



**UNIVERSIDAD NACIONAL DE LOJA**

**Facultad de la Energía, las Industrias y los Recursos  
Naturales no Renovables**

**Carrera de Ingeniería en Electrónica y  
Telecomunicaciones**

**“IMPLEMENTACIÓN DE UN INFOCHANNEL UTILIZANDO  
SERVICIOS WEB PARA LA VISUALIZACIÓN DE CONTENIDO  
MULTIMEDIA PARA EL SIS ECU 911 DE LA CIUDAD DE LOJA”**

TRABAJO DE TITULACIÓN PREVIO A LA  
OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y TELECOMUNICACIONES.

**Autor:**

*Luis Alberto Cartagena Soto*

**Director:**

*Ing. Santiago Abraham Medina León, Mg. Sc.*

*Loja - Ecuador*

*2019*

## CERTIFICACIÓN

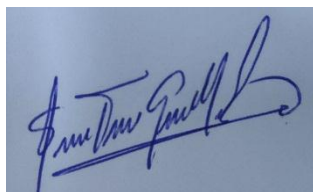
Ing. Santiago Abraham Medina León, Mg. Sc.

**DIRECTOR DE TESIS**

### CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación cuyo tema versa en **“IMPLEMENTACIÓN DE UN INFOCHANNEL UTILIZANDO SERVICIOS WEB PARA LA VISUALIZACIÓN DE CONTENIDO MULTIMEDIA PARA EL SIS ECU 911 DE LA CIUDAD DE LOJA”**, previa a la obtención del título de **Ingeniero en Electrónica y Telecomunicaciones**, realizado por el señor egresado: Luis Alberto Cartagena Soto, mismo que cumple con la reglamentación y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, 13 de junio de 2019.

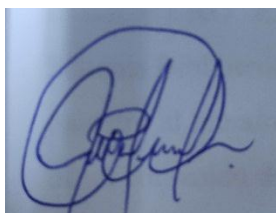
A rectangular box containing a handwritten signature in blue ink. The signature is cursive and appears to read 'Santiago Abraham Medina León'.

Ing. Santiago Abraham Medina León, Mg. Sc.  
**DIRECTOR DE TESIS**

## **AUTORÍA**

Yo, **LUIS ALBERTO CARTAGENA SOTO**, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional – Biblioteca Virtual.

A square image showing a handwritten signature in blue ink. The signature is stylized and appears to be 'L. Cartagena Soto'.

**Firma:**

**Cédula:** 1900523117

**Fecha:** 26 de agosto del 2019

## CARTA DE AUTORIZACIÓN

### CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo, **LUIS ALBERTO CARTAGENA SOTO** declaro ser autor de la tesis titulada: **“IMPLEMENTACIÓN DE UN INFOCHANNEL UTILIZANDO SERVICIOS WEB PARA LA VISUALIZACIÓN DE CONTENIDO MULTIMEDIA PARA EL SIS ECU 911 DE LA CIUDAD DE LOJA”**, como requisito para obtener el grado de **INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en RDI, en redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja a los veintiséis días del mes de agosto del dos mil diecinueve.



**Firma:**

**Autor:** Luis Alberto Cartagena Soto.

**Cédula:** 1900523117

**Dirección:** Zamora (Av. Mayaicu y Eugenio Espejo)

**Correo electrónico:** luis.cartagena@unl.edu.ec

**Teléfono:** 2605918

**Celular:** 0981827104

**Director de Tesis:** Ing. Santiago Abraham Medina León, Mg. Sc.

**Tribunal de Grado:** Ing. Juan Gabriel Ochoa Aldeán, Mg. Sc.

Ing. Luis Eduardo Rodríguez Montoya, Mg. Sc.

Ing. Marcelo Fernando Valdivieso Condolo, Mg. Sc.

## **DEDICATORIA**

*EL presente trabajo está dedicado principalmente a mis padres Luis y Kelly, mis mayores referentes en esta vida, sin su ejemplo de constancia, responsabilidad, dedicación y trabajo duro, no me habría sido posible lograr esta meta.*

*También se lo dedico a mis hermanos Dayra, Miguel y Kiara, quienes me han apoyado y han estado siempre conmigo en cada momento de mi vida.*

*Por supuesto, también se lo dedico a mis abuelos Alonso y Gloria, a quienes considero mis segundos padres, y que desde pequeño me inculcaron los valores de la perseverancia y la humildad.*

***Luis Alberto Cartagena Soto***

## **AGRADECIMIENTO**

*Agradezco profundamente a mis padres y hermanos por el apoyo incondicional que siempre me han brindado.*

*Agradezco a todos y cada uno de los docentes de la Carrera de Ingeniería en Electrónica y Telecomunicaciones de la Universidad Nacional de Loja, por brindarme una excelente formación académica.*

*Doy gracias a la vida y a Dios, por todas las personas que he podido conocer en mi paso por esta bella carrera, y por todos los buenos momentos que hemos compartido.*

*Un agradecimiento especial al Ing. Santiago Medina León y al Ing. Juan Pablo Cabrera, por su compromiso con el presente proyecto de tesis, y por supuesto, por su excelente trabajo como guías hasta su finalización.*

***Luis Alberto Cartagena Soto***

## TABLA DE CONTENIDOS

|  |     |
|--|-----|
| PORTADA .....  | I   |
| CERTIFICACIÓN.....   | II  |
| AUTORÍA .....  | III |
| CARTA DE AUTORIZACIÓN .....  | IV  |
| DEDICATORIA.....   | V   |
| AGRADECIMIENTO .....   | VI  |
| TABLA DE CONTENIDOS .....  | VII |
| ÍNDICE DE FIGURAS .....  | XII |
| ÍNDICE DE TABLAS.....  | XV  |
| <br>   |     |
| 1. TÍTULO.....   | 1   |
| 2. RESUMEN.....  | 2   |
| 2.1 SUMMARY.....   | 3   |
| 3. INTRODUCCIÓN.....   | 4   |
| 4. REVISIÓN DE LA LITERATURA.....                                    | 7   |
| 4.1 SISTEMAS DE TRANSMISIÓN DE VIDEO ( <i>VIDEO STREAMING</i> )..... | 7   |
| 4.1.1 Tipos de difusión de video afluente.....                       | 8   |
| 4.1.1.1 Difusión en vivo (Live).....                                 | 8   |
| 4.1.1.2 Difusión sobre demanda (on-demmand).....                     | 10  |
| 4.1.2 Servicios de trasmisión de video afluente.....                 | 10  |
| 4.1.2.1 Servidor de almacenamiento en red (NAS).....                 | 11  |
| 4.1.2.2 Aplicaciones móviles para video streaming.....               | 12  |

|         |  |    |
|---------|--|----|
| 4.1.2.3 | Aplicaciones web para video streaming. ....            | 14 |
| 4.2     | SERVIDORES. ....                                       | 16 |
| 4.2.1   | Servidores web. ....                                   | 18 |
| 4.2.2   | Tipos de servidores web. ....                          | 20 |
| 4.2.2.1 | Apache. ....   | 20 |
| 4.2.2.2 | Nginx. ....  | 20 |
| 4.2.2.3 | LiteSpeed. ....  | 21 |
| 4.2.2.4 | Microsoft IIS. ....                                    | 21 |
| 4.2.2.5 | GWS (Google Web Server). ....                          | 21 |
| 4.2.3   | Comparativa entre servidores web. ....                 | 22 |
| 4.2.4   | Hardware para la implementación del servidor web. .... | 24 |
| 4.2.4.1 | Unidad Central de Procesamiento. ....                  | 24 |
| 4.2.4.2 | Memoria RAM: ....                                      | 24 |
| 4.2.4.3 | Memoria de almacenamiento. ....                        | 24 |
| 4.2.5   | Sistema de gestión de base de datos. ....              | 25 |
| 4.2.5.1 | Objetivos y servicios de los SGBD. ....                | 25 |
| 4.2.5.2 | Elementos de un SGBD. ....                             | 27 |
| 4.2.5.3 | Administración y funciones dentro de un SGBD. ....     | 27 |
| 4.2.5.4 | Bases de datos relacionales. ....                      | 28 |
| 4.2.5.5 | MySQL. ....  | 30 |
| 4.3     | COMPRESIÓN DE VIDEO. ....                              | 31 |
| 4.3.1   | Códecs de video. ....                                  | 31 |
| 4.3.1.1 | Formato Mp4 (H.264/AAC). ....                          | 33 |
| 4.3.1.2 | Formato Ogv/Ogg (Theora/Vorbis). ....                  | 34 |
| 4.3.1.3 | Formato Webm (VP8/Vorbis). ....                        | 34 |
| 4.4     | ENTORNO DE DESARROLLO WEB. ....                        | 36 |
| 4.4.1   | Python. ....   | 36 |
| 4.4.2   | Framework. ....  | 38 |
| 4.4.3   | Framework Django. ....                                 | 40 |
| 4.4.3.1 | Componentes de Django. ....                            | 40 |
| 4.4.3.2 | Configuración de las rutas en Django. ....             | 42 |



|         |   |    |
|---------|---|----|
| 4.4.3.3 | Integración de Django con la base de datos. ....  | 43 |
| 4.4.3.4 | Interfaz de administración de Django.....   | 44 |
| 4.4.4   | Diseño de la interfaz de usuario ( <i>front-end</i> ).....  | 44 |
| 4.4.4.1 | HTML5.....  | 45 |
| 4.4.4.2 | CSS3.....   | 46 |
| 4.4.4.3 | JavaScript. ....  | 46 |
| 4.4.4.4 | JetBrains PyCharm Community.....  | 46 |
| 5.      | MATERIALES Y MÉTODOS.....   | 48 |
| 5.1     | MATERIALES.....   | 48 |
| 5.1.1   | Herramientas físicas. ....  | 48 |
| 5.1.2   | Herramientas de software. ....  | 49 |
| 5.2     | MÉTODOS:.....   | 51 |
| 5.2.1   | Método experimental.....  | 51 |
| 5.2.2   | Método de programación extrema.....   | 51 |
| 5.2.3   | Método deductivo.....   | 51 |
| 5.3     | PROCESO DE DISEÑO DEL SISTEMA: .....  | 51 |
| 5.3.1   | Selección e instalación del servidor web y la base de datos.....  | 52 |
| 5.3.2   | Instalación del entorno Python y el Framework Django. ....  | 53 |
| 5.3.3   | Inicio del proyecto en Django, creación de la base de datos en MySQL y configuración de la conexión entre ellas. .... | 54 |
| 5.3.4   | Migraciones a la base de datos, usuarios de administración y ejecución del servidor local de desarrollo. ....         | 62 |
| 5.3.5   | Agregar aplicaciones al proyecto. ....  | 66 |
| 5.3.6   | Configuración de modelos e interfaz de administración. ....   | 68 |
| 5.3.7   | Diseño de la interfaz de usuario. ....  | 72 |
| 5.3.8   | Configuración de las URLs y Vistas. ....  | 75 |
| 5.3.9   | Configurar el acceso externo a la aplicación. ....  | 77 |
| 5.3.10  | Determinación de parámetros para el correcto funcionamiento del sistema.  |    |

