizando Apache HTTP server 114
7.1.1 Configuración de SELinux para Apache 114
7.1.2 Configuración de Proxy reverso: 114
7.2 Flujos de trabajo de Opencast 115
7.2.1 Descripción de un flujo de trabajo 115
7.3 Configuración de Publicación en YouTube para Opencast 117
7.3.1 Configuración en Google 118

7.3.2 Configuración de YouTube en Opencast 119

7.3.3 Habilitación de YouTube en Opencast 120

7.4 Códigos de configuración de Opencast 121

7.4.1 custom.properties 121

7.4.2 org.opencastproject.organization-mh_default_org.cfg 127

7.5 Códigos de configuración de Galicaster 133

7.5.1 Archivos de configuración 133

7.5.2 Ejemplos de archivos de perfiles 138

7.6 Fichas técnicas de equipamiento utilizado 141

7.6.1 Servidor de difusión 141

7.6.2 Agente de captura estacionario 142

7.6.3 Agente de captura móvil 145

7.7 Planos de diseño 3D módulo móvil 147

II ÍNDICE DE TABLAS

Tabla 1: Comparación de cumplimiento de requerimientos 27
Tabla 2: Componentes necesarios de HW para galicaster..... 34
Tabla 3: Componentes de la Arquitectura de SW Opencast 35
Tabla 4: Características Dell PowerEdge R430 40
Tabla 5: Características Servidor Spielberg 42
Tabla 6: Características Servidor Cameron..... 42
Tabla 7: Características Servidor Storage 42
Tabla 8: Lista de componentes utilizados para la construcción del Agente de Captura Estacionario..... 58
Tabla 9: Componentes a utilizados para agente de captura móvil 67
Tabla 10: Configuración de grabación en Galicaster 98
Tabla 11: Configuración de codificación en Opencast 99
Tabla 12: Lista de operadores de flujos de trabajo..... 116

III ÍNDICE DE FIGURAS

Figura 1: Diagrama de sistema de captura 3
Figura 2: Esquema del sistema de difusión..... 4
Figura 3: Salas de clases con el sistema Webcast instalado..... 6
Figura 4: Sitio Web de Webcast..... 7
Figura 5: Implementación del sistema Webcast..... 8
Figura 6: Presentación de vídeos en plataforma Webcast. 8
Figura 7: Diagrama de Casos de uso de Replay..... 11
Figura 8: Interfaz de usuario de Oxford Replay..... 12
Figura 9: Equipos del agente de captura, universidad de Leeds..... 13
Figura 10: Sala de clases con el sistema de captura instalado..... 13
Figura 11: Kaltura CaptureSpace 14
Figura 12: Menú de administración de Kaltura CE..... 15
Figura 13: Agente de Captura Panopto 17
Figura 14: Sistema de difusión de videos Panopto..... 17

Figura 15: Software del agente de captura de TechSmith Relay 18
Figura 16: Interfaz de usuario de Cisco WebEx 19
Figura 17: Entradas y salidas del sistema de captura y difusión de presentaciones..... 28
Figura 18: Flujo de trabajo del sistema completo 28
Figura 19: Pantalla de grabación de Galicaster..... 30
Figura 20: Administrador de medios de Galicaster..... 31
Figura 21: Arquitectura de Galicaster 32
Figura 22: Diseño de Arquitectura Agente de captura..... 33
Figura 23: Composición de funcionamiento de Opencast 36
Figura 24: Diseño para la implementación de Opencast..... 37
Figura 25: Servidor Dell PowerEdge R430 instalado en dependencias del Departamento de Electrónica UTFSM 40
Figura 26: Diagrama de Red Servidor Opencast..... 41
Figura 27: Pantalla de acceso repositorio Opencast..... 44
Figura 28: Sitio de bienvenida al panel de administración de opencast..... 54
Figura 29: Presentación de módulo de medios (Presentación) de Opencast..... 55
Figura 30: Diagrama de arquitectura de Hardware de Galicaster 56
Figura 31: Cámara Logitech C920..... 59
Figura 32: Tarjeta Capturadora Datapath Vision RGB 1ES 59
Figura 33: Monitor Asus VT207N..... 60
Figura 34: Agente de Captura en auditorio Feick 61
Figura 35: Cámara instalada en el auditorio Feick..... 62
Figura 36: Blackmagic Design Mini Recorder 68
Figura 37: Primer diseño del agente de captura móvil..... 69
Figura 38: Diseño versión 2 módulo de captura móvil 70
Figura 39: Vista frontal y lateral módulo de captura móvil V2 70
Figura 40: Vista superior Módulo de captura V2..... 71
Figura 41: Agente de Captura, vista exterior, sin pintar 72
Figura 42: Agente de Captura, Vista Interior, Sin pintar 73
Figura 43: Proceso de barnizado Agente de Captura 73
Figura 44: De izquierda a derecha: Bandeja de audio, bandeja de PC, ordenadores de cable, cámara de grabación. 74
Figura 45: Cámara usando trípode. Distribución interior del módulo 75
Figura 46: Módulo de captura armado..... 75
Figura 47: Conexión del PC presentador al agente de captura 77

Figura 48: Encendiendo proyector de la sala 77
Figura 49: Izq: Encendido de equipos de audio; Der: Instalación de micrófono al presentador 78
Figura 50: Inicio de grabación en Galicaster..... 78
Figura 51: Galicaster en proceso de grabación 79
Figura 52: Acción de pausar en Galicaster..... 79
Figura 53: Confirmación de detención de grabación en Galicaster 80
Figura 54: Preparación de Galicaster para una nueva grabación 80
Figura 55: Apagado de sistemas de audio y microfonía 81
Figura 56: Agente de Captura Móvil en funcionamiento..... 82
Figura 57: Cámara grabando..... 83
Figura 58: Panel de admintración de Opencast..... 84
Figura 59: Ventana de metadatos al crear un evento..... 85
Figura 60: Ventana de origen al crear un evento “Subir Archivos” seleccionado 86
Figura 61: Ventana de origen al crear un evento “¨Programar un único evento” seleccionado 87
Figura 62: Ventana de origen al crear un evento “¨Programar múltiples eventos” seleccionado 88
Figura 63: Ventana de procesado al crear un evento..... 88
Figura 64: Ventana de política de acceso al crear un evento 89
Figura 65: Ventana de resumen al crear un evento 90
Figura 66: Modo de revisión y edición en Opencast..... 91
Figura 67: Panel de edición de grabaciones en Opencast..... 91
Figura 68: Selección de tareas para grabaciones..... 92
Figura 69: Confirmación de grabaciones a modificar..... 93
Figura 70: Ventana de comenzar tarea, selección de tarea..... 93
Figura 71: Lista de tareas que se pueden realizar..... 94
Figura 72: Ventana de comenzar tarea, opciones de tarea "Publish" 94
Figura 73: Ventana de comenzar tarea, opciones de tarea "Republish Metadata" 95
Figura 74: Ventana de comenzar tarea, opciones de tarea "Retract"..... 95
Figura 75: Ventana de comenzar tarea, opciones de tarea "Update Previews" 96
Figura 76: Ventana de comenzar tarea, resumen de acciones a realizar 96
Figura 77: Elección de flujo de trabajo para publicación en YouTube..... 97
Figura 78: Grabación publicada automáticamente utilizando YouTube..... 98
Figura 79: Equipo Galicaster en espera..... 100
Figura 80: Uso de recursos de Galicaster mientras graba 101

Figura 81; Uso de recursos al finalizar la grabación 102
Figura 82: Gráfico de tiempo de procesado vs duración de grabación 104
Figura 83: Porcentaje de uso de cpu grabación 1 105
Figura 84: Porcentaje de uso de CPU grabación 2 105
Figura 85: Porcentaje de uso de RAM en procesamiento de grabaciones 106

1 INTRODUCCIÓN

Prácticamente desde la invención de la grabación de video, se han capturado variadas exposiciones, charlas y un sin número de eventos. Sin embargo, desde hace más de una década, capturar el audio y video de una exposición y luego habilitar esas grabaciones al público se ha vuelto una rutina. Estas grabaciones publicadas varían desde producciones altamente sofisticadas a simples grabaciones de audio que acompañan a un archivo separado de diapositivas. La cantidad de formas de publicar y distribuir estas grabaciones también es muy amplia desde el envío a través de medios análogos (grabaciones en VHS, por ejemplo) hasta el uso de sistemas web que son capaces de proveer estos recursos bajo demanda.

1.1 PROBLEMÁTICA ACTUAL

Hoy en día, cuando la educación tiene un rol primordial en la sociedad, cuando cada vez se realizan y llevan a cabo más conferencias, charlas y presentaciones de diversas investigaciones científicas, es inconcebible que, con la tecnología disponible, una persona que desee asistir al evento se pierda la actividad, y toda la discusión y debate desarrollado en ella, por problemas de distanciamiento geográfico y/u otros compromisos contraídos para ese mismo horario.

Actualmente, existen diversas soluciones para poder grabar y transmitir contenidos en la red, sin embargo, estas soluciones carecen de un sistema unificado y de fácil uso, tanto para personas que se desarrollan en el área de las comunicaciones, como usuarios comunes y corrientes, sin experiencia en este tipo de sistemas. En casi la totalidad de las soluciones disponibles en el mercado, se requieren personas especialmente dedicadas para el proceso de grabación, edición y publicación, lo cual implica un alto costo en recurso humano para el desarrollo de estas tareas.

Los sistemas de captura de video y presentaciones automatizados han sido estudiados y desarrollados en numerosas universidades en el mundo. Una de las problemáticas actuales que buscan solucionar con este tipo de propuestas es el creciente costo por la mantención y construcción de infraestructura necesaria para la creciente cantidad de alumnos que asisten a las aulas de clases.

1.2 LA EFECTIVIDAD DE LA GRABACIÓN DE CLASES

En el caso particular de la grabación en las salas de clases, se han hecho varios estudios para determinar si la implementación de este recurso produce impacto entre los estudiantes, ya sea en la asistencia, el rendimiento académico o en la moral de estos.

Las investigaciones que se han llevado a cabo, han determinado que el formato que más ha favorecido a los estudiantes es el que presenta el audio del instructor en sincronía con las diapositivas [1] [2] [3], pero también el tener la grabación del instructor cuando este explica temas más complejos. En otras investigaciones, los estudiantes se ven menos estresados y ansiosos cuando ellos pueden tomar notas detalladas de las clases atendidas más tarde a su propio ritmo o lograr ponerse al día con las clases perdidas [4]. En una encuesta realizada a 7200 estudiantes de la universidad de Wisconsin-Madison [5], el 82% de los estudiantes indicaron su fuerte preferencia por las clases grabadas, publicadas en Internet para complementar sus estudios.

En cuanto al impacto académico, los estudios aún no son concluyentes; algunos estudios sugieren que la captura de clases ayuda a los estudiantes a obtener mejores calificaciones [5] [6] y también hay evidencia que los estudiantes que utilizan las grabaciones rinden de manera similar o mejor y se involucran más en las actividades en clases, comparados con los estudiantes que solo atienden a clases más tradicionales [7] [8]. Otros estudios han mostrado que no existe impacto significativo desde el uso de estos sistemas en el rendimiento en calificaciones y exámenes de los estudiantes [4]. Una investigación que contempló un total de 439 estudiantes, evidenció que la grabación de clases no ayuda significativamente a los mejores alumnos, sin embargo, los alumnos que mayormente se benefician de este sistema son los alumnos promedio [9].

1.3 ¿QUÉ ES UN SISTEMA DE CAPTURA Y DISTRIBUCIÓN DE VIDEOS?

A continuación, se describe cómo es un sistema de captura y distribución de vídeos y los elementos que lo componen.

1.3.1 Sistema de Captura

El Sistema de Captura (Figura 1) permite controlar una grabación audiovisual – ya sea de manera manual o automática – y almacenar y/o transmitir esta información a otros dispositivos. Se compone de dos elementos, un elemento de hardware y un elemento de software.

El hardware consiste en una cámara de video, un micrófono y un dispositivo para almacenar las grabaciones y coordinar la captura con los otros elementos antes descritos. Adicionalmente, puede haber otros dispositivos para capturar la presentación directamente del expositor, micrófonos adicionales para el público, entre otros.

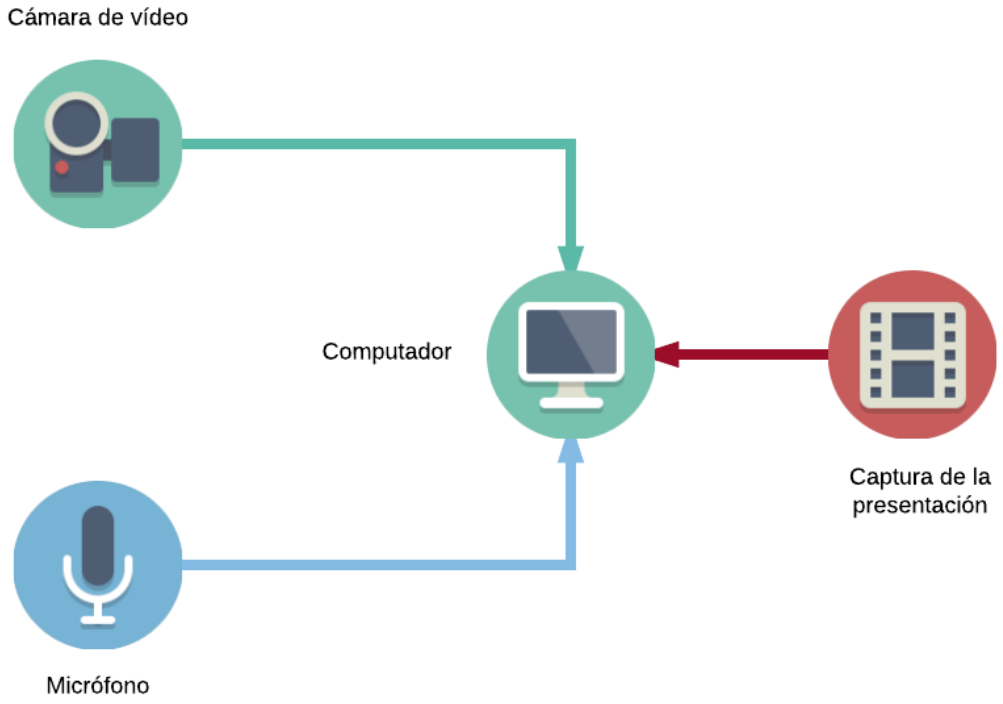


Figura 1: Diagrama de sistema de captura

En cuanto al software, se compone de un programa que sirve para comenzar la grabación, detenerla, administrar las grabaciones anteriores de manera local, y puede transmitir y/o exportar las grabaciones para su tratamiento posterior.

El sistema más básico existente para hacer esta tarea son las videograbadoras caseras y con los adelantos de tecnologías actuales, esto se puede hacer también con los teléfonos celulares.

Los sistemas de captura que funcionan bajo una plataforma de grabación y distribución de contenido multimedia son llamados **“Agentes de Captura”**.

1.3.2 Sistema de difusión

Un sistema de difusión de videos, es un sistema que permite el acceso de las grabaciones obtenidas previamente por el sistema de captura. Este sistema se compone por un servidor para el almacenamiento de las grabaciones y su procesamiento, y el software encargado de hacer estas tareas.

La distribución de las grabaciones ya procesadas por el sistema se hace generalmente a través de internet y/o la red de área local de la institución.

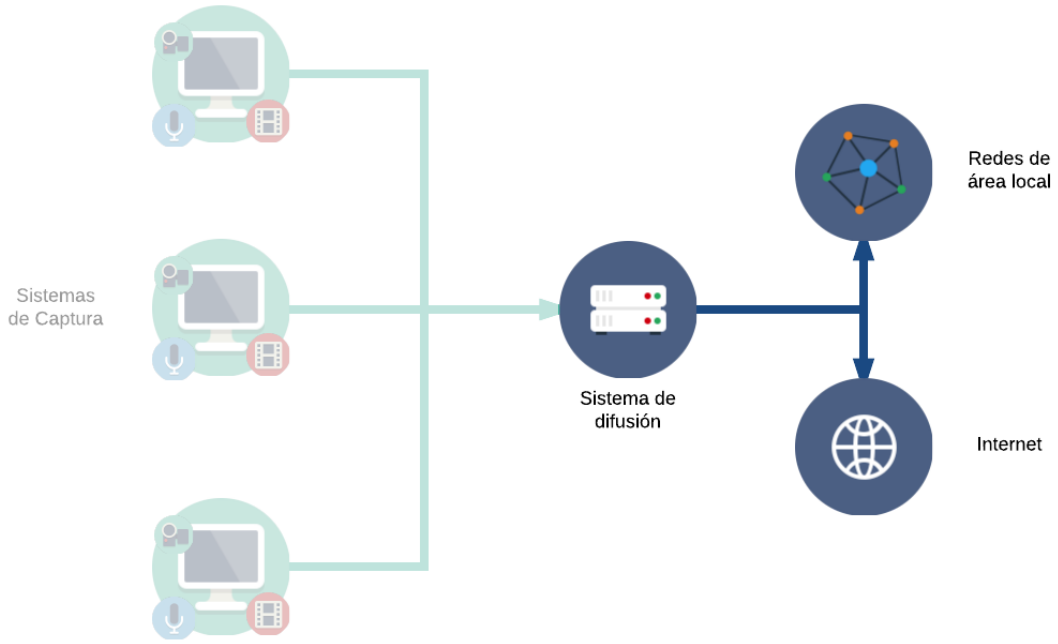


Figura 2: Esquema del sistema de difusión

Al sistema de difusión normalmente está implementado dentro de un servidor o en un servicio en la nube. Por ello también se le conoce como **“Servidor de Difusión”**.

1.4 OBJETIVOS DEL TRABAJO

Este trabajo tiene la finalidad de diseñar e implementar un sistema automatizado de grabación y difusión de presentaciones utilizando internet, manejando recursos de código abierto, en particular, las soluciones llamadas Galicaster [10] y Opencast [11], las cuales son una plataforma de distribución y publicación de contenido audiovisual, y un software de captura respectivamente, junto con el diseño e implementación de esta plataforma.

En este trabajo se desarrollan los siguientes temas: Reportar el estado de arte de sistemas de este tipo, implementar un módulo de captura de una salida a monitor, el video y audio del presentador usando software Galicaster; desarrollar un equipo (módulo) de captura portátil (móvil), configurar una máquina virtual, implementar un servidor de almacenamiento y difusión de medios, configurar un portal Opencast para el acceso del contenido grabado, reportar el rendimiento del sistema en escenarios de prueba, evaluar el uso de sistemas de video bajo demanda tipo YouTube para almacenamiento y difusión del contenido.

2 TECNOLOGÍAS Y CASOS DE USO ACTUALES

2.1 SISTEMAS DE CAPTURA DE VIDEO Y DISTRIBUCIÓN EN EL MUNDO

En distintas partes del mundo se han desarrollado sistemas para solucionar la problemática planteada en la sección anterior, algunos de estos casos son Webcast, de la Universidad de Berkeley:

2.1.1 Webcast de la universidad de California Berkeley

UC Berkeley ha ofrecido el servicio de captura de clases desde 1995 en sus campus, a través de un modelo de publicación de grabaciones, proveyendo al mundo con una ventana de la experiencia dentro de la sala de clases de esta universidad. Este programa ha tenido millones de visitas entre las más de 16.000 horas de contenido disponible a través del programa Webcast [12].

Las grabaciones son puestas a disposición de los alumnos a través de los canales de distribución de YouTube y iTunes U. La plataforma agrega las grabaciones de estos canales y los presenta de una manera ordenada para buscar por curso. Se tiene la posibilidad de hacer los cursos accesibles a todo público o "Sólo estudiantes".

Berkeley cuenta con 49 salas habilitadas para la grabación de clases, como las que se encuentran en la Figura 3, 46 de estas salas solo ofrecen grabar audio o grabar audio más la grabación de las diapositivas. Las otras 3 salas son auditorios que ofrecen grabación de las diapositivas, el instructor y audio. Se contratan estudiantes para manejar las cámaras de video en caso de que se elija la opción pertinente.

2.1.1.1 Composición del sistema

Se captura el audio y video, se envía a cinco computadores "puertas de enlace de video" que se encuentran en la sala de control maestro. Cada computador tiene tarjetas de audio y video que digitalizan las clases programadas basado en el horario agendado en la base de datos central [13].

Figura 3: Salas de clases con el sistema Webcast instalado

Las puertas de enlace de video reenvían la grabación al servidor de streaming para transmisión en vivo. Cuando la grabación finaliza, una copia de la clase es transferida al servidor de streaming para ser visto bajo demanda. Tanto la versión grabada como la de versión en vivo

aparecen en páginas web dinámicamente generadas que enumeran todos los cursos disponibles (Figura 5).

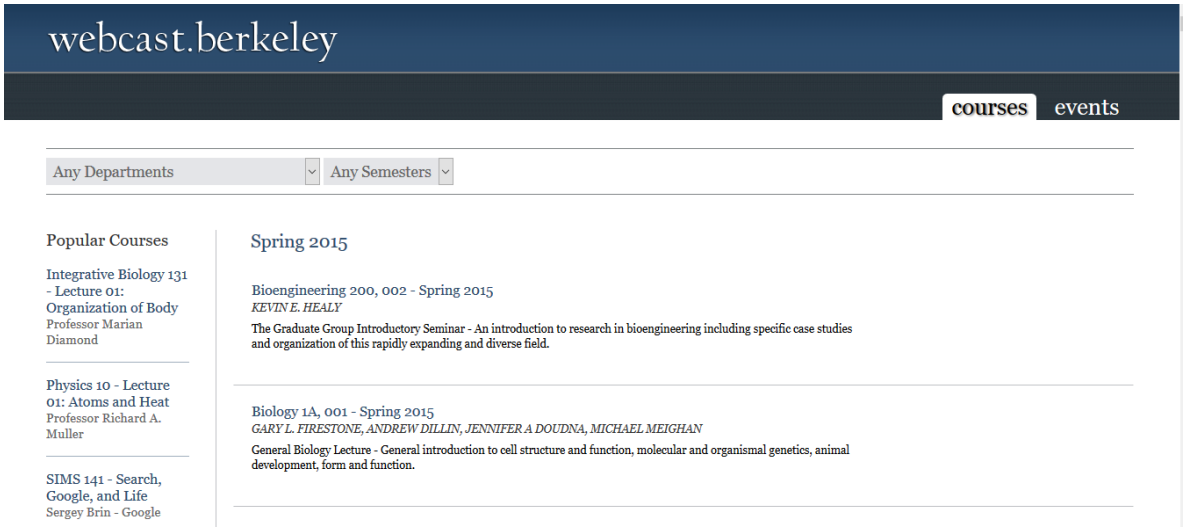


Figura 4: Sitio Web de Webcast

2.1.1.2 Implementación del sistema:

La primera versión del sistema se implementó en un servidor Sun E450 con 4 CPUs y 4 GB de RAM, el sistema operativo utilizado fue Solaris 8 , el programa Real Server 8 para proveer los medios en streaming y un servidor web Apache. La base de datos que contiene la información sobre los cursos en el Webcast es Postgres y el código del programa está escrito en PHP [13].

Una representación gráfica del sistema es la que se puede ver en la Figura 5, en donde se distinguen calramente las relaciones entre los clientes, el servidor y sus servicios y la puerta de enlace de vídeo.

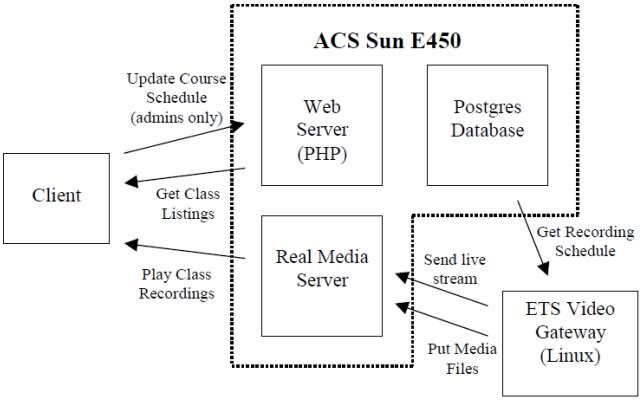


Figura 5: Implementación del sistema Webcast

Este sistema tenía la capacidad de crear páginas dinámicas en función de los vídeos y grabaciones sonoras que se iban añadiendo a la plataforma (Figura 6).

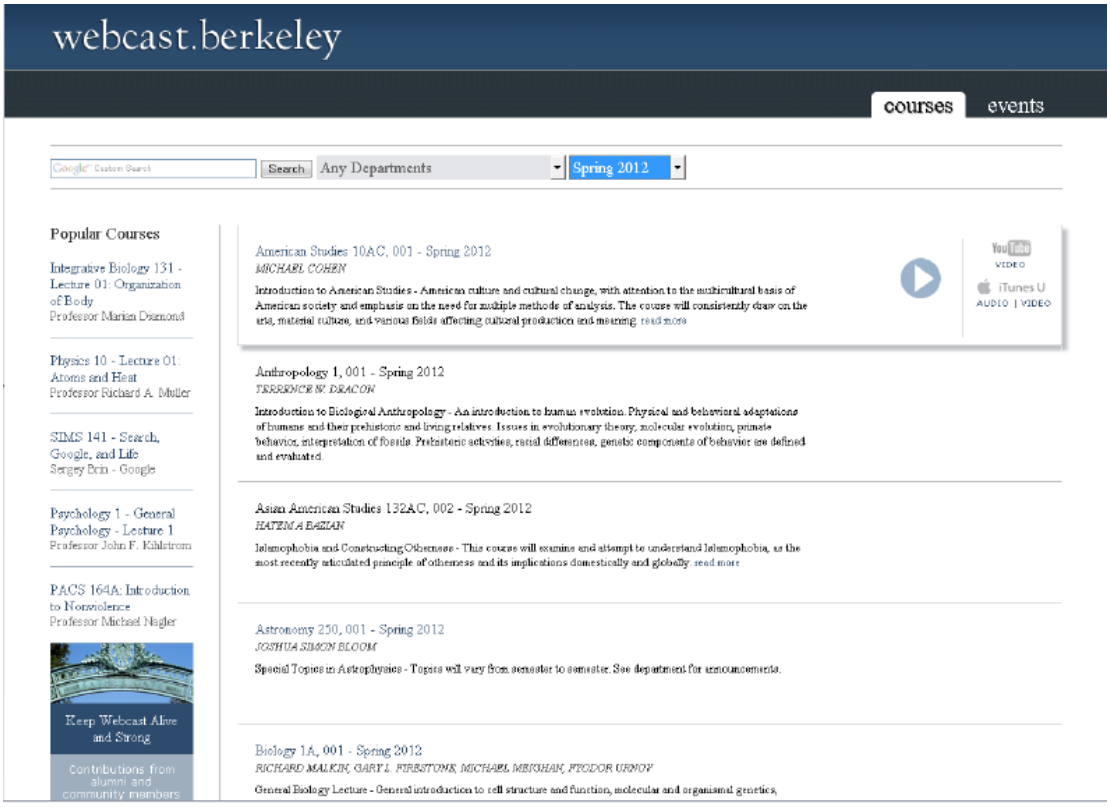


Figura 6: Presentación de vídeos en plataforma Webcast.

2.1.1.3 Uso Actual:

Webcast ha sido progresivamente actualizado con las herramientas que se implementaron en este trabajo de título, siendo relegado el material producido en los años anteriores a estar solo disponible en YouTube y iTunes U.

El servidor de distribución Sun, fue reemplazado por 6 servidores con funciones específicas:

- 1 Servidor de administración.
- 2 Servidores de Trabajo (para la transcodificación de los videos).
- 1 Servidor de llegada (hace de puente con los agentes de captura).
- 2 Servidores de Streaming.

En cuanto a las puertas de enlace de video, estas fueron reemplazadas por computadores más modernos corriendo Ubuntu 12.04 LTS. Para la transmisión en vivo se utiliza un servidor de streaming de la empresa Wowza[1].

En cuanto al sistema de publicación, se utiliza el sistema de administración de aprendizaje (Canvas por Instructure), para publicar las nuevas grabaciones.

2.1.2 Replay, universidad de Oxford.

Replay es el sistema de captura de lecciones de la Universidad de Oxford. Este servicio empezó como prueba piloto en 2014 y parte en 2015 [14].

El sistema está automatizado, lo que permite que un departamento o expositor pueda iniciar o administrar las grabaciones sin la necesidad de personal encargado del sistema que esté físicamente presente durante la lección.

Desde su puesta en marcha en enero de 2014, hasta agosto de 2016, se han creado más de 7837 horas de grabación y sobre 100.000 vistas individuales por estudiantes desde 15 departamentos diferentes [14].

[1] https://www.wowza.com/

2.1.2.1 Composición del sistema

Trabaja de la siguiente forma:

- Un expositor puede grabar automáticamente su lección en su computador o su laptop personal.
- Las grabaciones son subidas automáticamente y transferidas a un servidor inmediatamente después de la lección.
- Las grabaciones pueden ser también revisadas para hacer algunos ajustes pequeños antes de que sean lanzadas a los estudiantes.

El software utilizado por la plataforma es una solución comercial llamada Panopto, la cual se habla en detalle en la el punto 2.2.2.

Las grabaciones son lanzadas a través de la plataforma de enseñanza propia de la universidad de Oxford (WebLearn).

En cuanto al sistema de servidor, no es hospedado por la propia universidad, si no que utiliza un sistema en la Nube para este propósito. El diagrama de casos de uso de la solución se puede ver en la Figura 7.

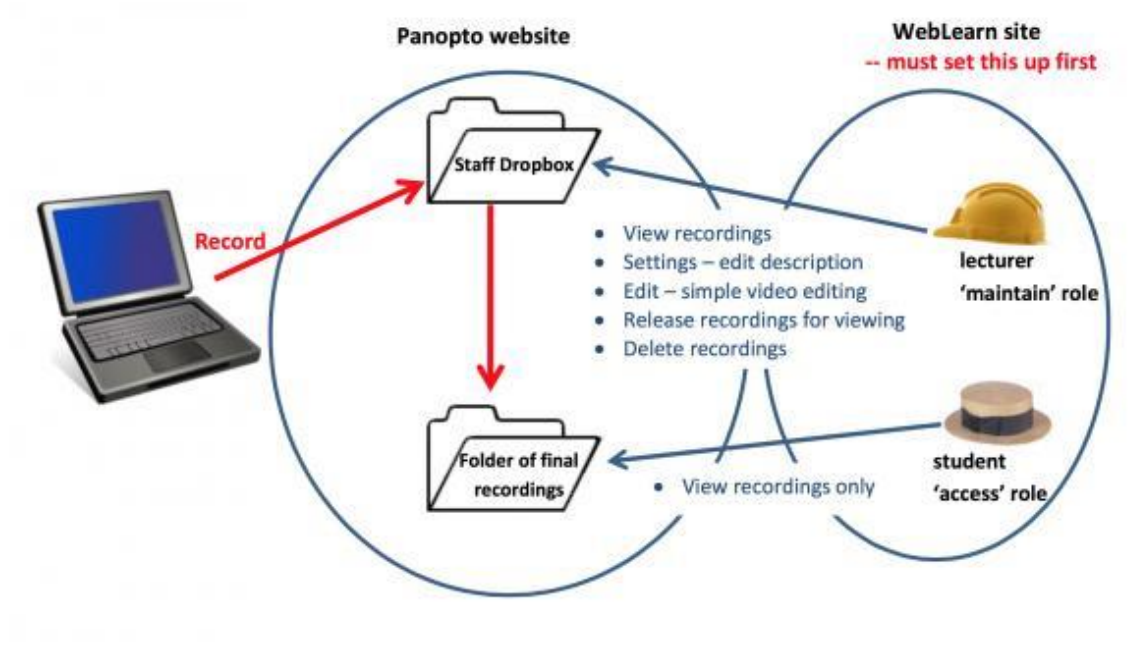


Figura 7: Diagrama de Casos de uso de Replay

El usuario al ingresar a una clase ve lo siguiente:

- Audio + diapositivas
- Video (opcional)
- El espectador puede saltar a cualquier diapositiva, con su audio correspondiente.
- Tanto los metadatos de la diapositiva como del audio se pueden buscar.
- Las grabaciones pueden ser descargadas como archivos MP3 o MP4 para su uso offline.

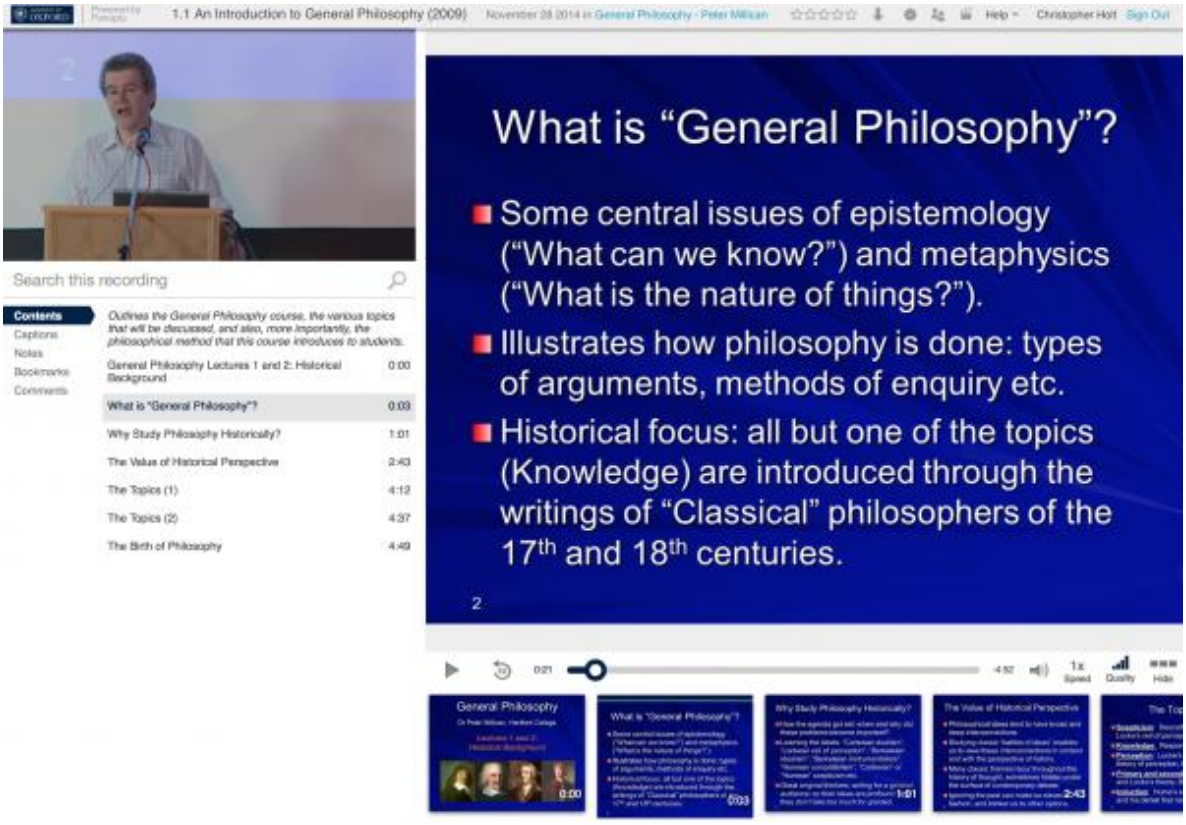


Figura 8: Interfaz de usuario de Oxford Replay

2.1.3 Sistema de captura de clases de la universidad de Leeds

La Universidad de Leeds, en 2014 lanzó un ambicioso proyecto de una magnitud de £1.1 millones para actualizar sus instalaciones y permitir la instalación de un sistema de captura de lecciones (el despliegue más grande hecho en Europa). Se hizo la instalación en 201 salas de clases.

El despliegue comenzó en marzo de 2014, terminando en julio del mismo año, para luego comenzar a utilizar el servicio en septiembre.

2.1.3.1 Composición del sistema

El sistema se desplegó en las distintas salas. Se crearon grupos de salas a las cuales se les clasificó por tipo de equipamiento que se instaló, la clasificación fue de la siguiente manera:

- Bronce: Captura de la presentación y Audio.

- Plata: Captura de la presentación, audio y vídeo desde una cámara fija.
- Oro: Un sistema completo de rastreo por vídeo para el presentador, presentación y audio. Además de una botonera para los pies para que el expositor pudiera crear acercamientos a la pizarra.

Para el agente de captura, se utilizó el Appliance Mediasite junto con equipos de captura de audio y de video Vaddio Audio Presenter. (Figura 9)

Figura 9: Equipos del agente de captura, universidad de Leeds

2.1.3.2 Uso Actual:

En la Universidad de Leeds ya se han grabado más de 16.500 lecciones, donde cada una ha tenido más de 230.000 visitas individuales.

Figura 10: Sala de clases con el sistema de captura instalado.

2.2 APLICACIONES DISPONIBLES PARA LA CAPTURA AUDIOVISUAL Y DIFUSIÓN POR INTERNET

2.2.1 Kaltura

La empresa Kaltura, es una empresa de software fundada el 2006, con la misión de potenciar cualquier experiencia de video. Entre sus productos tiene un sistema de captura de presentaciones, tanto una versión comercial (Kaltura CaptureSpace) como una versión de gratuita (Kaltura Community Edition).

2.2.1.1 Kaltura CaptureSpace

Esta es la solución comercial, combina grabación de forma fácil, publicación automatizada y vistas interactivas para generar una fácil captura de video en clases, en la oficina, en casa o en otros ambientes.

Figura 11: Kaltura CaptureSpace

Los creadores de contenido pueden capturar grabaciones en multi-streaming de hasta 3 entradas de video, sincronizadas con una presentación de PowerPoint o una captura de pantalla [15].

Los espectadores pueden navegar al contenido deseado, buscar a través de las diapositivas, cambiar entre streams de video, y verlo en modo “picture in picture”. Adicionalmente permite que las grabaciones sean vistas en cualquier dispositivo en cualquier momento.

A diferencia de los otros sistemas, en donde se utiliza un equipo separado que captura la pantalla del computador del presentador, este programa trabaja directamente desde el equipo del presentador capturando la pantalla de su equipo al mismo tiempo que las cámaras que tenga conectadas.

En cuanto al servidor de difusión, puede ser administrado en la nube por el personal de la empresa o puede ser un servidor propio, en el cual se puede instalar un portal de videos o utilizar el sistema de administración de aprendizaje para publicar el material multimedia generado.

2.2.1.2 Kaltura Community Edition:

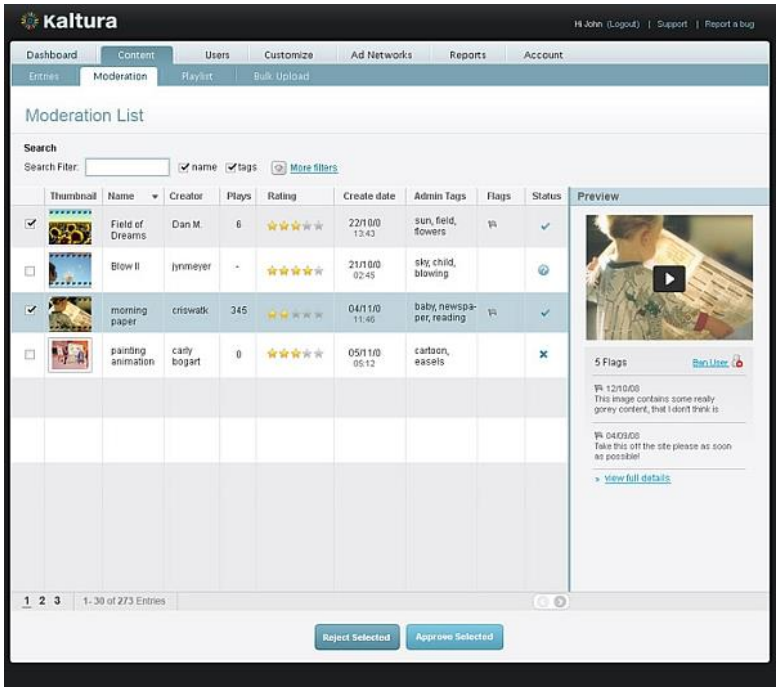


Figura 12: Menú de administración de Kaltura CE

También conocida como Kaltura CE, es la plataforma gratuita de código abierto de la compañía. Esta edición se desarrolla junto con los esfuerzos de la comunidad y la misma empresa.

Las capacidades que tiene la edición comunitaria de Kaltura son [16]:

- Capacidad de construir su propio servicio de video.
- Disfrutar de tecnología libre, de código abierto.
- Acceder a la Suite de API Kaltura, para habilitar el desarrollo de diferentes soluciones de video, integraciones y extensiones.
- Desplegar e integrar la plataforma de video con infraestructura propia o un servicio en la nube.
- Integración con widgets de código abierto, aplicaciones y extensiones para CMS y LMS.
- Tener un control total del contenido y su seguridad.
- Posibilidad de integrar CDN a elección.

2.2.2 Panopto

Este software comercial de la empresa del mismo nombre, provee servicios de grabación de lecciones, Screencasting, video streaming y software de administración de videos.

El software de captura y difusión de presentaciones fue desarrollado en la Universidad de Carniege Mellon y actualmente lo utilizan más de 600 instituciones a lo largo del mundo [17].

Entre sus características tiene un software de agente de captura (Figura 13), que permite capacidades de grabación remota, agendar grabaciones y soporte para multi-cámaras, lo que permite una fácil captura del expositor, el contenido de su pantalla y desde otros dispositivos.

Figura 13: Agente de Captura Panopto

El sistema de distribución de videos (Figura 14), permite subir, administrar y compartir todos los archivos de audio y video. Incluye análisis de los videos, editor de video web y codificación automática que asegura que los archivos multimedia se puedan reproducir en cualquier dispositivo. También permite conexión con sistemas de administración de enseñanza.

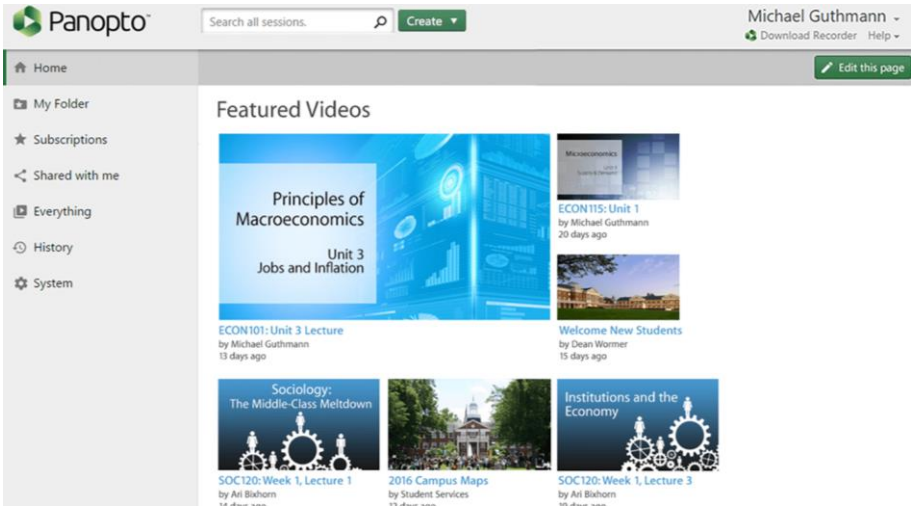


Figura 14: Sistema de difusión de videos Panopto

2.2.3 TechSmith Relay

La empresa TechSmitch, han incursionado en el software de captura y difusión de presentaciones. TechSmith Relay es una solución comercial que ofrece tanto el software para el agente de captura, como la plataforma del sistema de distribución de los videos.

Entre las características que ofrece son [18]:

- Integración con sistemas de administración de enseñanza (Moodle, por ejemplo).
- Cuestionarios en los videos.
- Generación de subtítulos automática.
- Análisis de uso de la plataforma.
- Servidor de difusión que puede ser en la nube de TechSmith o puede ser auto contenido en un servidor de la organización que lo vaya a utilizar.

El agente de captura (Figura 15) se puede utilizar tanto para grabar las diapositivas como también para grabar al instructor. No posee la característica de agendar grabaciones con el fin de que empiece a grabar en cuanto inicie la presentación de manera automática.

Figura 15: Software del agente de captura de TechSmith Relay

2.2.4 Cisco WebEx

Cisco WebEx es una plataforma de comunicación a distancia comprada por la empresa Cisco. Esta plataforma se inició como un sistema de videoconferencias corporativo que luego se fue expandiendo a otras áreas de la transmisión de Video a distancia.

WebEx permite que se puedan hacer grabaciones de los expositores y difundir este contenido generado a través de los sistemas en la nube de Cisco.

Figura 16: Interfaz de usuario de Cisco WebEx

Algunas de las características que tiene esta plataforma son [19]:

- Supervisión de atención individual y grupal.
- Pruebas y cuestionarios integrados en el sistema, como se ve en la Figura 16.
- Grabación de exposiciones y difusión de estas.
- Video en Alta definición.

La solución entregada no permite hacer grabaciones automatizadas ni tener hospedado el servicio en sistemas propios.

2.2.5 Opencast + Galicaster

Esta solución se compone de dos componentes, OpenCast, que es el módulo que se encarga como servidor de difusión de contenidos y Galicaster, encargado de la captura en audio y video de las exposiciones.

Opencast se define como: "Un sistema de distribución y captura de video flexible y personalizable para instituciones modernas" [11]. Es un proyecto de código abierto, en el que cualquiera puede participar y aportar a él. Hay debe acotar que esta definición oficial está ya desactualizada, Opencast desde su versión 2.0 dejó de lado su agente de captura para centrar recursos en la plataforma de difusión de vídeos [20].

Características destacadas:

- Calendarizar eventos automáticamente en función de una tabla de tiempo pre definida y capturar tanto el video del presentador como la pantalla del PC.
- Editar grabaciones por lotes y hacer cortes a estas.
- Procesar los videos, codificándolos y generando metadatos e imágenes de previsualización.
- Publicar y distribuir los videos para descarga o para acceso por sistemas de video bajo demanda como YouTube, RSS, Atom-Feeds.
- Administrar el contenido, configurar el status de los videos y los canales de contenido y distribución.

Galicaster es un software que tiene por objetivo ser una solución abierta, flexible y una solución al nivel del estado del arte para grabar contenidos multimedia educacionales [21].

Aunque Opencast puede utilizar distintos agentes de captura, Galicaster ofrece compatibilidad nativa con Opencast. Entre las características más importantes se encuentran:

- Puede registrarse como agente de captura en el servidor.
- Permite recibir órdenes de grabaciones calendarizadas.
- Envía automáticamente las grabaciones en el formato necesario al servidor.

En el capítulo siguiente se ahondará más en la solución descrita por estos dos componentes.

2.3 EN SÍNTESIS

En este capítulo se han visto tanto casos de estudio como software que permite hacer la implementación y dar solución a la problemática actual. De los casos de uso vistos, sólo uno utilizó tecnología de código abierto para implementar su plataforma a gran escala, mientras que en los otros dos casos (Universidades de Leeds y de Oxford) utilizaron soluciones comerciales.

En cuanto a la automatización del sistema, sólo la universidad de Berkeley y la universidad de Oxford contaban que su solución permitía hacer grabación automatizada de las lecciones y sin intervención mayor de un operador dentro de la sala de clases.

Del software estudiado, todos excepto dos tenían una versión libre de código abierto, en el caso de Kaltura, investigaciones posteriores dieron a conocer que ha sido prácticamente abandonada por los desarrolladores para poner énfasis a la versión comercial del producto [22]. Para satisfacer las necesidades de captura de lecciones para la Universidad Técnica Federico Santa María, se desarrolló una solución utilizando Opencast y Galicaster como se explica en el capítulo posterior.

3 DESARROLLO DE LA SOLUCIÓN EN OPENCAST Y GALICASTER

3.1 LEVANTAMIENTO DE REQUERIMIENTOS

En esta primera etapa es necesario establecer los límites del proyecto, para conseguirlo y no importando el proceso de desarrollo de la metodología que se utilice, es necesario realizar un levantamiento de requerimientos. Ya que cada requerimiento es una característica que el sistema debe tener.

El proceso de levantamiento de requerimientos implica un trabajo de depuración, en el sentido de dejar libre de ambigüedad cada una de las definiciones que se realicen. Además, es conveniente en separarlos en funcionales y no funcionales, ya que el modelado del sistema resultará de la revisión exhaustiva de los requerimientos funcionales.

3.1.1 Requerimientos funcionales

Como se ha dicho anteriormente, el objetivo principal del proyecto es implementar un sistema automatizado de captura y difusión por internet de presentaciones. El acceso al contenido grabado se hace a través de un portal Web. Para realizar la grabación el expositor contará con un agente de captura que se define a lo largo de este trabajo.

Este sistema se desarrolla para que sea desplegado dentro del Departamento de Electrónica de la Universidad, se considera que los alumnos, profesores, exalumnos y funcionarios de dicha institución serán los usuarios del sistema en primera instancia además del público en general.

A continuación, se enumeran el conjunto de requerimientos.

OBJ-01 El sistema debe grabar audio y vídeo

Req-01 El sistema debe grabar el audio y vídeo de la puesta en escena del expositor

Req-02 Debe grabar la pantalla del proyector de forma que esté conectado directamente a la salida de vídeo del dispositivo que esté utilizando el presentador.

Obj-02 El contenido grabado debe enviarse automáticamente al servidor de difusión

Req-03 El sistema debe dar la posibilidad de enviar la grabación inmediatamente después de que esta termina y también de enviarla al servidor a una hora programada por el administrador.

Obj-03 Debe poder visualizarse el contenido grabado a través de un portal web que contenga todas las grabaciones hechas.

Req-04 Los contenidos mostrados deben poder reproducirse por parte de los usuarios

Req-05 El portal debe permitir que el administrador se pueda autenticar para visualizar contenido no disponible para el público

Obj-04 Las grabaciones tienen que tener la posibilidad de iniciarse automáticamente agendando previamente el evento

Req-06 En la consola de administrador del servidor, tiene que tener la opción de agregar los metadatos del evento a grabar (Expositor, tema, categoría), para permitir una indexación automática posteriormente.

Obj-05 El Servidor de difusión debe permitir la administración de los vídeos ya grabados

Req-07 La administración de los vídeos ya grabados debe ser previa autenticación.

3.1.2 Requerimientos no funcionales

Se detallan a continuación los requerimientos no funcionales, ordenados según el modelo de requerimientos FURPS (Functionality, Usability, Realiability, Performance and Support) [23].

3.1.2.1 Requerimientos de Implementación

Req-08 El servidor de difusión debe utilizar como sistema operativo Linux.

Req-09 Los agentes de captura utilizarán el sistema operativo Linux Ubuntu.

Req-10 El navegador web para ingresar al portal tendrá que tener soporte para CSS y HTML 5.

Req-11 El código de los programas a implementar debe ser Open Source

3.1.2.2 Requerimientos de Desempeño

Req-12 Los Agentes de captura deben ser capaces de grabar a una resolución mínima de 720p para el caso del vídeo y las diapositivas.

Req-13 El servidor debe ser capaz de poder procesar las grabaciones al mismo tiempo de que se realiza la tarea de reproducción de vídeos de parte de los usuarios.

Req-14 Los Agentes de Captura deben ser capaces de grabar en menos de un minuto después de haber finalizado la grabación anterior.

3.1.2.3 Requerimientos de Operación

Req-15 Los agentes de captura deben indicar al usuario el estado en que se encuentran (Grabando, pausado, error, etc).

Req-16 El usuario no necesitará plugins adicionales para ver los vídeos.

Req-17 No se necesitará acceso físico al servidor. Se utilizará un portal Web para la administración del sistema.

Req-18 Los agentes de captura pueden ser utilizados por cualquier usuario (No necesitará credenciales de acceso).

3.1.2.4 Requerimientos de Confiabilidad

Req-19 Los Agentes de Captura guardarán una copia local de las grabaciones realizadas.

Req-20 El servidor funcionará bajo una máquina virtual, para un respaldo más robusto e independiente de la máquina física y escalabilidad futura.

Req-21 El servidor deberá contar con una unidad de respaldo de energía (UPS).

3.2 ALCANCE DEL PROYECTO

Como se mencionó durante la introducción, el alcance del proyecto sólo tomará en consideración los siguientes objetivos en sus etapas de diseño e implementación:

- Implementar un módulo de captura de un monitor, el video del presentador y audio usando software Galicaster.
- Desarrollar un módulo de captura móvil.
- Configurar un portal Opencast para el acceso del contenido grabado.
- Configurar una máquina virtual e implementar en ésta un servidor de almacenamiento y difusión de medios
- Reportar el rendimiento del sistema en escenarios de prueba
- Evaluar el uso de sistemas de video bajo demanda tipo YouTube para almacenamiento y difusión del contenido.

Además, este trabajo tiene las siguientes consideraciones:

- El sistema no transmite una señal de streaming en vivo (Como servicios ofrecidos por Livestream o UStream).
- La comunicación entre los agentes de captura y el servidor de difusión será de manera de que todos los equipos de red que intervengan en este proceso ya estén debidamente configurados.

3.3 DISEÑO

Como se vio en el punto 3.1, el diseño tiene que estar en función a los requerimientos levantados y los objetivos del alcance del proyecto. Este capítulo definirá la estructura completa de la plataforma. Cada una de las actividades del diseño de módulos irá en objetivo de dividir la plataforma completa en subsistemas con el objeto de manejar la complejidad del sistema, para luego, en la etapa de implementación definir la arquitectura del sistema.

3.3.1 Actividades del diseño de módulos

La actividad del diseño de módulos toma como entrada y salida los distintos procesos con el fin de determinar los módulos que se necesitarán para luego implementarlos.

3.3.1.1 Identificando metas de diseño

Las metas de diseño se obtienen directamente de los requerimientos no funcionales y el alcance del proyecto. Al revisar los requerimientos no funcionales descritos en 3.1.2, se puede orientar las siguientes características:

- **Plataforma de simple uso para los usuarios**: La plataforma debe ser de un uso muy simple, en la cual no se necesiten conocimientos específicos en alguna materia en particular ni tener que leer un complejo manual para utilizar el sistema de grabación de charlas.
- **Bajo coste de implementación:** Los costes de la implementación de la plataforma están dados principalmente por el uso de horas hombre en la puesta en marcha del sistema, del hardware asociado para su correcto funcionamiento y el software necesario para su funcionamiento. Para abaratar costes se utilizará software gratuito de código abierto.
- **Tolerante a fallas:** La plataforma debe proveer sistemas de fiabilidad en cuanto haga acciones como: Transmisión sobre la red, autenticación de usuarios. Para lidiar con estos problemas, la plataforma debe ser capaz de detectar estos problemas e informar al usuario sobre el estado actual, además debe notificar al usuario en caso de que se produzcan estos problemas.

3.3.1.2 Elección de la plataforma

Dadas las soluciones vistas en el capítulo anterior, hace necesario hacer la elección de la plataforma según los requerimientos levantados. Se generó la siguiente tabla con cada una de las soluciones en función si cumplían o no con todos los requerimientos:

Tabla 1: Comparación de cumplimiento de requerimientos

Matriz de requerimientos

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Kaltura	X	X		X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	-
Panopto	X	X	X	X	X	X	X			X		X	X	X	X	X	X		X	X	-
TechSmith Relay	X			X	X		X			X		X	X	X	X	X	X	X	X	X	-
Cisco WebEx	X	X		X	X		X			X		X	X	X	X	X	X				-
Opencast + Galicaster	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-

Como se puede ver en la Tabla 1, la única solución que cumple con las exigencias es Opencast + Galicaster. Kaltura cumplía todos los requerimientos a excepción de poder calendarizar los eventos y programar la subida del contenido automáticamente.

Entre las otras soluciones, varias no cumplían con otros puntos y en el Caso de WebEx, no existe la posibilidad de tener servidor propio, por tanto, no se tiene control de donde está montado. En el caso del requerimiento número 21, no aplicaba, ya que es una característica que pueden tener todas las soluciones, es independiente de ellas.

Con esta comparación hecha, queda elegida la solución Opencast + Galicaster como la seleccionada para implementar.

3.3.1.3 Descomposición de la plataforma elegida

Opencast + Galicaster se puede ver como un gran bloque donde entra el audio y vídeo tanto del presentador como de la propia presentación a la plataforma y esta sale de forma codificada y publicada en un sitio web, todo esto previamente configurado, como lo muestra la imagen a continuación:

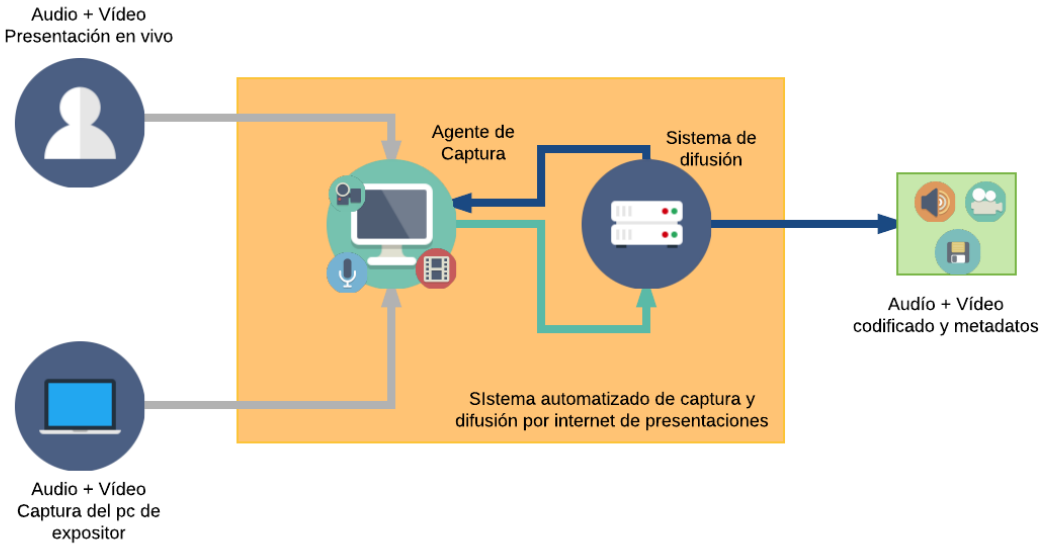


Figura 17: Entradas y salidas del sistema de captura y difusión de presentaciones

El agente de captura y el servidor de difusión, son tal como se vio en los diagramas del punto 1.3, los módulos principales de todo el sistema. La descomposición de cada módulo se define de la siguiente forma:

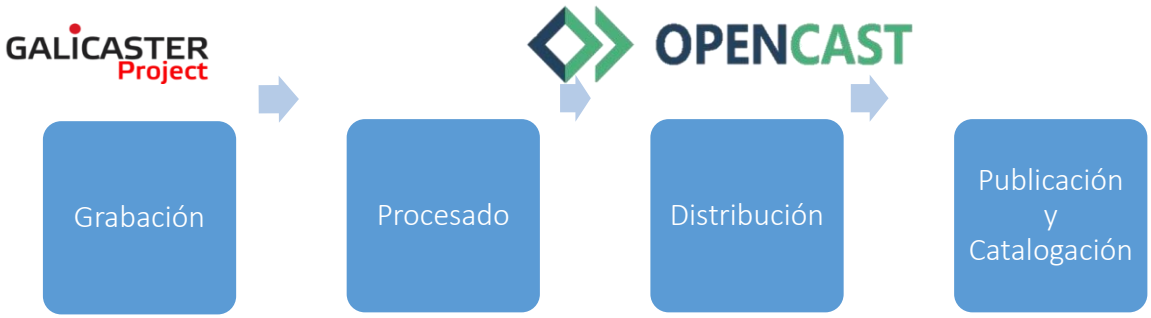


Figura 18: Flujo de trabajo del sistema completo

En donde el Agente de Captura sólo se encarga de la grabación del contenido, mientras que el servidor de difusión del resto de los otros elementos de la producción del material.

3.3.1.3.1 Agente de captura

El agente de captura es un dispositivo que se compone de parte de hardware y software, para cumplir con las metas de diseño y los requerimientos funcionales se creó el esquema tomando como base las entradas y salidas.

Este módulo tiene cinco entradas, las cuales son:

- Vídeo del presentador.
- Audio del presentador + Audio del computador del presentador.
- Vídeo de la captura del computador.
- Órdenes del servidor de difusión (Calendarización de los eventos y grabación automática).
- Interacción del usuario con la interfaz del agente de captura.

En cuanto a sus salidas son:

- Audio, Vídeo tanto del computador del presentador como del mismo presentador.
- Estado actual del Agente de captura hacia el servidor de difusión.
- Estado actual del agente de captura.

<u>Descripción del Software:</u>

El Software a utilizar para controlar el agente de captura es Galicaster. Es un software que tiene por objetivo la grabación de múltiples pistas de audio y vídeo.

Galicaster se encarga de la grabación del contenido, este es un punto crítico, ya que depende directamente del factor humano, define la calidad del material y los errores que se susciten en este punto, se van a transmitir al resto del flujo de trabajo de todo el sistema.

Algunos de los factores que pueden ocurrir a la hora de grabar una presentación son los siguientes:

- Micrófono desenchufado, silenciado o sin batería.

- Computador del presentador mal configurado o desconectado
- Mala iluminación
- Presentador situado fuera de plano

Este software mitiga estos factores con una interfaz gráfica que muestra alertas de grabación y el estado actual. Grabación automática y flexible, lo que permite una calendarización automática de los eventos, y un control básico sin operadores.

Figura 19: Pantalla de grabación de Galicaster

En la Figura 19, se muestra la pantalla de grabación de Galicaster, en ella se encuentran los siguientes elementos:

1. Estado de la grabación.
2. Imagen tanto del presentador como de la presentación.
3. VU meter, que permite monitorear el volumen adecuado del micrófono.
4. Controles básicos para manejar el sistema.
5. Una descripción en caso de que el evento haya sido programado y de los eventos próximos.

Una vez hecha la grabación y dependiendo de la configuración previa del sistema se “ingesta” al servidor de difusión. Galicaster cuenta con un administrador de medios para ver el estado actual de las grabaciones para hacer la ingesta de forma manual y/o verificar si esta acción se hizo automáticamente:

Figura 20: Administrador de medios de Galicaster

Otra de las características que tiene el programa es su gran flexibilidad de hardware compatible que puede ser utilizado para la grabación, algunos de los cuales son:

- Cámaras TV.
- Captura de salida de vídeo desde PC (VGA, HDMI, etc).
- Webcams.
- Cámaras IP y/o robotizadas.
- Microfonía y/o audio directo.

Esta aplicación es del tipo modular abarcando una serie de tecnologías para su funcionamiento, su diseño consta de tres módulos encargados de las siguientes funciones [24]:

- Interfaz de usuario (GUI), creada utilizando GTK.
- Núcleo del programa (Core), programada en Python.
- Sistemas Multimedia (Multimedia), basada en el framework GStreamer.

La interacción de estos módulos es de la siguiente manera:

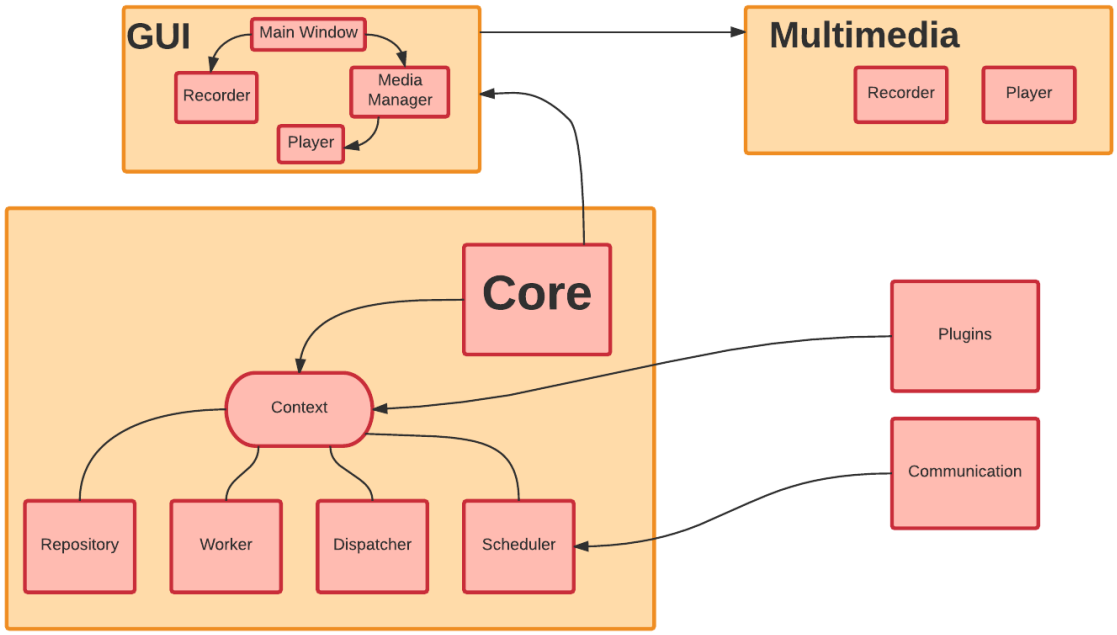


Figura 21: Arquitectura de Galicaster

Como se puede ver en la Figura 21, el núcleo del sistema (Core), empieza a obtener los datos contextuales del entorno, al mismo tiempo, éste inicializa a la interfaz de usuario (GUI). La interfaz de usuario controla directamente al módulo Multimedia, encargado de grabar y reproducir el contenido generado.

Una descripción más detallada de la arquitectura interna del software mostrado en la Figura 21, se puede encontrar en el sitio web del programa [21].

<u>Diseño de Hardware:</u>

Para que Galicaster funcione de manera correcta y en concordancia de los requerimientos y metas de diseño, se necesita que sea implementado en función del siguiente diagrama:

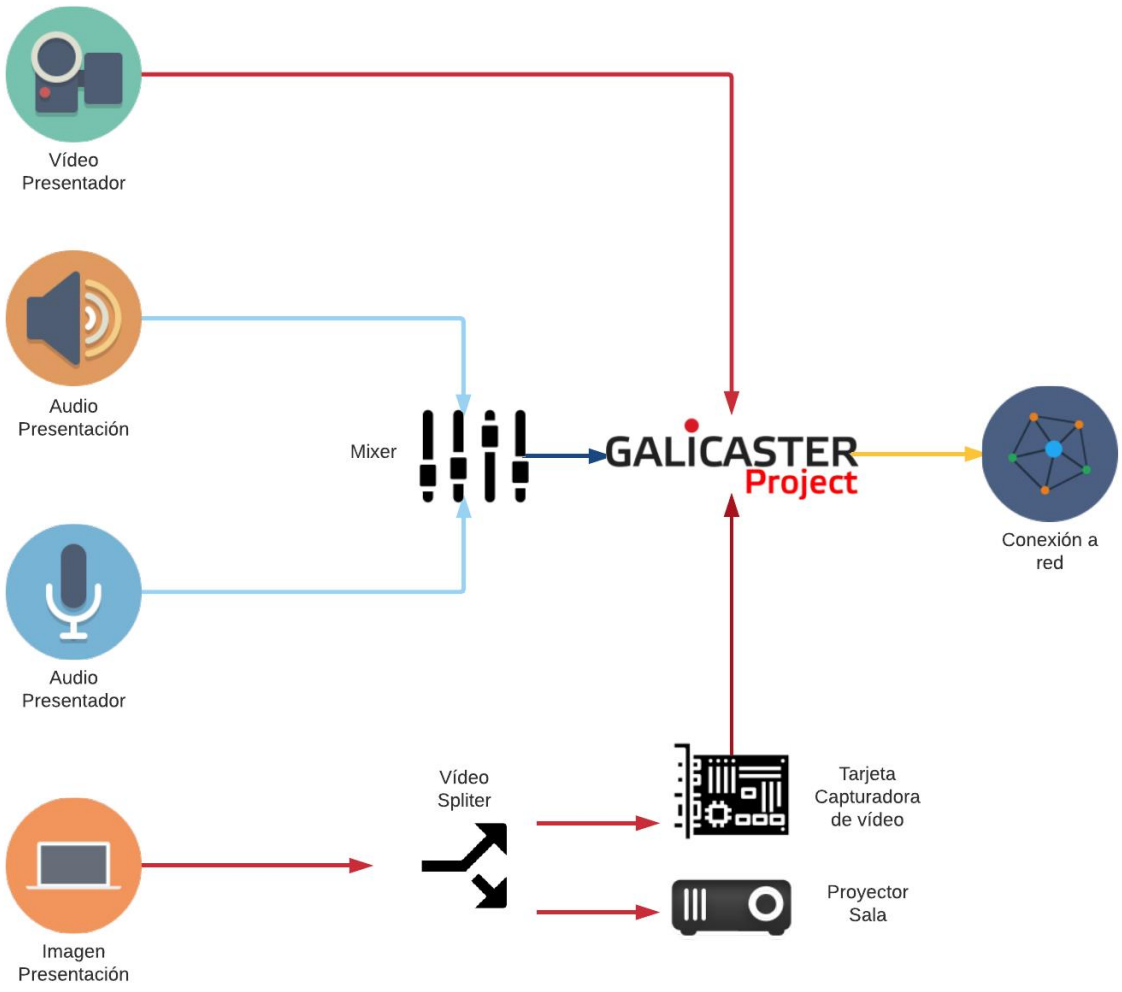


Figura 22: Diseño de Arquitectura Agente de captura

La descripción de los componentes es como sigue:

Tabla 2: Componentes necesarios de HW para galicaster.

Módulo	Descripción
Vídeo presentador	Imagen en tiempo real del presentador
Vídeo presentación	Imagen en tiempo real de la pantalla del computador del presentador
Audio presentador	Captura del audio del presentador en tiempo real
Audio presentación	Captura del audio del computador del presentador en tiempo real
Mixer (Mezclador)	Dispositivo utilizado para mezclar las señales de audio en un solo canal estéreo
Vídeo Splitter	Dispositivo utilizado para duplicar la señal de vídeo desde una fuente configurada previamente.
Proyector Sala	Muestra la imagen de la presentación en la sala
Tarjeta capturadora de vídeo	Tarjeta especializada en la captura del vídeo de la presentación para su procesamiento posterior
Conexión a red	Conexión necesaria para la comunicación con el servidor de difusión

Cabe mencionar que Galicaster funciona bajo un computador que utilice el sistema operativo Ubuntu Linux.

3.3.1.3.2 Servidor de difusión

Las entradas que deberá tener el servidor son las siguientes:

- Entrada de audio, vídeo y metadatos encapsulados en un paquete de datos desde los agentes de captura enviados por la red.
- Configuraciones del administrador.
- Estado de los agentes de captura conectados.
- Peticiones de los usuarios para reproducir las grabaciones ya publicadas.

Las salidas del servidor de difusión son:

- Generación del portal Web para visualizar las grabaciones.
- Streaming bajo demanda de las grabaciones ya procesadas.
- Control de los agentes de captura (Calendarización de grabaciones).

El Servidor de difusión será hecho bajo la solución Opencast, esta solución tiene un diseño modular basado el lenguaje JAVA utilizando el esquema OSGi (Open Services Gateway initiative) [25] para la modularización de sus componentes. El esquema OSGi corre sobre Apache Karaf, el cual es un ambiente OSGi que adiciona nuevas herramientas a una implementación estándar de este sistema.

La arquitectura de Opencast está dada por los siguientes nodos:

Tabla 3: Componentes de la Arquitectura de SW Opencast

Nodo	Descripción
Admin	Encargado de controlar el resto de los módulos y entregar el panel administrativo de estado del sistema.
Worker	Módulo encargado de transformar, recodificar los vídeos para el uso en los reproductores de vídeo objetivo y medios de difusión definidos.
Ingest	Recibe los paquetes de medios desde los agentes de captura. Adiciona información adicional de ellos y los guarda en el Storage
Presentation and Media	Encargado de la difusión del contenido ya procesado en un portal web para que los usuarios puedan utilizar el contenido producido
Storage	El Storage oficialmente no está catalogado como un nodo de Opencast, pero se toma como tal, ya que es la parte encargada de guardar tanto los elementos en procesamiento como los elementos ya procesados listos para distribuir y los flujos de datos que llegan desde el módulo de ingesta.

La interacción de estos módulos es de la siguiente forma:

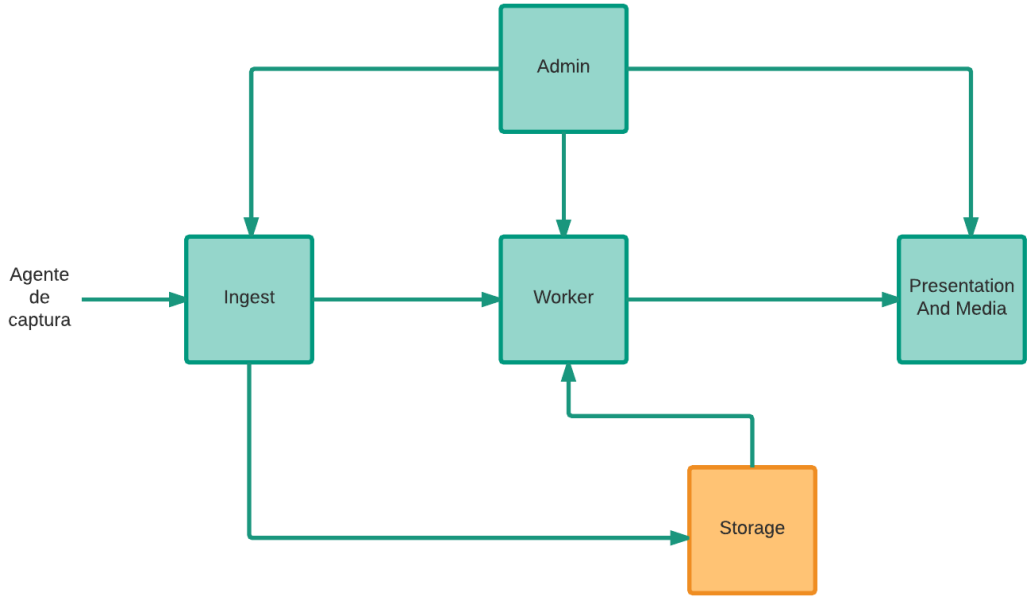


Figura 23: Composición de funcionamiento de Opencast

En cuanto al hardware necesario por Opencast sólo necesita una máquina física para poder funcionar y conexión a una red. Cabe destacar que Opencast permite instalar sus módulos en máquinas distintas como configuración opcional.

Para cumplir la meta de diseño de bajo coste de implementación, se implementó que el sistema Opencast se ejecute sobre un servidor de máquinas virtuales, en el cual existan dos máquinas, una para albergar Opencast con todos sus módulos y otra máquina virtual que albergase el almacenamiento de los vídeos del sistema.

La idea sobre esta solución radica que en el caso que hubiera que aumentar la capacidad de disco, fuera de manera sencilla sin tener que bajar el servicio de difusión de contenido para hacer esta tarea. También permite que la máquina que alberga a Opencast sea de menor tamaño y fácil de respaldar.

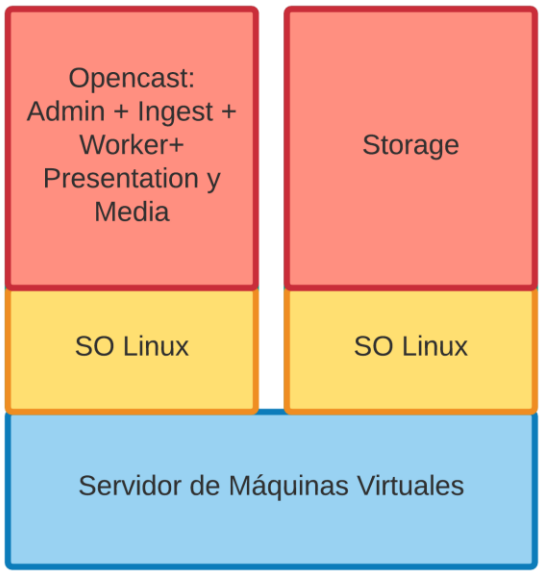


Figura 24: Diseño para la implementación de Opencast

3.4 IMPLEMENTACIÓN

Con el diseño ya finalizado, se hace la implementación de la plataforma. La implementación que se logró en este trabajo fue la de un servidor de difusión y un agente de captura. Adicionalmente, gracias al Programa de Iniciativas Estudiantiles Académicas, PIE>A, también se desarrolló e implementó un agente de captura móvil.

3.4.1 Servidor de difusión

3.4.1.1 Software utilizado

En el caso particular de opencast, se implementó tanto un servidor de producción con Opencast 2.0 y un servidor de pruebas con Opencast 2.2.2.

Además de Opencast, para la implementación del servidor se utilizaron los siguientes elementos:

- Servidor de máquinas virtuales

- Sistema Operativo
- Base de datos para Opencast
- Servidor Web

La descripción y elección de los elementos nombrados es a continuación:

3.4.1.1.1 Servidor de Máquinas virtuales

Como se explicó en el punto 3.3.1.3.2, se opta por una máquina virtual para alojar Opencast, se utilizó Citrix XenServer, el cual es una plataforma de virtualización de código abierto para la administración de infraestructuras virtuales [26].

La elección de esta solución fue por el amplio conocimiento y experiencia que tiene la Red de Computadores de Electrónica (RCE) en la utilización de esta plataforma de virtualización. XenServer, como es de código abierto, permite que sea utilizado sin coste.

3.4.1.1.2 Sistema Operativo

El sistema Operativo elegido a utilizar fue CentOS (**C**ommunity **ENT**erprise **O**perating **S**ystem), el cual es una bifurcación a nivel binario de la distribución Linux Red Hat Enterprise Linux (RHEL), compilado por voluntarios a partir del código fuente publicado por Red Hat.

El objetivo es ofrecer al usuario un software de “clase empresarial” gratuito. Se define como robusto, estable y fácil de instalar y utilizar. La versión actual es el número 7, la utilizada en este trabajo.

La elección de este sistema operativo recae en que los desarrolladores de Opencast, aunque, lo hicieron compatible para varias distribuciones de Linux, es la más utilizada por los usuarios de Opencast para el despliegue de este sistema.

3.4.1.1.3 Base de datos para Opencast

Uno de los requerimientos de Opencast para funcionar correctamente es el utilizar una base de datos relacional. Opencast incluye la base de datos H2 (HSQLDB). Esta base de datos es utilizada por defecto y no necesita configuración.

Los desarrolladores recomiendan fuertemente que esta base de datos sea reemplazada y se utilice MariaDB 10.0 (O MySQL 5.6). Las ventajas radican en que MariaDB tiene un mayor rendimiento [27], mejores herramientas de depuración y documentación.

Otras ventajas ocultas de la utilización de una base de datos independiente son: Permite la instalación distribuida de Opencast y la capacidad de actualizar a nuevas versiones del programa.

Se decide utilizar MariaDB sobre MySQL ya que la primera posee scripts de migración más robustos que el segundo.

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL. Está desarrollado por Michael (Monty) Widenius (fundador de MySQL) y la comunidad de desarrolladores de software libre. 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.

3.4.1.1.4 Servidor Web

Para que al sistema se pueda acceder de manera directa, sin tener que digitar directamente el puerto en la barra de direcciones del navegador, se despliega un servidor web HTTP para que haga de proxy reverso entre Opencast y el usuario que quiere acceder al contenido. El servidor utilizado es Apache HTTP server.

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. 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) [28].

3.4.1.2 Hardware Utilizado

El Hardware Utilizado para la Implementación fue un servidor Dell PowerEdge R430. Este servidor se instaló dentro del data center del Departamento de Electrónica de la UTFSM.