|         |  |     |
|---------|--|-----|
| 6.      | RESULTADOS.....  | 85  |
| 6.1     | Características del formato de video. ....   | 85  |
| 6.1.1   | Interoperabilidad de los códecs HTML5.....   | 85  |
| 6.1.1.1 | Códecs de video.....   | 85  |
| 6.1.1.2 | Códecs de audio.....   | 87  |
| 6.1.2   | Tasa de compresión de códecs. ....   | 88  |
| 6.1.3   | Características generales de los formatos de video para el servicio de video streaming. .... | 89  |
| 6.2     | Sistema de “Infochannel”.....  | 89  |
| 6.2.1   | Interfaz de administración. ....   | 89  |
| 6.2.2   | Interfaz de usuario. ....  | 91  |
| 6.2.3   | Diagrama físico del sistema implementado.....  | 93  |
| 6.2.4   | Diagrama lógico del sistema implementado.....  | 93  |
| 6.3     | Parámetros para el correcto funcionamiento del sistema. ....                                 | 94  |
| 6.3.1   | Políticas de uso. ....   | 94  |
| 6.3.2   | Requisitos para su despliegue:.....  | 94  |
| 6.3.2.1 | Requisitos de software:.....   | 94  |
| 6.3.2.2 | Requisitos de Hardware:.....   | 95  |
| 6.4     | Proceso de implementación del Infochannel. ....  | 97  |
| 7.      | DISCUSIÓN.....   | 98  |
| 8.      | CONCLUSIONES.....  | 104 |
| 9.      | RECOMENDACIONES. ....  | 105 |
| 10.     | BIBLIOGRAFÍA.....  | 106 |
| 11.     | ANEXOS.....  | 110 |

|   |     |
|---|-----|
| ANEXO 1: FUNCIONAMIENTO DEL INFOCHANNEL SOBRE DIFERENTES PLATAFORMAS DE USUARIO.....                        | 111 |
| ANEXO 2: IMPLEMENTACIÓN DEL INFOCHANNEL SOBRE LA RED LOCAL DEL SIS ECU 911 DE LA CIUDAD DE LOJA. ....       | 112 |
| ANEXO 3: CERTIFICACIÓN DEL SISTEMA DE INFOCHANNEL POR PARTE DEL SIS ECU 911 DE LA CIUDAD DE LOJA. ....      | 113 |
| ANEXO 4: CARACTERÍSTICAS TÉCNICAS DEL STB RECOMENDADO PARA SU INTALACIÓN EN EL SISTEMA DE INFOCHANNEL. .... | 116 |
| ANEXO 5: GUÍA DE USUARIO DEL INFOCHANNEL.....   | 117 |
| ANEXO 6: GUÍA DE ADMINISTRACIÓN DEL INFOCHANNEL.....  | 120 |
| ANEXO 7: GUÍA DEL PROGRAMADOR DEL INFOCHANNEL. ....   | 124 |
| ANEXO 8: CÓDIGO DEL BACKEND DEL SISTEMA DESARROLLADO EN PYTHON.....   | 141 |
| ANEXO 9: CÓDIGO DEL FRONTEND DEL SISTEMA DESARROLLADO EN HTML5, CCS3 Y JAVASCRIPT.....                      | 145 |

## ÍNDICE DE FIGURAS

|   |    |
|---|----|
| Figura 1.- Comunicación cliente-servidor .....  | 7  |
| Figura 2.- Comparación entre transferencia clásica y afluente. ....                         | 8  |
| Figura 3.- Difusión afluente en vivo.....   | 8  |
| Figura 4.- Clasificación de señales en vivo según su fuente. ....                           | 9  |
| Figura 5.- Clasificación de señales en vivo según su tipo de transmisión .....              | 9  |
| Figura 6.- Difusión afluente sobre demanda. ....  | 10 |
| Figura 7.- Diagrama de conexión de un NAS. ....   | 11 |
| Figura 8.- Modelo cliente-servidor.....   | 16 |
| Figura 9.- Interacciones entre navegador web y servidor web.....                            | 18 |
| Figura 10.- Transferencia de aplicaciones web entre cliente y servidor. ....                | 19 |
| Figura 11.- Reporte oficial de Netcraft de octubre de 2018. ....                            | 22 |
| Figura 12.- Sistema de gestión de bases de datos.....                                       | 25 |
| Figura 13.- Proceso básico de codificación y decodificación de video.....                   | 31 |
| Figura 14.- Formatos contenedor de video para HTML5.....                                    | 33 |
| Figura 15.- Diferenciación del back-end y front-end de un sistema. ....                     | 36 |
| Figura 16.- Características Python.....   | 37 |
| Figura 17.- Objetivos de un Framework.....  | 38 |
| Figura 18.- Generaciones de desarrollo web. ....  | 40 |
| Figura 19.- Ejemplo de modelo en Django. ....   | 41 |
| Figura 20.- Ejemplo de Vista en Django. ....  | 41 |
| Figura 21.- Ejemplo de Plantilla HTML en Django.....  | 42 |
| Figura 22.- Funcionamiento del MTV de Django. ....  | 43 |
| Figura 23.- Ejemplo de las configuraciones iniciales de la conexión a la base de datos..... | 44 |
| Figura 24.- Proceso de diseño del sistema.....  | 52 |
| Figura 25.- Instalación de XAMPP 3.2.3. ....  | 53 |
| Figura 26.- Instalación del entorno Python 3.7.2. ....                                      | 54 |
| Figura 27.- Instalación del Framework Django 2.1.7.....                                     | 54 |
| Figura 28.- Directorio "htdocs" del servidor XAMPP. ....                                    | 55 |
| Figura 29.- Acceso al directorio "htdocs" desde el cmd de Windows. ....                     | 55 |
| Figura 30.- Creación del proyecto en Django. ....   | 55 |
| Figura 31.- Archivos de Python base para el proyecto con Django. ....                       | 56 |

|   |    |
|---|----|
| Figura 32.- Acceso a MySQL para la creación de la base de datos. ....                               | 57 |
| Figura 33.- Interfaz web de administración de MySQL. ....   | 57 |
| Figura 34.- Creación de la base de datos en phpMyAdmin. ....  | 58 |
| Figura 35.- Acceso al gestor de librerías de Pycharm. ....  | 59 |
| Figura 36.- Integración de la librería "pymysql" desde el gestor de librerías de Pycharm.<br>.....  | 60 |
| Figura 37.- Instalación de la librería "pymysql" desde el CLI de Windows. ....                      | 60 |
| Figura 38.- Adhesión de la librería "pymysql" al proyecto. ....                                     | 60 |
| Figura 39.- Configuración de la conexión entre Django con la base de datos. ....                    | 61 |
| Figura 40.- Aplicaciones base de Django. ....   | 62 |
| Figura 41.- Primeras migraciones del proyecto. ....   | 63 |
| Figura 42.- Vista de las migraciones realizadas desde phpMyAdmin. ....                              | 64 |
| Figura 43.- Creación de un usuario administrador del sistema. ....                                  | 64 |
| Figura 44.- Activación del servidor de desarrollo de Django. ....                                   | 65 |
| Figura 45.- Activación del servidor de desarrollo con parámetros cambiados. ....                    | 65 |
| Figura 46.- Interfaz inicial de la aplicación web en Django. ....                                   | 66 |
| Figura 47.- Creación de una nueva aplicación dentro de un proyecto Django. ....                     | 67 |
| Figura 48.- Archivos base de una aplicación en Django. ....   | 67 |
| Figura 49.- Configuración del archivo "Settings.py" para agregar nueva aplicación. ...              | 68 |
| Figura 50.- Declaración de modelos. ....  | 69 |
| Figura 51.- Validación de modelos. ....   | 70 |
| Figura 52.- Migración de modelos a la base de datos. ....   | 70 |
| Figura 53.- Vista de modelos agregados desde el administrador de la base de datos. ...              | 70 |
| Figura 54.- Configuración de la visualización de modelos en la interfaz de administración.<br>..... | 71 |
| Figura 55.- Modelo de subida de videos al Repositorio (vista desde el administrador web).<br>.....  | 72 |
| Figura 56.- Modelo de subida de videos al Infochannel (vista desde el administrador web).<br>.....  | 72 |
| Figura 57.- Configuración del directorio de plantillas. ....  | 73 |
| Figura 58.- Configuración de directorios para archivos estáticos y multimedia. ....                 | 73 |
| Figura 59.- Plantillas HTML utilizadas. ....  | 74 |

|   |     |
|---|-----|
| Figura 60.- Etiquetas para la herencia de plantillas en Django. ....  | 75  |
| Figura 61.- Configuración de las URLs del sistema.....  | 76  |
| Figura 62.- Configuración de las Vistas del sistema. ....   | 77  |
| Figura 63.- Acceso al archivo de configuración del servidor Apache.....                                       | 78  |
| Figura 64.- Archivo de configuración de Apache modificado para el acceso remoto al servidor. ....             | 79  |
| Figura 65.- Configuración de la dirección IP del sistema en el dispositivo servidor. ....                     | 80  |
| Figura 66.- Configuración de la dirección IP del servidor en la aplicación. ....                              | 80  |
| Figura 67.- Arranque del servidor Django sobre red real.....  | 81  |
| Figura 68.- Red de pruebas del sistema.....   | 82  |
| Figura 69.- Calificación de parámetros HTML5 de los navegadores utilizados. ....                              | 83  |
| Figura 70.- Red del SIS ECU 911 de la ciudad de Loja sobre la cual se instaló el sistema de Infochannel. .... | 84  |
| Figura 71.- Códecs de video determinados con la herramienta "HTML5 TEST". ....                                | 85  |
| Figura 72.- Compatibilidad de códecs de video con navegadores web.....  | 86  |
| Figura 73.- Códecs de audio determinados con la herramienta "HTML5 TEST". ....                                | 87  |
| Figura 74.-Compatibilidad de códecs de audio con navegadores web.....   | 88  |
| Figura 75.- Tasa de compresión de códecs HTML. ....   | 88  |
| Figura 76.- Interfaz de acceso al panel de administración del Infochannel. ....                               | 90  |
| Figura 77.- Interfaz de administración del Infochannel. ....  | 90  |
| Figura 78.- Pantalla de inicio del Infochannel.....   | 91  |
| Figura 79.- Pantalla de reproducción del Infochannel.....   | 91  |
| Figura 80.- Pantalla de visualización del Repositorio. ....   | 92  |
| Figura 81.- Diagrama físico del sistema de Infochannel dentro del SIS ECU911 Loja. 93                         |     |
| Figura 82.- Diagrama lógico del sistema de Infochannel dentro del SIS ECU911 Loja.93                          |     |
| Figura 83.- Políticas de uso del sistema Infochannel ....   | 94  |
| Figura 84.- Formato de video óptimo para el sistema. ....   | 100 |

## ÍNDICE DE TABLAS

|  |    |
|--|----|
| Tabla 1.- Plataformas de video streaming sobre redes sociales. ....                                      | 13 |
| Tabla 2.- Plataformas de video streaming colaborativas. ....   | 14 |
| Tabla 3.- Plataformas web para video streaming. ....   | 15 |
| Tabla 4.- Reporte oficial de Netcraft de octubre de 2018 .....   | 22 |
| Tabla 5.- Parámetros comparativos para servidores web .....  | 22 |
| Tabla 6.- Comparativa de SGBD.....   | 29 |
| Tabla 7.- Ventajas y desventajas de MySQL .....  | 30 |
| Tabla 8.- Formatos contenedor de video. ....   | 32 |
| Tabla 9.- Características de formatos MP4, WEBM Y OGG .....  | 35 |
| Tabla 10.- Ventajas y desventajas de Python. ....  | 38 |
| Tabla 11.- Características generales de un Framework. ....   | 39 |
| Tabla 12.- Herramientas físicas. ....  | 48 |
| Tabla 13.- Herramientas de software.....   | 49 |
| Tabla 14.- Compatibilidad de códecs de video con navegadores web. ....                                   | 86 |
| Tabla 15.- Compatibilidad de códecs de audio con navegadores web. ....                                   | 87 |
| Tabla 16.- Características de los códecs estándar para HTML5. ....                                       | 89 |
| Tabla 17.- Características referenciales de hardware para el despliegue del sistema de Infochannel. .... | 95 |
| Tabla 18.- Especificaciones técnicas del servidor Lenovo Think Server RD350. ....                        | 96 |

**1. TÍTULO.**

**“IMPLEMENTACIÓN DE UN INFOCHANNEL UTILIZANDO  
SERVICIOS WEB PARA LA VISUALIZACIÓN DE CONTENIDO  
MULTIMEDIA PARA EL SIS ECU 911 DE LA CIUDAD DE LOJA”**



## **2. RESUMEN.**

A día de hoy, los servicios de difusión de contenido multimedia mediante redes de datos tienen una gran acogida en la sociedad. Este tipo de mecanismos, desde su concepción y conforme han ido desarrollándose, han jugado un papel muy importante dentro de los ámbitos del entretenimiento, las redes sociales, la difusión de información por medios audiovisuales, las llamadas y videoconferencias a largas distancias, transmisiones en directo, entre otras cosas. Concretamente uno de los servicios para la difusión de contenido multimedia actual, y bastante popular a nivel mundial, es el servicio de video streaming.

El video streaming consiste en una tecnología de difusión digital de contenido multimedia por medio de una red de computadoras, de forma que los usuarios del servicio, son capaces de consumir el producto al mismo tiempo que se descarga. Haciendo uso de esta creciente tecnología y el desarrollo de sistemas web, en el presente proyecto de tesis se propone la implementación de un Infochannel utilizando servicios web para la visualización de contenido multimedia para la institución pública ECU 911 de la ciudad de Loja. Este sistema tiene como objetivo facilitar la administración de contenido informativo institucional, concretamente los videos institucionales, por medio de un sistema centralizado de control y difusión de contenido multimedia aprovechando la red local del edificio.

El desarrollo del sistema se llevó a cabo teniendo en cuenta los criterios de transmisión de video afluente, la codificación de video y el uso de herramientas de software libre para el desarrollo de aplicaciones web.

En el presente trabajo se analizan ciertas tecnologías base para la implementación del sistema, tales como Python, HTML5, CSS3 y JavaScript, como lenguajes de programación, Apache como servidor web, MySQL para la gestión de bases de datos, y el Framework Django como nexo de todas estas tecnologías de desarrollo. Mediante el uso de estas herramientas se logró un sistema de difusión multimedia basado en servicios web, que consta fundamentalmente de dos tipos de interfaces, la una de control y administración del contenido, y la segunda como interfaz de usuario y visualización.

## **2.1 SUMMARY.**

Today, multimedia content dissemination services through data networks have a great reception in society. These types of mechanisms, since their conception and as they have been developed, have played a very important role within the ambits of entertainment, social networks, the dissemination of information by audiovisual means, calls and videoconferences over long distances, live broadcasts, among other things. Concretely, one of the services for the dissemination of current multimedia content, and quite popular worldwide, is the video streaming service.

The video streaming consists in a technology of digital diffusion of multimedia content through a computer network, so that the users of the service are able to consume the product at the same time as it is downloaded. Making use of the current multimedia broadcasting technology and the development of web systems, in this thesis project the implementation of an Infochannel using web services for the visualization of multimedia content for the public institution ECU 911 of the city of Loja is proposed. The objective of this system is to facilitate the administration of institutional information content, specifically institutional videos, by means of a centralized system of control and dissemination of multimedia content taking advantage of the LAN network of the building.

The development of the system was carried out taking into account the criteria of affluent video transmission, video coding and the use of free software tools for the development of web applications.

In this paper we analyze certain basic technologies for the implementation of the system, such as Python, HTML5, CSS3 and JavaScript, such as programming languages, Apache as a web server, MySQL for database management, and the Django Framework as a nexus. of all these development technologies. Through the use of these tools, a multimedia diffusion system based on web services was achieved, consisting basically of two types of interfaces, one of control and administration of content, and the second as a user interface and visualization.

### **3. INTRODUCCIÓN.**

Hace algunos años, la retransmisión en directo de un evento requería de un equipo técnico que se ocupase del transporte de los equipos, de las grabaciones, de la edición del contenido y del alquiler de un satélite de telecomunicaciones para la retransmisión del video. Hoy en día, gracias a las nuevas tecnologías, la retransmisión de video, más comúnmente conocida como servicio de video streaming, resulta mucho más simple y económica ya que se aprovechan las redes de telecomunicación globales (Internet), permitiendo a los usuarios y consumidores del servicio acceder a estas transmisiones en cualquier momento y desde cualquier lugar.

El video streaming consiste en una tecnología de difusión digital de contenido multimedia por medio de una red de computadoras, de forma que los usuarios del servicio, son capaces de consumir el producto al mismo tiempo que se descarga. A día de hoy, los servicios de difusión de contenido multimedia mediante redes de datos tienen una gran acogida en la sociedad. Este tipo de mecanismos, desde su concepción y conforme han ido desarrollándose, han jugado un papel muy importante dentro de los ámbitos del entretenimiento, las redes sociales, la difusión de información por medios audiovisuales, las llamadas y videoconferencias a largas distancias, transmisiones en directo, entre otras cosas (Cajamarca Sacta & Medina Pesántes, 2011).

Atendiendo a la difusión de información por medio de videos en la red; para una empresa o institución, ya sea pública o privada, este mecanismo de comunicación les permite transparentar sus actividades y labores como institución, así como actualizar de forma rápida dicho contenido. Actualmente esta es una práctica muy común y bastante popular a nivel mundial, lo que asegura el poder llegar a una gran cantidad de usuarios.

La institución pública SIS ECU 911 de la ciudad de Loja, cuenta con diferentes sistemas de difusión de información entre los cuales se tiene una página web institucional, 17 cuentas de Twitter, una Fan Page de Facebook, un usuario en Instagram y videos informativos institucionales internos (ECU 911, 2017).

Los videos institucionales internos consisten en una serie de grabaciones en las cuales se reflejan información como: servicios que presta la institución y como acceder a ellos, horarios de atención, planes y programas de la institución, entre otras cosas. Este tipo de

información multimedia es difundida por medio de redes sociales como Twitter, Facebook o Instagram, e incluso internamente dentro de la institución.

La institución SIS ECU 911 de la ciudad de Loja, internamente realiza la difusión de información multimedia (videos) colocándola dentro de varias unidades de almacenamiento USB y reproduciendo su contenido en sus Smart TVs (Televisores Inteligentes). Sin embargo, este mecanismo no permite gestionar y controlar dicho contenido de forma rápida y remota, sin mencionar que está limitado únicamente a ser reproducido por medio de sus Smart TVs. Es por ello que, atendiendo a esta necesidad por parte de la institución, se ha planteado un sistema denominado “Infochannel” o Canal de Información interno, que permita difundir toda la información multimedia de manera fácil y rápida aprovechando sus Smart TVs y la red interna de la institución, facilitando al mismo tiempo el acceso a la misma desde diferentes dispositivos de usuario.

El objetivo general de esta tesis es “implementar un sistema de visualización y control de contenido multimedia o Infochannel para los Smart TVs dentro de las instalaciones del SIS ECU 911 utilizando la red LAN del edificio” y para ello se han planteado los siguientes objetivos específicos:

Investigar la tecnología de servidores de aplicaciones de software para la implementación del Infochannel y códecs de video para disminuir la carga sobre el ancho de banda disponible mediante el uso de estándares de información compatibles con el software a utilizar.

Analizar y establecer los servicios web que se utilizaría en la comunicación de los recursos multimedia del Infochannel.

Implementar la interfaz de control y presentación del sistema multimedia con las tecnologías de software investigadas.

Realizar pruebas y validación del sistema determinando la calidad del servicio y recomendar los parámetros necesarios para establecer el correcto funcionamiento del mismo.

Teniendo presentes estos objetivos, en el inciso 4 se introducen los conceptos fundamentales sobre un sistema de transmisión de video, las características de un servidor

web, la compresión de video y el entorno de programación web requerido para implementar un servidor de contenido multimedia mediante el uso de servicios web.