Figura 25: Servidor Dell PowerEdge R430 instalado en dependencias del Departamento de Electrónica UTFSM

Sus Características son las siguientes:

Tabla 4: Características Dell PowerEdge R430

Característica	Especificación
CPU	Intel® Xeon® Processor E5-2620 (15M Cache, 2.00 GHz, 7.20 GT/s Intel® QPI) 6 núcleos
RAM	16 GB RDIMM
Disco Duro	2 X Discos duros Sata de 7200rpm 1TB configurados en RAID 1
Red	Dos puertos Ethernet de 1 Gbps cada uno + un puerto de administración iDRAC

3.4.1.3 Configuración de la arquitectura

3.4.1.3.1 Diagrama de red

En la red que está instalado el servidor, de manera simplificada es como se muestra en la Figura 26:

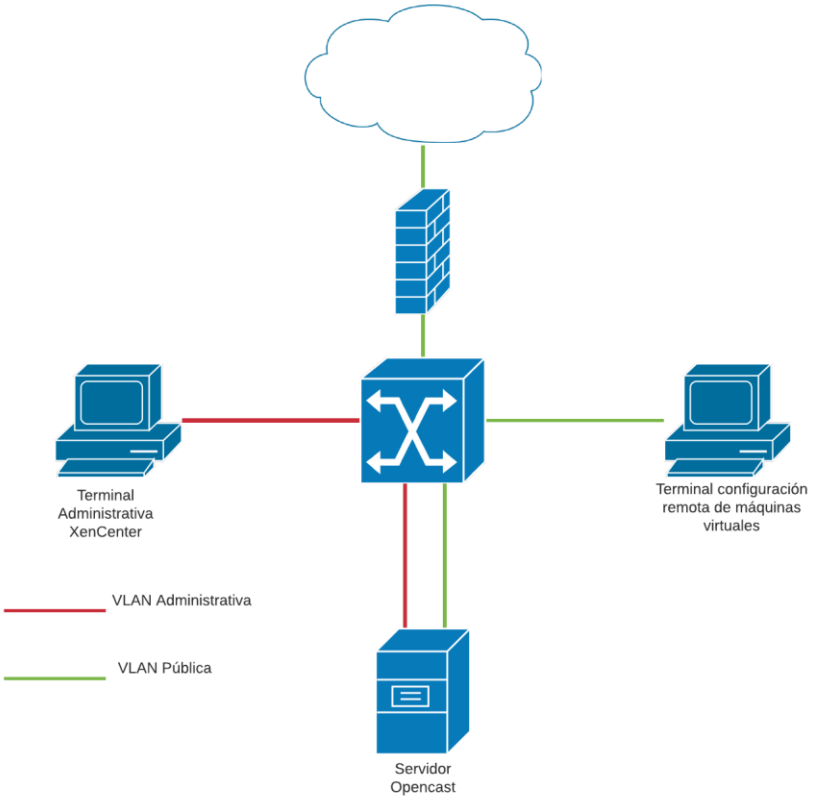


Figura 26: Diagrama de Red Servidor Opencast

Como se ve en la Figura 26, el servidor está conectado a dos VLAN, la primera, la VLAN administrativa es una VLAN dedicada sólo para la mantención y estado de los componentes de red, servidores y UPS que existen en el Departamento de electrónica, esta red no se puede acceder directamente por internet y está aislada del resto.

La VLAN pública tiene asignadas un rango de las IP públicas que tiene la universidad, teniendo comunicación directa hacia el mundo exterior previa configuración del firewall institucional. Para el caso de este trabajo, el firewall ya se encuentra debidamente configurado.

Para poder configurar las máquinas virtuales, sólo es posible a través de un terminal conectado a la VLAN administrativa, utilizando el software llamado XenCenter de Citrix.

3.4.1.3.2 Instalación del Servidor de máquina virtual y configuración de las máquinas virtuales.

Lo primero que se hace con el servidor, es instalar el servidor de máquinas virtuales, para luego preparar el entorno de cada máquina virtual.

La instalación de XenServer, es de manera directa siguiendo las instrucciones en pantalla. Una vez instalado y configurado el sistema se proceden a configurar las máquinas virtuales:

Tabla 5: Características Servidor Spielberg

Servidor Opencast Producción (Spielberg)	
Servidor Opencast 2.0, corriendo CentOS 7	
Número de núcleos CPU	4
RAM asignada	4 GB
Capacidad disco duro virtual	500 GB

Tabla 6: Características Servidor Cameron

Servidor Opencast de Pruebas (Cameron)	
Servidor Opencast 2.2.2, corriendo CentOS 7	
Número de núcleos CPU	4
RAM asignada	4 GB
Capacidad disco duro virtual	50 GB

Tabla 7: Características Servidor Storage

Servidor de almacenamiento NFS (Storage)	
Sistema Operativo CentOS 7	
Número de núcleos CPU	4
RAM asignada	4 GB
Capacidad disco duro virtual	100 GB

A todas las máquinas virtuales se les asigna una puerta de red virtual con acceso a la VLAN pública de la UTFSM.

En cuanto a la instalación del sistema operativo, el único resguardo que hay que tener es que en las dos máquinas exista un usuario de nombre Opencast con el mismo id de usuario y el mismo id de grupo. Esto será necesario para la configuración del servicio NFS.

3.4.1.3.3 Despliegue y configuración de Opencast

La configuración de Spielberg como de Cameron son prácticamente iguales, por tanto, se mostrará solo la configuración de los servidores Cameron y Storage, debido a que esta configuración lleva varias mejoras con respecto a Spielberg como:

- Nueva Versión de Opencast (En específico, la versión 2.2.2).
- Utilización de base de datos independiente.
- Almacenamiento en servidor independiente.

Las configuraciones a continuación están hechas como usuario ROOT.

Configuración del entorno

Antes de iniciar la instalación, se deben configurar los repositorios, esto se hace de la siguiente manera:

1. Se necesita registrarse previamente en el repositorio de Opencast, para ello hay que acceder al sitio web http://pkg.opencast.org

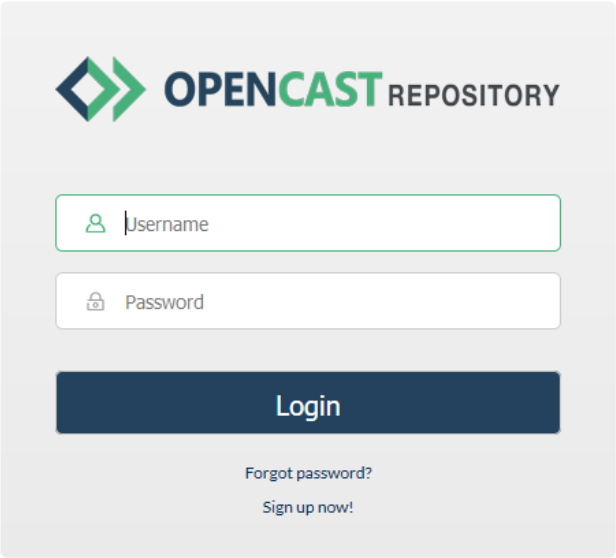


Figura 27: Pantalla de acceso repositorio Opencast

2. Se activa el repositorio de Opencast utilizando las credenciales creadas en el sitio web mencionado, utilizando las siguientes líneas de comando en el Shell del sistema:

```
# cd /etc/yum.repos.d curl -O https://pkg.opencast.org/opencast.repo \
   -d os=el -d version=7 -u [YOUR_USERNAME]
```

Donde `[YOUR_USERNAME]` es el nombre de usuario registrado previamente. Cuando se ejecuta, pide la contraseña generada en el primer paso.

3. Se agrega el repositorio EPEL con la siguiente línea de comando:

```
# yum install epel-release
```

4. Instalar Apache ActiveMQ

Apache ActiveMQ es un intercambiador de mensajes esencial para el funcionamiento de Opencast desde su versión 2.0. Este se instala con el comando

```
# yum install activemq-dist
```

La configuración de ActiveMQ se ve más adelante.

Instalación Opencast

Para la instalación de Opencast se utiliza el paquete “All-in-One” que lleva todos los nodos del sistema para ser utilizados en una sola máquina.

```
# yum install opencast22-allinone
```

Esto instalará Opencast con todas sus dependencias, incluyendo las herramientas de terceros.

Aunque Opencast está listo para ser ejecutado, este no está correctamente configurado, se necesita antes configurar el intercambiador de mensajes, la base de datos, el almacenamiento en red y finalmente preparar los archivos de configuración de Opencast.

Configuración ActiveMQ

La configuración de ActiveMQ tiene dos objetivos:

- Preparar las colas de mensajes requeridas para Opencast.
- Apuntar a todos los nodos de Opencast al intercambiador de mensajes.

Pasos de la configuración:

1. Copiar el archivo de configuración de Opencast a la carpeta de ActiveMQ

Opencast ya viene con un archivo de configuración listo para ActiveMQ, para utilizarlo se debe hacer el siguiente paso:

```
# cp /usr/share/matterhorn/docs/scripts/activemq/activemq.xml \
 /etc/activemq/activemq.xml
```

2. Configurar seguridad del Intercambiador de mensajes.

El paso final es asegurar las colas de ActiveMQ limitándose a un grupo, esto se hace editando el archivo de configuración activemq.xml en el directorio de configuración de ActiveMQ (Localizado en `/etc/activemq/`).

Añadir la siguiente configuración entre los tags XML `<broker></broker>`:

```
<plugins>
    <jaasAuthenticationPlugin configuration="activemq" />
    <authorizationPlugin>
        <map>
            <authorizationMap>
                <authorizationEntries>
                    <authorizationEntry queue=">" read="admins"
write="admins" admin="admins" />
                    <authorizationEntry topic=">" read="admins"
write="admins" admin="admins" />
                    <authorizationEntry topic="ActiveMQ.Advisory.>"
read="admins" write="admins" admin="admins"/>
                </authorizationEntries>
            </authorizationMap>
        </map>
    </authorizationPlugin>
</plugins>
```

3. Probar ActiveMQ

ActiveMQ ya está listo para funcionar, para probar su funcionamiento, primero se debe desactivar el firewall (O añadir los puertos 61616 y 8161 como excepciones), luego iniciar (Y habilitar para que se inicie junto con el encendido de la máquina) el servicio con los comandos:

```
# systemctl enable activemq
# systemctl start activemq
```

Luego entrar a `http://<ip_servidor>:8161/admin` con el nombre de usuario admin, contraseña admin, se ve el servicio ActiveMQ ya corriendo.

<u>Instalar y configurar la base de datos</u>

Como se mencionó en el punto 3.4.1.1.3, la base de datos MariaDB fue la escogida para ser utilizada para Opencast, las instrucciones de instalación y configuración son las siguientes:

1. Instalar MariaDB

La instalación se hace utilizando el siguiente comando:

```
# yum install mariadb mariadb-server
```

Luego de la instalación ejecutar:

```
# systemctl start mariadb.service
# systemctl enable mariadb.service
```

Ahora que la base de datos está corriendo, hay que configurar los permisos de la cuenta administrativa (ROOT) de la base de datos, para ello hay que ejecutar el comando:

```
# mysql_secure_installation
```

Este comando guía paso a paso para preparar la cuenta ROOT de la base de datos

2. Crear una base de datos Opencast

Para crear la base de datos se utiliza el comando para usar el cliente Shell de MariaDB

```
mysql> Mysql –u root –p
```

Una vez adentro, crear una base de datos de nombre opencast ejecutando:

```
mysql> CREATE DATABASE opencast CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Luego crear el usuario opencast con el password opencast_password y concederle todos los derechos necesarios:

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,INDEX ON opencast.*
  TO 'opencast'@'localhost' IDENTIFIED BY 'opencast_password';
```

Nota: Se puede escoger tanto como otro nombre de usuario de la base de datos como contraseña, lo cual es recomendado por razones de seguridad.

Se vuelve al Shell del sistema y se reinicializa el servicio de la base de datos utilizando:

```
# systemctl restart mariadb.service
```

3. Preparar la estructura de la base de datos:

Para preparar la estructura de la base de datos, se utilizan los scripts dll de Opencast, estos se encuentran en la carpeta de documentación de la instalación del programa, en el caso particular de este trabajo se encuentran en:

```
/usr/share/opencast/docs/scripts/ddl
```

Para ejecutar el script, se debe escribir la siguiente línea:

```
# mysql -u opencast -p opencast < /usr/share/opencast/docs/scripts/ddl
```

4. Configuración base de datos en Opencast

Abrir el archivo con el editor de preferencia (para este trabajo se utiliza el editor NANO):

```
# nano etc/opencast/custom.properties
```

Cambiar la siguiente llave de configuración a “falso” como se expresa a continuación:

```
org.opencastproject.db.ddl.generation=false
```

Cambiar la equivalencia de la línea `org.opencastproject.db.vendor`, por (Descomentar en caso de que esté hecho):

```
org.opencastproject.db.vendor=MySQL
```

Configurar el driver JDBC (Java DataBase Connection) para MariaDB/MySQL:

```
org.opencastproject.db.jdbc.driver=com.mysql.jdbc.Driver
```

Configurar la dirección donde Opencast debiera encontrar la base de datos y el nombre de la base de datos. En este caso la base de datos es local (localhost) y su nombre es “opencast”:

```
org.opencastproject.db.jdbc.url=jdbc:mysql://localhost/opencast
```

Configurar el mismo nombre de usuario con el cual Opencast debiera acceder a la base de datos:

```
org.opencastproject.db.jdbc.user=opencast
org.opencastproject.db.jdbc.pass=opencast_password
```

Con estos pasos concluye la puesta en marcha de la base de datos.

Configuración del Servidor NFS:

Como se menciona dentro de la Tabla 3, el nodo Storage estará compuesto por un servidor dedicado al almacenamiento de los recursos multimedia que trabaje Opencast, la configuración para la puesta en marcha de este sistema es de la siguiente manera:

1. *Configuración servidor Storage:*

En el servidor Storage, el cual su entorno del sistema operativo e interfaces de red ya fueron correctamente configuradas, se ejecutan los siguientes comandos [29] [30]:

```
# yum install nfs-utils nfs-utils-lib
# systemctl enable rpcbind
# systemctl start rpcbind
# systemctl enable nfs-server
```

```
# systemctl start nfs-server
```

Se crea la carpeta en donde se guardarán localmente los recursos del servidor Opencast:

```
# mkdir -p /opencast
# chown opencast:opencast /opencast
```

Luego con el editor de texto, se ingresa al archivo de /etc/exports para configurar las reglas de uso compartido de la carpeta antes creada:

```
/opencast  <ip_del_servidor_opencast>(rw,sync,no_subtree_check)
```

Siendo `<ip_del_servidor_opencast>` la ip de la máquina que tendrá acceso a esa carpeta.

Se reinicia el servicio NFS para que tome la nueva configuración:

```
# systemctl restart nfs-server
```

Finalmente, se agrega el servicio NFS al firewall del servidor:

```
# firewall-cmd –permanent --zone=public --add-service=nfs
# firewall-cmd –reload
```

2. *Configuración NFS en servidor Opencast*

La configuración del servidor Opencast sigue de la misma manera que el servidor storage, se ejecutan los siguientes comandos [29] [30]:

```
# yum install nfs-utils nfs-utils-lib
# systemctl enable rpcbind
# systemctl start rpcbind
# systemctl enable nfs-server
# systemctl start nfs-server
```

Se monta el almacenamiento en red de manera manual para probar el funcionamiento

```
# mount -t nfs <ip_del_servidor_storage>:/opencast /srv/opencast/
```

Ahora que están conectados se hace un chequeo como se indica:

```
# df -kh
```

Devuelve lo siguiente:

```
S.ficheros                        Tamaño Usados  Disp Uso% Montado en
/dev/mapper/centos_dhcp239-root     46G   2,7G   44G   6% /
devtmpfs                           1,9G      0  1,9G   0% /dev
tmpfs                              1,8G      0  1,8G   0% /dev/shm
tmpfs                              1,8G    41M  1,7G   3% /run
tmpfs                              1,8G      0  1,8G   0% /sys/fs/cgroup
/dev/xvda1                         497M   153M  345M  31% /boot
<ip_storage>:/opencast             99G   1,4G   92G   2% /srv/opencast
tmpfs                              354M      0  354M   0% /run/user/1000
```

Lo cual confirma que están conectados. Se chequea que existan permisos de lectura/escritura en la carpeta compartida. En el servidor Opencast se entra el comando:

```
# touch /srv/opencast/test_nfs
```

Al utilizar "`ls /srv/opencast/`" devuelve que el archivo "`test_nfs`" existe.

3. *Montaje permanente NFS*

Para que el montaje se mantenga en caso de que el servidor sea apagado, se debe editar el archivo "`fstab`" del servidor Opencast:

```
# nano /etc/fstab
```

Y agregar la siguiente línea al final del archivo:

```
<ip_del_servidor_storage>:/srv/opencast     /srv/opencast              nfs
rw,hard,intr,rsize=32768,wsize=32768 0 0
```

Con esto queda concluida la implementación del servidor NFS para Opencast.

Configuración de Opencast:

Con la base de datos, el intercambiador de mensajes y el almacenamiento remoto ya preparados, se editan los archivos de configuración de Opencast:

1. *Custom.properties*

Este archivo se encuentra en `/etc/opencast/`, y es el encargado de configurar las personalizaciones necesarias para cada instalación de Opencast.

Lo primero es editar la línea:

```
org.opencastproject.server.url=http://<URL>:8080
```

Configurar la URL del servidor, la URL puede ser tanto un nombre de dominio como una dirección IP, en el caso del servidor Cameron se utilizó una IP pública previamente asignada por la Red de Computadores de Electrónica. En el caso del servidor Spielberg se utiliza la URL "opencast.elo.utfsm.cl".

Atención: Este parámetro es muy importante, y se recomienda encarecidamente utilizar una URL en vez de una dirección IP. La razón de esto es debido a que cuando se registran nuevos recursos en la Base de Datos, estos quedan apuntados utilizando la dirección que indica este parámetro. El cambio de este parámetro durante la operación de la máquina resultaría en una pérdida de las rutas.

Al no tener un dominio definido para el servidor de pruebas, se dejó con su dirección IP.

Luego hay que editar el parámetro `org.ops4j.pax.web.listening.addresses`, este parámetro se utiliza para que Opencast sólo se pueda acceder desde ciertas direcciones IP, se debe dejar en 0.0.0.0 para dar acceso general a toda la red

```
org.ops4j.pax.web.listening.addresses=0.0.0.0
```

Configurar la locación del directorio compartido:

```
org.opencastproject.storage.dir=/srv/opencast
```

Definir qué repositorio de archivos se accederán todos los archivos localmente

```
org.opencastproject.file.repo.url=${org.opencastproject.admin.ui.ur
l}
```

Finalmente se configuran las credenciales de acceso, estas sirven tanto para acceder a través del panel administrativo como para que los agentes de captura se puedan registrar en el sistema, los parámetros a cambiar son:

```
org.opencastproject.security.admin.user =admin
org.opencastproject.security.admin.pass =opencast
org.opencastproject.security.digest.user =opencast_system_account
org.opencastproject.security.digest.pass =CHANGE_ME
```

Los valores "`admin`" son para el panel administrativo, mientras que los valores "`digest`" corresponden a los que utilizan los agentes de captura para conectarse al servidor, así como los otros nodos del sistema en caso de que sea una arquitectura distribuida.

2. *org.opencastproject.organization-mh_default_org.cfg*

Opencast tiene la capacidad de poder administrar varios sitios de distintas organizaciones, esto es ideal cuando se tienen distintos departamentos a los cuales se quiera brindar servicio, sin tener que armar un nuevo sistema de servidores para este propósito. El caso mostrado en este trabajo sólo utiliza una sola organización, y, por tanto, las configuraciones por defecto, sin embargo, este archivo configura las rutas de acceso para los logotipos del Módulo de medios lo que permite cierta personalización del portal donde se publican las grabaciones.

Las líneas modificadas son las siguientes:

```
prop.logo_mediamodule=http://opencast.elo.utfsm.cl/datos/logo_depto
_elo.png
prop.logo_player=http://opencast.elo.utfsm.cl/datos/logo_depto_elo_
grande.jpg
```

3.4.1.3.4 Puesta en Marcha del sistema

Con todos los módulos ya configurados, se puede dar partida al servidor Opencast, para que funcione de manera adecuada debe haber sido configurado el Firewall del sistema previamente, las instrucciones de cómo se configuró se encuentran en el anexo de este documento.

Ejecutar en la línea de comandos del sistema:

```
# systemctl enable opencast.service
# systemctl start opencast.service
```

Opencast se inicia, al terminar de cargar, se puede acceder a la ventana de configuración (Figura 28) y al módulo de presentación desde la URL configurada previamente:

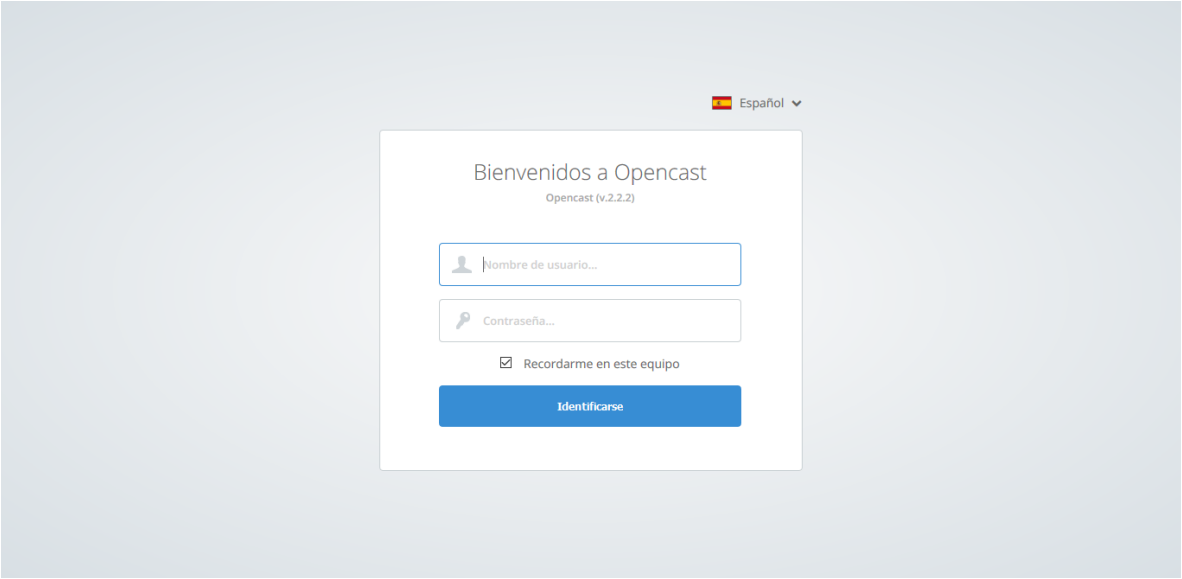


Figura 28: Sitio de bienvenida al panel de administración de opencast

Módulo de Medios (Figura 29):

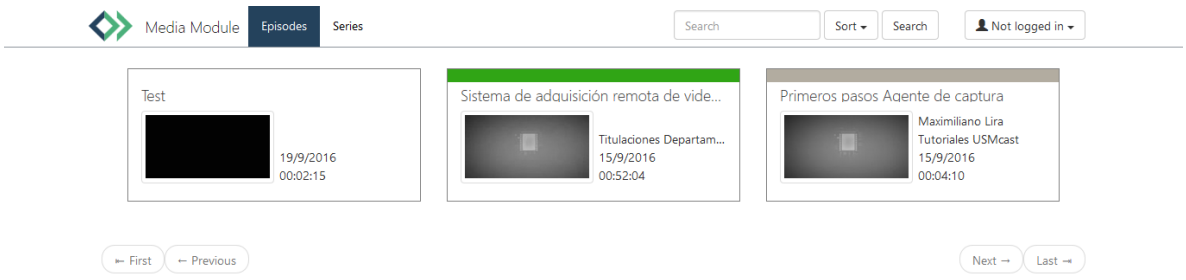


Figura 29: Presentación de módulo de medios (Presentación) de Opencast

La Figura 29, muestra un portal de medios, en el cual ya fueron integrados algunos vídeos para efectos de la demostración, cuando se inicia por primera vez, no hay recursos ya añadidos.

Para que se pueda ingresar a Opencast directamente sin tener que escribir explícitamente el puerto en la barra de direcciones se utiliza un proxy reverso creado por el servidor Apache, las instrucciones para instalar este servicio se encuentran en los anexos de este trabajo.

3.4.2 Agente de captura estacionario

El agente de captura estacionario se define como un dispositivo capaz de grabar tanto al presentador como la presentación y enviarla al servidor de difusión una vez terminado el evento, también puede recibir órdenes desde el servidor de difusión para iniciar grabaciones automatizadas del sistema.

3.4.2.1 Software utilizado

El software utilizado además de Galicaster (El cual fue instalado en su versión 1.4.2) como se definió en las secciones anteriores solo contempla el sistema operativo.

En este caso se utilizó como sistema operativo Linux Ubuntu en su versión 12.04. La razón principal del uso de esta versión y distribución es la compatibilidad que tiene con Galicaster y sus dependencias.

Ubuntu es un sistema operativo basado en GNU/Linux y que se distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity. Está orientado al usuario promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia del usuario.

Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto [31].

3.4.2.2 Hardware Utilizado

El hardware utilizado se puede ver en la siguiente figura 42 y la Tabla 8 que viene a continuación:

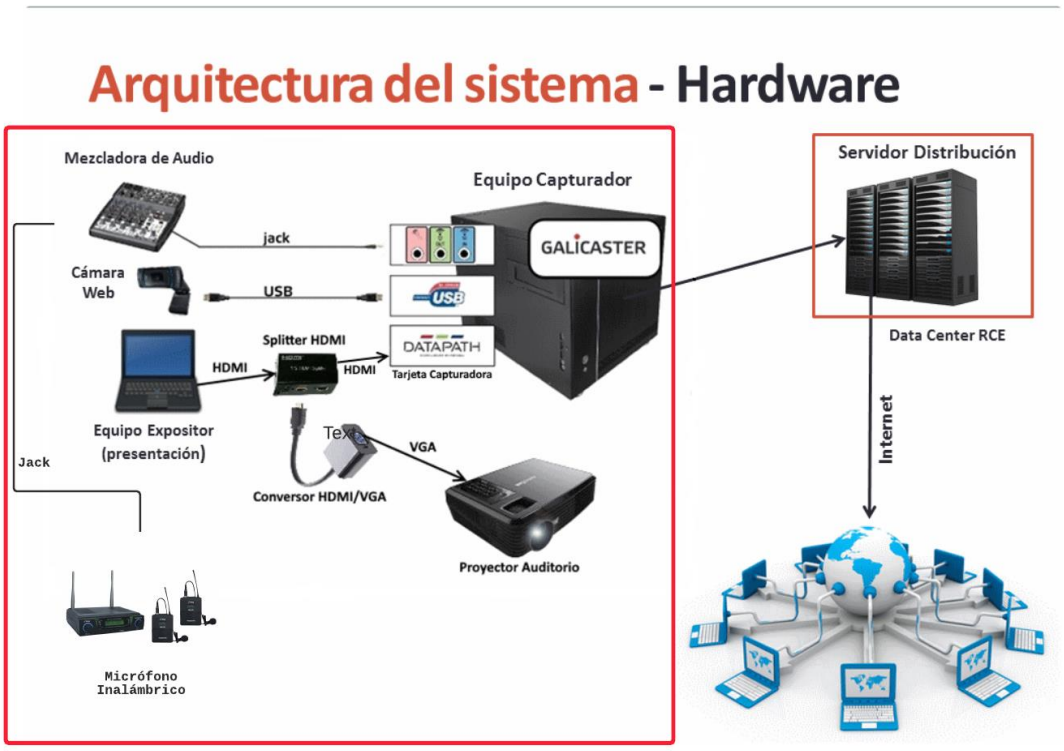


Figura 30: Diagrama de arquitectura de Hardware de Galicaster

Como se puede ver en la Figura 30, los componentes siguen la misma estructura que en el diagrama de la Figura 22, a excepción de la entrada de audio del computador del presentador.

Existe una limitación de Hardware. Los equipos de los presentadores solo pueden enviar audio ya sea por el cable HDMI o por la salida análoga de audio. El enviar audio por el HDMI permitiría la grabación en el sistema, pero no permitiría que la audiencia escuchase las grabaciones sonoras. Si se obtiene el audio por la salida análoga, este se podría introducir al

mezclador de audio, pero no sería un audio cómodo para la audiencia ya que la salida del mezclador estaría junto con la voz del presentador, lo cual además podría producir un efecto de retorno y una merma en la calidad de la grabación. Para este problema existen soluciones como utilizar mezcladores de audio más sofisticados o hubs de audio. Esto quedó fuera del alcance de este trabajo.

Tabla 8: Lista de componentes utilizados para la construcción del Agente de Captura Estacionario

Componente	Marca	Modelo	Cantidad
Computador			
Procesador	Intel	Core i5-4690	1
Placa madre	Gigabyte	GA-B85M-DS3H	1
Memoria RAM	Kingston	HyperX Fury HX318C10FB/4 (1x4 GB | DIMM DDR3-1866)	2
Disco duro	Western Digital	Blue 1000 GB (WD10EZEX)	1
Fuente de poder	Thermaltake	SMART SP-550P	1
Gabinete	Sentey	Stealth GS-6008	1
Teclado y mouse inalámbrico			1
Tarjeta capturadora	DataPath	VisionRGB 1es	1
Cámaras web	Logitech	c920	1
Audio			
Mesa de audio	--	--	1
Monitor táctil 19,5''	Asus	VT207N Touchscreen	1
Accesorios necesarios			
Cable HDMI	--	--	4
Adaptador HDMI-VGA	--	--	1
Splitter HDMI	--	--	1
Cable Extensión USB	--	--	1
Baterías recargables	--	--	2

Fuera de la configuración del computador utilizada, la cual se eligió en función de las recomendaciones del equipo desarrollador de Galicaster, para tener una grabación fluida y sin problemas, hay componentes que se eligieron en particular para asegurar la compatibilidad y la experiencia del usuario.

3.4.2.2.1 Cámara Logitech C920

Figura 31: Cámara Logitech C920

Esta cámara web de interfaz web se eligió debido a la gran cantidad de ejemplos y usos en ambientes de producción de Galicaster alrededor del mundo, además de ser recomendada por los propios desarrolladores por su bajo costo y poder llegar a grabar hasta una resolución de 1080p.

3.4.2.2.2 Capturadora de Vídeo Datapath VisionRGB 1ES

Figura 32: Tarjeta Capturadora Datapath Vision RGB 1ES

Esta tarjeta tiene una compatibilidad probada con Galicaster, permite tanto grabar sistemas análogos como video componente o señales VGA como señales digitales por DVI o HDMI. En digital puede capturar una resolución de hasta 1080p y 2048x1536 para señales análogas.

También provee señal en blanco (Si no hay señal de entrada entrega una señal de prueba). Lo cual es útil para que el sistema no se bloquee esperando que el dispositivo tenga señal y escala la resolución de la imagen en caso de que ocurra un cambio en esta.

3.4.2.2.3 Pantalla Táctil Asus VT207N

Figura 33: Monitor Asus VT207N

Es un monitor de 19.5 pulgadas de una resolución de 1600x900 con 10 puntos de toque táctil. La interfaz táctil de Galicaster fue diseñada para que fuera utilizada en monitores de pantalla táctil con el fin de que el usuario pudiera fácilmente iniciar y para la grabación sin tener que usar algún periférico adicional y simplificar el sistema.

3.4.2.2.4 Instalación física del Agente de Captura

La instalación de este equipo se hizo dentro del Auditorio Feick del Departamento de Electrónica de la Universidad Técnica Federico Santamaría, en la Casa Central ubicada en Valparaíso, Chile.

La implementación del sistema quedó de la siguiente manera:

Figura 34: Agente de Captura en auditorio Feick

De la imagen:

1. Sistemas de audio: Micrófono inalámbrico, Mezclador.
2. Computador Corriendo Galicaster.
3. Pantalla Táctil.

En la imagen siguiente se ve la posición donde quedó instalada la Cámara Web, la cual está sobre el proyector de la sala:

Figura 35: Cámara instalada en el auditorio Feick

Además, existe una UPS conectada al computador para evitar la pérdida de grabaciones que estén en curso si ocurre algún incidente con el suministro eléctrico.

3.4.2.3 Configuración de la arquitectura

Con el sistema ya configurado en sus interfaces de red, se procede a instalar Galicaster en su versión 1.4.2 y posterior configuración, para ellos se siguen estos pasos:

3.4.2.3.1 Descarga Galicaster

En primer lugar, es necesario descargar Galicaster el archivo .deb desde el siguiente enlace: https://wiki.teltek.es/display/Galicaster/Download+Galicaster

3.4.2.3.2 Agregar los repositorios

Antes de instalar el archivo recién descargado, se debe agregar el siguiente repositorio:

```
$ sudo add-apt-repository ppa:mc3man/trusty-media
$ sudo apt-get update
$ sudo apt-get install ffmpeg gstreamer0.10-ffmpeg
```

Y luego se ejecuta el archivo .deb previamente descargado. Con lo cual instala Galicaster en nuestro sistema.

3.4.2.3.3 Configuración del entorno de Galicaster

Para que Galicaster siempre pueda encontrar los dispositivos de grabación independiente de los cambios de hardware que pueda tener la máquina, se crean vínculos simbólicos con el punto de montaje de estos dispositivos en la configuración.

Se ejecuta la línea:

```
$ udevadm info --attribute-walk --name=${device}
```

Donde `${device}` es el punto de montaje del dispositivo a utilizar, por ejemplo, en el caso de la cámara Logitech C920 queda montada en `/dev/video0`

El comando udevadm nos devolverá una serie de atributos del dispositivo, luego se elige uno de esos atributos que sea único para el dispositivo (normalmente se utiliza el nombre), que fue el utilizado en este trabajo, este comando devolvió la clave:

```
ATTR={name}==”HD webcam pro c920”
```

Para la capturadora Datapath se hace el mismo procedimiento

Luego esto se crea el archivo `/etc/udev/rules.d/galicaster.rules`

Donde se introduce el siguiente texto, utilizando la clave dada anteriormente.

```
KERNEL=="video[0-9]*", ATTR{name}==" HD webcam pro c920", GROUP="video",
SYMLINK+="camera"
KERNEL=="video[0-9]*", ATTR{name}=="DATAPATH RGB ES1", GROUP="video",
SYMLINK+="screen"
```

Se reinicia el PC y los vínculos simbólicos quedan creados, luego estos vínculos se usarán en los archivos de configuración de Galicaster.

Adicionalmente y por un tema de permisos, se cambia la propiedad de las carpetas de Galicaster para que no ocurran errores de permisos:

```
$ sudo chown -R <user>:<group> /etc/galicaster
$ sudo chown -R <user>:<group> /usr/share/galicaster
```

Donde `<user>` es el nombre de usuario que ejecutará Galicaster, y `<group>` es el grupo al que pertenece el usuario (usualmente es el mismo nombre de usuario).

3.4.2.3.4 Creación de Perfiles de captura

Para que Galicaster funcione de manera satisfactoria, necesita que se configuren perfiles de grabación, estos perfiles le dicen al programa qué fuentes multimedia tiene que utilizar, la resolución de grabación y a qué pista pertenece al presentador o a la presentación.

Los perfiles se guardan en archivos de texto plano con extensión ".ini" dentro de la carpeta `/etc/galicaster/profiles/`

El perfil creado para su uso por defecto es uno que captura tanto la presentación como al presentador.

```
[data]
name = Screen-Cam-Mixer

[track1]
name = Webcam
device = v4l2
location = /dev/camera
file = WEBCAM.avi
flavor = presenter
caps = image/jpeg,framerate=24/1,width=1280,height=720

[track2]
name = AudioSource
device = pulse
location = alsa_input.pci-0000_00_1b.0.analog-stereo
file = sound.mp3
flavor = presenter
vumeter = True
player = True
amplification = 1.0

[track3]
Name = Slides
device = datapath
location = /dev/screen
file = SCREEN.avi
flavor = presentation
caps = video/x-raw-yuv,framerate=30/1,width=1920,height=1080
```

El archivo de configuración de perfil tiene "pistas" (Tracks), para cada flujo multimedia, así se tiene un flujo para el Audio del presentador, el vídeo del presentador y la señal de la captura de pantalla del computador del presentador.

El detalle de cómo se arman estos archivos se puede ver en el sitio web del desarrollador y en el Anexo 7.2 se encuentra una serie de perfiles de ejemplo que se utilizaron en el trabajo durante la etapa de implementación para comprobar el funcionamiento del equipo.

3.4.2.3.5 Conexión con Opencast

Para que la comunicación entre Galicaster y Opencast funcione de manera adecuada, hay que hacer la comunicación entre los puntos REST de cada programa.

Para lograr esto, se necesita editar el archivo "`config-dist.ini`" ubicado en `/usr/share/galicaster/config-dist.ini` modificando las siguientes líneas:

```
[ingest]
active = false
manual =  none
scheduled = none
```

```
host = http://fakeadmin.matterhorn.com:80
```

Por las siguientes:

```
[ingest]
active = true
manual =  immediately
scheduled = immediately
username = matterhorn_system_account
password = CHANGE_ME
host = http://[ip o URL  del servidor opencast ]:[Puerto del servidor opencast]
```

Como se puede ver en las líneas de configuración, se activa el sistema de ingesta de vídeos, luego tanto la ingesta manual (Que es cuando se activa la grabación de manera directa en Galicaster), como la ingesta programada (Scheduled), si se van a enviar las grabaciones inmediatamente después de que estas terminen o programadas a una hora definida previamente.

Un aspecto importante es el nombre de usuario y la contraseña, estos deben ser exactamente iguales que en los valores de configuración `org.opencastproject.security.digest.user` y `org.opencastproject.security.digest.pass` respectivamente que se configuraron en el punto 3.4.1.3.3.

Con esto últimos parámetros, Galicaster ya está preparado para grabar y enviar los flujos multimedia y recibir las programaciones de eventos de Opencast.

3.4.3 Agente de captura móvil

Gracias al apoyo del fondo I+D+i del Programa de Iniciativas Estudiantiles Académicas (PIE>A), se obtuvieron recursos para la construcción de un agente de captura móvil.

El agente de captura estacionario tiene la problemática de que no se puede mover hacia otro salón de la universidad, complicando a las personas que quieran utilizar este elemento en sus clases o eventos. Con el sistema de captura móvil se busca que la plataforma pueda ser usada en otras salas y auditorios de la universidad, logrando grabar y publicar más actividades que se generen dentro de la institución.

3.4.3.1 Elección de la plataforma de captura

La elección de la plataforma, al igual que en el módulo estacionario consiste en el mismo esquema de diseño, el gran cambio es la versión del sistema operativo, que pasa de Ubuntu 12.04 a la última versión estable 16.04 y Galicaster a la versión 2.0 RC1.

La elección de esta nueva versión del programa sobre la anterior radica en la elección del Hardware de captura adquirido para la construcción del equipo, que se habla a continuación:

3.4.3.1.1 Hardware utilizado:

El hardware elegido para la puesta en marcha del agente móvil fue la siguiente:

Tabla 9: Componentes a utilizados para agente de captura móvil

Componente	Marca	Modelo	Cantidad
Computador			
Procesador	Intel	Procesador Intel® Core™ i3-6100	1
Placa madre	Asus	Placa Madre Asus H110M-A-D3	1
Memoria RAM	Kingston	Memoria Ram DDR3 4GB	1
Disco duro	Western Digital	Blue 1000 GB (WD10EZEX)	1
Fuente de poder	Thermaltake	SMART SP-550P	1
Teclado y mouse inalámbrico	Logitech	Teclado Logitech K400 Inal. C/Touchpad	1
Tarjeta capturadora	Blackmagic	Decklink Mini Recorder	1
Cámaras web	Logitech	c920	1
Audio			
Mesa de audio	AM	AM 120 MKII Mixer Análogo	1
Micrófono Inalábrico	SKP	Microfono VHF 0755 inal. Solapa	1
Monitor 22”	ViewSonic	Monitor ViewSonic VA2261-2	1
Accesorios necesarios			
Cable HDMI	--	--	4
Adaptador HDMI-VGA	--	--	1
Splitter HDMI	--	--	1
Cable Extensión USB	--	--	1

Mueble de Transporte	--	--	1
Accesorios Mueble	--	--	Varios

Las diferencias que tiene, el agente móvil y el estacionario destacan:

- **Cambio de pantalla táctil por pantalla estática:** Esta decisión ocurre por restricciones de presupuesto y por la experiencia de uso de la pantalla táctil, la cual no ha jugado un valor relevante en la interacción de la interfaz de usuario del agente de captura estacionario.
- **Capturadora Blackmagic Design Decklink Mini:** Esta capturadora de la marca Blackmagic, tiene la gran ventaja de ser de un valor mucho menor (200 USD) [32] comparado con el precio (782 USD) de la tarjeta de Datapath [33]. Esta capturadora tiene las mismas ventajas que la tarjeta Datapath, a excepción de enviar una señal piloto en caso de que no existan entradas, hacer escalado de manera automática si hay un cambio de resolución y una compatibilidad más restringida con equipos que se conectan en la interfaz HDMI.

Figura 36: Blackmagic Design Mini Recorder

- **Mueble contenedor:** Un mueble diseñado a medida para el transporte de todo el agente de captura.

3.4.3.2 Diseño mecánico del agente

Para poder transportar el agente de captura, se diseñó un mueble que fuera capaz de transportar el equipo completo y tuviera la capacidad de ser desplegado como guardado de manera simple

y rápida, además de sólo dar las entradas y salidas necesarias para su conexión sin complicar al usuario más allá del uso del equipo.

El primer diseño consistió en un mueble, donde la pantalla de tipo táctil estaba anclada al mueble sobre un mástil. La cámara de grabación sobre el monitor, tuviera ruedas para el desplazamiento del agente de captura y cajones para separar el equipamiento de audio, el equipo computacional y los accesorios.