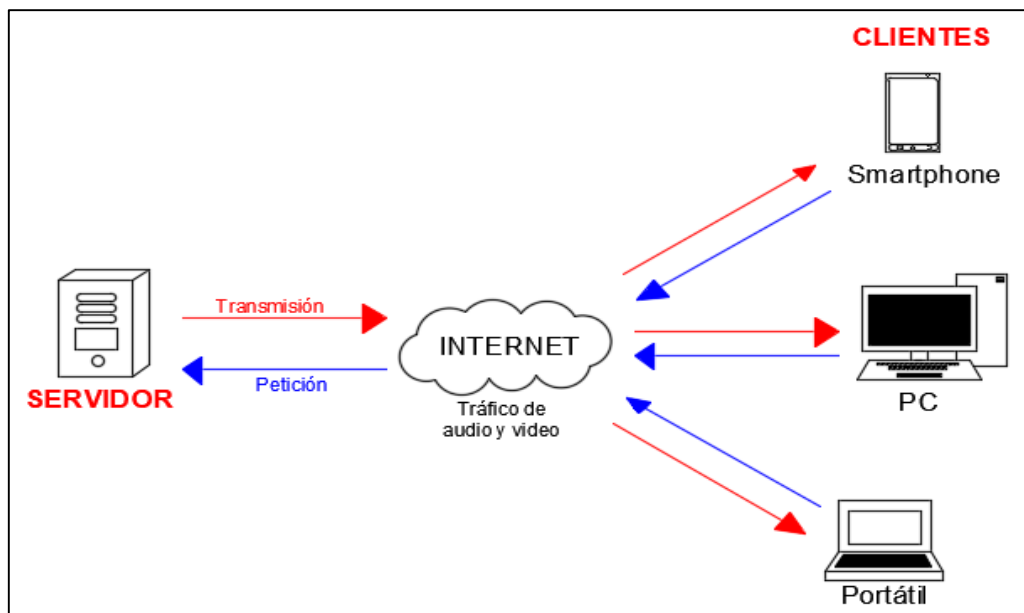
En el inciso 5 se describe la metodología usada para el desarrollo de este proyecto, al igual que las herramientas tanto de software como de hardware empleadas. También dentro de este apartado se describe a detalle el proceso de diseño del sistema de “Infochannel”.

En los incisos 6 y 7, se presentan los resultados del proceso de diseño y la discusión de estos en base a los objetivos planteados, mientras que en los incisos 8, 9, 10 y 11, se exponen las conclusiones a las que se ha llegado, las recomendaciones para su correcta ejecución y posibles mejoras, la bibliografía utilizada, y los diferentes anexos de este proyecto de tesis.

#### 4. REVISIÓN DE LA LITERATURA.

##### 4.1 SISTEMAS DE TRANSMISIÓN DE VIDEO (*VIDEO STREAMING*).

En la actualidad, la difusión de archivos multimedia a través de la red, ya sea local o por internet, maneja generalmente una arquitectura de tipo cliente-servidor (ver Figura 1).



*Figura 1.- Comunicación cliente-servidor*  
*Fuente: Colores Vargas, 2010.*

La información multimedia que reciben los clientes puede ser accedida de manera distinta, estos métodos son (Kashyap, Member, Bing, & Member, 2010)(Prendes, 2011):

- **Transferencia clásica:** es un método de transferencia de archivos, en el cual la información se descarga en la memoria local del cliente y una vez completada la descarga total de dicho archivo (recepción de todos sus paquetes), esta se reproduce (ver Figura 2).
- **Transferencia afluente:** se trata de una tecnología más eficiente para la difusión de archivos multimedia a través de la red. Gracias a esta, se aligeran las descargas que fluyen del servidor a los clientes, consiguiendo así una recepción instantánea y sin esperas, haciendo posible reproducir su contenido durante la descarga sin necesidad de guardar el archivo. En una difusión afluente, la información multimedia es enviada desde el servidor en forma de un flujo controlado de paquetes. Éstos al ser recibidos por los clientes, son reproducidos al instante y luego desechados (ver Figura 2).

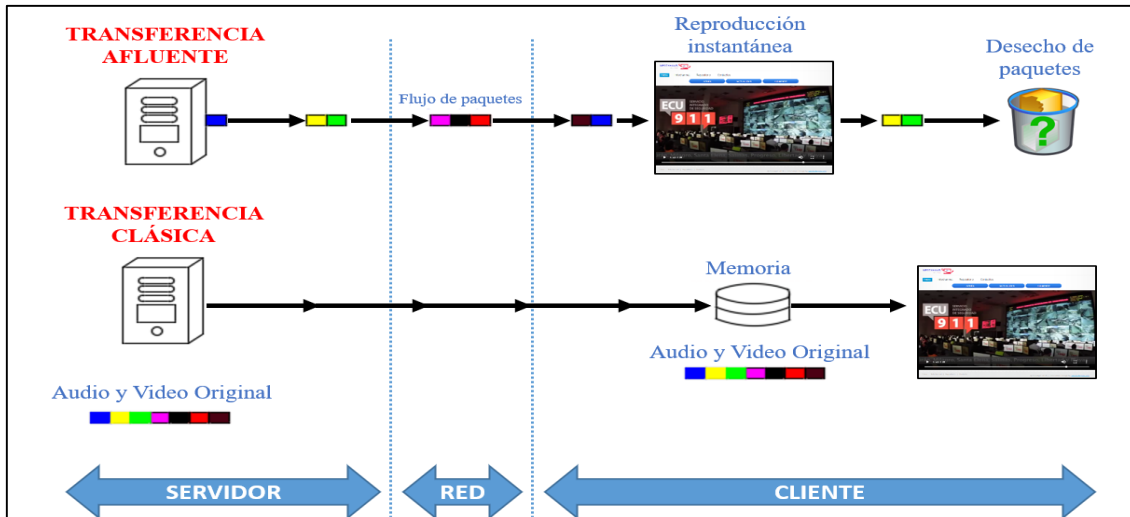


Figura 2.- Comparación entre transferencia clásica y afluente.  
Fuente: Colores Vargas, 2010.

#### 4.1.1 Tipos de difusión de video afluente.

El tipo de servicio es un parámetro importante, ya que a medida que aumenta la interactividad con el cliente, también se incrementa la complejidad del sistema de difusión afluente y, por lo tanto, el valor agregado del servicio ofrecido a los clientes. Los posibles servicios que pueden ofrecer un sistema de video afluente son:

##### 4.1.1.1 Difusión en vivo (*Live*).

Es un sistema orientado a la multidifusión (*multicast*), siendo este el tipo de servicio utilizado por los canales de televisión en vivo. En la Figura 3, se ilustra el funcionamiento de este sistema, en el cual los clientes se conectan y pueden ver únicamente la información que se esté transmitiendo en ese instante. En este servicio la interactividad está limitada, por lo que al cliente sólo se le permite pausar la reproducción (Prendes, 2011).

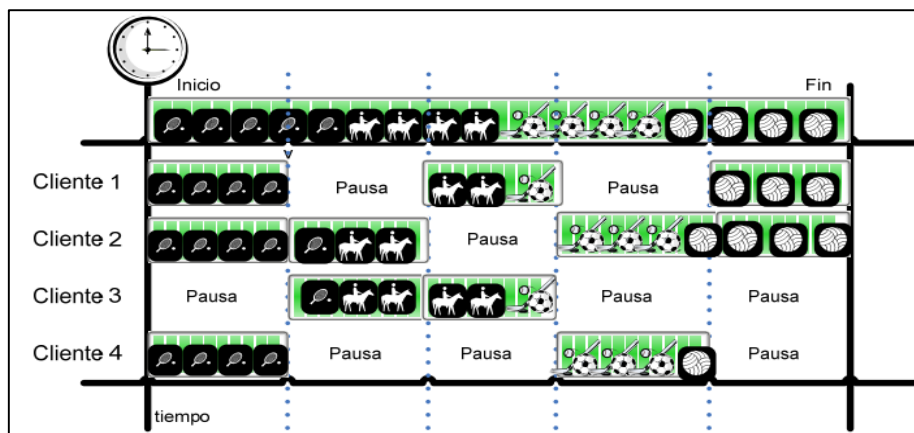


Figura 3.- Difusión afluente en vivo.  
Fuente: Colores Vargas, 2010.

De acuerdo a la fuente de video, el servicio de difusión en vivo se puede clasificar en:

1. **Por el origen de las señales de audio y video:** esto se debe a si la transmisión puede ser con información en vivo proveniente de un dispositivo de captura o con información almacenada en un disco duro (ver Figura 4) (Kashyap et al., 2010).

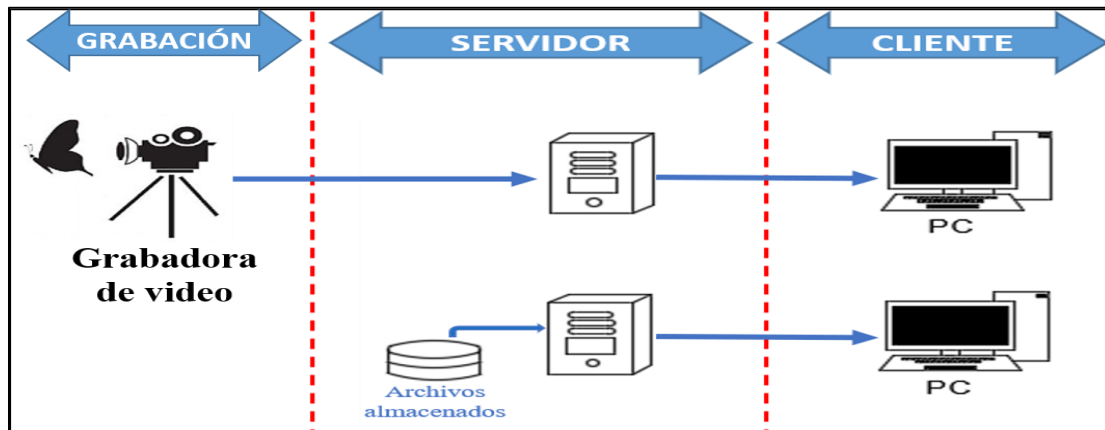


Figura 4.- Clasificación de señales en vivo según su fuente.  
Fuente: Colores Vargas, 2010.