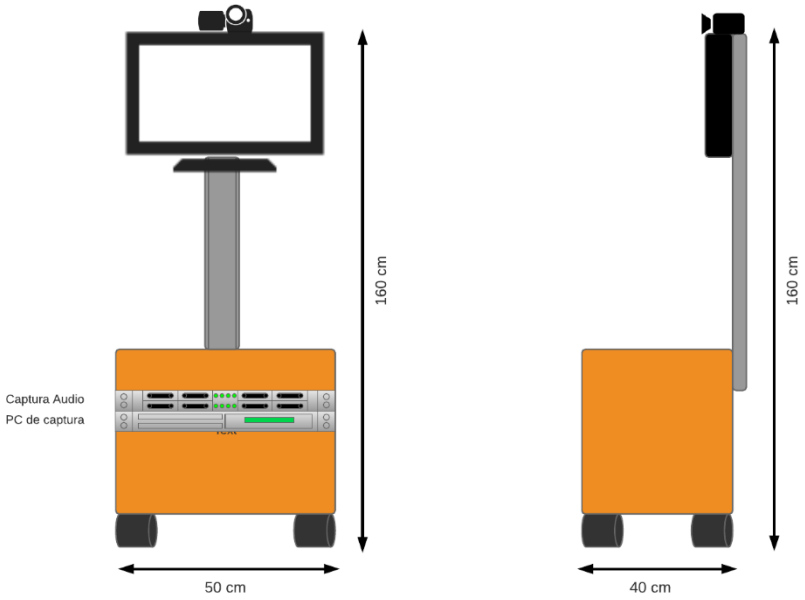


Figura 37: Primer diseño del agente de captura móvil

Luego de discutir con distintos académicos de la universidad y estudiando las salas de clases que existen en la UTFSM (Los auditorios no son planos y no permiten que la cámara sujeta al equipo pueda hacer una toma completa), se decide por hacer la cámara extraíble utilizando un trípode para poder colocarla en cualquier lugar de la sala. Esta se guardará en un espacio especial en un costado del mueble.

El modelo resultante de la retroalimentación obtenida fue el siguiente:

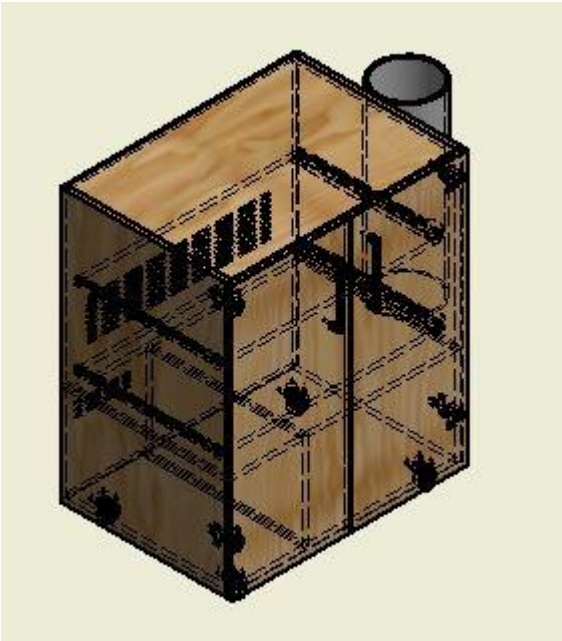


Figura 38: Diseño versión 2 módulo de captura móvil

Figura 39: Vista frontal y lateral módulo de captura móvil V2

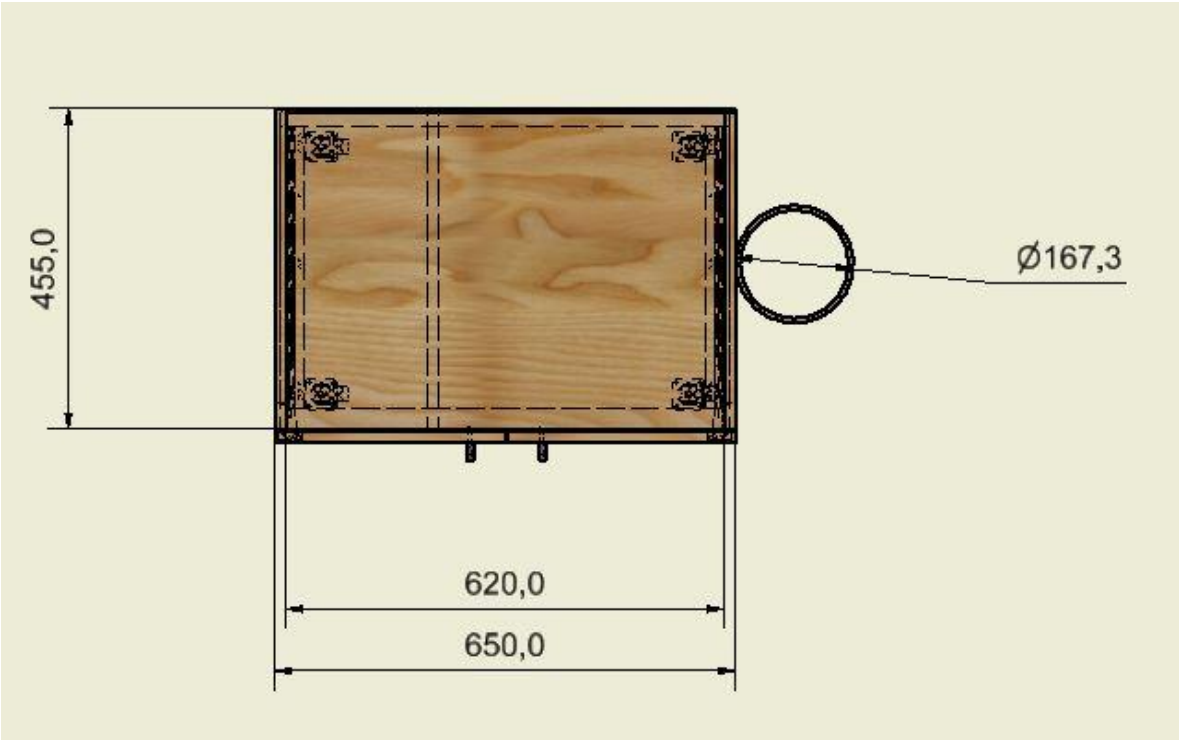


Figura 40: Vista superior Módulo de captura V2

Los cajones que se tenían pensado en la primera iteración son reemplazados por bandejas móviles, para facilitar la mantención del equipo y simplificar la construcción, el tubo lateral fue pensado para guardar el trípode con su cámara, además de facilitar el almacenamiento de este.

Los planos completos se encuentran en el anexo 7.7.

3.4.3.3 Esquema de configuración

3.4.3.3.1 Construcción del diseño

El diseño fue construido utilizando madera de terciado mueblista y conforme a los planos mostrados en la sección anterior.

Figura 41: Agente de Captura, vista exterior, sin pintar

La parte interior del contenedor, tiene cuatro separaciones, una bandeja para los sistemas de captura de audio, una bandeja para los sistemas computacionales, Un espacio para colocar una UPS y otro espacio para las conexiones eléctricas y para almacenamiento misceláneo.

Figura 42: Agente de Captura, Vista Interior, Sin pintar

Una vez acabada la construcción se procede a dar un acabado en barniz para proteger la madera de la humedad y dar una mejor terminación.

Figura 43: Proceso de barnizado Agente de Captura

Finalmente se instalan los equipos:

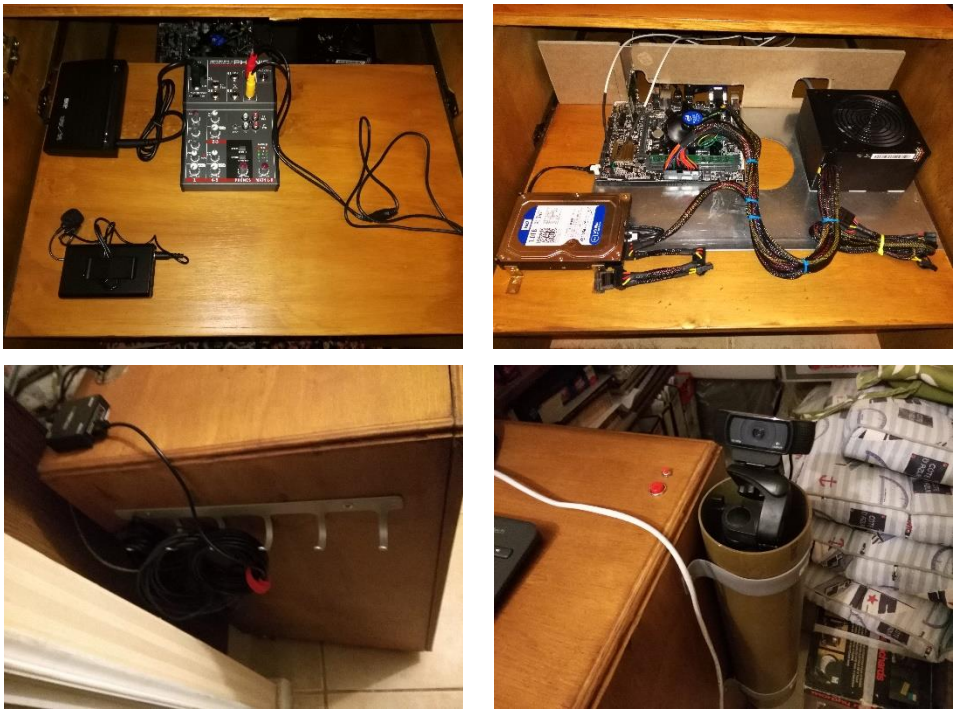


Figura 44: De izquierda a derecha: Bandeja de audio, bandeja de PC, ordenadores de cable, cámara de grabación.

Otros aditamentos que tiene el equipo:

- Rejillas de ventilación.
- Botón de encendido y reinicio en la parte superior del agente.
- Llave de seguridad para evitar el ingreso de personal no autorizado a los equipos.
- Ruedas de desplazamiento con freno.

El Agente de Captura terminado:

Figura 45: Cámara usando trípode. Distribución interior del módulo

Figura 46: Módulo de captura armado

Por tema de presupuesto, a este equipo no se pudo agregar una UPS y el espacio asignado se utiliza como lugar de almacenamiento. La configuración de software es la misma del agente estacionario, por tanto, los mismos comandos son aplicables para este caso.

4 PRUEBAS DEL SISTEMA

4.1 DESCRIPCIÓN DEL AMBIENTE DE PRUEBAS

El ambiente de pruebas utilizado para validar el sistema es el siguiente:

- Se utilizaron los dispositivos y configuraciones descritos en el capítulo anterior.
- Los equipos no estaban bajo una carga de procesamiento al realizar las pruebas más allá de los programas involucrados en las tareas dadas.
- La comunicación entre los equipos se hizo dentro de la red del Departamento de Electrónica de la universidad.
- Se realizaron las pruebas en altitud del mar en la zona de Valparaíso, Chile.

4.2 PRUEBAS PARA USUARIO

4.2.1 Grabación utilizando agente de captura estático

Lo primero es conectar el computador de presentaciones al splitter HDMI:

Figura 47: Conexión del PC presentador al agente de captura

Luego Encender el proyector de la sala:

Figura 48: Encendiendo proyector de la sala

El paso siguiente es encender los equipos de audio e instalar el micrófono inalámbrico al usuario:

Figura 49: Izq: Encendido de equipos de audio; Der: Instalación de micrófono al presentador

Una vez preparados los equipos se enciende la pantalla, donde aparecerá el menú de Galicaster, se selecciona grabar, el equipo comenzará inmediatamente.

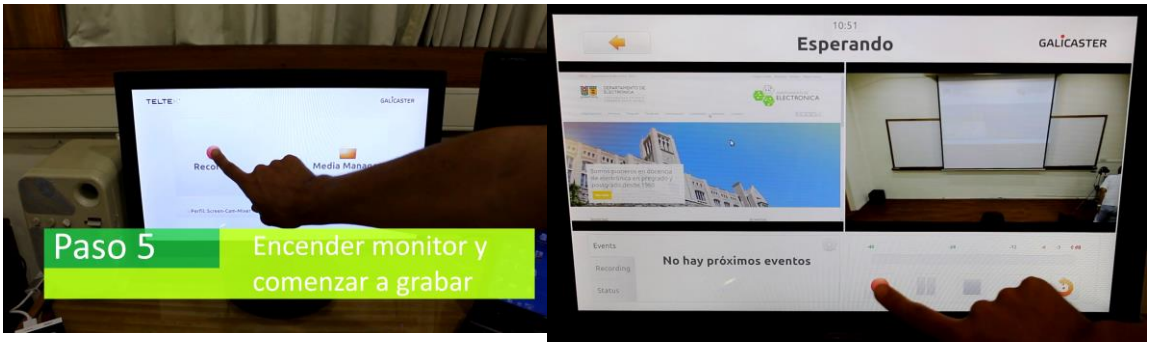


Figura 50: Inicio de grabación en Galicaster

Figura 51: Galicaster en proceso de grabación

Si se desea se pueden hacer pausas momentáneas sin detener la grabación:

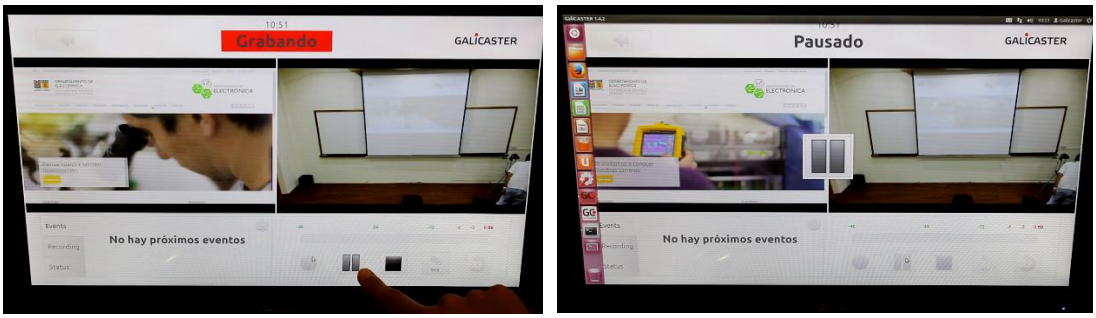


Figura 52: Acción de pausar en Galicaster

Una vez finalizado, se presiona el botón Stop, el cual dará una advertencia de confirmación:

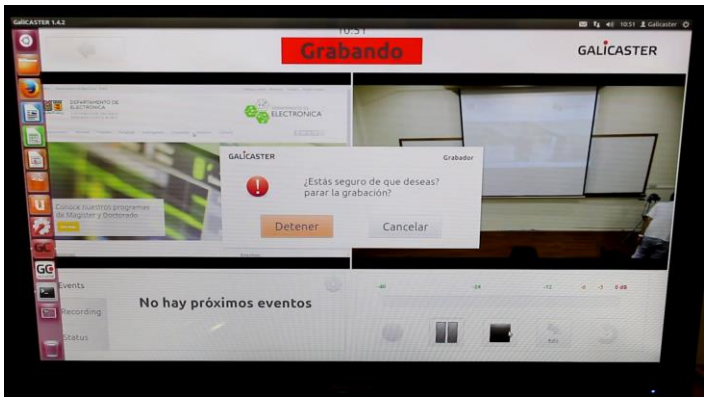


Figura 53: Confirmación de detención de grabación en Galicaster

En la siguiente imagen se puede ver que Galicaster queda listo para grabar en unos pocos segundos luego de haber terminado la grabación anterior:

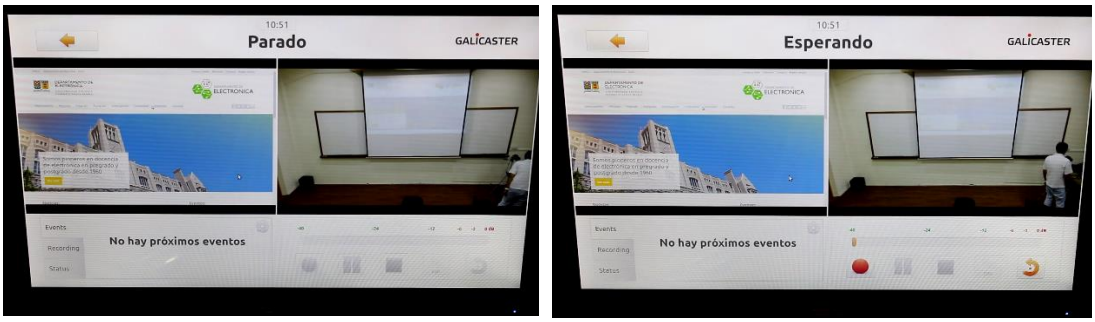


Figura 54: Preparación de Galicaster para una nueva grabación

Finalmente se desconectan y apagan los equipos (Exceptuando el computador):

Figura 55: Apagado de sistemas de audio y microfonía

Con esto finaliza el uso de grabación por parte del usuario, la grabación es enviada al servidor de difusión donde es procesada y publicada de manera automática.

4.2.2 Grabación utilizando agente de captura móvil

El proceso de grabación con el agente de captura móvil es prácticamente igual, solamente tiene algunas diferencias específicas que se darán a continuación:

El módulo móvil no cuenta con pantalla táctil, por tanto, debe utilizar el teclado y mouse (Incluídos) para seleccionar los menús que aparecen en pantalla.

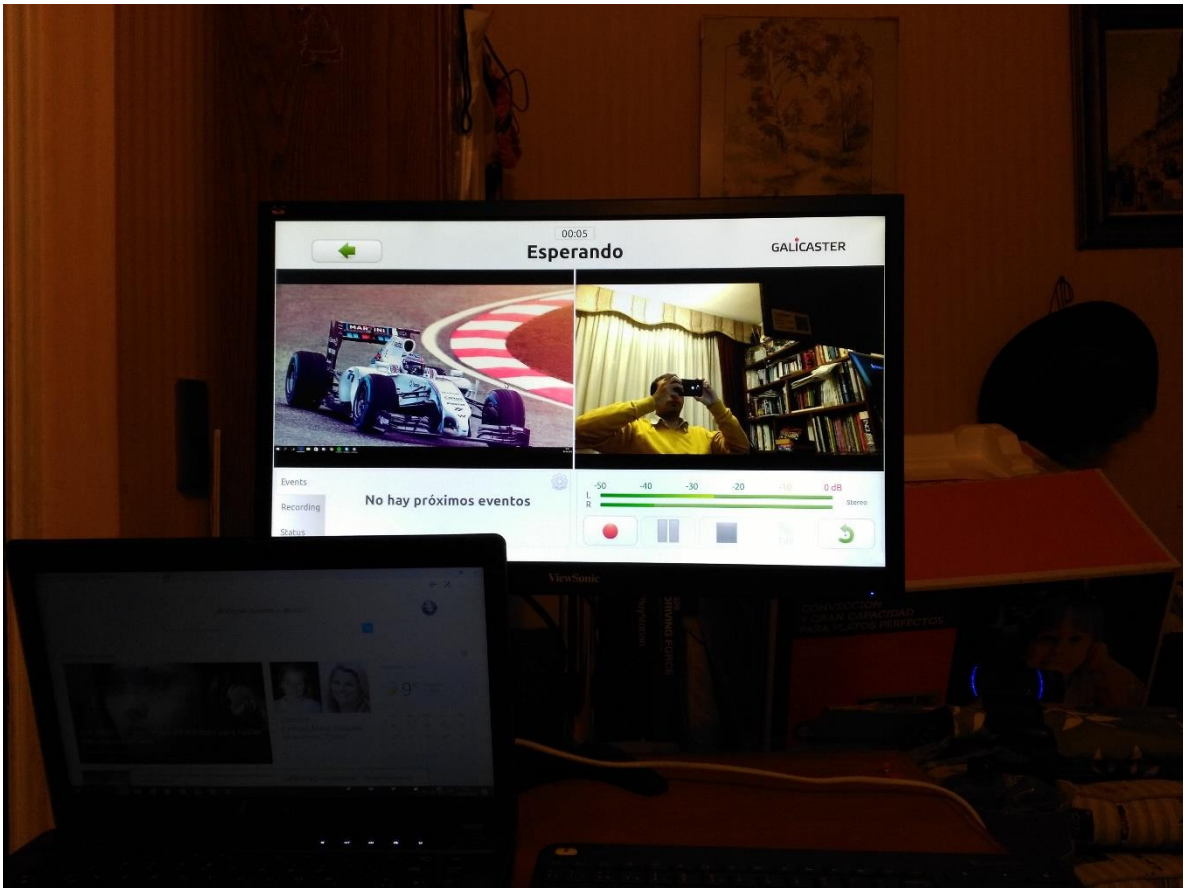


Figura 56: Agente de Captura Móvil en funcionamiento

Como se aprecia en la Figura 56, el módulo móvil está listo para grabar, utiliza exactamente las mismas conexiones y pasos del agente estacionario.

Cabe destacar que ocurre un problema con la tarjeta capturadora y el splitter HDMI, estos dos elementos son incapaces de comunicarse entre sí, no enviando señal al computador. Este problema radica en que tanto como la capturadora, como el splitter no son capaces de hacer el "Handshake" necesario para el envío de vídeo. Existen varias ideas de cómo solucionar este problema, una de ellas es utilizar la entrada SDI de la capturadora y un conversor HDMI-SDI, lamentablemente, este último dispositivo llegó defectuoso, por tanto, no se resolvió este inconveniente. El agente de captura móvil, al momento de la escritura del trabajo, sólo puede capturar sin replicar la imagen a un proyector.

Figura 57: Cámara grabando

Si es necesario, el equipo permite que la cámara grabe desde su lugar de almacenaje sin tener que sacarla de su posición.

Finalmente, como este es un equipo móvil, necesita conexión a la red para enviar la grabación al servidor de difusión. El módulo no necesita conexión durante las grabaciones. Para enviar la grabación se ha colocado un cable de red de 10 metros, con tal que personal que retire el módulo móvil pueda conectarlo y dar la orden de enviar la grabación.

4.3 PRUEBAS PARA ADMINISTRADOR

Las pruebas siguientes se hacen desde que el administrador entra al sitio de administración de Opencast, empezando por la primera pantalla que aparece luego de entrar en el sistema.

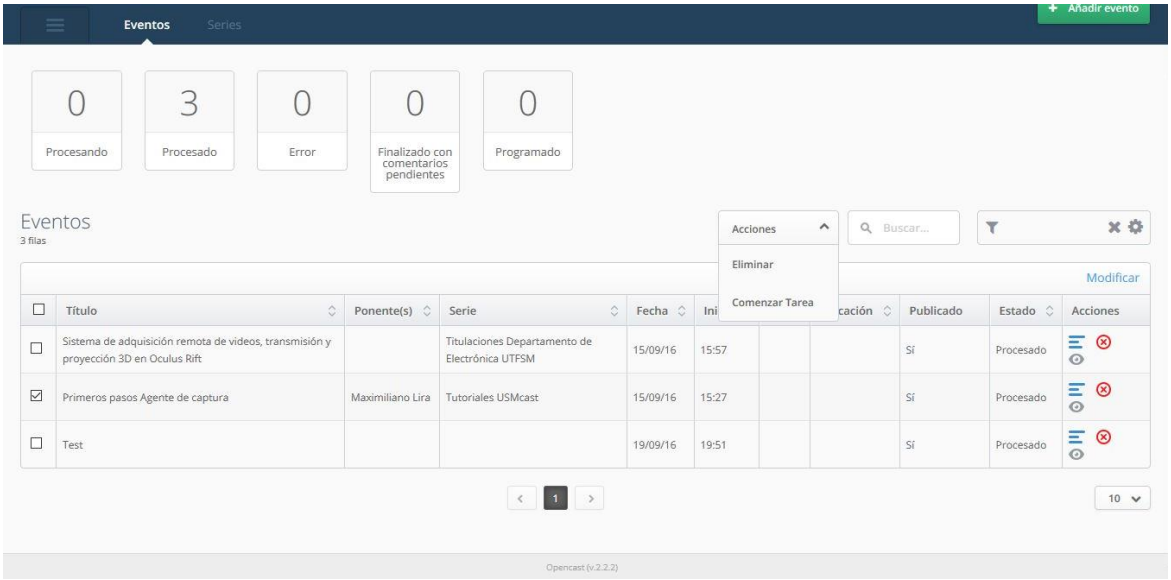


Figura 58: Panel de admintración de Opencast

4.3.1 Ingreso de evento

Para ingresar un evento existen 4 formas:

1. Desde el Agente de Captura: Simplemente se hace una grabación no programada.
2. Subiendo la grabación manualmente.
3. Programando un evento único.
4. Programando una serie de eventos que ocurrirán de manera periódica durante un lapso de tiempo establecido.

Dicho esto, la primera forma se vio en el punto anterior. En el caso de hacerlo desde el panel administrativo de Opencast es de la siguiente forma:

Se hace clic en el botón llamado “+ Añadir Evento” ubicado en la esquina superior derecha de la ventana, aparecerá el siguiente cuadro:

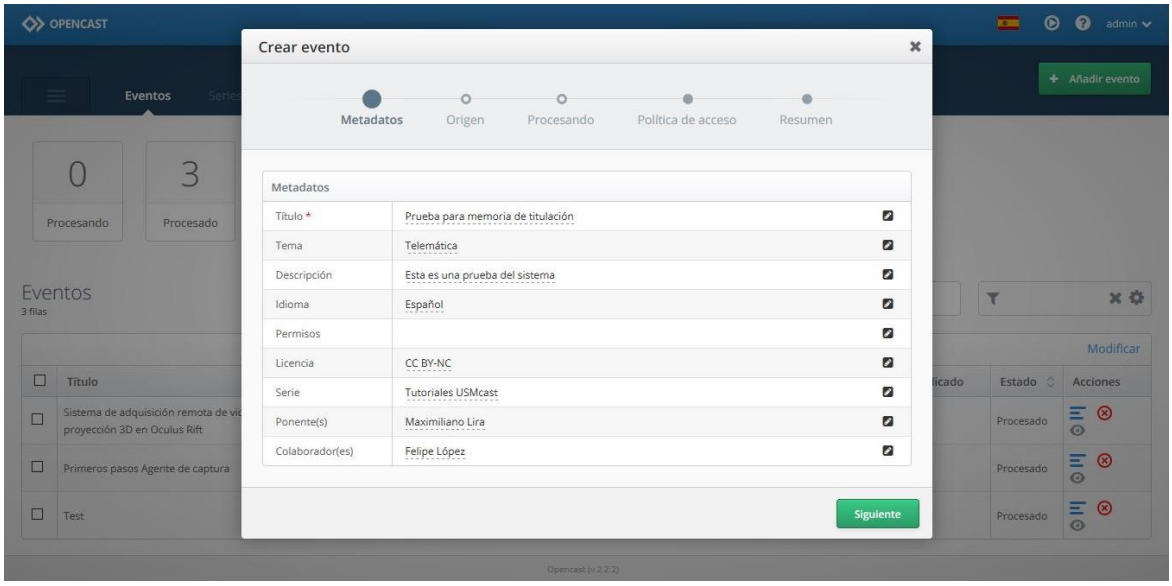


Figura 59: Ventana de metadatos al crear un evento

Este cuadro pide los metadatos del vídeo, Título de la presentación, tema a tratar, descripción, Permisos, Licencia (Con derechos reservados, Creative Commons, otra), a qué serie pertenece, nombre del ponente y colaboradores.

La Serie es un componente opcional, permite dejar una cortina para abrir y/o cerrar la grabación y clasificar en el sistema a todos esos vídeos en una misma categoría. Muy útil si se quiere clasificar las grabaciones en Charlas, Clases de algún ramo en particular u otro tipo de vídeos, por ejemplo.

Al presionar el botón siguiente aparecerá la siguiente ventana:

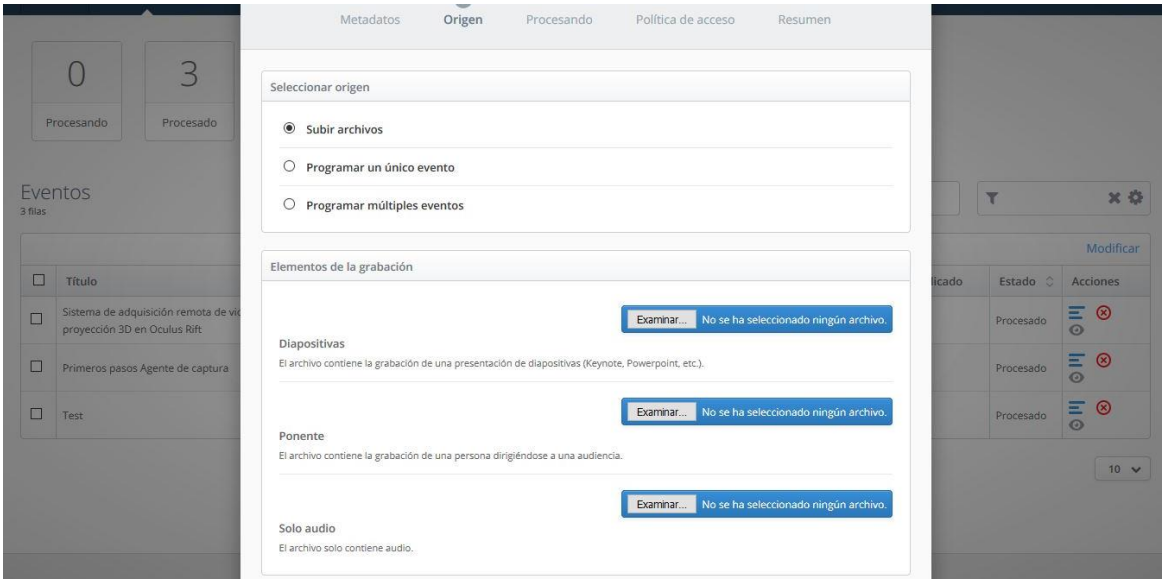


Figura 60: Ventana de origen al crear un evento "Subir Archivos" seleccionado

En esta ventana se pueden escoger las tres opciones que se comentaban anteriormente, en la imagen en particular está seleccionada la opción "Subir Archivos". Esta opción permite subir tanto la grabación de las diapositivas (O captura del computador del presentador), la captura del ponente y un canal de audio. El canal de Audio se usa en el caso de que no se haya hecho grabación de vídeo del ponente.

En el caso de que se elija la opción "Programar un único evento" aparecen las siguientes opciones:

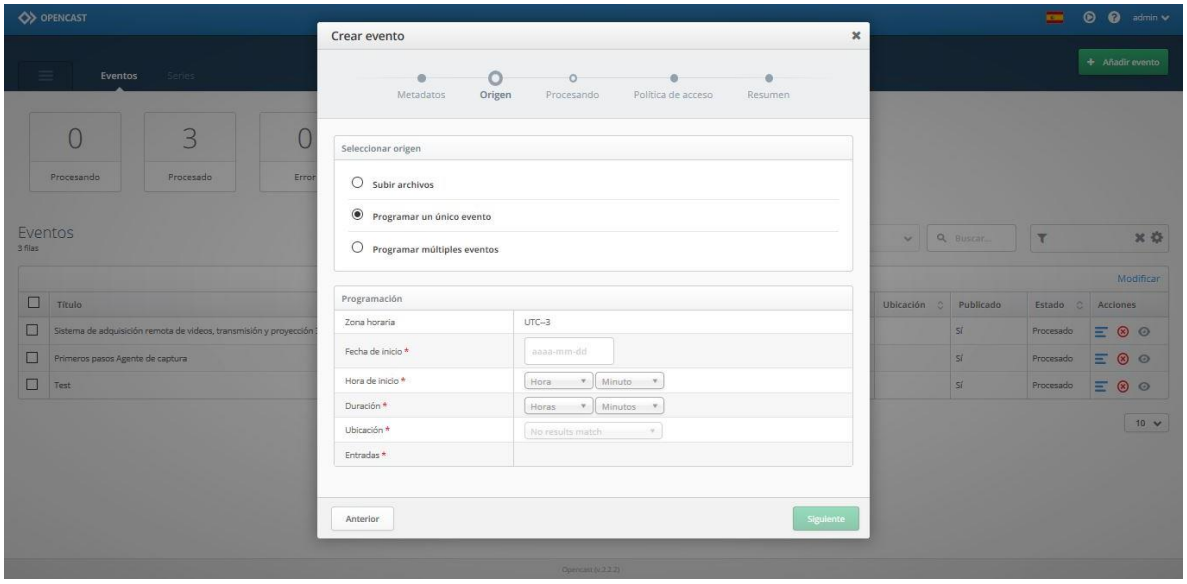


Figura 61: Ventana de origen al crear un evento "¨Programar un único evento" seleccionado

La fecha del evento, la hora de inicio de este evento, la duración y la ubicación.

La zona horaria está dada por la hora del servidor, mientras que la ubicación es el nombre que se le ha asignado al agente de captura, esto es importante para notificar al agente de captura cuando tiene que empezar a grabar.

Si se elige la opción "Programar múltiples eventos" aparece este menú:

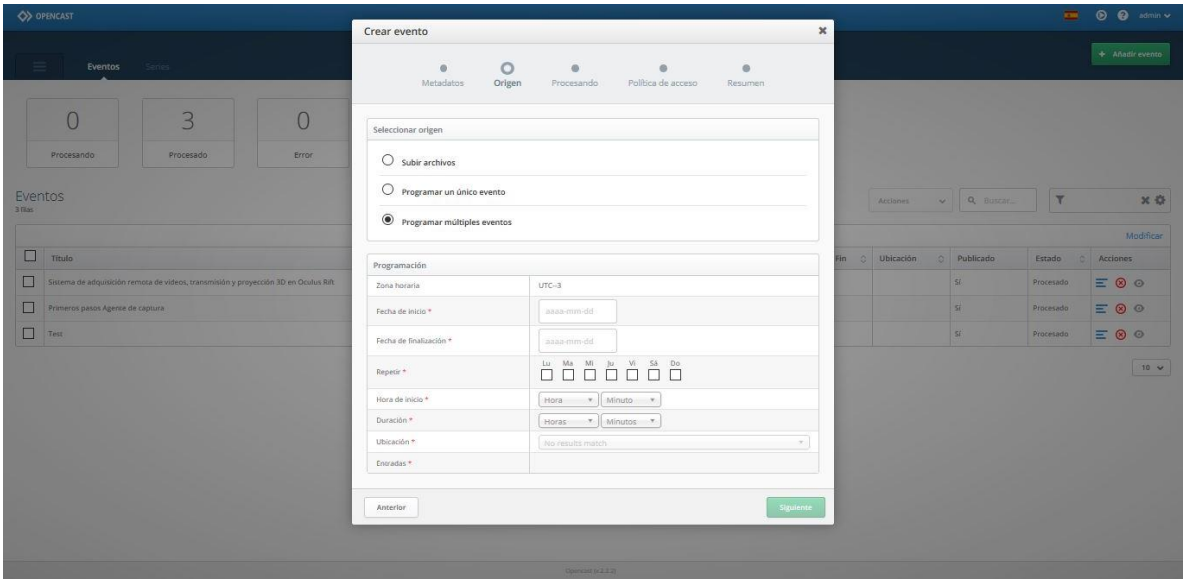


Figura 62: Ventana de origen al crear un evento "¨Programar múltiples eventos" seleccionado

Es prácticamente igual que el anterior, pero agrega los controles de fecha de inicio y finalización, que días de la semana debe ser repetido.

Presionando siguiente (Indiferente de la opción que se haya elegido aparecen los controles de procesado de la grabación.

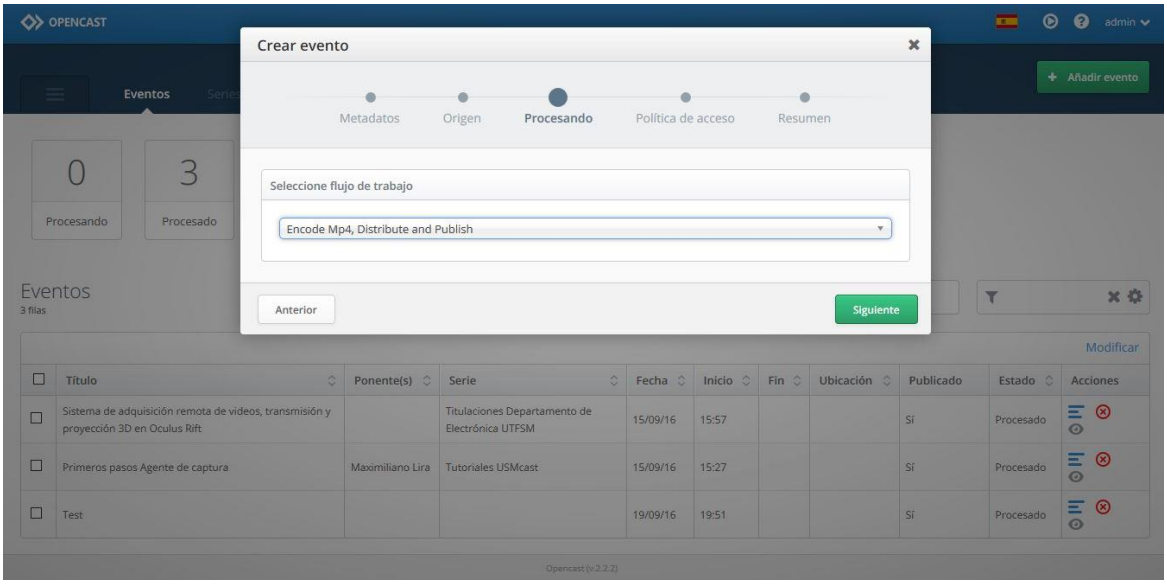


Figura 63: Ventana de procesado al crear un evento

En esta ventana se elige el flujo de trabajo a utilizar. El flujo de trabajo (También conocido como workflow), es una parte muy importante de Opencast, le indica al servidor todos los pasos que debe hacer con el vídeo y a qué herramientas llamar en el paso de procesamiento de los datos.

La configuración de los flujos de trabajo se ve en el anexo 7.2 de este documento.

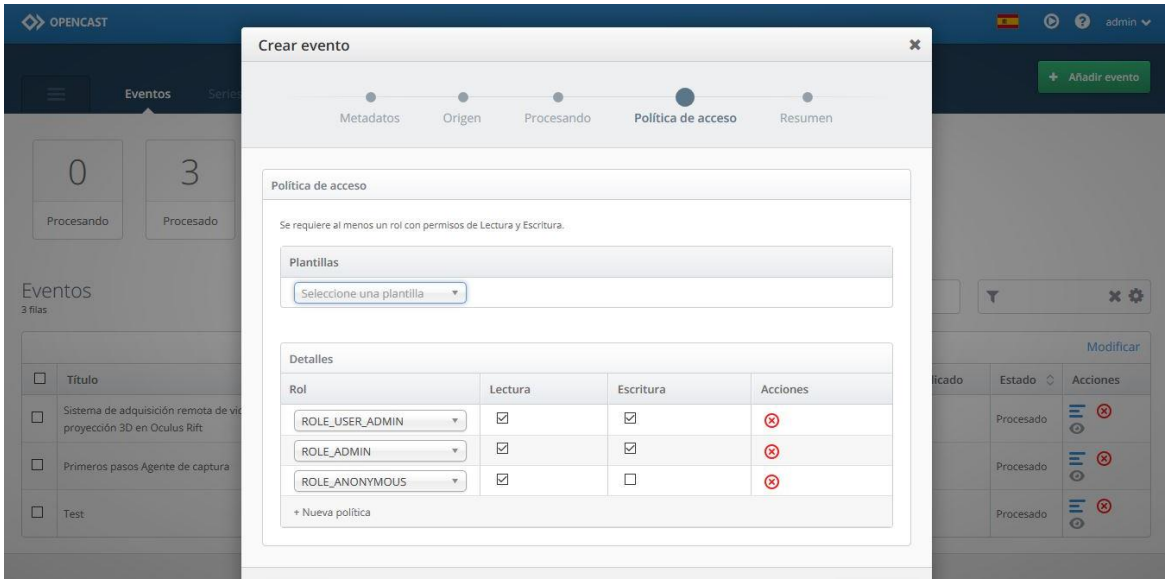


Figura 64: Ventana de política de acceso al crear un evento

En la Figura 64 se configuran los permisos a la o las grabaciones a subir/previstas, se pueden crear distintos niveles de control de acceso y grupos de usuarios para que puedan acceder a distintos vídeos. En el caso de este trabajo solo se utilizaron dos perfiles, el perfil de administrador y el perfil de usuario anónimo.

Finalmente aparece una ventana que resume todas las entradas ingresadas para la creación del evento, una vez apretado el botón “Terminar”, comienza el proceso de edición del evento ingresado (Si es un evento que se subió inmediatamente) o esperará hasta que un agente de captura le envíe la grabación.

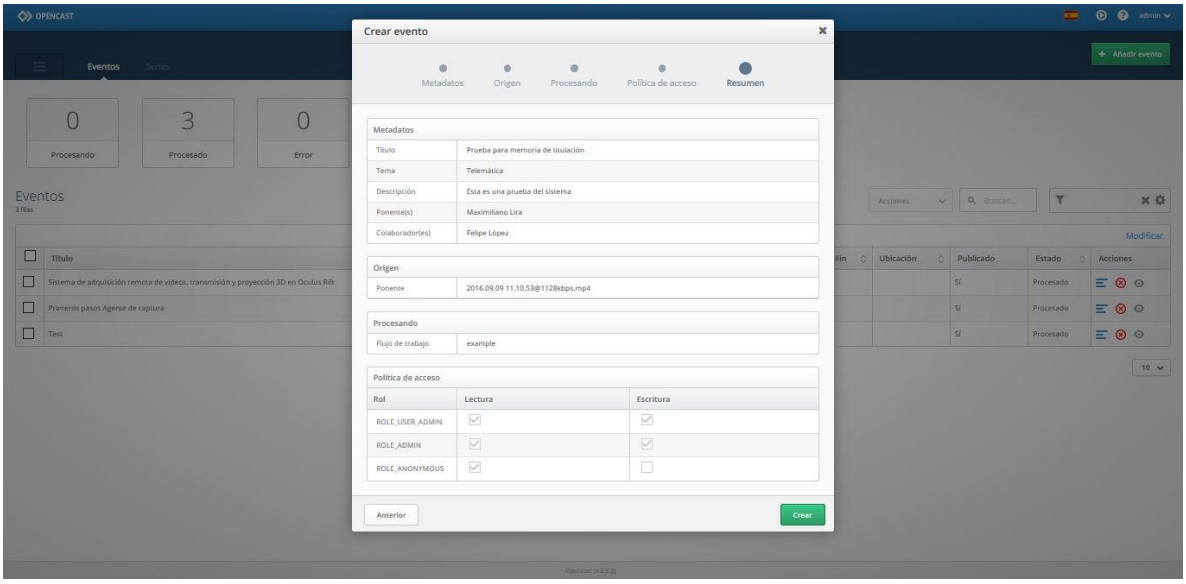


Figura 65: Ventana de resumen al crear un evento

4.3.2 Edición de evento ya ingresado

Una vez que un evento fue ingresado y procesado, se pueden hacer varias tareas adicionales, como hacer edición de la grabación, republicar o eliminar de la publicación.

4.3.2.1 Edición de la grabación:

En el menú principal del panel del administrador, existe un botón para la vista previa de cada una de las grabaciones y la edición de las mismas, que tiene la forma de un ojo y se encuentra en el lado derecho de cada grabación, al hacer clic en ese icono, aparece la siguiente pantalla:

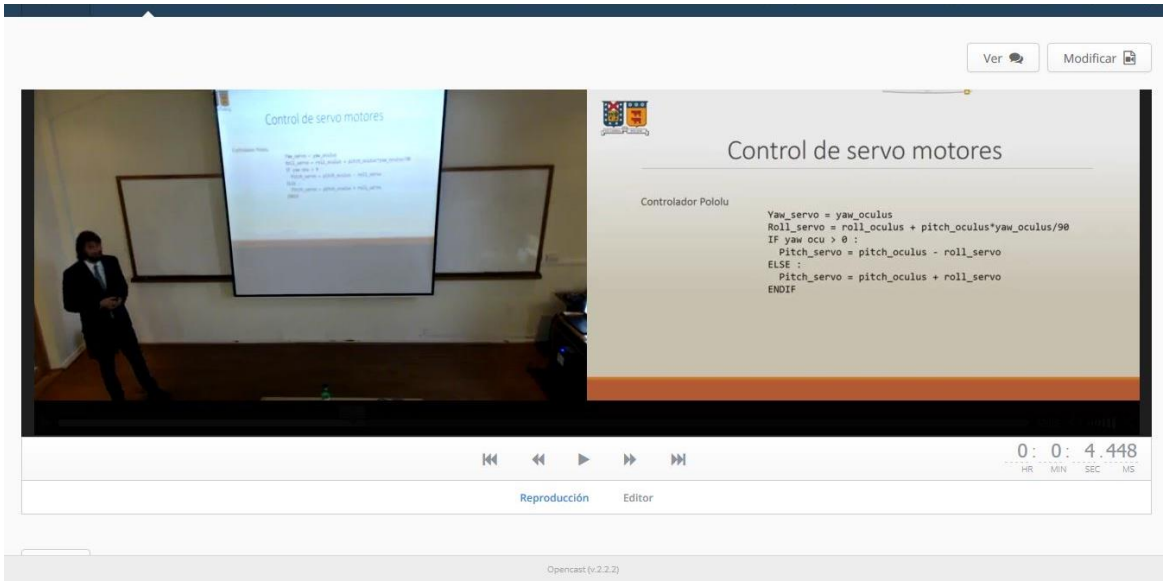


Figura 66: Modo de revisión y edición en Opencast

La Figura 66 muestra una pantalla de reproducción de la grabación que ya fue indexada por el sistema, en el cual permite verla por completo. Si se hace clic en el vínculo llamado “editor” se lleva a la siguiente pantalla:

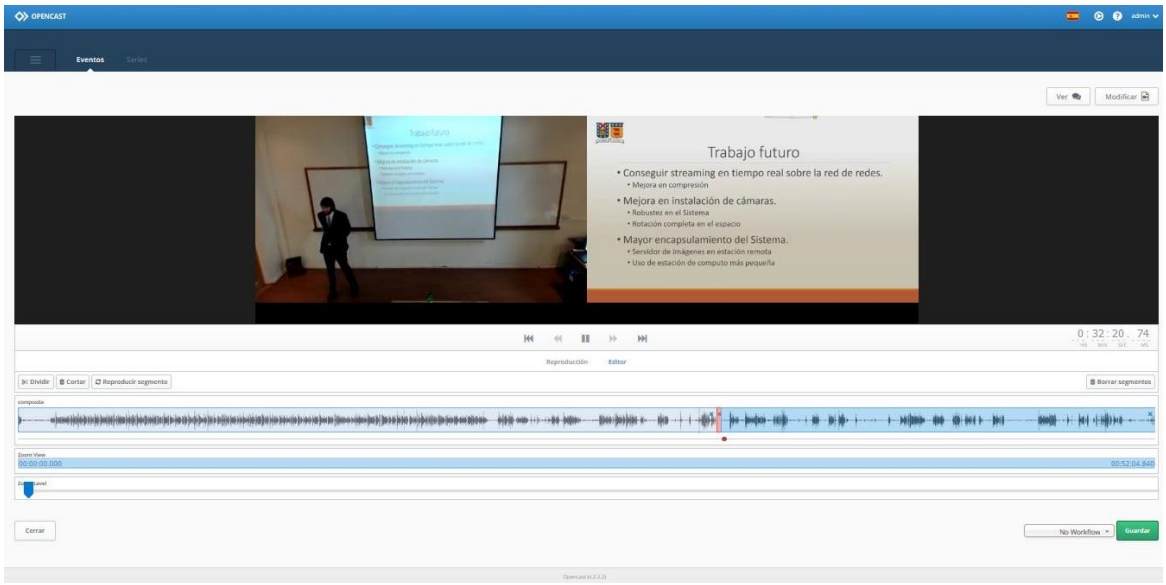


Figura 67: Panel de edición de grabaciones en Opencast

La Figura 67 muestra el panel de edición de Opencast, el cual permite hacer ediciones básicas al vídeo, las cuales son sólo cortar y seleccionar.

Adicionalmente cuenta con un sistema para agregar comentarios internos, esto es útil cuando existe más de un administrador y para llevar un registro de los cambios hechos o que hay que hacer a la grabación. También está pantalla permite hacer la edición de los metadatos a través del botón “modificar”. Para guardar los cambios hechos, simplemente se debe presionar el botón “Guardar” y elegir el flujo de trabajo deseado para que se encargue de la recodificación de la grabación editada.

4.3.2.2 Re-publicación, eliminación de las grabaciones y edición de los metadatos

Estos dos aspectos son muy importantes, el primero permite que los cambios hechos a los metadatos se puedan re-publicar tanto a YouTube como al módulo de medios de Opencast.

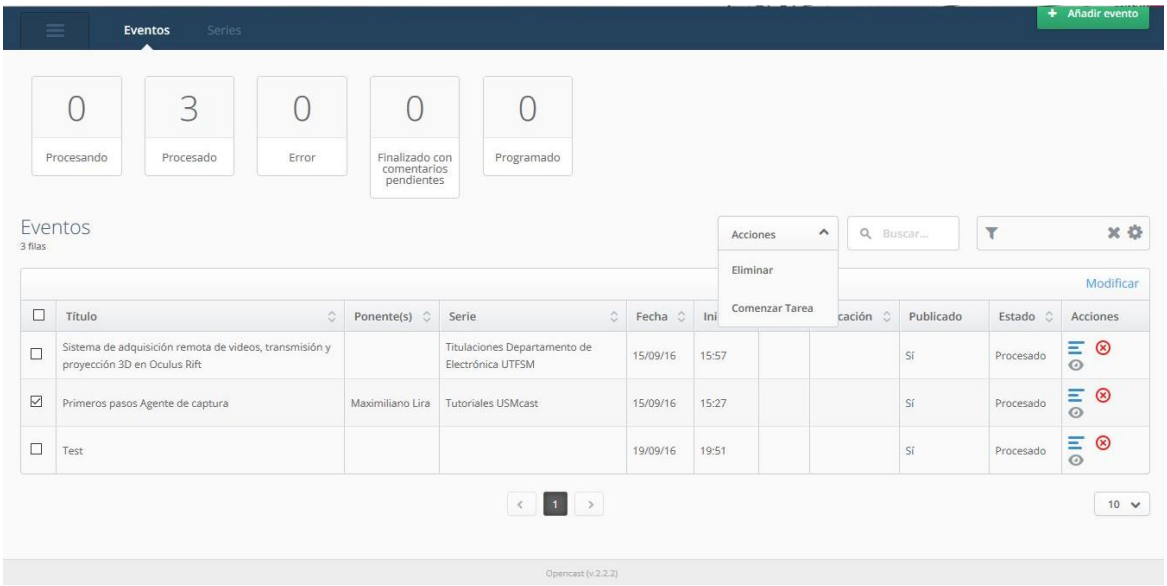


Figura 68: Selección de tareas para grabaciones

Primero hay que modificar los metadatos, para ello, se presiona primero en el ícono de las 3 barras azules, que se encuentra en la columna “Acciones” de cada grabación. Luego, se debe hacer como en la Figura 68, en la cual se selecciona una de las grabaciones (o las grabaciones) que se editó (editaron) y a continuación, se abre el menú de “Acciones”, para hacer clic en “Comenzar Tarea”.

Aparece la siguiente ventana:

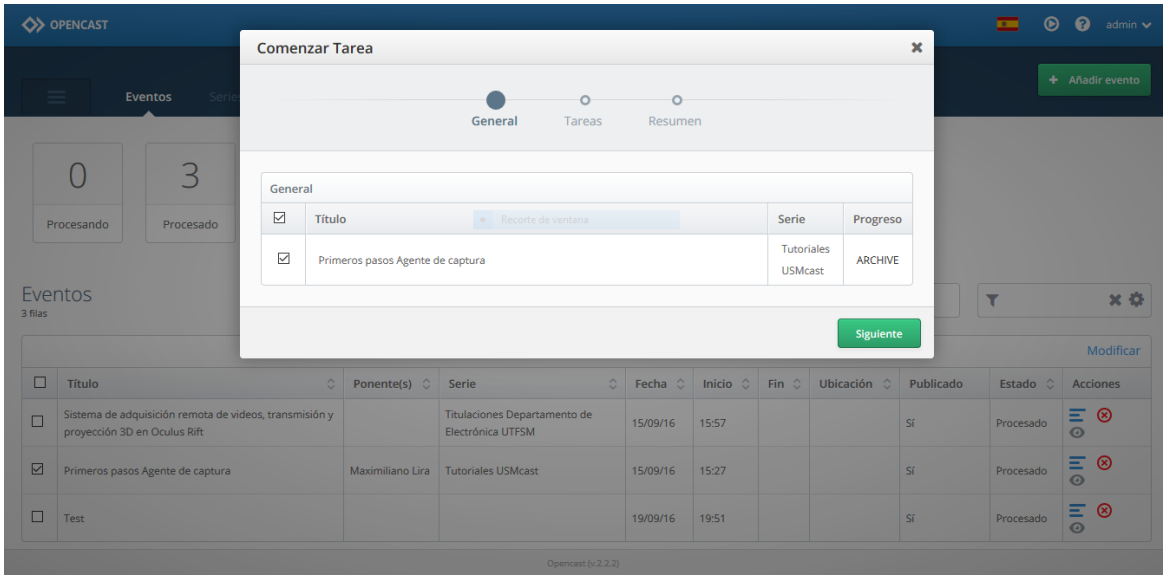


Figura 69: Confirmación de grabaciones a modificar

En la Figura 69, se confirman la o las grabaciones a editar, la ventana muestra a qué serie y el estado en que se encuentra la grabación actualmente.

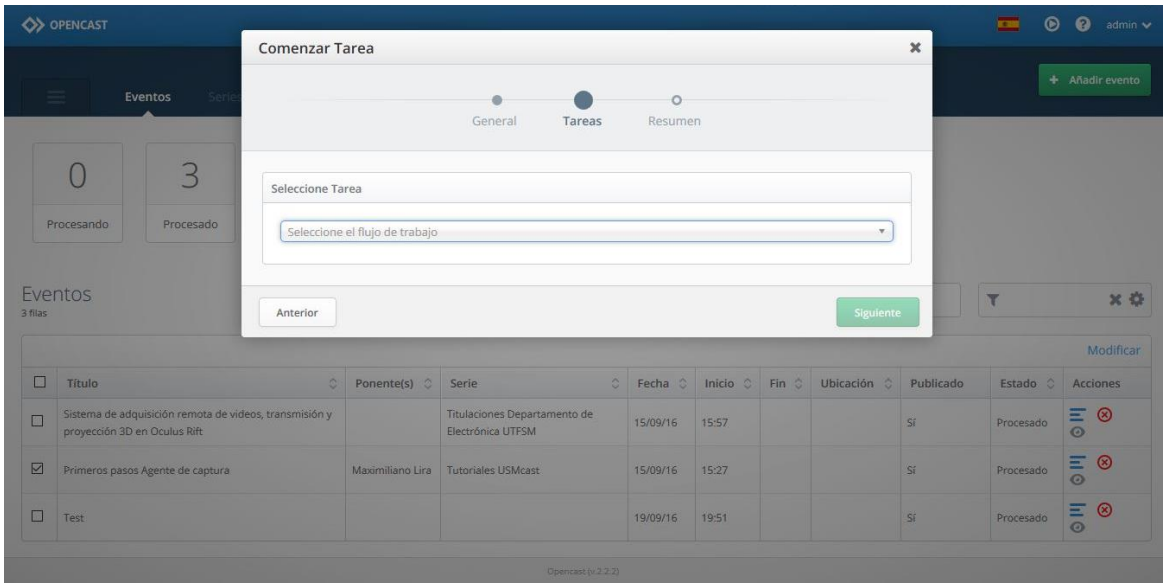


Figura 70: Ventana de comenzar tarea, selección de tarea

Se elige la tarea según la que se quiere llevar a cargo, puede ser una de las 4 opciones: Publicar (Publish), Republicar metadata (Republis metadata), Retractar (Retract) y actualizar vista previa (Update Previews).

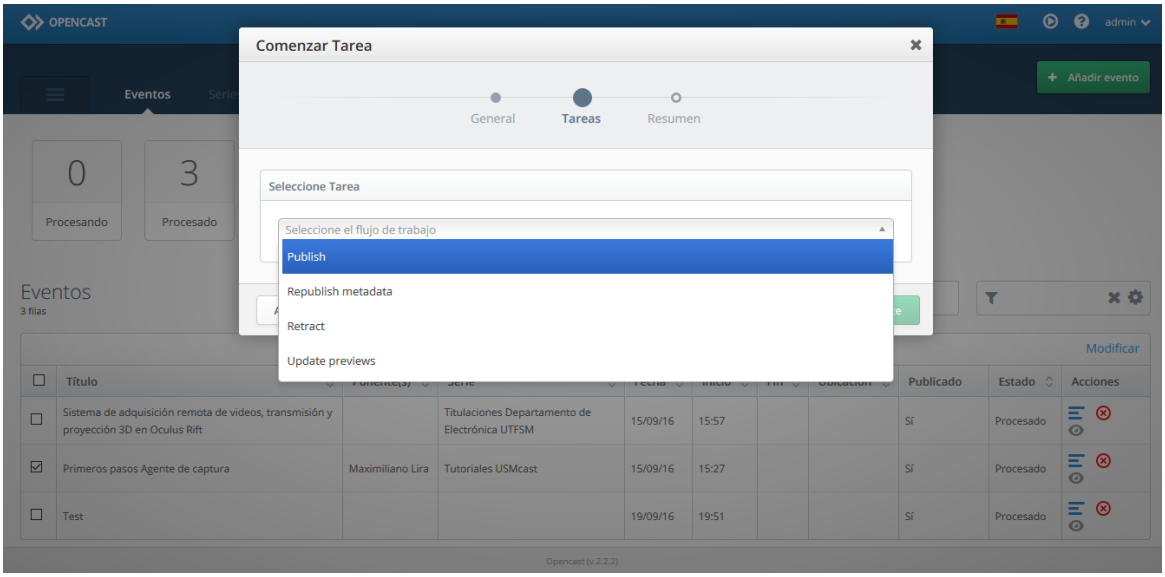


Figura 71: Lista de tareas que se pueden realizar

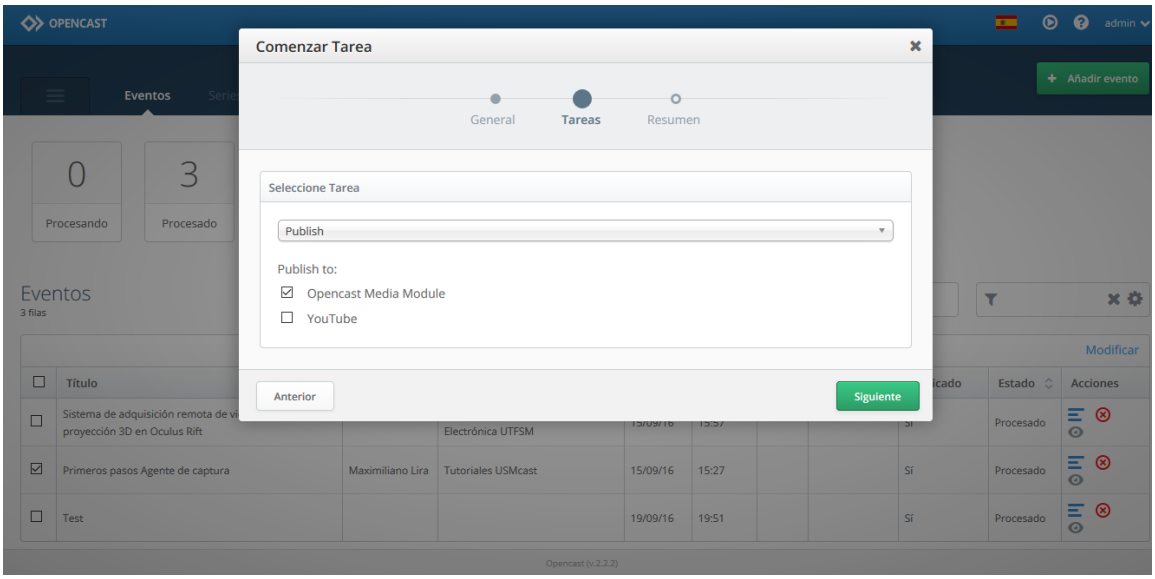


Figura 72: Ventana de comenzar tarea, opciones de tarea "Publish"

La tarea “Publicar” permite publicar una grabación en el módulo de presentación y/o en YouTube, esta opción es útil si se dejó la grabación para revisión o se editaron las cortinas de inicio y final de la serie.

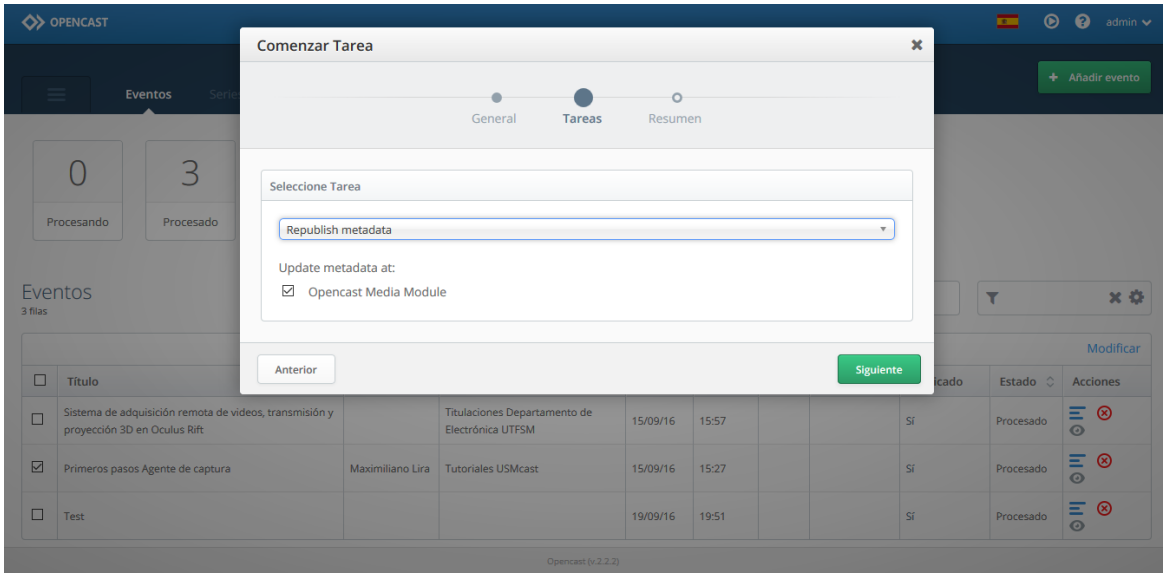


Figura 73: Ventana de comenzar tarea, opciones de tarea "Republish Metadata"

Si se editaron los metadatos y no se quiere pasar por el proceso de publicar (Que implica la recodificación del vídeo), existe la opción de sólo publicar el cambio de los metadatos. Esta opción sólo afecta al módulo de medios de Opencast. En el caso de YouTube este cambio debe hacerse manual desde el mismo sitio web.

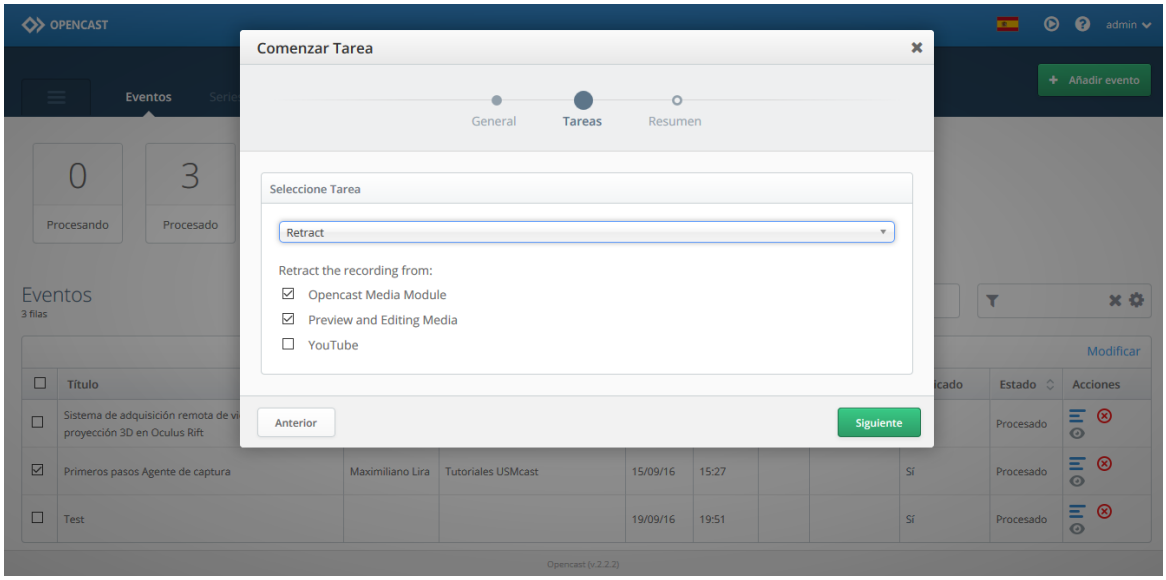


Figura 74: Ventana de comenzar tarea, opciones de tarea "Retract"

La opción retractar sirve para eliminar un vídeo tanto del módulo de medios como de YouTube.

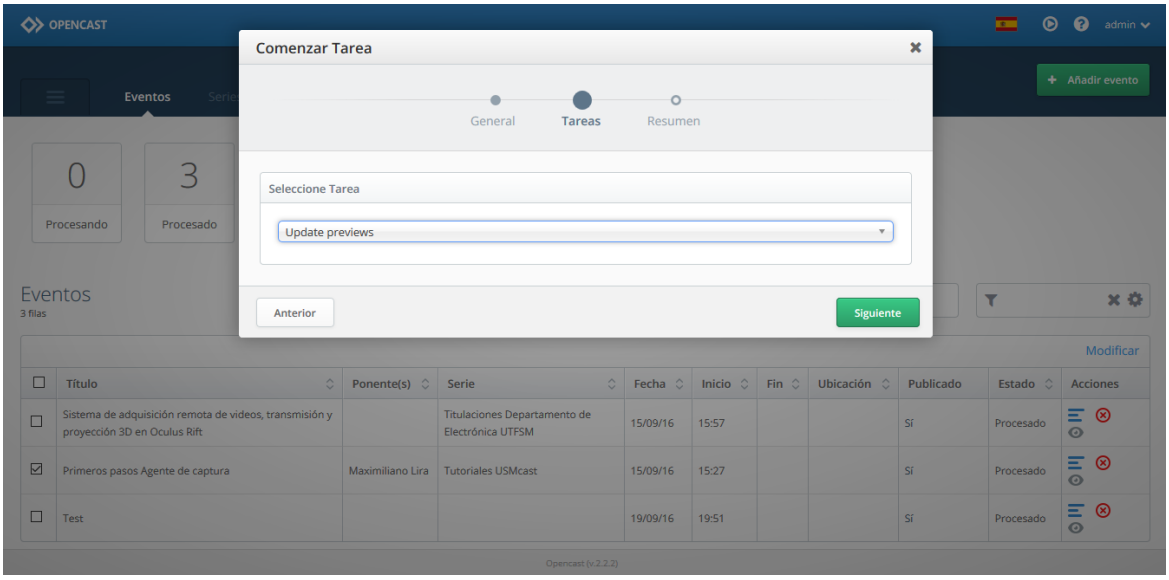


Figura 75: Ventana de comenzar tarea, opciones de tarea "Update Previews"

Finalmente, la opción de actualizar la pre-visualización permite actualizar las miniaturas que aparecen en el módulo de medios.

Una vez elegida la acción, se da un resumen de esta y aparece el botón para empezar con la tarea asignada.

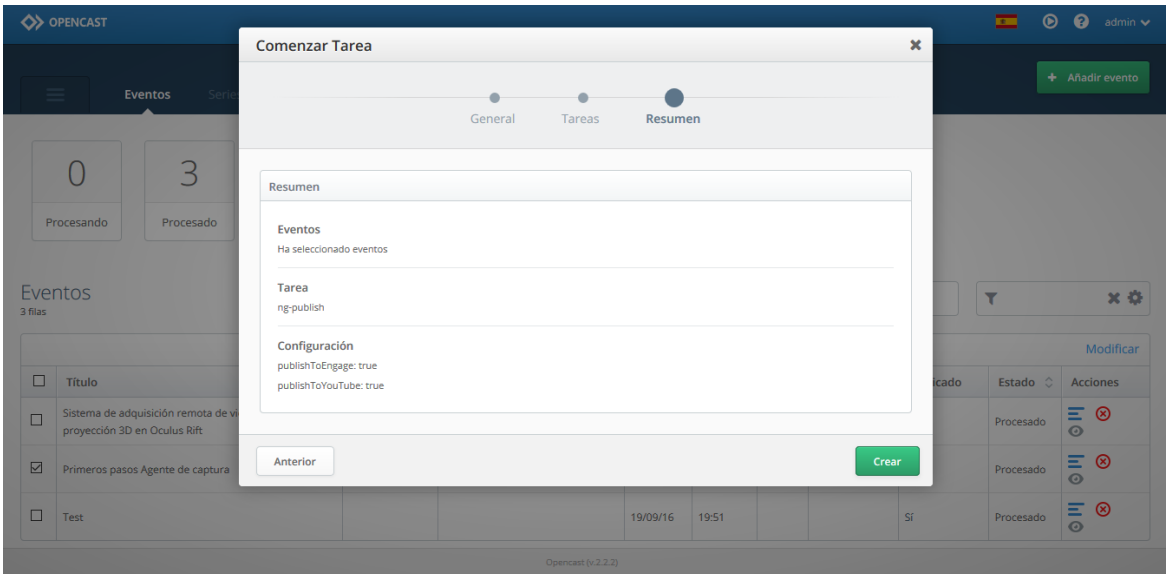


Figura 76: Ventana de comenzar tarea, resumen de acciones a realizar

4.3.3 Publicación de evento a sistemas de video bajo demanda (YouTube)

Opencast tiene la capacidad de conectarse a sistemas de vídeo bajo demanda, en particular a YouTube, la implementación de esta característica se encuentra explicada en el punto 7.3 de este trabajo.

Para publicar en Youtube, antes se necesita haber creado un flujo de trabajo que admita la publicación en este servicio, configurado las credenciales de acceso a la cuenta de Google (Necesarias para la publicación en YouTube) y el canal de YouTube donde se publicarán los vídeos.

Con la configuración ya hecha, se procede a subir o calendarizar el evento utilizando el grupo de trabajo para YouTube.

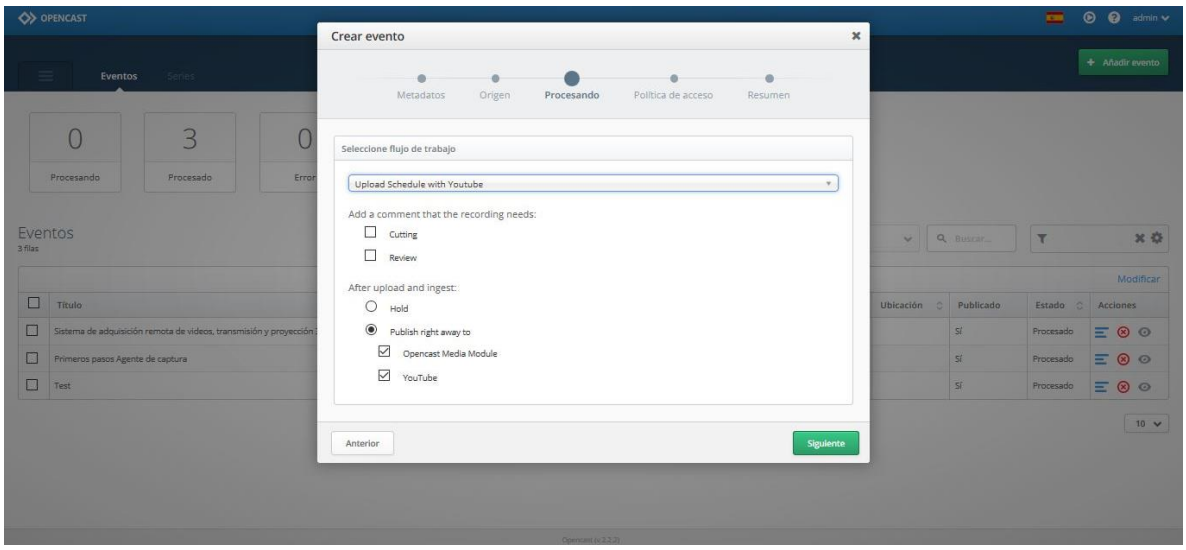


Figura 77: Elección de flujo de trabajo para publicación en YouTube

En la Figura 77 se aprecia cómo se selecciona el flujo de trabajo creado para este propósito, en la cual muestra que se activa la opción de publicación de YouTube para este efecto.

Una vez procesado y publicado en YouTube, este vídeo queda de la siguiente forma:

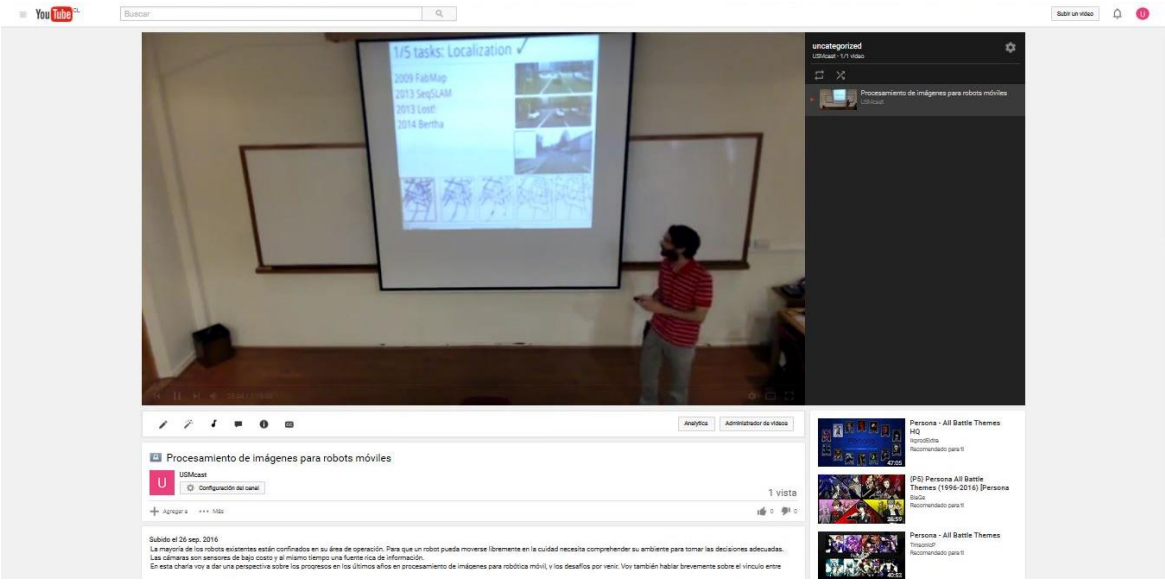


Figura 78: Grabación publicada automáticamente utilizando YouTube

La configuración actual, sólo sube el vídeo del presentador, aunque está la posibilidad de poder hacer una edición que una en un formato "Side-by-Side" a las dos grabaciones [34] (presentación y grabación), mejoras con respecto a esta característica fueron descartadas debido a fallas que presentaba el código [35], que permitía hacer la publicación en YouTube más robusta.

4.4 PRUEBAS DE DESEMPEÑO

4.4.1 Tiempo y recursos utilizados en el procesamiento de grabación y codificación

Estas pruebas de desempeño se hicieron usando la siguiente configuración de grabación:

Tabla 10: Configuración de grabación en Galicaster

Presentación	
Resolución	1920x1080
Contenedor	AVI
Bitrate	----
Cuadros por segundo	30
Cámara Presentador	
Resolución	1280x720
Contenedor	AVI
Bitrate	---
Cuadros por segundo	24

La codificación utilizada por Opencast para los vídeos fue la siguiente:

Tabla 11: Configuración de codificación en Opencast

Presentación	
Resolución	800x450
Contenedor - Codificador	MP4 – H264
Bitrate	Variable
Cuadros por segundo	25
Cámara Presentador	
Resolución	640x360
Contenedor - Codificador	MP4 – H264
Bitrate	Variable
Cuadros por segundo	25

La resolución utilizada es la dada por defecto del sistema Opencast, esta puede ser cambiada por una resolución mayor o crear varios perfiles de resolución de archivos. Se dejó esta configuración ya que ha dado excelentes resultados y el espacio en memoria utilizado por las grabaciones es muy bajo, en el caso de la grabación del presentador es de alrededor de 130 MB por hora de vídeo y en el caso de las presentaciones, como son principalmente imágenes estáticas, se da un uso de espacio de alrededor de 40 MB por hora de vídeo.

4.4.1.1 Agente de Captura

En el caso del tiempo que utiliza Galicaster para el procesamiento de la grabación es el mismo que el tiempo de la grabación en sí. Galicaster codifica al mismo tiempo que va grabando. Ello le da la posibilidad de comenzar una nueva grabación prácticamente desde el momento de terminar la grabación anterior.

En cuanto al uso de recursos, el mayor uso está durante el proceso de grabado. Las imágenes a continuación fueron obtenidas desde agente de captura móvil:

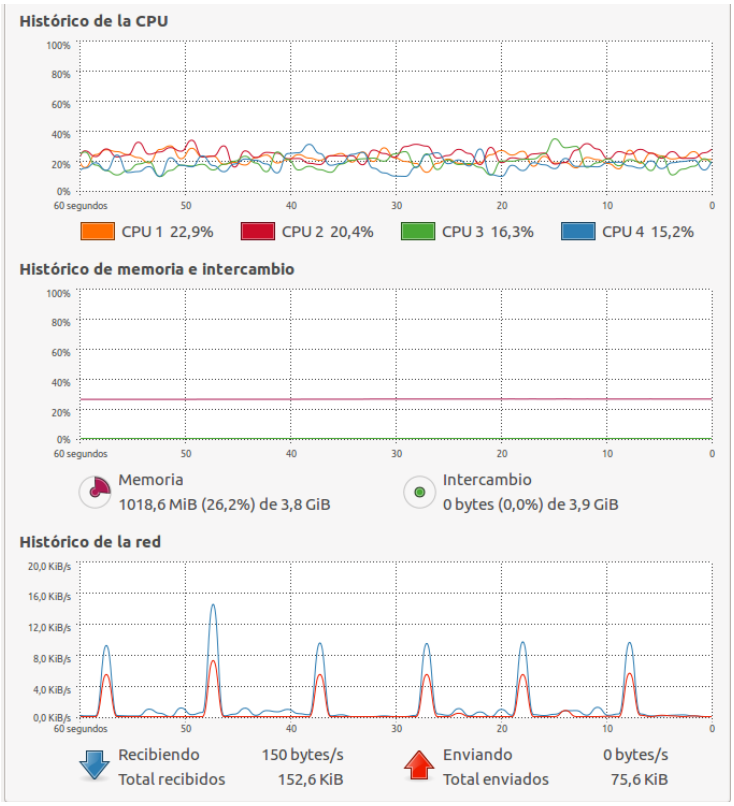


Figura 79: Equipo Galicaster en espera

Galicaster en modo de espera no hace uso intensivo de la CPU, lo que se aprecia en la Figura 79 es sólo la utilización de una CPU del sistema mientras se mantiene en espera, la memoria RAM está utilizada al 26,2 % y el tráfico de red es despreciable.