2. **Por el tipo de transmisión:** La transmisión puede ser orientada a uno o a muchos clientes (ver Figura 5). Existen 3 tipos de transmisión:

- **Transmisión Unicast:** Consiste en la transmisión dedicada a cada cliente. Se envían distintos archivos afluentes a distintos clientes, dividiendo el ancho de banda entre ellos.
- **Transmisión Multicast:** Consiste en la transmisión por igual a un grupo de clientes, enviando el archivo afluente por la red a todos los clientes que pertenezcan al grupo y que deseen tomarlo.

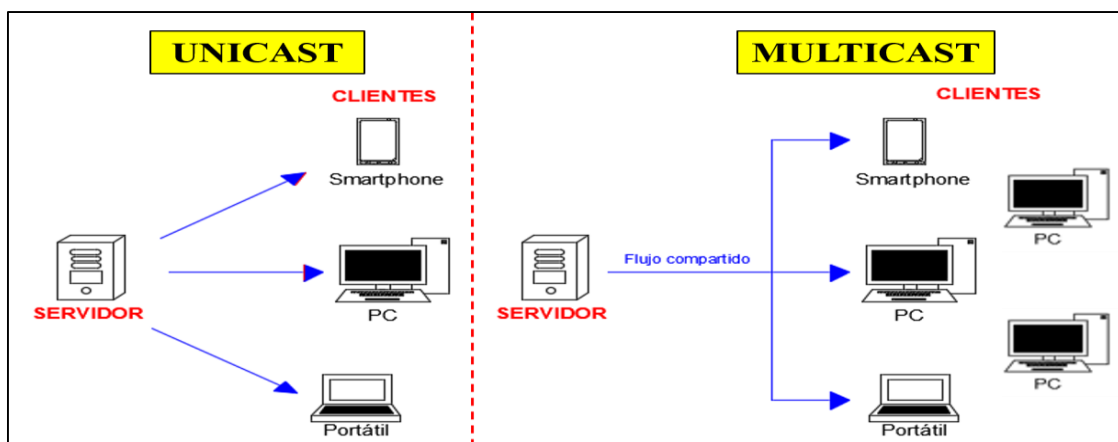


Figura 5.- Clasificación de señales en vivo según su tipo de transmisión  
Fuente: Colores Vargas, 2010.



- **Transmisión Broadcast:** Consiste en la transmisión por igual a todos los clientes, enviando el archivo afluente por la red a todos los dispositivos que pertenezcan a la red y que deseen tomarla (Kashyap et al., 2010).

#### 4.1.1.2 Difusión sobre demanda (on-demand).

En este tipo de servicio, los clientes solicitan el envío de información en el instante que lo deseen, siendo esta información personalizada para cada cliente. Para cualquier instante de tiempo, cada uno de los clientes tiene la capacidad de realizar peticiones del contenido que desee (ver Figura 6). El servicio sobre demanda es la base de la televisión interactiva, es decir que permite a los usuarios realizar acciones como pausar el video, adelantarlos o retrasarlos a su gusto.

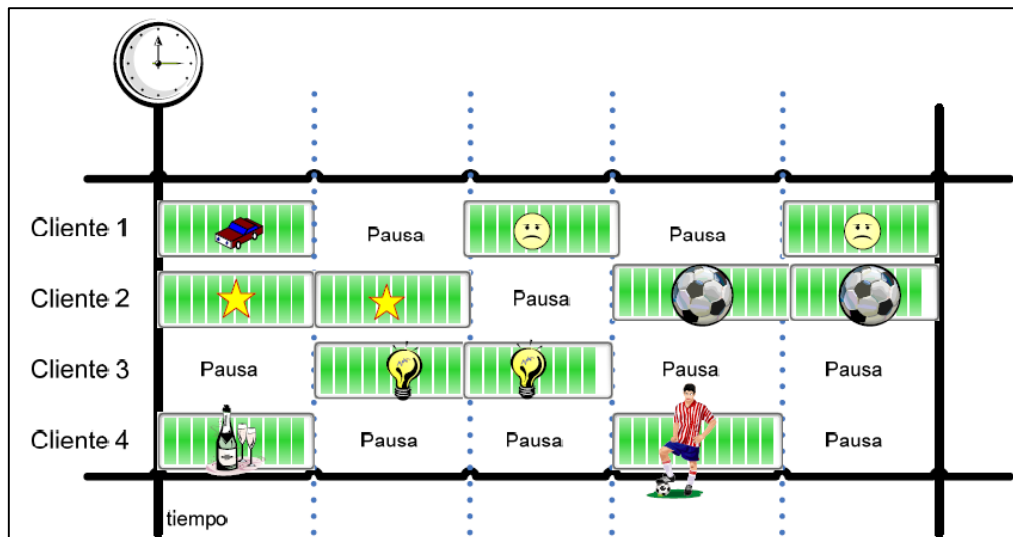


Figura 6.- Difusión afluente sobre demanda.  
Fuente: Colores Vargas, 2010.

#### 4.1.2 Servicios de transmisión de video afluente.

El sistema de Infochannel planteado en esta tesis, está orientado a la difusión de contenido multimedia dentro de la institución ECU 911 Loja mediante una plataforma que permita controlar y transmitir dicho contenido, por lo que en un principio es importante conocer las tecnologías ya existentes que permiten implementar sistemas de difusión similares y que puedan resultar como alternativas viables para los objetivos del Infochannel.

Las tecnologías de transmisión de video desde sus inicios hasta la actualidad, han ido evolucionando poco a poco, y con el tiempo han surgido diferentes plataformas y aplicaciones para la reproducción y difusión de video. Actualmente con el auge de los servicios por Internet y los sistemas de difusión digitales, era obvio que los sistemas de

difusión de video también hallarían su sitio en este campo (Bárcena, Prado, Cimoli, & Pérez, 2015). A continuación, se enuncian algunas plataformas ya existentes y que permiten realizar la difusión de video sobre redes de datos:

- Sistemas de almacenamiento en red (NAS o Network Attached Storage).
- Aplicaciones móviles para video streaming.
- Aplicaciones web para video streaming.

#### 4.1.2.1 Servidor de almacenamiento en red (NAS).

Una solución sencilla y de fácil configuración para implementar un sistema de compartición de archivos, es mediante un dispositivo conectado a la red denominado NAS (ver Figura 7).

Un sistema NAS es un dispositivo de almacenamiento conectado a la red y cuya función principal es la de hacer copias de seguridad de los archivos seleccionados al momento de su configuración, ya sea desde un ordenador personal u otro dispositivo móvil, aunque también posee otras funcionalidades, las cuales estarían disponibles de acuerdo al fabricante y al modelo de NAS que se elija (Western Digital Corporation, n.d.).

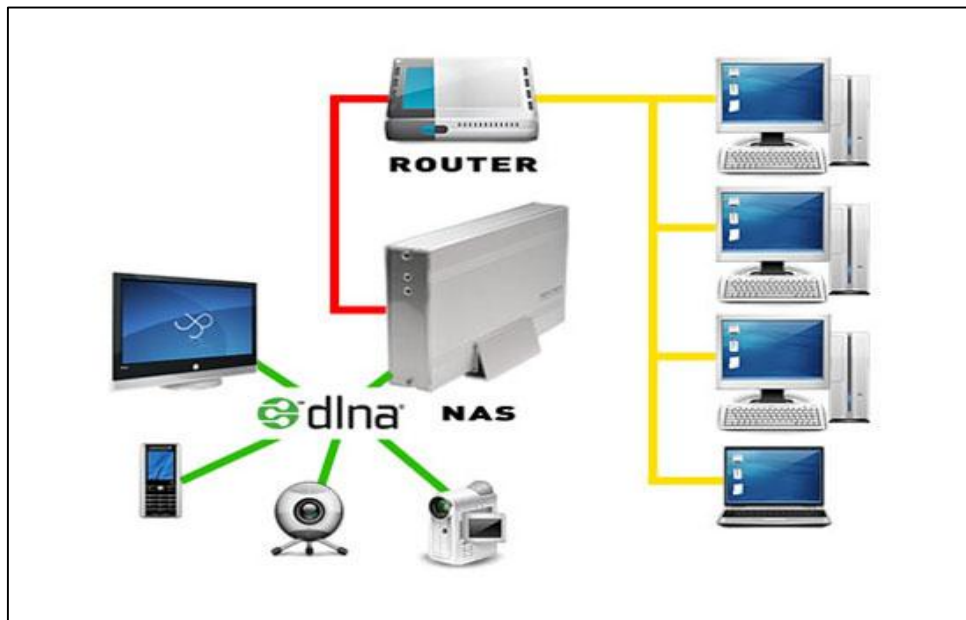


Figura 7.- Diagrama de conexión de un NAS.  
Fuente: (Tecnología&Informática, n.d.) .

A efectos prácticos la función principal de estos dispositivos es la de actuar como unidad de almacenamiento, haciendo las veces de disco duro externo o permitiendo crear un sistema propio de almacenamiento en la nube.

Un NAS está diseñado para tener un funcionamiento de tiempo continuo, y está compuesto por un procesador, uno o más discos rígidos, normalmente en modo RAID (*Redundance Array of Independent Disks* o Arreglo Redundante de Discos Independientes), y también integra un sistema operativo propio para el control de las transacciones y las tareas de red. Este grado de autonomía le permite ser extremadamente independiente y permite tener un total control de los datos y archivos. Un NAS al no disponer de interfaces de entrada de datos como teclado o ratón, ni tampoco de visualización, el usuario deberá acceder y gestionar al NAS mediante un cliente web.

Un NAS también permite la compartición de archivos multimedia como imágenes, videos y música con cualquiera de los dispositivos disponibles y que tengan acceso a la red, gracias a la tecnología de acceso DLNA (*Digital Living Network Alliance* o Alianza para el estilo de vida digital en red). También puede utilizarse como servidor web, servidor de bases de datos, servidor de impresión, servidor multimedia, servidor de descargas para Torrents y servidor de vigilancia a través de cámaras, entre otros (Tecnología&Informática, n.d.).

Uno de los inconvenientes, desde el punto de vista de usuario, es que las interfaces de acceso a los archivos almacenados en el NAS son bastante simples y sin posibilidad de personalización visual o agregación de funciones de usuario personalizadas, dado que dichas funciones vienen predeterminadas por el fabricante. Otro inconveniente a tener en cuenta es el alto costo de este tipo de dispositivos, sin tomar en cuenta el coste de los discos duros que deben acoplarse al mismo (Growing et al., 2012).