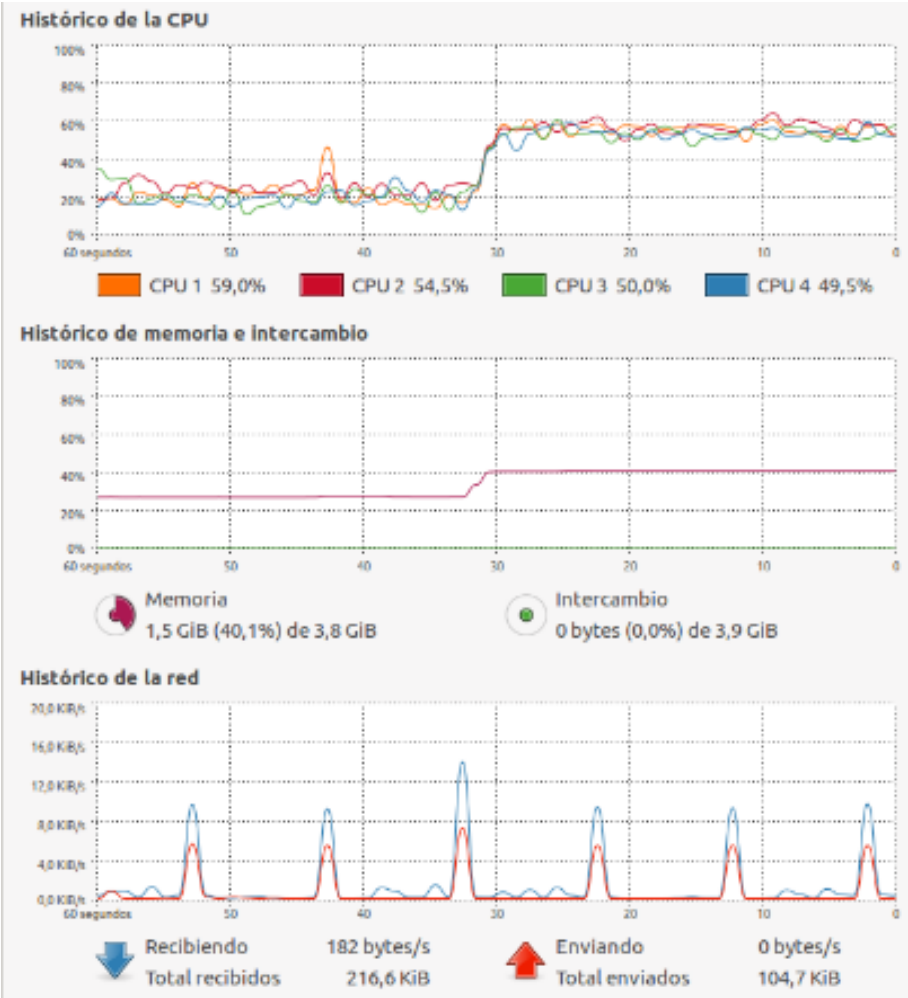


Figura 80: Uso de recursos de Galicaster mientras graba

La Figura 80 y muestra como es el estado de la CPU, y memoria RAM mientras graba Galicaster, como se puede apreciar, la utilización total de la CPU aumentó al 60% de su capacidad, demostrando que Galicaster hace un uso exhaustivo de este recurso. En cuanto a la memoria RAM, subió un 13,9 % (518 MiB de memoria adicional).

Otro recurso que necesita Galicaster es el uso de la red. El programa cuando envía la grabación tiene el siguiente comportamiento:

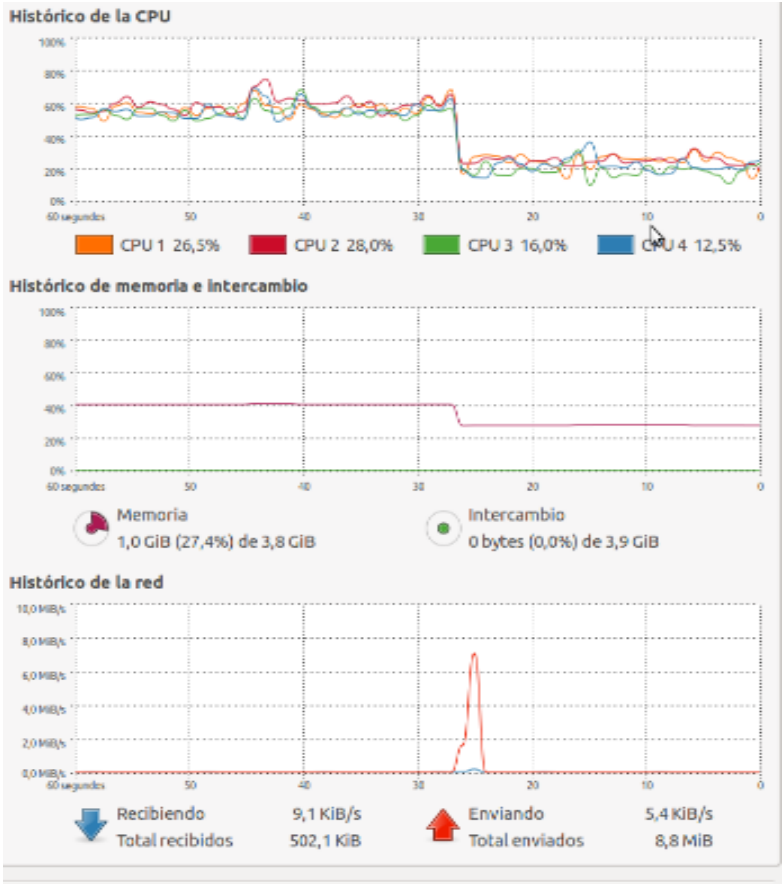


Figura 81; Uso de recursos al finalizar la grabación

La Figura 81, muestra que al finalizar la grabación, se liberan de inmediato los recursos utilizados, y que inmediatamente ocurre la transmisión del paquete de medios a Opencast para su procesado automático.

El tiempo de envío de la grabación dependerá del largo de esta, de la congestión de red actual y de la rapidez que tiene el enlace entre el servidor y el agente de captura. Para el caso del agente de captura estacionario, este cuenta con un enlace de 100 Mbps por la red del Departamento de Electrónica hasta el servidor de difusión.

4.4.1.2 Servidor de difusión

En el caso del servidor de difusión, una de las pruebas más importantes que se evalúa, es el tiempo de procesamiento que tarda una grabación en publicarse en el módulo de medios de

Opencast, para tomar esta medida, se utilizaron las marcas de tiempo del procesamiento del servidor versus el tiempo de duración de la grabación.

Los datos adquiridos fueron desde el servidor de producción (Spielberg), las grabaciones escogidas cumplen el siguiente criterio:

- Cortina de introducción y término
- Grabación del presentador como de sus diapositivas
- Usan el mismo perfil de codificación tanto en la captura de Galicaster como en el procesamiento de Opencast.
- Cada grabación fue procesada de manera independiente

Con estos datos se pudo genera la Figura 82:

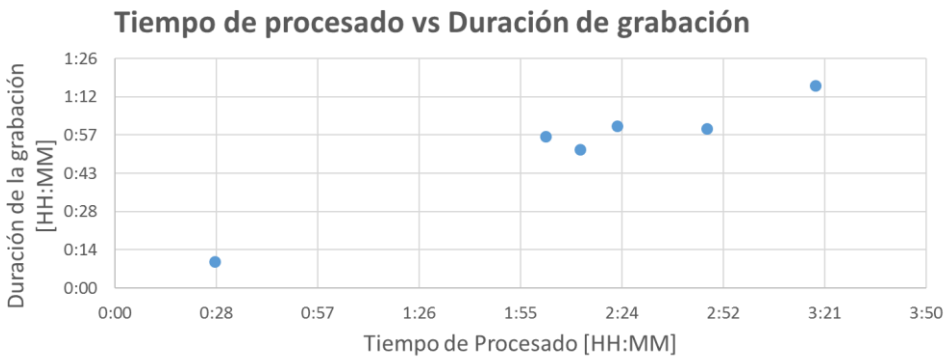


Figura 82: Gráfico de tiempo de procesado vs duración de grabación

La Figura 82, muestra el tiempo que se demora una grabación, desde que Galicaster la transmite al servidor, hasta que esta publicada. Se obtiene que para una grabación de menos de 15 minutos se procesa en menos de media hora, mientras que, para grabaciones de una hora aproximadamente, este procesamiento puede llevar desde 2 horas hasta un poco más de 3 horas, con un promedio de 2 horas con 30 minutos.

En cuanto al uso de los recursos de la máquina mientras procesa los vídeos, se puede ver en el siguiente gráfico donde se toman los datos de procesamiento de dos vídeos distintos. Cabe recalcar que estos vídeos se procesaron de forma separada (no simultáneos).

Las grabaciones para analizar la carga de la máquina utilizadas fueron extraídas desde el servidor de pruebas (Cameron), se extrajo información del uso de la CPU y RAM durante su uso utilizando el registro de sistema “SAR”, en el caso de la medición de porcentaje de uso de la CPU, este porcentaje es por núcleo.

Los datos obtenidos fueron los siguientes:

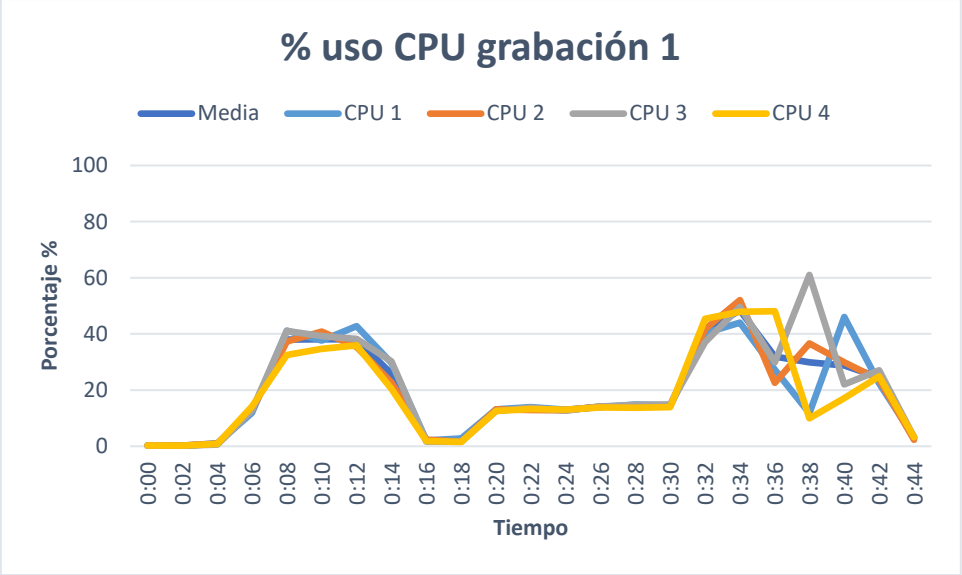


Figura 83: Porcentaje de uso de cpu grabación 1

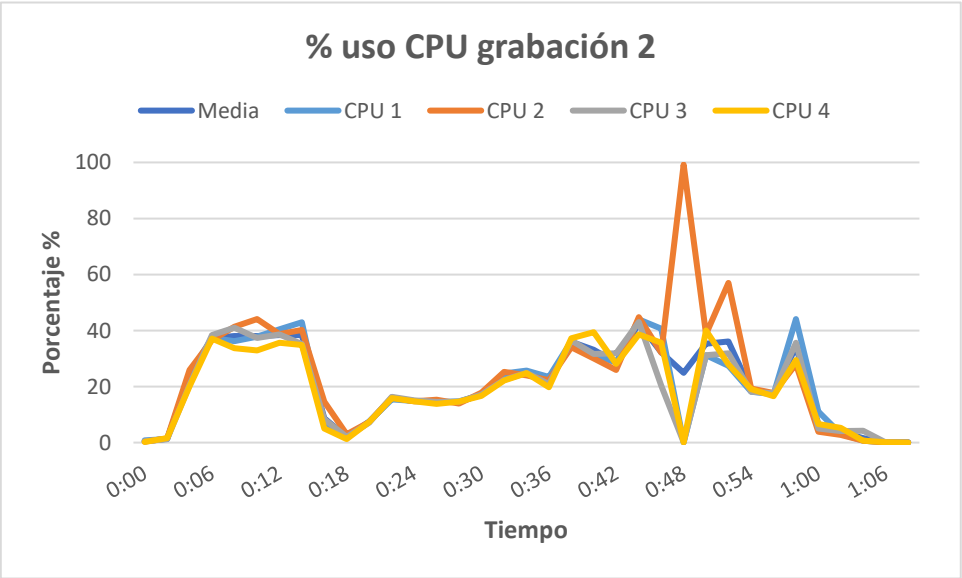


Figura 84: Porcentaje de uso de CPU grabación 2

Como se puede ver en las dos figuras, el porcentaje de utilización de CPU es muy parejo en el uso de las arquitecturas multi-CPU, lo que habla de una buena optimización en este tipo de esquemas.

En cuanto al uso de los procesadores, la mayor carga se tuvo en las dos grabaciones cuando se utilizaba el módulo “Compose” de Opencast. Este módulo es el que se encarga de la codificación, concatenación u otra tarea ligada al tratamiento de los vídeos.

En las dos grabaciones tienen un alza del uso de la CPU entre los minutos 4 y 16 a 18, que corresponde a la tarea de crear la vista previa de las grabaciones entregadas por Galicaster. Luego existe otra alza de uso de los procesadores en los ¾ del procesamiento de la grabación, este punto del procesado es debido a que el sistema está codificando en un formato de vídeo apto para la publicación (En este caso MP4) y también hace la segmentación del vídeo de la presentación para detectar el cambio de diapositiva.

El uso de RAM, al igual que el de la CPU, fue obtenido con el programa de registro “SAR”, el valor mostrado a continuación es el porcentaje de RAM utilizado para el usuario actual. La carga de memoria RAM durante el procesamiento es de la siguiente forma:

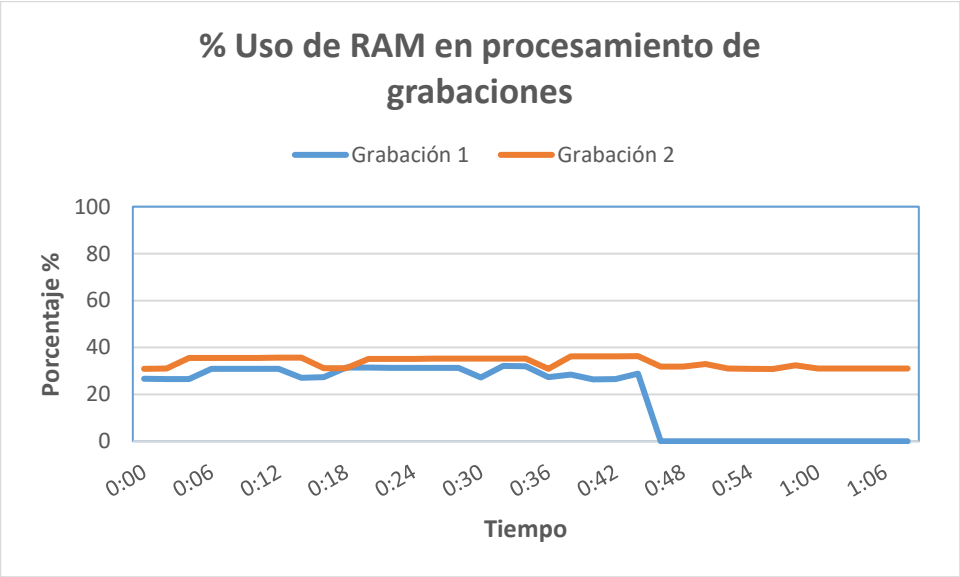


Figura 85: Porcentaje de uso de RAM en procesamiento de grabaciones

Tanto la grabación uno, como la dos utilizan entre un 30 a un 40 por ciento de la memoria RAM del equipo, lo que en este equipo en particular se traduce en 1,6 GB de memoria que se utiliza en todo el proceso. El nivel de uso de RAM por Opencast es aceptable, permitiendo que la máquina pueda hacer otras tareas de impacto medio mientras se procesa la grabación.

5 CONCLUSIONES Y TRABAJO FUTURO

Con los resultados obtenidos en las pruebas realizadas en la sección 4, es posible concluir que la solución implementada cumple tanto con los requerimientos funcionales como con los no funcionales que fueron planteados en la sección 3.1. Permitiendo hacer una grabación y publicación automatizada de las charlas que utilicen el sistema implementado. En cuanto al agente de captura móvil, no se logró que capturara vídeo al mismo tiempo que se envía al proyector, esto debido a una incompatibilidad de hardware entre el splitter HDMI y la capturadora BlackMagic del equipo.

La utilización de Galicaster junto con Opencast, se encargan de hacer la grabación, codificación y la publicación de las grabaciones permite que personas con poca experiencia en tecnología y sin tener que configurar nada adicional en sus computadores, puedan difundir sus exposiciones fuera de la sala donde hacen su puesta en escena. En el caso particular de la UTFSM, ya se encuentra en producción desde noviembre de 2015, logrando 9 presentaciones grabadas desde esa fecha hasta la escritura de este trabajo.

El sistema de captura de Galicaster, su codificación de las grabaciones se hace de manera simultánea a la grabación, lo cual tiene la ventaja de que, al momento de terminar la grabación, el agente puede enviarla inmediatamente para su difusión por parte de Opencast. Por otro lado, la cantidad de recursos utilizados durante la grabación (Sobre el 60%), hace que no sea recomendable usarlo para otra tarea además de ésta.

Opencast demostró ser una plataforma robusta tanto para la codificación como la publicación de las grabaciones, las muchas opciones que entrega permiten que se adecúe a las necesidades de las organizaciones que necesiten de este tipo de plataformas. Al ser una plataforma con la capacidad de ser distribuida, podría aumentar la capacidad de cómputo y disminuir los tiempos de codificación que se obtuvieron con la capacidad actual.

Sobre el módulo móvil, se cumplió el objetivo de poder empaquetar todo el equipamiento en un módulo fácil de llevar para las salas y auditorios. Por otro lado, debido a su diseño y peso, no es apto para subir ni bajar escaleras, lo cual hace que este módulo sea difícil de transportar entre los distintos edificios de la UTFSM casa central. Una idea para resolver esto es crear un

nuevo diseño, con materiales más ligeros, pero igual de resistentes, y que sea fácil de llevar por escaleras.

Como trabajo futuro, se pueden hacer varias mejoras. Por el lado del servidor, se podría mejorar la codificación del vídeo optimizando y creando nuevos perfiles junto con un sistema de autoedición que permita la colocación de un banner de título dentro de la grabación y el logotipo de la universidad o el departamento al cual se expone la charla. En el caso del agente de captura se puede crear un sistema de arreglos de micrófono dentro de la sala, lo que permitiría que el expositor no necesitase colocarse un micrófono de solapa y tampoco la necesidad de tener un micrófono para los asistentes cuando quieran hacer una pregunta.

Otras características que se podrían agregar al sistema son generar un Stream en Vivo, crear un portal utilizando otra plataforma más completa (eDx, Pumukit por ejemplo), generar un sistema de rastreo para las cámaras, con la idea que puedan seguir al expositor mientras este se desplaza por la sala.

6 REFERENCIAS

[1] J. C. W. Debuse, A. Hede y M. Lawley, «Learning efficacy of simultaneous audio and on-screen text in online lectures,» *Australian Journal of Educational Technology,* vol. 25, nº 5, pp. 748-762, 2009.

[2] D. K. Griffin, D. Mitchell y S. J. Thompson, «Podcasting by synchronising PowerPoint and voice: What are the pedagogical benefits?,» *Computers & Education,* vol. 53, nº 2, p. 532–539, 2009.

[3] D. MCKINNEY, J. L. DYCK y E. S. LUBER, «iTunes University and the classroom: Can podcasts replace Professors?,» *Computers & education,* vol. 52, nº 3, pp. 617-623, 2009.

[4] D. N. HARPP y e. al., «Lecture retrieval via the Web: Better than being there?,» *Journal of Chemical Education,* vol. 85, nº 5, p. 688, 2004.

[5] R. Veeramani y S. Bradley, «Online-learning study: Insights regarding undergraduate preference for lecture capture,» University of Wisconsin-Madison: E-Business Institute, 2008. [En línea]. Available: http://www.uWebi.org/news/uw-online-learning.pdf. [Último acceso: 25 Mayo 2010].

[6] K. e. a. WOO, «Web-based lecture technologies: blurring the boundaries between face-to-face and distance learning,» *Research in Learning Technology,* vol. 16, nº 2, 2008.

[7] J. DAY y J. FOLEY, «Evaluating web lectures: A case study from HCI,» *CHI'06 Extended Abstracts on Human Factors in Computing Systems. ACM,* pp. 195-200, 2006.

[8] T. Traphagan, «Class lecture Webcasting, fall 2004 and spring 2005: A case study,» The University of Texas, Austin, 2005.

[9] R. Owston, D. Lupshenyuk y H. Wideman, «Lecture capture in large undergraduate classes: Student perceptions and academic performance,» *Internet and Higher Education,* vol. 14, nº 4, pp. 262-268, 2011.

[10] Galicaster, «Galicaster project Home - GALICASTER - Teltek Video Research WIKI | Lecture Capture systems,» 22 Junio 2015. [En línea]. Available: https://wiki.teltek.es/display/Galicaster/Galicaster+project+Home#. [Último acceso: 20 Octubre 2016].

[11] Opencast, «Opencast | Media & Distribution,» 2016. [En línea]. Available: www.opencast.org. [Último acceso: 04 Septiembre 2016].

[12] Opencast, «Adopter Highlight: University of California Berkeley,» [En línea]. Available: http://www.opencast.org/users/adopter-highlight-university-of-california-berkeley. [Último acceso: 25 Agosto 2016].

[13] UC Berkeley , «Nomination of the UC Berkeley Webcast Project for the Larry Sautter Award,» 2002.

[14] Oxford IT Services , «Replay Capture Lecture Service,» 13 Julio 2016. [En línea]. Available: http://help.it.ox.ac.uk/replay/index. [Último acceso: 2016 Agosto 29].

[15] Kaltura, «CaptureSpace | Kaltura,» 2016. [En línea]. Available: http://corp.kaltura.com/Products/Video-Applications/Kaltura-CaptureSpace. [Último acceso: 29 Agosto 2016].

[16] Kaltura, «Kaltura Video Platform | Kaltura Community Edition,» 2016. [En línea]. Available: http://corp.kaltura.com/Deployment-Options/Kaltura-Community-Edition. [Último acceso: 28 Agpsto 2016].

[17] Panopto, «Panopto Video Platform for Education,» 2016. [En línea]. Available: https://www.panopto.com/panopto-for-education/. [Último acceso: 29 Agosto 2016].

[18] TechSmith, «Higher Education Video and Lecture Capture Platform,» 2016. [En línea]. Available: https://www.techsmith.com/techsmith-relay.html. [Último acceso: 30 Agosto 2016].

[19] Cisco, «Lleve la formación Online al máximo nivel,» San Francisco, 2015.

[20] Opencast Community, «Release Notes - Opencast - Administration Guide,» 17 Julio 2015. [En línea]. Available: https://docs.opencast.org/r/2.0.x/admin/release.notes/. [Último acceso: 26 Octubre 2016].

[21] Teltek Video Research, «Galicaster project Home - GALICASTER - Teltek Video Research WIKI | Lecture Capture Systems,» 05 Julio 2016. [En línea]. [Último acceso: 04 Septiembre 2016].

[22] T. Owens, «Tim Owen's Blog,» 9 Enero 2013. [En línea]. Available: https://blog.timowens.io/why-im-getting-rid-of-kaltura-and-why-you-should-too/. [Último acceso: 27 Agosto 2016].

[23] R. B. Grady, Practical Software Metrics for Project Management and Process Improvement, Prentice Hall, 1992.

[24] H. Canto, «Developing with Galicaster de Héctor Canto en Prezi,» 26 Junio 2013. [En línea]. Available: https://prezi.com/-ixh-cl4b25j/developing-with-galicaster/. [Último acceso: 13 Septiembre 2016].

[25] S. Patil, «Hello, OSGi, Part 1: Bundles for beginners | JavaWorld,» 04 Marzo 2008. [En línea]. Available: http://www.javaworld.com/article/2077837/java-se/java-se-hello-osgi-part-1-bundles-for-beginners.html. [Último acceso: 20 Septiembre 2016].

[26] XenServer Wiki, «XenServer Wiki,» 19 Mayo 2014. [En línea]. Available: https://wiki.xenserver.org/index.php?title=Main_Page. [Último acceso: 20 Septiembre 2016].

[27] Opencast Community, «Database - Opencast - Administration Guide,» 2015. [En línea]. Available: https://docs.opencast.org/latest/admin/configuration/database/. [Último acceso: 01 Noviembre 2016].

[28] Wikipedia, «Servidor HTTP Apache - Wikipedia, la enciclopedia libre,» 29 Junio 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Servidor_HTTP_Apache. [Último acceso: 20 Septiembre 2016].

[29] Opencast Community, «Multiple Servers - Opencast - Administration Guide,» 30 Marzo 2016. [En línea]. Available: https://docs.opencast.org/latest/admin/installation/multiple-servers/#step-2-set-up-nfs-server. [Último acceso: 21 Septiembre 2016].

[30] S. Kishore, «NFS server and client on CentOS 7,» How to Forge, Linux Tutorials, 16 Diciembre 2014. [En línea]. Available: https://www.howtoforge.com/nfs-server-and-client-on-centos-7. [Último acceso: 21 Septiembre 2016].

[31] Wikipedia, «Ubuntu - Wikipedia, la enciclopedia libre,» 02 Septiembre 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Ubuntu. [Último acceso: 21 Septiembre 2016].

[32] B&H Photo Video, «Blackmagic Design DeckLink Mini Recorder,» 2016. [En línea]. Available: https://www.bhphotovideo.com/c/product/964122-REG/blackmagic_design_decklink_mini_recorder.html. [Último acceso: 22 Septiembre 2016].

[33] B&H Photo Video, «DATAPATH VisionRGB-E1S DVI/RGB/HD Video Capture Card,» 2016. [En línea]. Available: https://www.bhphotovideo.com/c/product/1152002-REG/datapath_visionrgb_e1s_dvi_rgb_hd_video_capture.html. [Último acceso: 22 Septiembre 2016].

[34] L. Kiesow, «[OC Users] Re: Help with YouTube integration - Grupos de Google,» 20 Septiembre 2016. [En línea]. Available: https://groups.google.com/a/opencast.org/forum/#!topic/users/FzPY2X7PXm0. [Último acceso: 20 Septiembre 2016].

[35] J. Crossman, «opencast-community/Matterhorn/ Oull request #596: MH-10863 YouTube v3 integration with configurable retry, more operations,» 24 07 2015. [En línea]. Available: https://bitbucket.org/opencast-community/matterhorn/pull-requests/596/mh-10863-youtube-v3-integration-with/diff. [Último acceso: 26 Septiembre 2016].

[36] J. Ellison, «Apache Mod_proxy '[Error] (13) Permission Denied' Error on RHEL - SysAdmin's Journey,» 1 Febrero 2010. [En línea]. Available: http://sysadminsjourney.com/content/2010/02/01/apache-modproxy-error-13permission-denied-error-rhel/. [Último acceso: 2 Noviembre 2015].

[37] bombcar, «apache 2.2 - Can you use Redirect and Proxypass at the same time - Server Fault,» 18 Enero 2016. [En línea]. Available: http://serverfault.com/questions/605931/can-you-use-redirect-and-proxypass-at-the-same-time. [Último acceso: 2015 Septiembre 25].

[38] Opencast Community, «Overview - Opencast - Administration Guide,» 27 Septiembre 2016. [En línea]. Available: https://documentation.opencast.org/r/2.2.x/admin/workflowoperationhandlers/. [Último acceso: 29 Septiembre 2016].

7 ANEXOS

7.1 INSTALACIÓN DE PROXY REVERSO UTILIZANDO APACHE HTTP SERVER

Apache puede ser utilizado para funcionar como proxy reverso. La utilidad de esto es hacer que el servidor obtenga los recursos pedidos por el cliente en vez de que este los requiera de forma directa. Aumentando la seguridad del servicio.

Para el caso de Opencast, configura para que las peticiones entren por el puerto 80 y luego éstas sean reenvíadas al puerto 8080. Esta configuración se hace al archivo /etc/httpd/conf.d

7.1.1 Configuración de SELinux para Apache

CentOS en particular, tiene un sistema de políticas de seguridad. Por ello, si se quiere utilizar el servidor apache en conjunto con Opencast para distintas tareas (Ya sea alojar logotipos o utilizarlo de proxy reverso), debe seguir los siguientes pasos [36]:

En el terminal del servidor donde se ejecutará Apache (En este caso es el mismo servidor de Opencast), se escribe lo siguiente:

```
# /usr/sbin/setsebool httpd_can_network_connect 1
```

Para que esta orden persista a los reinicios del sistema, ingresar el comando:

```
# /usr/sbin/setsebool -P httpd_can_network_connect 1
```

7.1.2 Configuración de Proxy reverso:

Editando el archivo `/etc/httpd/conf.d`, se procede a agregar las siguientes líneas [37]:

```
<VirtualHost *:80>
ServerName opencast.elo.utfsm.cl
ProxyRequests Off
ProxyPreserveHost On
```

```
    ProxyPass /player !
    ProxyPass /admin !
    ProxyPass /docs !
ProxyPass /datos !
ProxyPass / http://127.0.0.1:8080/
ProxyPassReverse / http://127.0.0.1:8080/
    Redirect permanent /player http://opencast.elo.utfsm.cl/engage/ui
    Redirect permanent /admin http://opencast.elo.utfsm.cl/admin-ng

</VirtualHost>
```

Las instrucciones que se dictan en la sección de código anterior son:

- Habilitar Proxy reverso para reenviar peticiones desde el puerto 80 al 8080.
- Los directorios /player, /admin, /docs y /datos no los manejará apache.
- Se hará redirección de los directorios anteriores (Excepto /datos) a otros sitios de manera de acortar las URL.
- /datos lo manejará directamente Apache.

7.2 FLUJOS DE TRABAJO DE OPENCAST

Cuando se ingresa una nueva grabación a Opencast, el servicio hace una serie de pasos para preparar y condicionar la grabación y su posterior publicación. Este proceso se hace a través de Flujos de trabajos que son previamente configurados por el administrador del equipo.

Los flujos de trabajos son una lista ordenada de operaciones, donde no hay límite para el número de operaciones ni su repetición en los flujos de trabajo dados.

Estas operaciones pueden ser configuradas usando “elementos de configuración”. El uso de reemplazo de frases en estos valores permite que los flujos de trabajo se adapten dinámicamente a una entrada dada o a la decisión del usuario.

Para que Opencast pueda utilizar estos flujos de trabajo, estos se deben guardar en un archivo con extensión XML, dentro de la carpeta `/etc/opencast/workflows`.

7.2.1 Descripción de un flujo de trabajo

Los flujos de trabajo están definidos en XML, la estructura general de un flujo de trabajo es de la siguiente forma:

```
<definition xmlns="http://workflow.opencastproject.org">

  <!-- Description -->
  <id></id>
  <title></title>
  <tags></tags>
  <description></description>

  <!-- Operations -->
  <operations>
    <operation></operation>
    ...
  </operations>

</definition>
```

En la zona de descripción, se utiliza para identificar y dar los parámetros que utilizará Opencast para la operación.

En la sección de operaciones existen las siguientes acciones que se pueden realizar:

Tabla 12: Lista de operadores de flujos de trabajo

Operador del flujo de trabajo	Descripción
analyze-tracks	Analiza las pistas en el paquete de medios
analyze-audio	Analiza el primer flujo de audio
append	Detiene para que el usuario pueda elegier el siguiente flujo de trabajo
archive	Archiva el estado actual del paquete de medios
caption	Espera al usuario para que suba subtítulos
cleanup	Limpia el repositorio a de trabajo utilizado
comment	Agrega, resuelve o elimina algún comentario
compose	Codifica los archivos de medios utilizando FFMPEG
composite	Compone dos vídeos en un lienzo
concat	Concatena múltiples pistas de vídeo en una sola
copy	Copia los elementos de un paquete de medios en un directorio
cover-image	Genera imagen de portada conteniendo metadatos
defaults	Aplica los valores por defecto del flujo de trabajo
editor	Espera al usuario para su revisión, luego corta el vídeo según la lista de cortes
email	Envía notificaciones por email en cualquier punto del flujo de trabajo
encode	Codifica archivos de medios en distintos formatos en paralelo
extract-text	Extrae el texto desde los segmentos de presentación

http-notify	Notifica el punto de término HTTP sobre el proceso en el flujo de trabajo
image	Extrae imágenes desde el vídeo utilizando FFmpeg
image-to-video	Crea una pista de vídeo desde una imagen fuente
incident	Prueba incidents desde un trabajo falso
ingest-download	Descarga los archivosdesde una URL externa para su ingesta
inspect	Inspecciona el medio (Chequea que sea válido)
normalize-audio	Normaliza la primera pista del flujo de audio
partial-import	Importa pistas parciales y las procesa de acuerdo al document SMIL
post-mediapackage	Envía el paquete de medios a un servicio remoto
prepare-av	Prepara las versiones de trabajo del audio y vídeo
publish-configure	Distribuye y publica los medios a la publicación configurada
publish-engage	Distribuye y publica los medios al engage player
republish	Republica los elementos en la búsqueda
retract-configure	Retracta los medios desde la publicación configurada
segment-video	Extrae los segmentos desde la presentación
segmentpreviews	Extrae imágenes de segmentos desde un video usando FFmpeg
series	Aplica la serie al paquete de medios
silence	Detección de silencio desde el audio del paquete de medios
tag	Modifica las etiquetas desde los elmentos contenidos
trim	Espera al usuario para revisión, luego corta la grabación
waveform	Crea una imagen de forma de onda del audio del paquete de medios
zip	Crea un archive ZIP del estado actual del paquete de medios

El detalle del funcionamiento de esta operaciones se puede revisar en la documentación oficial de Opencast [38].

7.3 CONFIGURACIÓN DE PUBLICACIÓN EN YOUTUBE PARA OPENCAST

La configuración para habilitar la publicación en Youtube de Opencast es la siguiente:

Para habilitarlo se necesita lo siguiente tener antes:

- Cuenta Google

- Canal de YouTube donde hacer la publicación

Una vez cumplido esos requisitos, se tiene que crear un nuevo proyecto de desarrollador, la idea de este proyecto es para proveer con las llaves de acceso a Opencast para que pueda hacer la publicación.

7.3.1 Configuración en Google

Crear un nuevo proyecto de Google:

- Entrar a la cuenta de google
- Navegar a la consola de desarrolladores de Google en: https://console.developers.google.com/project.
- Crear un nuevo proyecto y seguir las instrucciones de la página.
- Una vez terminado, en la página del nuevo proyecto elegir **API & Auth** y luego en **pantalla de consentimiento** en el panel de navegación.
- Colocar el **NOMBRE DEL PRODUCTO** y la **DIRECCIÓN DE EMAIL**

Habilitar API:

- Escoger APIs en el panel de navegación.
- Usar el filtro para encontrar y habilitar YouTube Data API v3.

Registrar la aplicación:

- Escoger Credenciales en el panel de navegación.
- Hacer clic en Crear una nueva ID de cliente para OAuth.
- Escoger la aplicación instalada para el tipo de aplicación como OTRO.
- Aceptar con el botón **Crear ID Cliente.**

Guardar el ID de cliente en formato JSON

- Descargar la información del cliente en JSON haciendo click **Descargar JSON.**
- Grabar el archivo JSON en `etc/youtube-v3/client-secrets-youtube-v3.json`.

7.3.2 Configuración de YouTube en Opencast

Con el archivo de las credenciales guardado en formato JSON dentro del servidor, se procede de la siguiente manera:

- Se inicia Opencast (En caso de que este corriendo se reinicia el servicio)

ATENCIÓN: Hasta que no esté correctamente configurado, Opencast no hará lectura ni escritura en la base de datos. En caso que se quiera abortar la configuración, solo se debe borrar el archivo JSON y reiniciar Opencast.

- En la línea de comandos, se lanza el comando para ver el estatus extendido del servicio Opencast:

```
# systemctl status opencast -l
```

  El comando entregará una URL que se debe copiar en un navegador web que tenga acceso a internet.
- La página web nos pedirá la credencial Google, se debe ingresar la cuenta con la cual se creó el proyecto de desarrollador de Google. Dará a elección al canal que se quiere publicar y si le damos acceso a la aplicación.
- Una vez dado el acceso, el navegador mostrará un error de conexión, esto es normal, ya que está pidiendo acceso a un sitio inexistente dentro del cliente. Se copia la dirección inválida y de ejecuta:

```
# curl [Dirección devuelta por el navegador]
```

- Una vez hecho eso, se obtiene la respuesta: `Received verification code. Closing...`
- Verificar que se haya escrito la llave en `work/opencast/youtube-v3/data-store/store`.
- Reiniciar Opencast.

7.3.3 **Habilitación de YouTube en Opencast**

Opencast ya se encuentra habilitado para publicar en YouTube, el último paso que falta es activar esta característica. Para ello se necesita crear un nuevo flujo de trabajo que lo permita.

En Opencast 2.2.2 esta opción se encuentra deshabilitada, para volver a activarla, se deben seguir los siguientes pasos:

Nota: Los pasos siguientes se hicieron en base del flujo de trabajo por defecto de Opencast.

- Se hace una copia del flujo de trabajo por defecto `"/etc/opencast/workflows/ng-schedule-and-upload.xml"`. y se crea la copia llamada `"/etc/opencast/workflows/ng-schedule-and-upload-youtube.xml"`.
- A esta copia se modifican los siguientes valores:
    - En `<configuration_panel>`, se habilita la opción de YouTube, la línea debe quedar de esta forma:

```
<input id="publishToYouTube" name="publishToYouTube" type="checkbox"
checked="checked" class="configField" value="true" />
<label for="publishToYouTube">YouTube</label>
```

    - En `<operation id="defaults" escription="Applying default configuration values">`, cambiar a "`true`" la llave `publishToYouTube`.
- En el flujo de trabajo `ng-partial-publish.xml`, se debe comentar la `operación` "`publish-youtube`" y pegar el siguiente código:

```
<operation
      id="publish-youtube"
      if="${publishToYouTube}"
      max-attempts="2"
      fail-on-error="true"
      exception-handler-workflow="error"
      description="Publish to YouTube">
      <configurations>
        <configuration key="source-
flavors">presenter/trimmed</configuration>
      </configurations>
    </operation>
```

Este código ha sido configurado para que se publique la grabación del presentador, si se quiere en cambio colocar la grabación de las diapositivas, se debe cambiar la línea "presenter/trimmed" por "presentation/trimmed".

Opencast detectará automáticamente el nuevo flujo de trabajo, con ello se puede ya seleccionar el nuevo flujo de trabajo con la opción de YouTube habilitada.