#### **4.1.2.2 Aplicaciones móviles para video streaming.**

Otra opción bastante popular para establecer de forma rápida un servicio de video streaming, es mediante el uso de aplicaciones móviles, dada la enorme acogida de los dispositivos inteligentes (*Smartphones, Smart TVs, Smartwatches o Tablets*)(Camuñas, n.d.). En las siguientes tablas se exponen algunas alternativas para para video streaming.

*Tabla 1.- Plataformas de video streaming sobre redes sociales.  
Fuente: Elaboración propia.*

| <b>Streaming de video sobre redes sociales</b> |                      |                           |                              |                                       |                                 |
|--|----------------------|---------------------------|------------------------------|---------------------------------------|---------------------------------|
| <b>Plataforma</b>                              | <b>Desarrollador</b> | <b>Licencia de acceso</b> | <b>Tipo de Streaming</b>     | <b>Sistemas operativos soportados</b> | <b>Acceso desde cliente web</b> |
| <b>YouTube</b>                                 | Google Inc.          | - Gratuita<br>- De pago   | - En vivo<br>- Sobre demanda | - Android<br>- IOS<br>- Windows Phone | Si                              |
| <b>Facebook</b>                                | Facebook Inc.        | - Gratuita                | - En vivo<br>- Sobre demanda | - Android<br>- IOS<br>- Windows Phone | Si                              |
| <b>Twitter</b>                                 | Twitter Inc.         | - Gratuita                | - En vivo<br>- Sobre demanda | - Android<br>- IOS<br>- Windows Phone | Si                              |
| <b>Periscope</b>                               | Twitter Inc.         | - Gratuita                | - En vivo                    | - Android<br>- IOS<br>- Windows Phone | Si                              |
| <b>Instagram</b>                               | Facebook Inc.        | - Gratuita                | - En vivo<br>- Sobre demanda | - Android<br>- IOS<br>- Windows Phone | Si                              |

Las opciones mostradas en la Tabla 1, son algunas de las aplicaciones más populares para ejecutar de forma rápida un servicio de streaming que pueda ser visualizado por una gran cantidad de usuarios. Este tipo de aplicaciones (en su mayoría) además de transmisiones de video en vivo, también permiten almacenar dichas transmisiones e incluso videos pregrabados y ponerlos a disposición de otros usuarios, los cuales tienen la posibilidad de interactuar con ellos y manifestar opiniones escritas acerca de los mismos. En cuanto a la accesibilidad y disponibilidad del servicio, para este tipo de plataformas es bastante elevada ya que los usuarios únicamente requieren de una cuenta registrada en la plataforma y una conexión activa a internet (Camuñas, n.d.).

Cabe recalcar que la aplicación de YouTube suele encontrarse ya preinstalada de fábrica en la mayoría de sistemas operativos dedicados a dispositivos inteligentes, por lo que no sólo puede hallarse en sistemas móviles como Android, IOS o Windows Phone, sino que también en sistemas operativos como WebOS, TizenOS, Android AOSP, Android TV, Firefox OS y Roku TV, los cuales son dedicados para Smart TVs.

Un punto importante a tener en cuenta con este tipo de aplicaciones de streaming es que, de acuerdo al sistema operativo del dispositivo usuario, este podrá tener acceso únicamente a las aplicaciones que sea compatibles con dicho sistema operativo, a excepción de las aplicaciones que son accesibles desde un cliente o navegador web. Sin

embargo, a pesar del soporte web que pueda tener la aplicación, hay que elegir muy bien el método que se desee usar para llevar a cabo la transmisión de video, teniendo en cuenta el tipo de usuario final al que vaya orientado el servicio y, por supuesto, también al tipo de dispositivos de usuario para el cual se diseña el sistema.

Otras de las alternativas móviles para el servicio de streaming de video son las que se muestran en la Tabla 2. La particularidad de estas aplicaciones es que están diseñadas específicamente para ofrecer servicios de video en vivo, tales como videoconferencias, e incluso seminarios, talleres, cursos o conferencias interactivas en vivo a través de internet (*webinars*) (Camuñas, n.d.).

*Tabla 2.- Plataformas de video streaming colaborativas.  
Fuente: Elaboración propia.*

| <b>Streaming de video mediante herramientas de trabajo colaborativas</b> |                           |                           |                          |                                       |                                 |
|--|---------------------------|---------------------------|--------------------------|---------------------------------------|---------------------------------|
| <b>Plataforma</b>  | <b>Desarrollador</b>      | <b>Licencia de acceso</b> | <b>Tipo de Streaming</b> | <b>Sistemas operativos soportados</b> | <b>Acceso desde cliente web</b> |
| <b>Skype</b>   | Microsoft Corporation     | - Gratuita<br>- De pago   | - En vivo                | - Android<br>- IOS<br>- Windows Phone | Si                              |
| <b>Zoom Cloud Meetings</b>   | Zoom Video Communications | - Gratuita<br>- De pago   | - En vivo                | - Android<br>- IOS<br>- Windows Phone | Si                              |
| <b>Hangouts</b>  | Google Inc.               | - Gratuita                | - En vivo                | - Android<br>- IOS<br>- Windows Phone | Si                              |
| <b>Whereby</b>   | Whereby Inc.              | - Gratuita<br>- De pago   | - En vivo                | - Android<br>- IOS<br>- Windows Phone | Si                              |

En conjunto todas las aplicaciones antes mencionadas ofrecen servicios bastante buenos y fáciles de implementar, sin embargo, la mayoría de ellas en sus versiones gratuitas, suelen tener limitantes en su funcionamiento, lo que hace que el usuario necesariamente deba pagar por una licencia para acceder a un servicio de mayor calidad.

#### **4.1.2.3 Aplicaciones web para video streaming.**

Actualmente, las aplicaciones de video streaming mediante un cliente web resultan una alternativa con un mayor índice de accesibilidad y disponibilidad frente a las aplicaciones móviles, dado que cualquier dispositivo que tenga acceso a una red IP, siempre contará con un navegador web que pueda permitirle consumir dicho servicio independientemente del sistema operativo que maneje. A continuación, en la Tabla 3 se enuncian algunas de las plataformas de streaming mediante un cliente web.

Tabla 3.- Plataformas web para video streaming.  
Fuente: Elaboración propia.

| Streaming mediante otras plataformas |                     |                         |                              |                                |                          |
|--------------------------------------|---------------------|-------------------------|------------------------------|--------------------------------|--------------------------|
| Plataforma                           | Desarrollador       | Licencia de acceso      | Tipo de Streaming            | Sistemas operativos soportados | Acceso desde cliente web |
| <b>YowiTV</b>                        | Yowi TV             | - Gratuita<br>- De pago | - En vivo<br>- Sobre demanda | Multiplataforma                | Si                       |
| <b>Everwebinar</b>                   | Genesis Digital LLC | - Gratuita<br>- De pago | - En vivo<br>- Sobre demanda | Multiplataforma                | Si                       |
| <b>StealthSeminar</b>                | StealthSeminar      | - Gratuita<br>- De pago | - En vivo<br>- Sobre demanda | Multiplataforma                | Si                       |
| <b>Vimeo</b>                         | Vimeo Inc.          | - Gratuita<br>- De pago | - En vivo<br>- Sobre demanda | Multiplataforma                | Si                       |
| <b>OBS</b>                           | OBS Project         | - Gratuita              | - En vivo<br>- Sobre demanda | Multiplataforma                | Si                       |
| <b>BeLive</b>                        | BeLive Studios      | - Gratuita              | - En vivo                    | Multiplataforma                | Si                       |
| <b>Class on live</b>                 | ClassOnLive S.L.    | - Gratuita<br>- De pago | - En vivo<br>- Sobre demanda | Multiplataforma                | Si                       |
| <b>Livestream</b>                    | Vimeo Inc.          | - De pago               | - En vivo<br>- Sobre demanda | Multiplataforma                | Si                       |

Al igual que las aplicaciones móviles, las aplicaciones web tienen una amplia variedad de funcionalidades, las cuales se encuentran programadas y configuradas de forma predeterminada por la empresa desarrolladora de las mismas, y en caso de querer modificar o adherir nuevas funcionalidades de usuario, esto resulta imposible, dado que no se tiene acceso al código fuente del servicio. Sin embargo, teniendo en cuenta las características de ser una opción de bajo coste y de fácil accesibilidad para cualquier tipo de dispositivo con acceso a una red IP, los sistemas basados en clientes web se consideraron como los más óptimos para el diseño y desarrollo del sistema de video streaming denominado “Infochannel”.

De acuerdo a los conceptos e información redactada hasta este punto y a los objetivos planteados en esta tesis, el sistema de Infochannel se desarrollaría como una aplicación basada en servicios web bajo los principios funcionales de la transferencia de video afluente de archivos almacenados localmente dentro un servidor web, ofreciendo así, un servicio de video sobre-demanda. En las secciones siguientes se detalla cada una de las partes fundamentales que conforman este sistema.

## 4.2 SERVIDORES.

Un servidor, como software, es una aplicación en ejecución capaz de atender las peticiones de uno o múltiples dispositivos denominados clientes y, por consiguiente, retornar una respuesta acorde a dichas solicitudes. Para su ejecución, los servidores requieren de un *hardware* que los aloje y por supuesto de acuerdo al tipo de servicio, se requerirá de una mayor o menor capacidad, por lo que un mismo dispositivo *hardware* puede ser capaz de alojar múltiples servicios. Una de las arquitecturas que manejan este tipo de sistemas se denomina cliente-servidor. En esta arquitectura la capacidad de procesos está repartida entre los clientes y los servidores, es decir, se reparte las responsabilidades, dejando del lado del servidor la gestión de la información y del lado del cliente la realización de peticiones (Bazán, 2017).

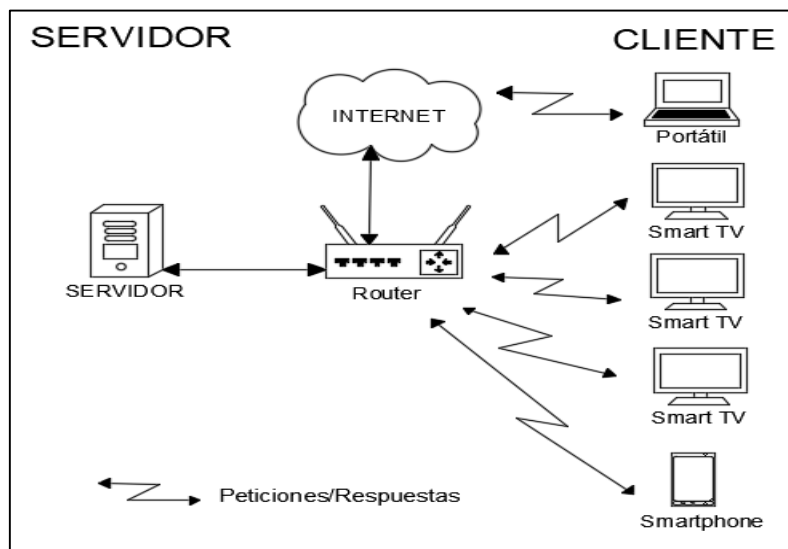


Figura 8.- Modelo cliente-servidor.  
Fuente: Elaboración propia.

Dentro de una red específica, los servidores son desplegados para proveer servicios esenciales dentro de la misma, ya sea para usuarios privados (pertenecientes a una empresa u organización) o para usuarios públicos a través de Internet (Tanenbaum & Wetherall, 2011). De acuerdo al tipo de servicio, algunos tipos de servidores son:

- **Servidores de Aplicaciones:** son considerados como un tipo de nexo que interconecta dos aplicaciones (*middleware*). Estos servidores ocupan una gran parte del territorio entre los servidores de bases de datos y el usuario, y a menudo los conectan (Uribe Sosa, 2015).

- **Servidores de Audio/Video:** este tipo de servidores agrega capacidades multimedia a los sitios web, permitiéndoles así reproducir contenido de flujo continuo (*streaming*) desde el servidor (Uribe Sosa, 2015).
- **Servidores de Chat:** Los servidores de chat permiten el intercambio información a una gran cantidad de usuarios ofreciendo la posibilidad de llevar a cabo conversaciones en tiempo real (Uribe Sosa, 2015).
- **Servidores FTP:** Gracias al Protocolo de transferencia de archivos (FTP o *File Transfer Protocol*), permiten mover uno o más archivos con seguridad entre distintos ordenadores proporcionando seguridad y organización de los archivos así como control de la transferencia (Uribe Sosa, 2015).
- **Servidores Groupware:** permite a sus usuarios, sin importar su localización, colaborar y trabajar vía internet o vía intranet corporativo bajo una atmósfera virtual (Uribe Sosa, 2015).
- **Servidores de Correo:** estos servidores transfieren y almacenan correos electrónicos por medio de redes corporativas o a través de internet (Uribe Sosa, 2015).
- **Servidores Proxy:** un servidor proxy o procurador se encuentra ubicado entre un cliente y un servidor de modo que pueda filtrar sus peticiones, mejorar el funcionamiento y compartir conexiones (Uribe Sosa, 2015).
- **Servidores Telnet/SSH:** Un servidor telnet o SSH permite a los usuarios acceder de forma remota a un dispositivo. Esto facilita las labores de administración de equipos dentro de una misma red (Uribe Sosa, 2015).
- **Servidores Web:** Un servidor web o servidor HTTP, sirve contenido estático o dinámico a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario o cliente. Este intercambio de datos es mediado por el navegador y el servidor que se comunican entre sí mediante HTTP (*Hypertext Transfer Protocol* o Protocolo de transferencia de hipertexto) (Bammidi, 2017).



De acuerdo al rol que asumen dentro de una red, los servidores se dividen en:

- **Servidor dedicado:** es aquel que invierte todos sus recursos administrativos a la red, es decir, su función primordial es la de atender las solicitudes por parte de los clientes (Hosting, 2015).
- **Servidor no dedicado:** es aquel cuyos recursos de procesamiento se hallan compartimentados entre los clientes conectados al servidor y ciertas funciones provenientes de un usuario local (Hosting, 2015).

#### 4.2.1 Servidores web.

El objetivo de este tipo de servidores es el de establecer conexiones bidireccionales o unidireccionales con el cliente, generando una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente es compilado y ejecutado por un navegador web (ver Figura 9). Para la transmisión de todos estos datos se requiere del protocolo HTTP o, para mayor seguridad, del protocolo HTTPS (*Hypertext Transfer Protocol Secure* o Protocolo seguro de transferencia de hipertexto), pertenecientes a la capa de aplicación del modelo OSI (*Open System Interconnection* o Interconexión de Sistemas Abiertos) (Incencio Piñeiro, 2014).

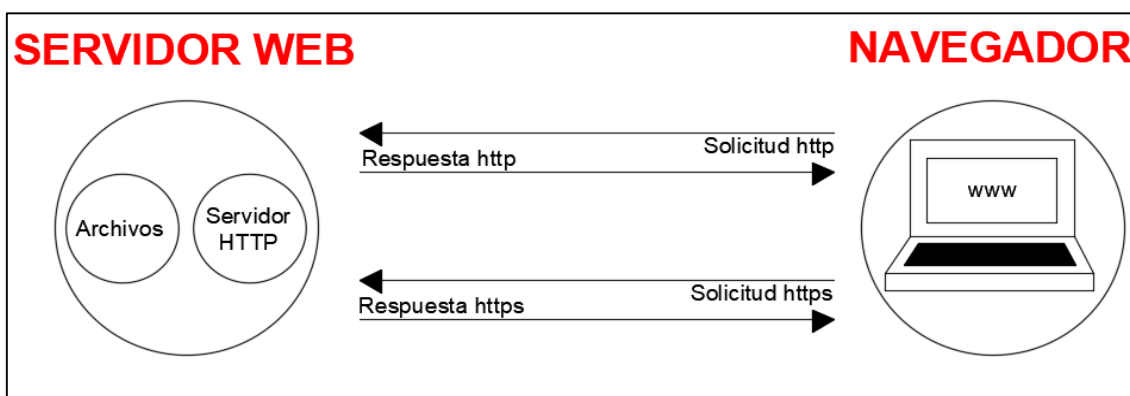


Figura 9.- Interacciones entre navegador web y servidor web.  
Fuente: Elaboración propia.

Los servidores HTTP suelen almacenar por lo general páginas web, que incluyen archivos de configuración en HTML (*HyperText Markup Language* o Lenguaje de Marcas de Hipertexto), PHP (*Hypertext Preprocessor* o Preprocesador de hipertexto), ASP (*Active Server Pages*), JavaScript, CSS (*Cascading Style Sheets* u Hojas de estilo en cascada), etc, y también contenido multimedia como imágenes, vídeos y audio (Huang, Xia, Sun, & Xue, 2015).

Los Servidores web pueden transferir también aplicaciones web, las cuales son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP (ver Figura 10) (de la Torre, 2012). Hay que distinguir entre:

- **Aplicaciones en el lado del cliente:** el cliente web recibe el código o script de las aplicaciones desde el servidor y mediante un navegador web las ejecuta en la máquina usuario. Los navegadores permiten ejecutar dichas aplicaciones escritas en lenguajes como JavaScript o Java (ver Figura 10) (de la Torre, 2012).
- **Aplicaciones en el lado del servidor:** en este caso, el servidor web es quien ejecuta la aplicación, para ello se sirve de lenguajes como Python, Ruby, ASP o PHP, entre otros (ver Figura 10). En este caso, una vez ejecutada la aplicación, se genera una respuesta HTML, la cual es enviada al cliente por medio del protocolo HTTP (de la Torre, 2012).

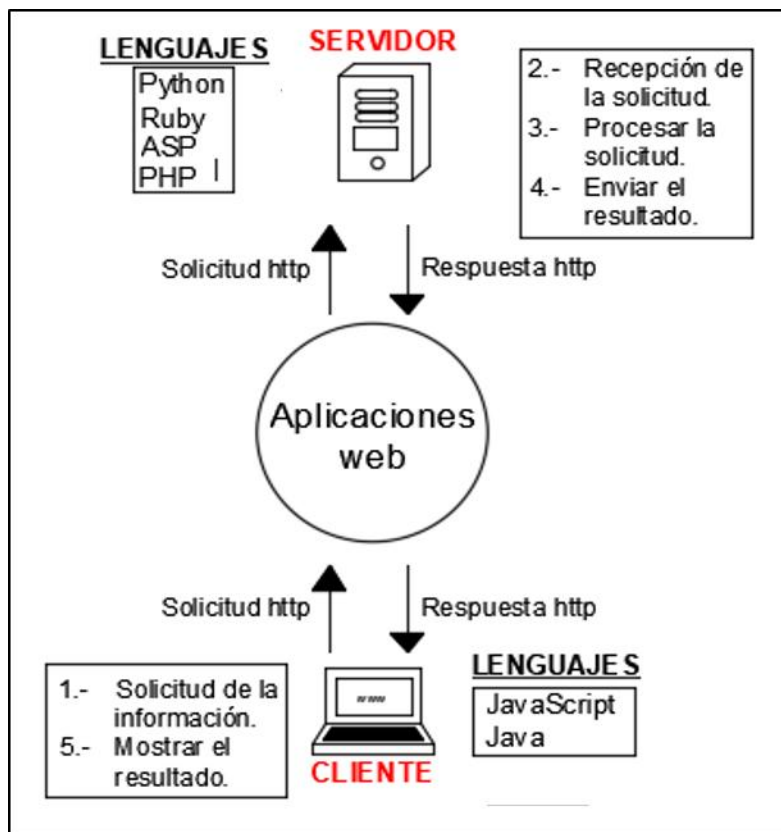


Figura 10.- Transferencia de aplicaciones web entre cliente y servidor.  
Fuente: Fernández Moreno, 2010.

## **4.2.2 Tipos de servidores web.**

### **4.2.2.1 Apache.**

Es el servidor web con mayor acogida de Internet. Apache surgió en abril de 1996 y perdura hasta el día de hoy. Durante todo este tiempo ha sido el líder indiscutido de los servidores web, hasta hace pocos años en los que ha ido perdiendo terreno frente a servidores como Microsoft IIS y Nginx (Systems, 2010).

Una de sus principales ventajas es que es de código abierto, gratuito y multiplataforma (Windows, Linux y Unix). El paquete de software LAMP (Linux, Apache, MySQL y PHP) lo popularizó muchísimo durante el auge de las aplicaciones hechas en PHP desde el año 2000 en adelante (Systems, 2010).

Apache ofrece la posibilidad de integrar intérpretes para lenguajes de programación como PHP, Perl, Python o Ruby por medio de módulos, permitiendo así la entrega de contenido dinámico sin la necesidad de un servidor de aplicaciones externo (Systems, 2010).