7.4 CÓDIGOS DE CONFIGURACIÓN DE OPENCAST

Los siguientes textos, son una copia completa de los archivos de configuración modificados para el funcionamiento de Opencast 2.2.2 en este trabajo:

7.4.1 custom.properties

```
#########################################
### Opencast configuration properties ###
#########################################

# The internal URL of this Opencast installation, used to locate services
running on this instance and for inter-node
# communication in distributed setups.
#
# Note that while the server.url is the public url of this instance, there
is the actual public url of an individual
# tenant, which is configured in
etc/load/org.opencastproject.organization-<tenant id>.cfg with the default
tenant id
# being "mh_default_org".
#
# Also note that if this felix installation is proxied behind an Apache
HTTPD reverse proxy, and communication is meant
# to go through that proxy, then server.url should point to the proxy's
port (usually 80).
org.opencastproject.server.url=http://200.1.17.239:8080

# This property specifies the comma separated list of addresses used to
for opencast to listen to.  (e.g. localhost or
# localhost,10.0.0.1). Host names or IP addresses can be used. Default
value is "0.0.0.0".
# Comment this out to make Opencast listen to all hosts.
org.ops4j.pax.web.listening.addresses=0.0.0.0

# The url of the remote service registry.  This is used in cases where
there is no direct connection to the service
# registry database such as capture agens running in protected
environments. This is typically true for capture agents
# and should be set to the url of a server running the actual
implementation of the service registry and the path to
# the service registry(admin, worker, etc. See the build profiles in
pom.xml for a complete list).
#org.opencastproject.serviceregistry.url=${org.opencastproject.server.url}
/services
```

```
# The base URL to use for publishing job locations. If left commented out,
the local server URL will be used.  Set this
# if you intend to support swapping servers with different IPs or host
names.
#org.opencastproject.jobs.url=${org.opencastproject.server.url}

# Whether to accept a job whose load exceeds the hostâ€™s max load
# Default: true
#org.opencastproject.job.load.acceptexceeding=true

# The directory where the system will store its processed files (including
temporary files).  This directory should
# be persistent between reboots (i.e., not /tmp)
org.opencastproject.storage.dir=/srv/opencast

# The directory where the configuration files for the Elasticsearch
indices are located. This directory should
# be persistent between reboots (i.e., not /tmp)
org.opencastproject.elasticsearch.config.dir=${karaf.etc}/index

# The username and password to present to other Opencast servers when
calling their REST endpoints.  The remote server
# must contain matching values.
org.opencastproject.security.digest.user=opencast_system_account
org.opencastproject.security.digest.pass=UTFSM16cast

# The number of times to retry a request if the nonce expires.
org.opencastproject.security.digest.nonce.retries=12

# The configuration property specifying the minimum amount of time in
seconds wait before retrying a request after
# a nonce timeout. Default is 300 seconds (5 minutes).
org.opencastproject.security.digest.nonce.base.time=300

# The maximum amount of time to wait in addition to the base time for a
random generator to add after a nonce timeout
# so that requests that timeout won't all try again at exactly the same
time. Default is 300 seconds (5 minutes).
org.opencastproject.security.digest.nonce.variable.time=300

# This changes the number of seconds from when an internal request is made
until a signed URL will expire. More
# specifically, the HTTP client needs access to internal storage areas
such as the working file repository as well as to
# distributed artifacts on the downloads and streaming servers, all of
which are protected by verification components.
# Default is 60 seconds as it shouldn't take longer than that to make a
request to a server. This will have no impact on
# a system where url signing is not configured. For more information
please see:
# http://docs.opencast.org/latest/admin/configuration/stream-
security/#configuration-of-url-signing-timeout-values
org.opencastproject.security.internal.url.signing.duration=60

# Optional account with administrative rights.
org.opencastproject.security.admin.user=admin
org.opencastproject.security.admin.pass=UTFSM16cast
org.opencastproject.security.admin.roles=ROLE_ADMIN,ROLE_USER,ROLE_OAUTH_U
SER

# Optional demo accounts with series memberships, but not administrative
rights.  If set to 'true', 1000 users will be
```

```
# loaded into the sample user directory. Usernames are user0, ...,
user999. Passwords are pass0, ..., pass999.
# Default: false
#org.opencastproject.security.demo.loadusers=false

# To enable the LDAP user provider, you must uncomment this line, and run
on a JVM that provides the
# "com.sun.jndi.ldap" package.
#org.osgi.framework.system.packages.extra=com.sun.jndi.ldap,com.sun.net.ss
l.internal.ssl

# The base URL of the streaming server (ususally
"rtmp://<SERVER_URL>/matterhorn-engage").
# ${org.opencastproject.server.url} can not be used, because the streaming
server does not use the HTTP protocol.
# Streaming is not included in the default workflow, since the Red5
streaming server is a 3rd party component that
# requires separate installation.
#org.opencastproject.streaming.url=rtmp://localhost/matterhorn-engage

# The directory where Opencast stores the streams
#org.opencastproject.streaming.directory=${org.opencastproject.storage.dir
}/streams

# The port to use for the streaming server, only used by a Wowza streaming
server.
#org.opencastproject.streaming.port=1935

# Some newer streaming server versions expect an "flv:" tag within the
rtmp URL.
# Not every RTMP-streaming server is compatible with this (i.e. nginx), so
this
# is the compatibility mode to the old syntax.
# true  = without "flv:" tag - old syntax
# false = with    "flv:" tag - new syntax
#org.opencastproject.streaming.flvcompatibility=true

# The directory to store media, metadata, and attachments for download
from the engage tool
org.opencastproject.download.directory=${org.opencastproject.storage.dir}/
downloads

# The base URL for media downloads.
org.opencastproject.download.url=${org.opencastproject.server.url}/static

# Relational Database configuration.  By default, Opencast uses an
embedded H2 database.  A standalone database server
# is recommended for production systems.  If you run the ddl script for
your db vendor (see docs/scripts/ddl/) manually,
# (this is recommended) set 'ddl-generation' to 'false'.
org.opencastproject.db.ddl.generation=false

# Enable ddl script generation.  This is used by developers to update the
ddl scripts, and should not be enabled in
# production.
org.opencastproject.db.ddl.script.generation=false

# db.vendor can be any of the values listed at under the
"eclipselink.target-database" section of
#
http://www.eclipse.org/eclipselink/documentation/2.4/jpa/extensions/p_targ
et_database.htm#target-database
# Common values include MySQL, PostgreSQL.
```

```
org.opencastproject.db.vendor=MySQL

# Opencast comes with the jdbc drivers for MySQL (com.mysql.jdbc.Driver)
and PostgreSQL (org.postgresql.Driver). To add
# other jdbcDrivers to the Opencast runtime, rebuild the matterhorn-db
module with your desired drivers.
org.opencastproject.db.jdbc.driver=com.mysql.jdbc.Driver

# The jdbc connection url, username, and password
org.opencastproject.db.jdbc.url=jdbc:mysql://localhost/opencast
org.opencastproject.db.jdbc.user=opencast
org.opencastproject.db.jdbc.pass=caster_feick

# This configures the connection to the Active MQ broker so that we can
send and receive messages.
# There are many settings for specifying the broker url:
http://activemq.apache.org/activemq-4-connection-uris.html
# Including with failover support details here:
http://activemq.apache.org/failover-transport-reference.html
# Default (url):
failover://(tcp://127.0.0.1:61616)?initialReconnectDelay=2000&maxReconnect
Attempts=2
# Default (username/password): None
activemq.broker.url=failover://(tcp://127.0.0.1:61616)?initialReconnectDel
ay=2000&maxReconnectAttempts=2
activemq.broker.username=admin
activemq.broker.password=password

# Default directory to use for embedded solr indexes.
# If not set, ${karaf.data}/solr-indexes will be used.
#org.opencastproject.solr.dir=${karaf.data}/solr-indexes

# Directory to store the embedded solr indices.  This should be a
persistent and stable directory.
# Default: ${org.opencastproject.solr.dir}/{archive, search, series,
scheduler, workflow}
#org.opencastproject.archive.solr.dir=${org.opencastproject.solr.dir}/arch
ive
#org.opencastproject.scheduler.solr.dir=${org.opencastproject.solr.dir}/sc
heduler
#org.opencastproject.search.solr.dir=${org.opencastproject.solr.dir}/searc
h
#org.opencastproject.series.solr.dir=${org.opencastproject.solr.dir}/serie
s
#org.opencastproject.workflow.solr.dir=${org.opencastproject.solr.dir}/wor
kflow

# URLs of dedicated Solr server to use.  Note that if thesw URLs are
specified, the local embedded Solr index as
# configured using `org.opencastproject.*.solr.dir` will be ignored. A
dedicated Solr server should be set up in order
# to enable running multiple instances of the related service. Please
consult http://lucene.apache.org/solr/ on how to
# set up a standalone Solr server.
#org.opencastproject.archive.solr.url=http://localhost:8983/solr/
#org.opencastproject.scheduler.solr.url=http://localhost:8983/solr/
#org.opencastproject.search.solr.url=http://localhost:8983/solr/
#org.opencastproject.series.solr.url=http://localhost:8983/solr/
#org.opencastproject.workflow.solr.url=http://localhost:8983/solr/

# The path to the archive repository
org.opencastproject.episode.rootdir=${org.opencastproject.storage.dir}/arc
hive
```

```
# The path to the repository of files used during media processing.
org.opencastproject.file.repo.path=${org.opencastproject.storage.dir}/file
s

# The base URL of the file server.  When using a shared filesystem between
servers, set all servers to use the same URL.
# Only then will hard linking between the working file repository and the
workspace be enabled to prevent downloads.
#org.opencastproject.file.repo.url=${org.opencastproject.server.url}

# The path to the working files (recommend using fast, transient storage)
org.opencastproject.workspace.rootdir=${org.opencastproject.storage.dir}/w
orkspace

# The scheduled period in seconds, at which a workspace cleanup operation
is performed.
# 86400 seconds equals 24 hours.
# Default value: -1 (Disable cleanup scheduler)
org.opencastproject.workspace.cleanup.period=86400

# The maximum age a file must reach in seconds before a deletion of the
file in the workspace cleanup operation is
# performed. 2592000 seconds equals 30 days.
# Default value: -1 (max age will never be reached)
org.opencastproject.workspace.cleanup.max.age=2592000

# The ID of the default workflow definition to run when media are ingested
org.opencastproject.workflow.default.definition=ng-schedule-and-upload

# The max number of ingests to allow at the same time. If more ingests try
than the max they will receive service
# unavailable. A value of 0 means that the server will accept all ingests.
# Default: 0
#org.opencastproject.ingest.max.concurrent=0

# Send server configuration data to the Opencast project to help us track
how people are using Opencast.  No security
# related information will be sent to the opencast project.  Comment this
out to disable this feature.
#org.opencastproject.anonymous.feedback.url=http://opencast.org/form/track
ing

# The maximum number of concurrent files to ingest from the inbox
directory
# Default: 1
#org.opencastproject.inbox.threads=1

# Engage Distribution delay between elements in milliseconds
# Default: 1000
#org.opencastproject.distribution.delay=1000

# Path to the ffmpeg binary. Its name is sufficient if the binary is in
the
# system path (default: ffmpeg)
#org.opencastproject.composer.ffmpeg.path=/opt/ffmpeg/ffmpeg

# Path to the ffprobe binary. Its name is sufficient if the binary is in
the
# system path (default: ffprobe)
#org.opencastproject.inspection.ffprobe.path=/opt/ffmpeg/ffprobe

# Path to the tesseract binary used by the text analyzer. Its name is
```

```
# sufficient if the binary is in the system path (default: tesseract)
#org.opencastproject.textanalyzer.tesseract.path=/opt/tesseract/tesseract

# Additional options for Tesseract like language or page segmentation
mode.
# The default are no additional options.
#org.opencastproject.textanalyzer.tesseract.options=-l eng -psm 3

# Path to the hunspell binary used by the matterhorn-dictionary-hunspell
# module. The default ist just "hunspell" which requires hunspell to be in
the
# search path.
#org.opencastproject.dictionary.hunspell.binary=/opt/hunspell/hunspell

# Command to use for filtering text by the matterhorn-dictionary-hunspell
# module. The command is appended to the hunspell binary path. It should
filter
# the text from stdin and print the recognized words to stdout. Usually
this
# should be a combination of "-G" and a list of dictionaries. The default
is to
# use "-d de_DE,en_GB,en_US -G".
#org.opencastproject.dictionary.hunspell.command=-i utf-8 -d
de_DE,en_GB,en_US -G

# The path for SoX command line used by audio normalization
#org.opencastproject.sox.path=/opt/sox/sox

# Email address of the server's admin.
org.opencastproject.admin.email=admin@localhost

# Location of the temporary directory to build zip archives.
# Default: ${karaf.data}/tmp/zip
#org.opencastproject.workflow.handler.workflow.ZipWorkflowOperationHandler
.tmpdir=${karaf.data}/tmp/zip

# Whether opencast should import test-data by the given csv file path.
# Default: false
#org.opencastproject.dataloader.testdata=false

# The path to the CSV file to import the data from it
org.opencastproject.dataloader.csv=${karaf.etc}/dataimport

# The location to store uploaded static files such as images and videos.
org.opencastproject.staticfiles.rootdir=${org.opencastproject.storage.dir}
/staticfiles

# True means to use a webserver to serve the static files but this will
not secure
# any of the files using user or organization security. If false it uses
# Opencast endpoints to serve and secure the files.
org.opencastproject.staticfiles.webserver.enabled=false

# The url to the location where the webserver serves the static file
uploads from. It will add the organization and uuid
# for the uploaded static file. If not set Opencast uses endpoints to
serve and secure the files.
org.opencastproject.staticfiles.webserver.url=${org.opencastproject.server
.url}/staticfiles/

# This is the maximum allowable size in bytes for a file to be uploaded.
If the property is missing or set to 0 it is
# disabled.
```

```
# Default is 1000000000 which is 1GB.
org.opencastproject.staticfiles.upload.max.size=1000000000

# The place for Karaf to put the lock file ensuring that Opencast is not
run twice at the same time.
karaf.lock.dir=${karaf.data}

# Setting or deactivating the remote shutdown port in Apache Karaf.
Commenting this out will make Karaf listen to a
# random shutdown port on localhost, announcing it via
${karaf.shutdown.port.file}. Setting this to -1 will deactivate
# the shutdown port. Note that the stop script is based on this and will
not work any longer if the port is deactivated.
#karaf.shutdown.port=-1

# Specifies the location of the port file for Opencast. It is used by the
shutdown script to send the shutdown command
# to the main process.
karaf.shutdown.port.file=${karaf.data}/port

# Specifies the location of the PID file for Opencast. It is used by the
shutdown script to synchronously shut down
# Opencast as it will wait for the process with the given process id.
Removing this will cause the network port to be
# used as fallback.
karaf.shutdown.pid.file=${karaf.data}/pid

# Command for shutting down Opencast. If the shutdown port is enabled,
Opencast will listen for this command to initiate
# the shut down procedure.
karaf.shutdown.command = 3500d4e3-ce93-4ae3-abb4-5e90cef4deb
```

7.4.2 org.opencastproject.organization-mh_default_org.cfg

```
# Organization (tenant) definition.
#
# With Opencast being capable of hosting multiple tenants per
installation, there needs
# to be one organization configuration per tenant.
#
# Therefore, in order to add another tenant to the installation, put a
file named
# org.opencastproject.organization-<tenant id>.cfg into Opencast's "load"
directory.
# Make sure to adjust at least the following fields to properly map the
new tenant to the
# installation:
#
#   - id
#   - server
#   - port
#   - prop.org.opencastproject.admin.ui.url
#   - prop.org.opencastproject.engage.ui.url
#

# Identifier of the tenant on this Opencast installation.
#
# The identifier is used to map data that belongs to this tenant only as
well as during
# lookup of related configuration entities such as the security settings
in etc/security.
#
# Value: a text identifier using [a-Z, 0..9, _, -] as characters.
```

```
#
id=mh_default_org

# Name of the organization (tenant).
#
# Value: <text>
#
name=Opencast Project

# Server name that identifies the organization (tenant) on this Opencast
installation.
#
# Use the public url that is assigned to this tenant. In out-of-the-box
installations
# with only one tenant, this value corresponds to the server.url in
etc/config.properties.
# Make sure to adjust these two keys accordingly:
#
#   - prop.org.opencastproject.admin.ui.url
#   - prop.org.opencastproject.engage.ui.url
#
# Value: <hostname without schema and port>
#
server=localhost

# Port for this tenant.
#
# When run behind a proxy server, set this to -1 since most proxies won't
forward the
# original port and set the corresponding field in the request to this
value.
#
# Value: <integer port number>
#
port=8080

# Identifier of the Administrative role.
#
# Value: a text identifier using [a-Z, 0..9, _, -] as characters.
# Default: ROLE_ADMIN
#
admin_role=ROLE_ADMIN

# Identifier of the Anonymous role.
#
# Value: a text identifier using [a-Z, 0..9, _, -] as characters.
# Default: ROLE_ANONYMOUS
#
anonymous_role=ROLE_ANONYMOUS

# Link to the Admin UI.
#
# Value: <a complete url with scheme and port>
# Default: ${org.opencastproject.server.url}
#
#prop.org.opencastproject.admin.ui.url=http://localhost:8080/admin

# Link to the Opencast documentation.
#
# If the property is specified, the admin ui help menu will link to the
documentation.
# The value needs to be a URL if set.
# Default: undefined
```

```
prop.org.opencastproject.admin.help.documentation.url=http://docs.opencast
.org

# Link to the Opencast REST service documentation for the local system.
#
# If the property is specified, the admin ui help menu will link to the
documentation.
# The value needs to be a URL if set.
# Default: undefined
prop.org.opencastproject.admin.help.restdocs.url=/rest_docs.html

# Link to the media module
#
# If the property is specified, the admin ui will show a media module icon
on the top-right that links
# to the media module.
# The value needs to be a URL if set.
# Default: undefined
prop.org.opencastproject.admin.mediamodule.url=${prop.org.opencastproject.
engage.ui.url}/engage/ui

# Link to a JIRA feedback widget collector.
#
# If the property is specified, the feedback button in the lower right
corner will
# link to the widget and allow to create issues right inside the project.
If the
# property is not specified, the feedback button will be hidden.
#
# Value: <a complete url with scheme and port>
#
#prop.org.opencastproject.admin.feedback.url=

# Link to the Engage UI.
#
# Value: <a complete url with scheme and port>
# Default: ${org.opencastproject.server.url}
#
#prop.org.opencastproject.engage.ui.url=http://localhost:8080

# Link to the RSS and atom feed base
#
# Value: <a complete url with scheme and port>
# Default: ${org.opencastproject.server.url}
#
#org.opencastproject.feed.url=http://localhost:8080

# Path to the logo that is displayed in the upper left part of the
# media module page.
#
# Value: <a complete url with scheme and port>
#
#prop.logo_mediamodule=/engage/ui/img/logo/opencast-icon.svg
prop.logo_mediamodule=http://opencast.elo.utfsm.cl/datos/logo_depto_elo.pn
g

# Path to the logo that is displayed in the Opencast 2.0 player
#
# Value: <a complete url with scheme and port>
#
#prop.logo_player=/engage/ui/img/logo/opencast.svg
prop.logo_player=http://opencast.elo.utfsm.cl/datos/logo_depto_elo_grande.
jpg
```

```
# Whether to show or hide the episodes tab in the admin UI.
#
# Values: true | false
# Default: true
#
prop.adminui.i18n_tab_episode.enable=true

# Whether to show or hide the users tab in the admin UI.
#
# Values: true | false
# Default: true
#
prop.adminui.i18n_tab_users.enable=false

# Pre- and Post-Roll value (in seconds) of the video editor
#
# Value: <integer>
# Default: 2
prop.adminui.prePostRoll=2

# Min. segment length value (in seconds) of the video editor
#
# Value: <integer>
# Default: 2
prop.adminui.minSegmentLength=2

# Define size of Chunksize on upload in KB.
#
# Value: <integer>
# Default: 2048
#
prop.adminui.chunksize=2048

# Define whether series metadata pre-population should occur when a
recording is scheduled.
#
# Values: true | false
# Default: false
#
prop.adminui.series_prepopulate.enable=false

# Defines whether to display the default login and username on the login
page
#
# Values: true | false
# Default: false
#
prop.login.display_defaults=true

# Flag for displaying the download link.
#
# Values: true, false
# Default:
#
prop.engageui.link_download.enable=false

# Flag for enabling annotations globally.
#
# Values: true, false
# Default: true
#
prop.engageui.annotations.enable=true
```

```
# Flag for displaying the media module.
#
# Values: true, false
# Default: true
#
prop.engageui.links_media_module.enable=true

# Flag for displaying a redirect when surfing with a mobile device
#
# Values: true, false
# Default: false
#
prop.engageui.link_mobile_redirect.enable=false

# URL for the mobile redirect.
#
# Value: A URL for a html URL tag
# Default: <none>
#
prop.engageui.link_mobile_redirect.url=http://opencast.org/

# Description for the mobile redirect.
#
# Value: <text>
# Default: <none>
#
prop.engageui.link_mobile_redirect.description=For more information have a
look at the official site.

# Choose the default video player
# comment in the following line for the old flash player
# prop.player=/engage/ui/player.html
# comment in the following line for the html5 player
prop.player=/engage/theodul/ui/core.html

# The default flavor of the master video (the video on the "left side" in
the video display)
prop.player.mastervideotype=presenter/delivery

# Play only videos that are tagged with one of these tags
# default: don't filter for tags at all
prop.player.allowedtags=engage-download,engage-streaming

# Play only videos that use these video/streaming formats.
# The format setting in the browser will be done for what is left after
these have been filtered
# with this setting.
# default: don't filter for video/streaming formats at all
# allowed values: hls,dash,rtmp,mp4,webm,audio
# prop.player.allowedformats=hls,dash,mp4

# Choose where the controls will be rendered
# Options:
#   - top
#   - bottom
prop.player.positioncontrols=bottom

# Choose which layout should be applied if none is stored in the local
storage of the browser
# Options:
#   - left
#   - right
```

```
#   - beside
#   - off (default)
prop.player.layout=off

# Choose which flavor (without subflavor) should be focused in the layout
if none is stored in the local storage.
# The focused flavor will only be used if prop.player.layout is not "off".
prop.player.focusedflavor=presentation

# Link back to Media Module from Theodul player (true|false)
# Default: false
prop.link_mediamodule=true

# Link back to Media Module from Theodul player (true|false)
# Default: false
prop.show_embed_links=true

# Shortcut definitions for the html5 player
# - modifier keys
#   - shift
#   - ctrl
#   - alt
#   - option
#   - meta
#   - command
#   - mod -- combines Mac 'command' and Windows/Linux 'ctrl'
# - special keys
#   - backspace
#   - tab
#   - enter
#   - return
#   - capslock
#   - esc
#   - escape
#   - space
#   - pageup
#   - pagedown
#   - end
#   - home
#   - left
#   - up
#   - right
#   - down
#   - ins
#   - del
# - any other key
#   - should be referenceable by name like a, /, $, *, or =
# - for more information have a look at the player docs
prop.player.shortcut.controls.playPause=space
prop.player.shortcut.controls.seekRight=right
prop.player.shortcut.controls.seekLeft=left
prop.player.shortcut.controls.jumpToBegin=backspace
prop.player.shortcut.controls.prevChapter=pagedown
prop.player.shortcut.controls.nextChapter=pageup
prop.player.shortcut.playbackrate.increase=t
prop.player.shortcut.playbackrate.decrease=T
prop.player.shortcut.volume.muteToggle=m
prop.player.shortcut.volume.up=9
prop.player.shortcut.volume.down=8
prop.player.shortcut.fullscreen.enable=mod+enter
prop.player.shortcut.fullscreen.cancel=escape
prop.player.shortcut.layout.focusNext=>
prop.player.shortcut.layout.focusPrev=<
```

```
prop.player.shortcut.layout.movePiP=p
prop.player.shortcut.layout.togglePiP=o
prop.player.shortcut.zoom.moveLeft=a
prop.player.shortcut.zoom.moveRight=d
prop.player.shortcut.zoom.moveUp=w
prop.player.shortcut.zoom.moveDown=s
prop.player.shortcut.zoom.in=+
prop.player.shortcut.zoom.out=-

prop.player.shortcut-
sequence=controls,volume,playbackrate,layout,zoom,fullscreen

# Shortcut definitions for video editor UI
#
# Format: prop.trim.<action>=<key>
#
prop.trim.previous_frame=left
prop.trim.next_frame=right
prop.trim.split_at_current_time=v
prop.trim.play_current_segment=c
prop.trim.play_current_pre_post=shift+c
prop.trim.set_current_time_as_inpoint=i
prop.trim.set_current_time_as_outpoint=o
prop.trim.play_pause=space
prop.trim.select_item_at_current_time=y
prop.trim.delete_selected_segment=delete
prop.trim.previous_marker=up
prop.trim.next_marker=down
prop.trim.play_ending_of_current_segment=n
```

7.5 CÓDIGOS DE CONFIGURACIÓN DE GALICASTER

7.5.1 Archivos de configuración

7.5.1.1 conf-dist.ini

Ubicación del archivo: `/usr/share/galicaster/`

Descripción: Archivo que contiene las configuraciones de conexión al servidor, la configuración de los registros, los plugins y configuración de notificaciones entre otras.

```
[basic]
admin = True
profile = default
stopdialog = True
quit = True
shutdown = False
swapvideos = True

[repository]
foldertemplate = gc_{hostname}_{year}-{month}-{day}T{hour}h{minute}m{second}

[logger]
```

```
path = /var/log/galicaster/galicaster.log
level = DEBUG
rotate = False
use_syslog = False

;; Metadata editor configuration
;; Two parameters are available:
;;   - blocked: a blank-separated list of metadata fields that will be non-
editable by the user
;;   - mandatory: a blank-separated list of metadata fields that MUST NOT be
blank in order to
;;                apply the changes to the mediapackage
;;
;; Both parameters admit the following values (unquoted):
;;   - 'title'
;;   - 'presenter' or 'creator'
;;   - 'description'
;;   - 'language'
;;   - 'series', 'ispartof' or 'isPartOf'
;[metadata]

; MATTERHORN COMMUNICATION
[ingest]
active = True
visible_tracks = False
legacy = False ;For MH 1.2 and 1.3 compatibility
manual =  immediately ;Possible values: none, immediately, nightly
scheduled = immediately ;Possible values: none, immediately, nightly
host = http://200.1.17.130:8080
username = matterhorn_system_account
password = CHANGE_ME
workflow = full
workflow-parameters = trimHold:true;videoPreview:true
multiple-ingest = False
connect_timeout = 2
timeout = 2

;; MATTERHORN SERIES
;; This section sets allows filtering series shown in the drop down list of the
metadata editor.
;; It accepts most of the filter values that Matterhorn endpoint accepts, namely:
;;    seriesId, seriesTitle, creator, contributor, publisher, rightsholder,
createdfrom,
;;    createdto, language, license, subject, abstract, description
;; According to Matterhorn documentation, the date-like filters (createdfrom and
createdto) must
;; follow the format yyyy-MM-dd'T'HH:mm:ss'Z'
;; In addition to the previous filters, the 'default' keyword accepts a series ID
that will appear
;; in the series list, no matter what.
;; The values to the parameters may include placeholders for certain environment
variables.
;; The only one supported currently is '{user}', that is substituted with the
current user name.
[series]
;default = XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
;contributor = {user}
; ... etc

[heartbeat]
short = 10
long = 60
```

```
night = 00:00

[sidebyside]
layout = sbs ;Side by side layout. Possible values: pip-screen, pip-camera

; OVERLAPPING AND RECORDINGS OPTIONS
[allows]
manual = True
start = False
stop = False
pause = True
overlap = False

; PLUGINS
[plugins]
nocursor = False
noaudiodialog = True
screensaver = False
cleanstale = False
forcedurationrec = False
shortcuts = True
checkrepo = False
rest = False
pushpic = False
setuprecording = False
hidetabs = False
retryingest = False
failovermic = False
notifycrash = False

[screensaver]
inactivity = 120

[cleanstale]
maxarchivaldays = 30
checkoninit = False

[forcedurationrec]
duration = 240

[audio]
min = -76
keep_closed = False

[rest]
host = 0.0.0.0
port = 8080

[notifycrash]
mailuser =
mailpass =
mailto =
mailsubject =
mailmessage =
smtpserver =
smtpport =

;; check_after is the minimum number of seconds to wait between checking
;; to see that all recordings are successfully ingested.
;; check_published is whether to check that the mediapackage is already published
;; on the matterhorn server and if so, sets the ingest state to succeeded
;; without reingesting.
```

```
;; check_nightly will mark the mediapackage as 'nightly' and postpone the ingest.
[retryingest]
check_after = 300
check_published = True
nightly = False

; 'device' is the pulse audio device that will be used to record the failover
audio track.
; 'failover_threshold' is the threshold rms amplitude at which the audio will be
replaced.
; this number is between -100 and 0
; the galicaster vumeter gives a rough visual indication of this value.
; 'audio_device' if multiple audio sources are used, this number corresponds
; to the audio track to replace. 1 = the first audio track.
[failovermic]
device = default
failover_threshold = -50
audio_device = 1

;; Configuration for the setuprecording plugin.
;; The following keys define the values that will be pre-filled in the metadata
editor
;; when the "REC" button is pressed:
;;   - title: Sets up the default value for the recording title
;;   - presenter or creator: Sets up the default "Presenter" value
;;   - description: Sets up the default "Description" value
;;   - language: Sets up the default "Language" value
;;   - series, ispartof or isPartOf: Sets up the default "Series" id. The ID must
exist,
;;                                   otherwise it will be ignored.
;; The following list of "placeholders" may be used in the previous values.
;;   - {user}: This string will be substituted by the current Unix login name
;;             For instance: "presenter = {user}" will set up the default
presenter
;;             value to the current user

;[setuprecording]
;; UI OPTIONS
;; This is the configuration section for the plugin hidetabs.py
;; The currently available configuration keys are:
;;   - hide: A space-separated list of tabs that will be hidden in the record UI
;;           Possible values are: 'events', 'recording' and 'status' (unquoted)
;;   - default: Name of the tab that will be initially displayed in the UI
;;              Possible values are: 'events', 'recording' and 'status'
;[hidetabs]

;; OPERATIONS OPTIONS
;; A space-separated list of operations that are to be hidden in the
;; corresponding pop up in the manager UI.
;; The possible values for both parameters are (unquoted):
;;     'ingest', 'exporttozip' and 'sidebyside'
;[operations]
;hide = ingest exporttozip sidebyside
;hide_nightly = ingest exporttozip sidebyside

;MEDIA MANAGER APPEARENCE
[color]
classic = false
none =  #FFF0AA ;yellow
nightly = #AEFFAE ; light green
pending = #AEFFAE ; light green
processing = #FFAE00 ; orange
```

```
done = #88FF88 ; green
failed = #FFAEAE ;red

[track1]
name = Bars
pattern = 0
caps = video/x-raw-yuv,framerate=25/1,width=640,height=480
color1 = 4294967295
color2 = 4278190080
location = default
file = CAMERA.avi
device = videotest
flavor = presenter
active = True

[track2]
name = Static
device = videotest
location = default
file = SCREEN.avi
flavor = presentation
caps = video/x-raw-yuv,framerate=25/1,width=640,height=480
pattern = 1
color1 = 4294967295
color2 = 4278190080
active = True

[track3]
name = Noise
device = audiotest
location = default
file = sound.mp3
flavor = presenter
pattern = pink-noise
frequency = 440
volume = 0.3
player = True
vumeter = True
amplification = 1.0
active = True
```

7.5.1.2 conf.ini

Ubicacion: `/etc/galicaster/`

Descripción: Archivo de configuración inicial con el cual inicia Galicaster, en este se agregan los parámetros de cómo se quiere que el programa inicie (Perfil de captura por defecto, tener confirmación al salir, etc)

```
[basic]
admin = True
quit = True
profile = Screen-Cam-Mixer
```

7.5.2 Ejemplos de archivos de perfiles

Estos archivos incluyen los perfiles de captura, que le indican a Galicaster donde se encuentran los dispositivos y en que formato los debe guardar. Cada dispositivo està separado por lo que se llama “track” y por su categoría (presenter o presentation), entre otras opciones.

Todos los perfiles se guardan en: `/etc/galicaster/profiles/`

Atención: Usted puede colocar cualquier nombre a estos archivos, solo deben cumplir tener la extensiòn “.ini”

7.5.2.1 capture.ini

Descripción: Perfil que sólo captura la presentación desde la tarjeta Datapath Vision y desde la entrada de línea de la tarjeta de audio.

```
[data]
name = Screen-Capture

[track1]
name = Slides
device = datapath
location = /dev/screen
file = SCREEN.avi
flavor = presentation
caps = video/x-raw-yuv,framerate=30/1,width=1920,height=1080

[track2]
name = AudioSource
device = pulse
location = alsa_input.pci-0000_00_1b.0.analog-stereo
file = sound.mp3
flavor = presentation
vumeter = True
player = True
amplification = 1.0
```

7.5.2.2 lecture.ini

Descripción: Perfil, que permite la captura tanto del presentador con la cámara, como al mismo tiempo de la presentación a través de la capturadora Datapath y audio por la entrada de línea del computador.

```
[data]
```

```
name = Screen-Cam-Mixer

[track1]
name = Webcam
device = v4l2
location = /dev/camera
file = WEBCAM.avi
flavor = presenter
caps = image/jpeg,framerate=24/1,width=1280,height=720

[track2]
name = AudioSource
device = pulse
location = alsa_input.pci-0000_00_1b.0.analog-stereo
file = sound.mp3
flavor = presenter
vumeter = True
player = True
amplification = 1.0

[track3]
name = Slides
device = datapath
location = /dev/screen
file = SCREEN.avi
flavor = presentation
caps = video/x-raw-yuv,framerate=30/1,width=1920,height=1080
```

lecture.ini (Módulo Móvil)

Cambia la tarjeta Datapath por una tarjeta Blackmagic Design Decklink mini recorder.

```
[data]
name=Screen+Cam+Mixer

[track1]
name = Webcam
file = webcam.avi
location = /dev/camera
caps = image/jpeg,format=YUY2,framerate=24/1,width=1280,height=720
device = v4l2
flavor = presenter

[track2]
name = AudioSource
device = pulse
location = alsa_input.pci-0000_00_1f.3.analog-stereo
file = sound.mp3
flavor = presenter
vumeter = True
player = True
amplification = 1.0

[track3]
name = blackmagic
file = screen.avi
input = hdmi
input-mode = 1080i60
location = /dev/blackmagic/io0
subdevice = 0
device = blackmagic
flavor = presentation
audio-input = none
vumeter = false


player = false
```

webcam.ini

Descripción: Perfil que sólo captura por la cámara, es igual a “lecture.ini”, pero sin la pista de la capturadora de presentaciones.

```
[data]
name = Webcam-Mixer

[track1]
name = Webcam
device = v4l2
location = /dev/camera
file = WEBCAM.avi
flavor = presenter
caps = image/jpeg,framerate=24/1,width=1280,height=720

[track2]
name = AudioSource
device = pulse
location = alsa_input.pci-0000_00_1b.0.analog-stereo
file = sound.mp3
flavor = presenter
vumeter = True
```

```
player = True
amplification = 1.0
```

7.6 FICHAS TÉCNICAS DE EQUIPAMIENTO UTILIZADO

En esta sección, se encuentran las fichas técnicas del hardware de captura y del servidor utilizado para esta tarea.

7.6.1 Servidor de difusión

- Dell PowerEdge R430

PowerEdge R430

Powerful, two-socket entry rack server delivers outstanding performance, configuration flexibility, high availability and intuitive management in a short-height (1U), short-depth (24-inch) chassis.

Feature	PowerEdge R430 technical specification
Form factor	1U rack server
Processor	Intel® Xeon® processor E5-2600 v4 product family **Processor sockets**: 2 **Chipset**: C610 **Internal interconnect**: Up to 9.6GT/s **Cache**: 2.5MB per core; core options: 4, 6, 8, 10, 12, 14, 16, 18, 20
Memory	DDR4 DIMMs at up to 2400MT/s 12 DIMM slots: 4GB/8GB/16GB/32GB
I/O slots	2 PCIe 3.0
Storage	SAS, SATA, nearline SAS, SSD Multiple R430 chassis available: • 10 x 2.5" hot-plug drives • 8 x 2.5" hot-plug drives • 4 x 3.5" hot-plug drives • 4 x 3.5" cabled drives
RAID controllers	**Internal controllers**: PERC S130 (SW RAID), PERC H330, PERC H730, PERC H730P **External HBAs (RAID)**: PERC H830
Communications	4 x 1GbE LOM Click here for R430 supported network interface cards (NICs) and host bus adapters (HBAs) and scroll to "Additional Network Cards" section.
Power supplies	450W, 550W PSU
Systems management	**Systems management:** IPMI 2.0 compliant; Dell OpenManage Essentials; Dell OpenManage Mobile; Dell OpenManage Power Center **Remote management** iDRAC8 with Lifecycle Controller, iDRAC8 Express (default), iDRAC8 Enterprise (upgrade) 8GB vFlash media (upgrade), 16GB vFlash media (upgrade) **Dell OpenManage Integrations:** • Dell OpenManage Integration Suite for Microsoft® System Center • Dell OpenManage Integration for VMware® vCenter™ **Dell OpenManage Connections:** • HP Operations Manager, IBM Tivoli® Netcool® and CA Network and Systems Management • Dell OpenManage Plug-in for Oracle® Database Manager
Optional supported hypervisors	Citrix® XenServer, VMware vSphere® ESXi®, Red Hat® Enterprise Virtualization®
Operating systems	Microsoft Windows Server® 2008 R2 Microsoft Windows Server 2012 Microsoft Windows Server 2012 R2 Novell® SUSE® Linux Enterprise Server Red Hat Enterprise Linux VMware vSphere ESXi For more information on the specific versions and additions, visit Dell.com/OSsupport.
Rack support	ReadyRails™ II sliding rails for tool-less mounting in 4-post racks with square or unthreaded round holes or tooled mounting in 4-post threaded hole racks, with support for optional tool-less cable management arm.
OEM-ready version available	From bezel to BIOS to packaging, your servers can look and feel as if they were designed and built by you. For more information, visit Dell.com/OEM.
Recommended support	Dell ProSupport Plus for critical systems or Dell ProSupport for premium hardware and software support for your PowerEdge solution. Consulting and deployment offerings are also available. Contact your Dell representative today for more information. Availability and terms of Dell Services vary by region. For more information, visit Dell.com/ServiceDescriptions.

7.6.2 Agente de captura estacionario

- Datapath Vision RGB e1s

VisionRGB-E1S

Single Channel RGB/DVI/HD Capture Card

Advanced Graphics Display Technology

SPECIFICATION

Board Format	PCI-e x4 low profile card, 68.9mm x 167.6mm. PCI-e bus master with scatter gather DMA providing. maximum data rate of 650MB/s.
Connectors	One DVI-I type connector.
Maximum Sample Rate	170 MHz analogue RGB or 165MHz DVI. Analog modes up to 340MHz pixel clock can be captured using dual-pass sampling.
Video Sampling	RGB: 24 bits per pixel / 8-8-8 format.
Video Capture Memory	32 MB, triple buffered.
Analog RGB Mode Support	640 x 480, 800 x 600, 1024 x 768, 1280 x 1024, 1600 x 1200, 1920 x 1080, 2048 x 1536, custom modes.
DVI Single Link Mode Support	640 x 480, 800 x 600, 1024 x 768, 1280 x 1024, 1600 x 1200, 1920x1080, 1920 x 1200, and custom modes.

HD Modes	1080p,1080i, 720p, 576p, 576i, 480p and 480i using a Component-DVI connector (HDCP not supported).
Input Mode Detection	Automatic detection of input modes in hardware, enabling the tracking of mode changes in the source signal.
Pixel Transfer Formats	RGB: 5-5-5, 5-6-5 or 8-8-8 (24bit/32bit) pixels. YUV: 4:2:2. MONO: 8bit.
Update Rate	User defined, captured frame rate will match the source.Providing max data rate (650MB/s) is not exceeded. Multi-buffered to eliminate tearing artifacts.
Video Format Options	Analog RGB plus HSync and VSync (5 wire). Analog RGB with Composite Sync (4 wire). Analog RGB with Sync on Green/YPbPr (3 wire). DVI Single Link.
Operating System Support	Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows Server 2012, Windows 7, Windows 8 and Linux support See www.datapath.co.uk for updates.

Power Requirements	Max current at +3.3V – 0.25A. Max current at +12V – 0.5A. Max power – 6.8 Watts.
Operating Temperature	0 to 35 deg C / 32 to 96 deg F
Storage Temperature	-20 to 70 deg C / -4 to 158 deg F
Relative Humidity	5% to 90% non-condensing.
Warranty	3 years

DESCRIPTION

The VisionRGB-E1S has a single capture channel supporting up to 1920x1080 DVI or 2048x1536 Analog resolution.

The VisionRGB-E1S captures the Analog/DVI data and triple buffers it into onboard storage. This data is then copied using DMA to the host system for display, storage or streaming.

When a Datapath graphics card is used, the VisionRGB-E1S transfers the data directly to the graphics card thereby increasing performance. The VisionRGB-E1S sends the relevant portions of each captured image to each display channel and instructs each channel to use its graphics engine to render the data. This fully utilises the hardware and dramatically increases performance.

When the RGB/DVI data is displayed on a non Datapath graphics card, the VisionRGB-E1S sends the data to system memory or direct to the graphics card, dependant on the software used for display.

The VisionRGB-E1S is an ideal solution for applications that require the capture of Analog or DVI sources in real time. Typical applications include:

- Viewing Analog or DVI sources from PCs, MACs, Industrial/medical equipment, cameras and other video equipment
- Streaming video applications
- Video/Data Wall Controllers

STREAMING SUPPORT

DirectShow drivers for WDM Streaming driver supports the following applications, to encode, record and stream video over networks or the Internet:

- Microsoft Media Encoder®
- VLC
- StreamPix
- VirtualDub
- Adobe Flash Encoder
- AMCap
- Any other DirectShow encoding software

- Logitech C920 HD Pro Webcam

Logitech

LOGITECH® HD PRO WEBCAM C920

Hi-Def Yourself.

Full HD 1080p video that's faster, smoother and works on more computers.

Say it bigger. Say it better. Full HD 1080p calls and clips in widescreen let friends and family see you in the sharpest, smoothest video quality available. Video call, vblog, and share your videos with ease. Plus it works seamlessly with your favorite applications. The term 'webcam' doesn't quite do it justice.

WORKS WITH

Windows® XP (SP2 or higher), Windows Vista® and Windows® 7 or (32-bit or 64-bit)

For HD 1080p video recording:

- 2.4 GHz Intel® Core 2 Duo processor
- 2 GB RAM or more
- Hard drive space for recorded videos
- USB 2.0 port

For full HD 1080p and 720p video calling:

Minimum upload and download at least a 1 Mbps connection for 720p video calling, and 2 Mbps for 1080p.

(Requirements for H.264 and MJPEG formats vary)

Visit your preferred video calling provider's website for information on video-calling software requirements.

WHAT'S IN THE BOX?

- Webcam with 6-foot cable
- Logitech webcam software with Logitech Vid HD (for PC)
- User documentation
- Tripod-ready base
- 2-year limited hardware warranty

Which Logitech HD Webcam is Right for You?

Find even more at logitech.com

	HD WEBCAM C310	HD WEBCAM C615	HD PRO WEBCAM C920
	GOOD	BETTER	BEST
What to look for			
High-definition (HD) video	HD 720p	Full HD 1080p	Full HD 1080p with H.264
Photo quality	5MP	8MP	15MP
Focus type	Always focused*	Autofocus**	20-step Autofocus**
Built-in noise-cancelling mic	Single	Single	Dual stereo
Auto light correction	Premium	Premium	Premium
1-click video upload to Facebook®, YouTube™ and Twitter™* **	HD 720p	Full HD 1080p	Full HD 1080p
Video Effects	Standard	Premium	Premium
360-degree full-motion camera rotation		•	
Portability		Fold-and-go webcam Tripod-ready base	Tripod-ready base
Compatibility			
Windows® XP, Windows Vista®, Windows® 7	•	•	•
Mac® OS X 10.5 or higher		•	

* 40 cm and beyond
** 10 cm and beyond
*** Twitter uploading requires software download on the C920. To download, go to logitech.com/support-downloads

7.6.3 Agente de captura móvil

- Blackmagic Design Decklink mini Recorder

DeckLink Mini Recorder

Record uncompressed 10-bit video from SDI and HDMI video sources. DeckLink Mini Recorder switches between SD and HD video formats and is perfect for building ingest servers or other video solutions where you need to capture only in a low profile card! Includes two PCIe shields for both full height and low profile slots.

$145

Connections

SDI Video Input
1 x 10-bit SD/HD switchable

HDMI Video Input
1 x HDMI type A connector.

SDI Audio Input
8 channels embedded in SD and HD.

HDMI Audio Input
8 Channels embedded in SD and HD.

Computer Interface
PCI Express 1 lane, compatible with 1, 4, 8 and 16 lane PCIe slots.

Video Standards

SD Format Support
525i29.97 NTSC, 625i25 PAL

HD Format Support
720p50, 720p59.94, 720p60
1080p23.98, 1080p24, 1080p25, 1080p29.97, 1080p30
1080PsF23.98, 1080PsF24, 1080PsF25, 1080PsF29.97, 1080PsF30
1080i50, 1080i59.94, 1080i60

SDI Compliance
SMPTE 259M and 292M.

SDI Metadata Support
VITC read for 3:2 pulldown removal. VANC capture using up to 3 lines of video in file. HD RP188 and closed captioning.

Audio Sampling
Television standard sample rate of 48 kHz and 24 bit.

Video Sampling
4:2:2

Color Precision
10-bit

Color Space
REC 601, REC 709

HDMI Configuration
HDMI automatically configures to connected display.

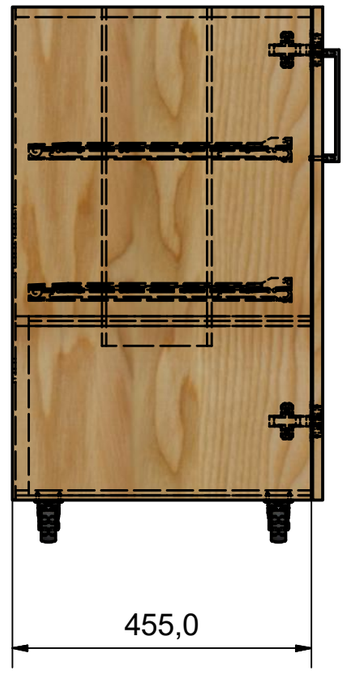
HDMI Resolution
Pixel for pixel HD resolution input to connected device.

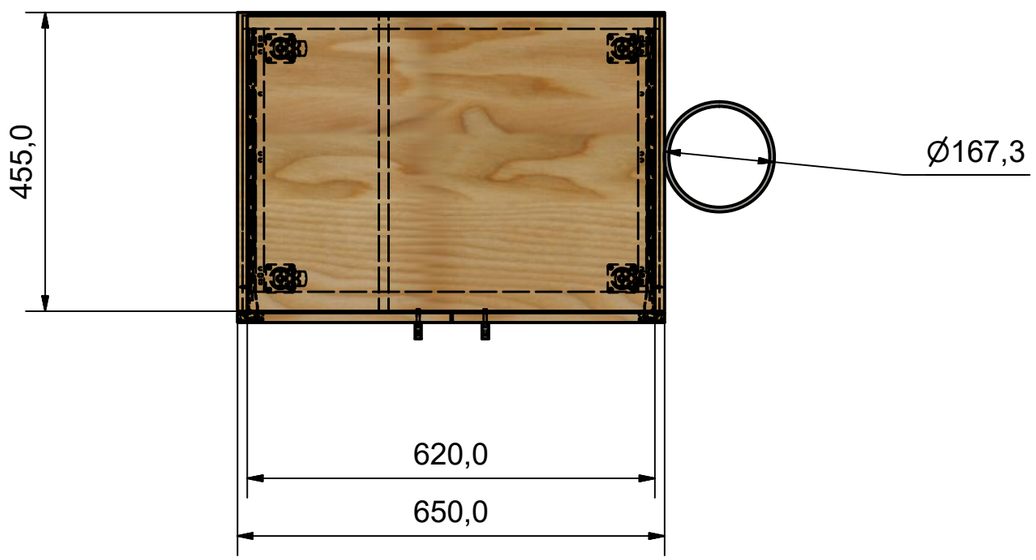
HDMI Color Precision
10-bit

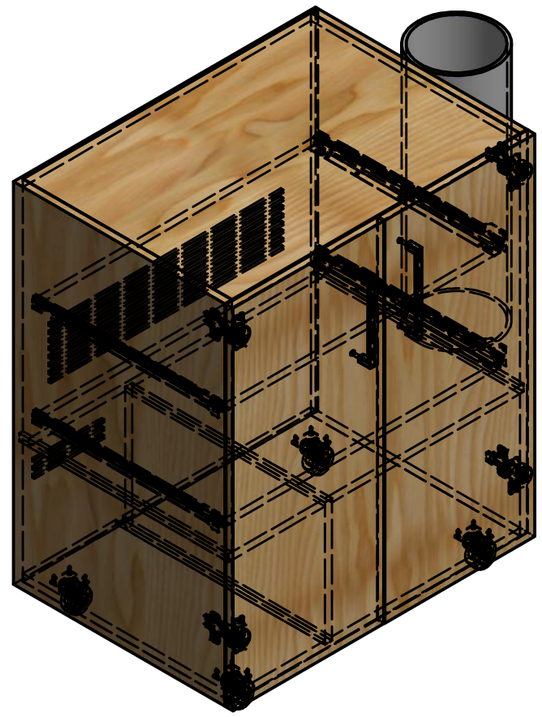
Multiple Rate support
SDI and HDMI are switchable between standard definition and high definition.

Copy Protection
HDMI input is unable to capture from copy protected HDMI sources. Always confirm copyright ownership before capture or distribution of content.

7.7 PLANOS DE DISEÑO 3D MÓDULO MÓVIL

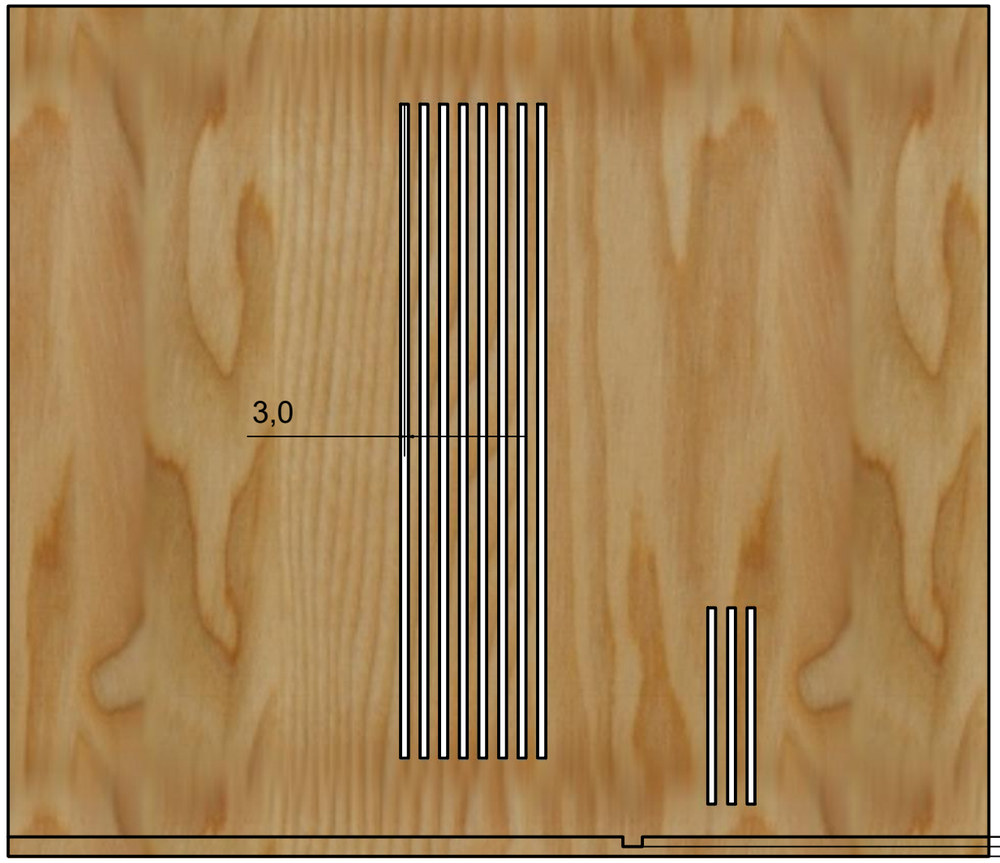


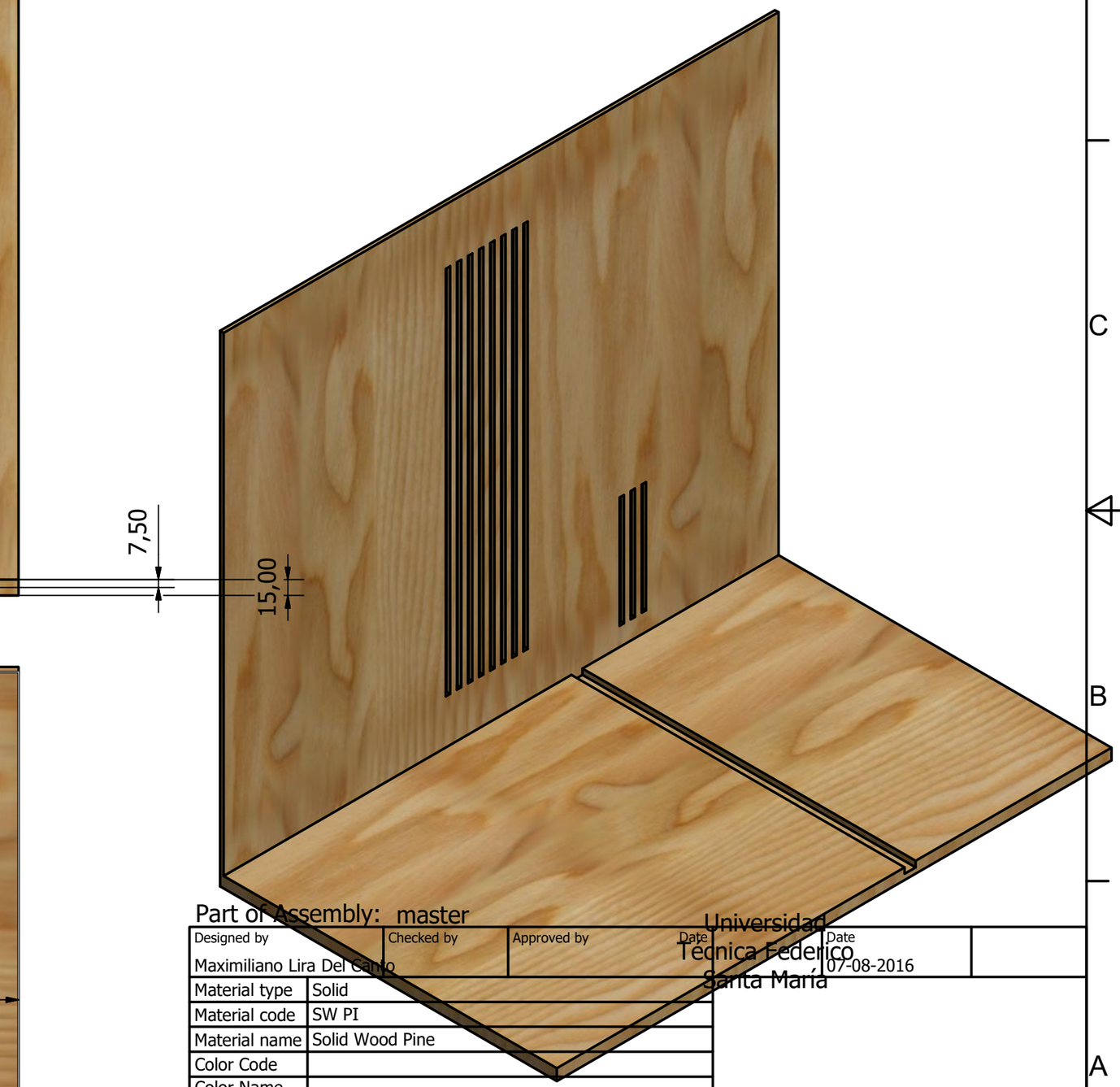


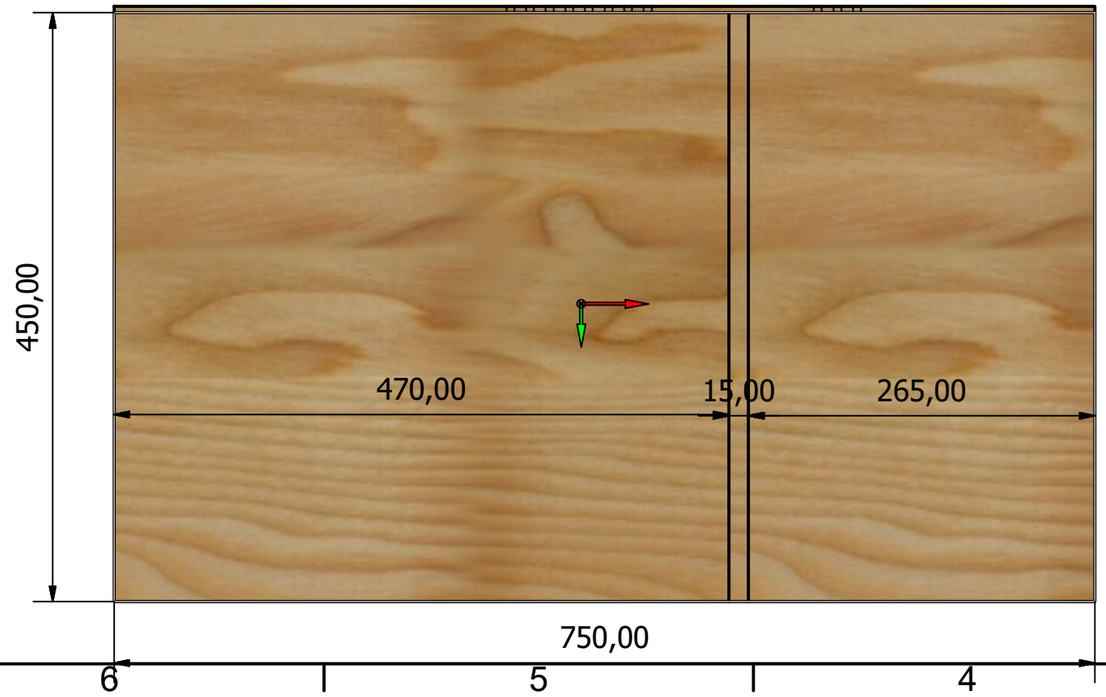


Designed by	Checked by	Approved by	Date		Date	
Maximiliano Lira Del	Canto				07-08-2016	

Universidad Técnica Federico Santa María	Módulo Móvil USMcast		
	master	Edition 2	Sheet 1 / 14







Part of Assembly: master

Designed by	Checked by	Approved by	Date	Date	
Maximiliano Lira Del Canto			Universidad Técnica Federico Santa María	07-08-2016	

Material type	Solid
Material code	SW PI
Material name	Solid Wood Pine
Color Code	
Color Name	
Material WP L	750
Material WP W	450
Material WP T	15
WP Type	Fill Workpiece

Lateral Izquierdo	Edition 2	Sheet 2 / 14

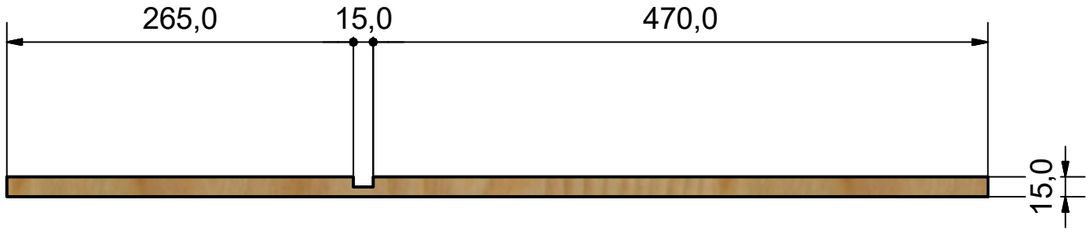
Part of Assembly: master

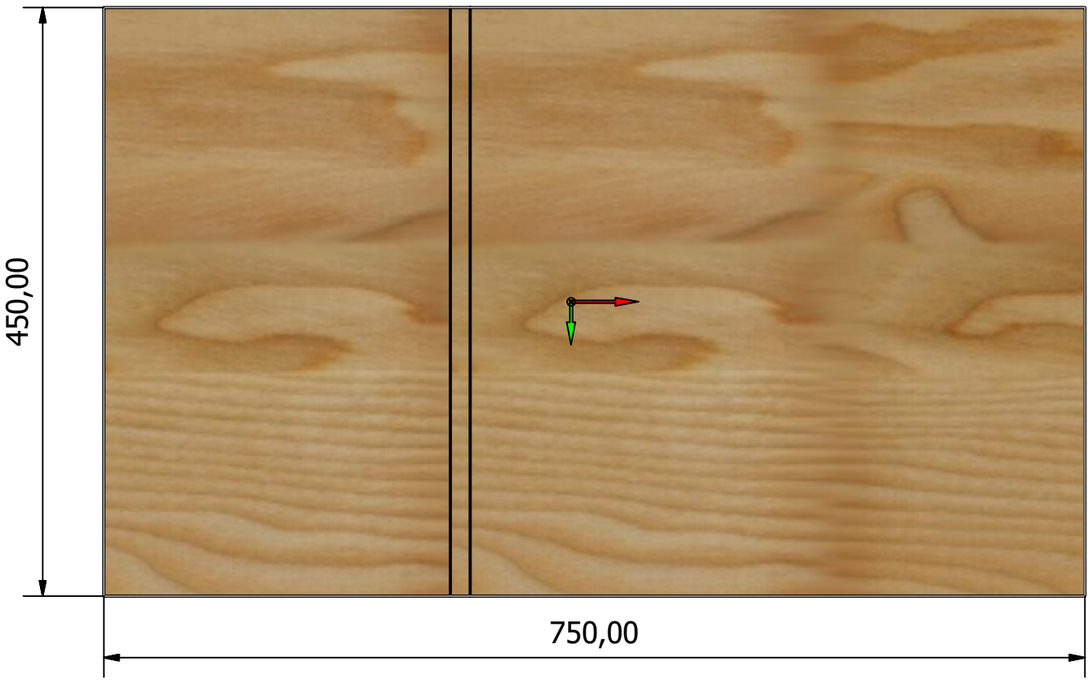
Designed by	Checked by	Approved by	Date	Date	
Maximiliano Lira Del Canto				07-08-2016	

Universidad Técnica Federico Santa María

Material type	Solid
Material code	SW PI
Material name	Solid Wood Pine
Color Code	
Color Name	
Material WP L	620
Material WP W	450
Material WP T	15
WP Type	Fill Workpiece

Tapa Superior	Edition	Sheet
	2	3 / 14





Part of Assembly: master

Designed by	Checked by	Approved by	Date	Date
Maximiliano Lira Del Canto				07-08-2016

Material type	Solid
Material code	SW PI
Material name	Solid Wood Pine
Color Code	
Color Name	
Material WP L	750
Material WP W	450
Material WP T	15
WP Type	Fill Workpiece

Universidad Técnica Federico Santa María

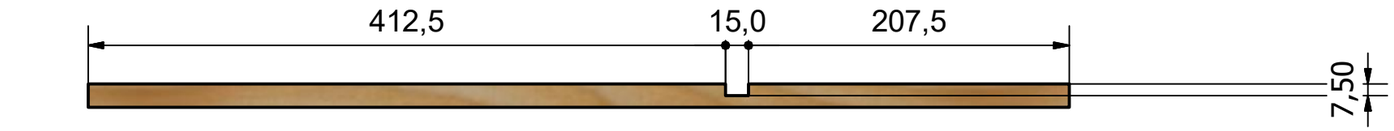
	Edition	Sheet
Lateral Derecho	2	4 / 14

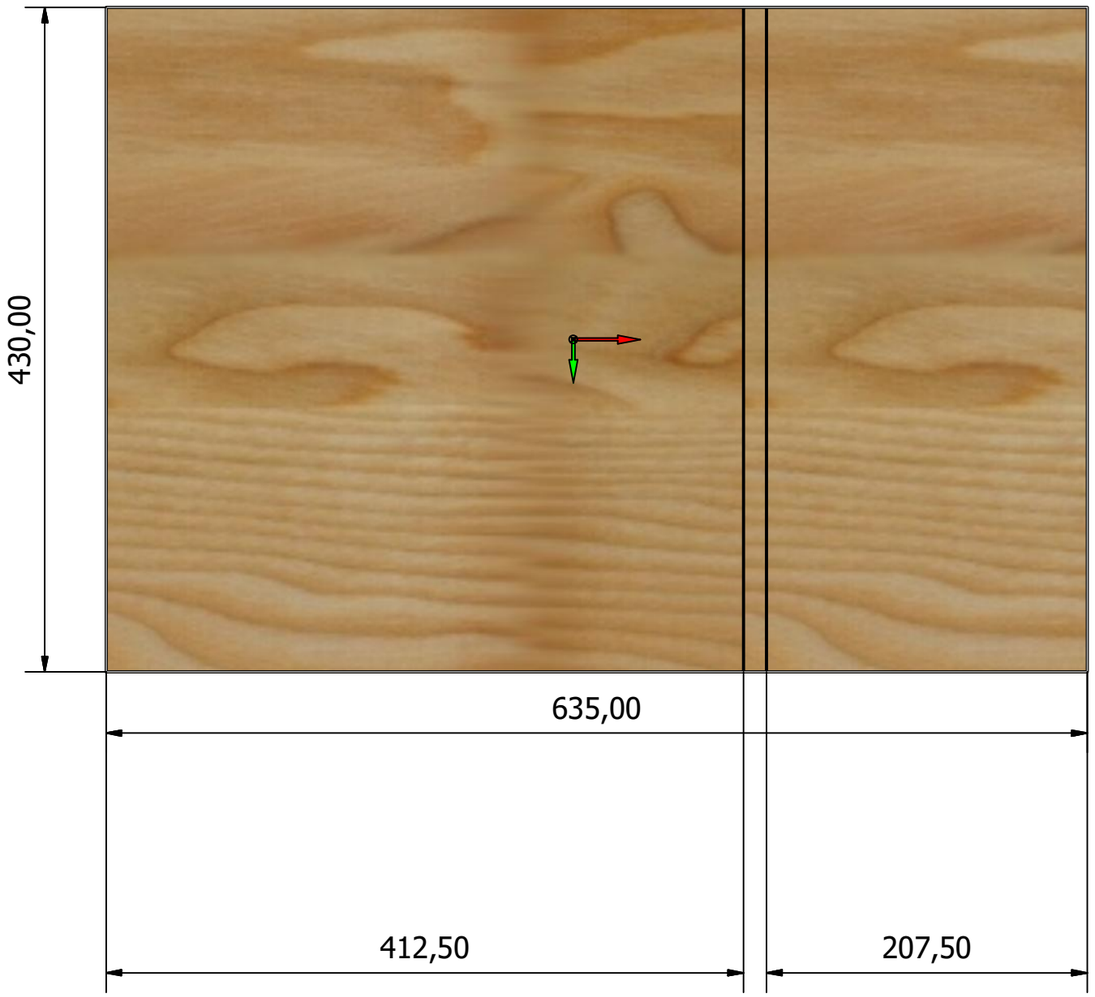
Part of Assembly: master

Designed by Maximiliano Lira Del Canto	Checked by	Approved by	Date	Date 07-08-2016	
Material type	Solid				
Material code	SW PI				
Material name	Solid Wood Pine				
Color Code					
Color Name					
Material WP L	620				
Material WP W	450				
Material WP T	15				
WP Type	Fill Workpiece				

Universidad Técnica Federico Santa María

Tapa Inferior	Edition 2	Sheet 5 / 14





Part of Assembly: master

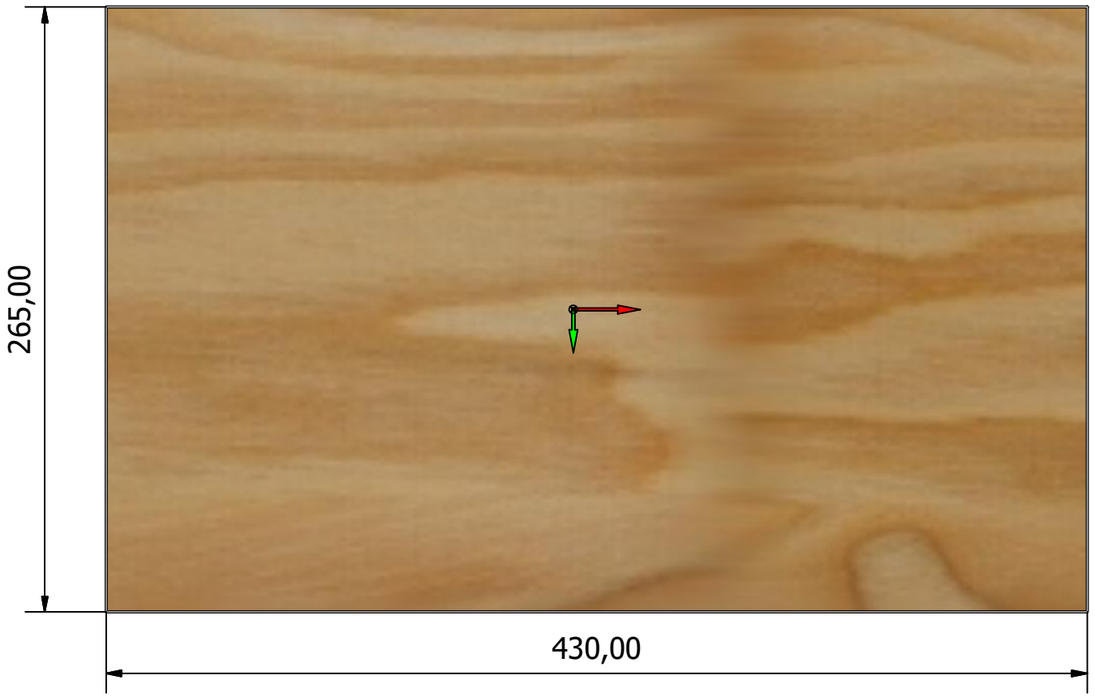
Designed by	Checked by	Approved by	Date	Date
Maximiliano Lira Del Canto				07-08-2016

Material type	Solid
Material code	SW PI
Material name	Solid Wood Pine
Color Code	
Color Name	
Material WP L	635
Material WP W	430
Material WP T	15
WP Type	Fill Workpiece

Universidad Técnica Federico Santa María

Separador Horizontal

Edition	Sheet
2	6 / 14



Part of Assembly: master

Designed by	Checked by	Approved by	Date	Date
Maximiliano Lira Del Canto				07-08-2016

Material type	Solid
Material code	SW PI
Material name	Solid Wood Pine
Color Code	
Color Name	
Material WP L	430
Material WP W	265
Material WP T	15
WP Type	Fill Workpiece

Universidad Técnica Federico Santa María

	Edition	Sheet
Separador Vertical	2	7 / 14

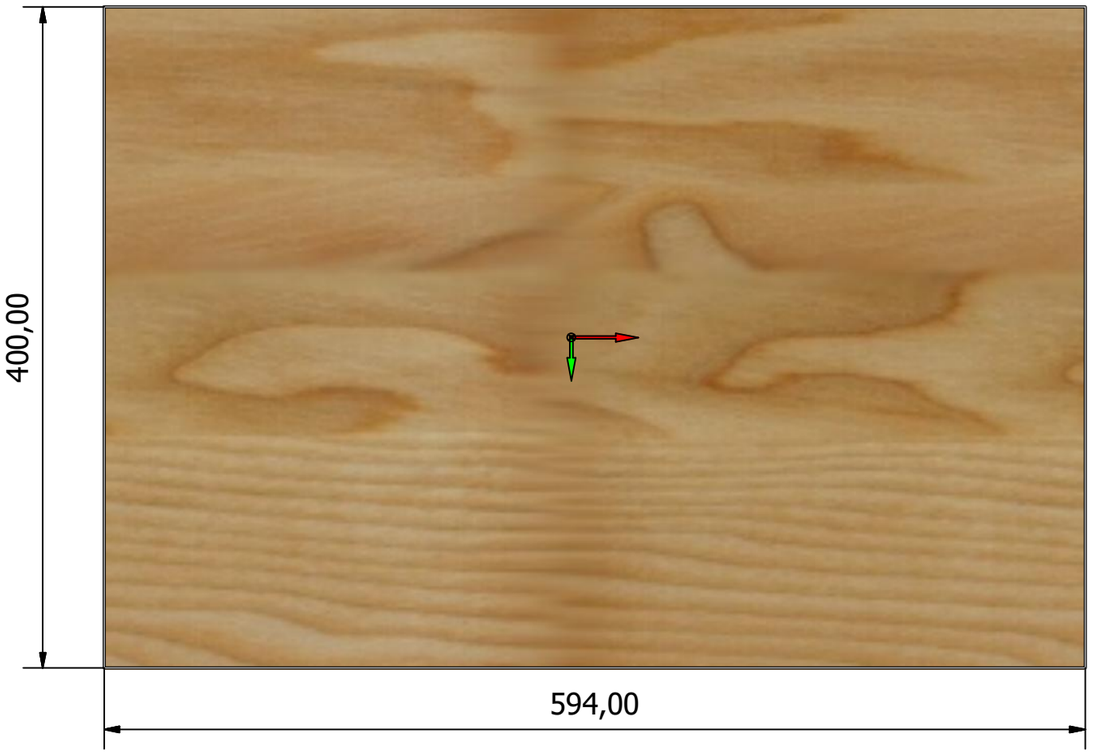
Part of Assembly: master

Designed by	Checked by	Approved by	Date	Date
Maximiliano Lira Del Canto				07-08-2016

Material type	Solid
Material code	SW PI
Material name	Solid Wood Pine
Color Code	
Color Name	
Material WP L	594
Material WP W	400
Material WP T	15
WP Type	Fill Workpiece

Universidad Técnica Federico Santa María

Bandeja Audio	Edition	Sheet
	2	8 / 14



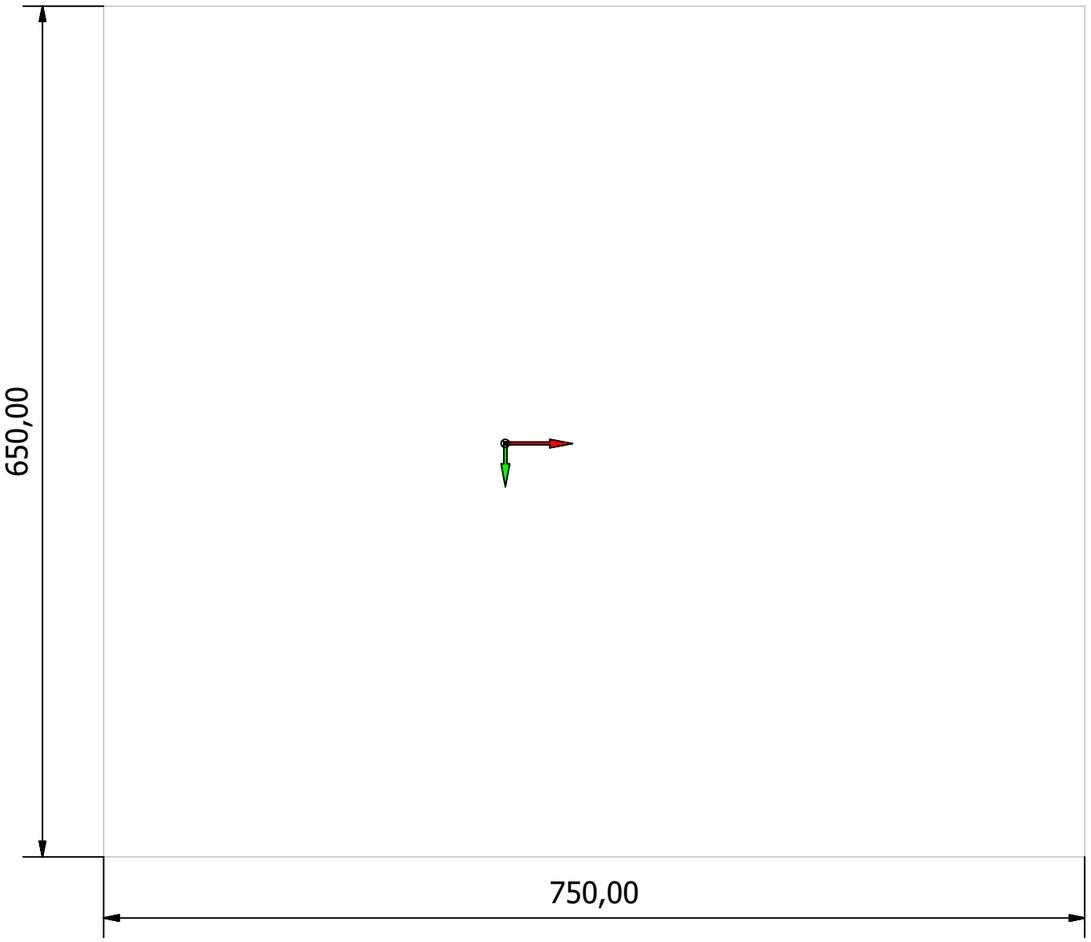
Part of Assembly: master

Designed by	Checked by	Approved by	Date	Date
Maximiliano Lira Del Canto				07-08-2016

Material type	Solid
Material code	SW PI
Material name	Solid Wood Pine
Color Code	
Color Name	
Material WP L	594
Material WP W	400
Material WP T	15
WP Type	Fill Workpiece

Universidad Técnica Federico Santa María

Bandeja PC	Edition	Sheet
	2	9 / 14



Part of Assembly: master

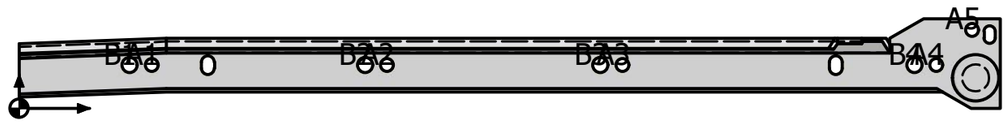
Designed by	Checked by	Approved by	Date	Date
Maximiliano Lira Del Canto				07-08-2016

Material type	Laminated Board
Material code	LCB
Material name	Laminated Cardboard
Color Code	F01
Color Name	Facade 01 group color
Material WP L	750
Material WP W	650
Material WP T	5
WP Type	Fill Workpiece

Universidad Técnica Federico Santa María

Tapa posterior	Edition	Sheet
	2	10 / 14

Holes Table										
Hole	Xdim	Ydim	Zdim	D	Depth	D2	Depth2	Type	XYang	Xang
A1	54,00	17,60	-8,90	5,30	1,30			T	180°	
A2	150,00	17,60	-8,90	5,30	1,30			T	180°	
A3	246,00	17,60	-8,90	5,30	1,30			T	180°	
A4	374,00	17,60	-8,90	5,30	1,30			T	180°	
A5	388,50	32,00	-8,90	5,30	1,30			T	180°	
B1	45,00	17,60	-8,90	6,30	1,30			T	180°	
B2	141,00	17,60	-8,90	6,30	1,30			T	180°	
B3	237,00	17,60	-8,90	6,30	1,30			T	180°	
B4	365,00	17,60	-8,90	6,30	1,30			T	180°	



Part of Assembly: master

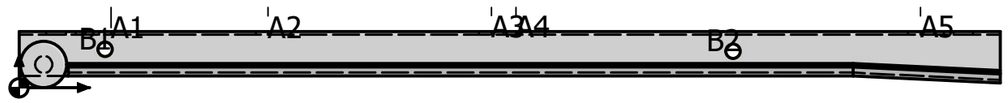
Designed by	Checked by	Approved by	Date	Date
Maximiliano Lira Del Canto				07-08-2016

Material type	
Material code	
Material name	
Color Code	
Color Name	
Material WP L	
Material WP W	
Material WP T	
WP Type	

Universidad Técnica Federico Santa María

Coulisses_13	Edition	Sheet
	2	11 / 14

Holes Table										
Hole	Xdim	Ydim	Zdim	D	Depth	D2	Depth2	Type	XYang	Xang
A1	35,00	21,65	-16,40	5,00	1,30			T	90°	90°
A2	99,00	21,65	-16,40	5,00	1,30			T	90°	90°
A3	190,00	21,65	-16,40	5,00	1,30			T	90°	90°
A4	200,00	21,65	-16,40	5,00	1,30			T	90°	90°
A5	365,00	21,65	-16,40	5,00	1,30			T	90°	90°
B1	35,00	15,65	-8,10	6,00	1,30			T	180°	
B2	291,00	14,90	-8,10	6,00	1,30			T	180°	



Part of Assembly: master

Designed by	Checked by	Approved by	Date	Date
Maximiliano Lira Del Canto				07-08-2016

Material type	
Material code	
Material name	
Color Code	
Color Name	
Material WP L	
Material WP W	
Material WP T	
WP Type	

Universidad Técnica Federico Santa María

Coulisses_37	Edition	Sheet
	2	12 / 14

Holes Table										
Hole	Xdim	Ydim	Zdim	D	Depth	D2	Depth2	Type	XYang	Xang
A1	54,00	18,90	-8,90	5,30	1,30			T	180°	
A2	150,00	18,90	-8,90	5,30	1,30			T	180°	
A3	246,00	18,90	-8,90	5,30	1,30			T	180°	
A4	374,00	18,90	-8,90	5,30	1,30			T	180°	
A5	388,50	4,50	-8,90	5,30	1,30			T	180°	
B1	45,00	18,90	-8,90	6,30	1,30			T	180°	
B2	141,00	18,90	-8,90	6,30	1,30			T	180°	
B3	237,00	18,90	-8,90	6,30	1,30			T	180°	
B4	365,00	18,90	-8,90	6,30	1,30			T	180°	

Part of Assembly: master

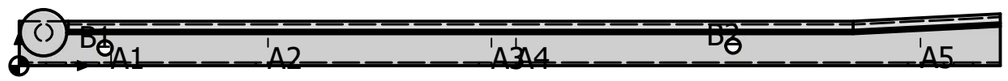
Designed by	Checked by	Approved by	Date	Date
Maximiliano Lira Del Canto				07-08-2016

Material type	
Material code	
Material name	
Color Code	
Color Name	
Material WP L	
Material WP W	
Material WP T	
WP Type	

Universidad Técnica Federico Santa María

Coulisses_14	Edition	Sheet
	2	13 / 14

Holes Table										
Hole	Xdim	Ydim	Zdim	D	Depth	D2	Depth2	Type	XYang	Xang
A1	35,00	1,30	-16,40	5,00	1,30			T	90°	270°
A2	99,00	1,30	-16,40	5,00	1,30			T	90°	270°
A3	190,00	1,30	-16,40	5,00	1,30			T	90°	270°
A4	200,00	1,30	-16,40	5,00	1,30			T	90°	270°
A5	365,00	1,30	-16,40	5,00	1,30			T	90°	270°
B1	35,00	7,30	-8,10	6,00	1,30			T	180°	
B2	291,00	8,05	-8,10	6,00	1,30			T	180°	



Part of Assembly: master

Designed by	Checked by	Approved by	Date	Date
Maximiliano Lira Del Canto				07-08-2016

Material type	
Material code	
Material name	
Color Code	
Color Name	
Material WP L	
Material WP W	
Material WP T	
WP Type	

Universidad Técnica Federico Santa María

Coulisses_38	Edition	Sheet
	2	14 / 14