Entre sus desventajas está su bajo rendimiento al momento de atender miles de peticiones simultáneos en procesamiento de peticiones de contenido dinámico o archivos estáticos, quedando rezagado por su arcaica arquitectura versus nuevas y mejores opciones como Nginx (Systems, 2010).

### **4.2.2.2 Nginx.**

Este es un servidor web de código abierto y gratuito destacado por su alto rendimiento. Incluye además funciones como servidor proxy reverso HTTP, balanceador de carga, así como POP3 (Post Office Protocol o Protocolo de oficina postal) e IMAP (Internet Message Access Protocol o Protocolo de acceso a mensajes de internet). Es un sistema multiplataforma disponible para Windows, Linux y Unix (Nginx, 2010).

Entre sus ventajas se tiene una configuración muy intuitiva y sencilla, pero al mismo tiempo muy potente, de forma que puede integrarse nativamente con casi cualquier tecnología y lenguaje de programación actual. Es ideal para despachar archivos tanto estáticos como dinámicos (Nginx, 2010).

Como desventaja se tiene que nginx no permite integrar los intérpretes correspondientes mediante módulos y, en cualquier caso, se necesita un servidor de aplicaciones separado, lo que puede suponer un gasto adicional innecesario para las páginas web de pequeña y

mediana envergadura. Esto resulta factible para diseños web grandes en los que el tráfico elevado requiere de una estructura de tales características (Nginx, 2010).

#### **4.2.2.3 LiteSpeed.**

El servidor LiteSpeed es un software HTTP propietario desarrollado por LiteSpeedTech, una empresa que se ha centrado en desarrollar una alternativa a Apache para entornos Linux.

Soporta grandes cantidades de conexiones simultáneas con un bajo consumo de recursos, incluso para aplicaciones demandantes como las que utilizan PHP. A nivel de archivos estáticos también está a la altura de Nginx.

Una de sus pocas desventajas es que la es un software propietario y no muy popular entre la comunidad de *webmasters* y desarrolladores.

#### **4.2.2.4 Microsoft IIS.**

*Internet Information Services* (IIS), es un tipo de servidor web creado por Microsoft específicamente para su plataforma de sistemas operativos Windows.

Permite el procesamiento y despacho de páginas desarrolladas en tecnología ASP / ASP.NET, aunque también vale aclarar que sirve para interpretar páginas programadas en Perl o PHP (Junta de Andalucía, n.d.).

No es sólo un servidor web, sino también una suite de servicios para la web, ya que ofrece también servicios de SMTP (*Simple Mail Transfer Protocol* o Protocolo para transferencia simple de correo) y FTP, entre otros (Junta de Andalucía, n.d.).

Tiene como gran desventaja que es un servidor web propietario exclusivo de Windows, y por lógica carece de integración para tantas tecnologías y lenguajes como otros servidores (Junta de Andalucía, n.d.).

#### **4.2.2.5 GWS (Google Web Server).**

*Google Web Server*, GWS o Servidor Web de Google, es un tipo de servidor web propiedad de Google, el cual no es posible descargar desde ningún lugar, ya que no está disponible para el público (Fitzpatrick, n.d.).

Se trata de un servidor web privado escrito en C++, desarrollado y utilizado por Google para la mayoría de su infraestructura web. De la poca información publicada acerca de este servidor, se sabe que originalmente se basó en el software del servidor web Apache y

actualmente se ejecuta en una versión muy modificada de Linux Debian (Fitzpatrick, n.d.).

### 4.2.3 Comparativa entre servidores web.

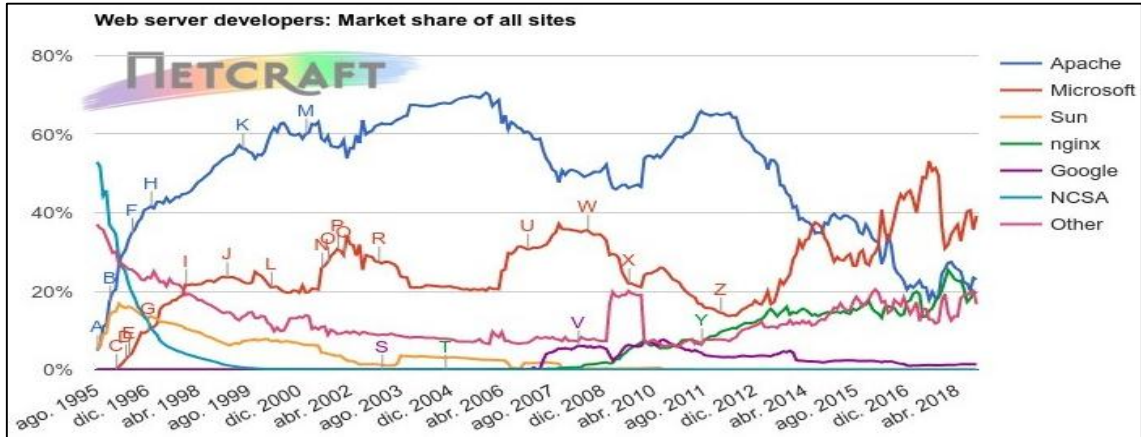


Figura 11.- Reporte oficial de Netcraft de octubre de 2018.  
Fuente: NETCRAFT, n.d.

Tabla 4.- Reporte oficial de Netcraft de octubre de 2018  
Fuente: NETCRAFT, n.d..

| Desarrollador | Septiembre de 2018 | Porcentaje | Octubre de 2018 | Porcentaje | Variabilidad |
|---------------|--------------------|------------|-----------------|------------|--------------|
| Microsoft     | 585,892,927        | 35.67%     | 656,395,388     | 39.22%     | 3.55         |
| Apache        | 384,521,189        | 23.41%     | 384,514,944     | 22.98%     | -0.44        |
| Nginx         | 319,330,263        | 19.44%     | 330,074,974     | 19.72%     | 0.28         |
| Google        | 22,210,093         | 1.35%      | 23,620,555      | 1.41%      | 0.06         |

Como se observa en la Figura 11 y en la Tabla 4, el claro dominante del mercado actual es Microsoft IIS, que ha ganado mucho terreno últimamente frente a su clásico rival Apache, el cual se ha mantenido como líder indiscutible durante décadas.

Nginx sigue creciendo lentamente, y Google tiene un *market share* (ver Figura 11) realmente pequeño pues hay que tener en cuenta que es usado primordialmente en las plataformas de Google search (Buscador de Google).

Además de los índices de popularidad y uso de estos servidores, existen otros parámetros a tener en cuenta a la hora de seleccionar un servidor con el cual trabajar. En la Tabla 5 se exponen ciertos parámetros a tener en cuenta a la hora de seleccionar un tipo de servidor web:

Tabla 5.- Parámetros comparativos para servidores web  
Fuente: Elaboración propia.

| CARACTERÍSTICAS                        | APACHE                                | NGINX                    | LITESPEED                                       | IIS                      | GWS    |
|--|---------------------------------------|--------------------------|---|--------------------------|--------|
| Empresa desarrolladora                 | Apache Software Foundation            | NGINX Inc. e Igor Sysoev | LiteSpeed Technologies                          | Microsoft                | Google |
| Año de lanzamiento                     | 1996                                  | 2009                     | 2003  | 1997                     | -      |
| Escrito en                             | - C<br>- Extensible Markup<br>- C++   | C                        | - C<br>- C++                                    | C++                      | C++    |
| Software libre                         | Si                                    | Si                       | No  | No                       | No     |
| Conexiones permanentes                 | Si                                    | Si                       | Si  | Si                       | -      |
| Autenticación                          | Si                                    | Si                       | Si  | Si                       | -      |
| Capacidad de ser empotrado             | No                                    | Si                       | Si  | No                       | -      |
| Multiplataforma                        | Si                                    | Si                       | No  | No                       | No     |
| Alojamiento virtual                    | Si                                    | Si                       | Si  | Si                       | -      |
| Integración de intérpretes de lenguaje | Si                                    | No                       | Si  | Si                       | -      |
| Lenguajes de programación soportados   | - Python<br>- Perl<br>- PHP<br>- Ruby | -                        | - PHP<br>- Perl<br>- Ruby<br>- Python<br>- Java | - ASP<br>- PHP<br>- Perl | -      |
| WSGI                                   | Si                                    | Si                       | No  | No                       | -      |
| SCGI                                   | Si                                    | Si                       | -   | Si                       | -      |
| Administración por consola             | Si                                    | Si                       | Si  | Si                       | -      |
| Conexiones seguras https               | Si                                    | Si                       | Si  | Si                       | Si     |
| QUIC (Quick UDP Internet Connections)  | -                                     | -                        | Si  | -                        | -      |
| Balanceador de carga                   | No                                    | Si                       | Si  | Si                       | -      |

Entre los tipos de servidores seleccionados se puede apreciar las grandes ventajas ofrecidas por el servidor web Apache, el cual, pese a sus falencias frente a conexiones muy concurrentes, ofrece grandes resultados y múltiples herramientas a la hora de desarrollar un sitio web.

#### **4.2.4 Hardware para la implementación del servidor web.**

Para la implementación de un servidor web, es necesario tener en consideración ciertos parámetros de hardware que aseguren el correcto funcionamiento de sus servicios. Algunas de las características de hardware que se deben tomar en consideración son las siguientes: Unidad Central de Procesamiento, Memoria RAM y Memoria de almacenamiento.

##### **4.2.4.1 Unidad Central de Procesamiento.**

La CPU (*Central Processing Unit*) en el servidor (también conocida como “procesador”) es lo que interpreta y ejecuta instrucciones, datos de procesamiento y tareas de rendimiento como mostrar páginas web, ejecutar consultas de la base de datos y ejecutar otros comandos de programas y computación. Cuantos más procesadores haya en un servidor, más rápido y más eficientemente es capaz de funcionar y mayor será la cantidad de instrucciones que se pueden ejecutar en un menor espacio de tiempo. La rapidez con la cual un procesador funciona depende en parte de la velocidad del reloj, la cual refleja la cantidad de instrucciones que puede ejecutar la CPU por segundo. También es importante considerar el número de núcleos que posee el procesador, dado que esto ayuda también en la velocidad de ejecución de instrucciones.

##### **4.2.4.2 Memoria RAM:**

La memoria RAM (Random Access Memory) es la memoria desde la cual el procesador central recibe sus instrucciones y en la que guarda los resultados de cálculo. Su velocidad de intercambio es muy superior a la de los discos duros. La capacidad de esta memoria se mide en Megabytes o Gigabytes, que indica la cantidad de datos que se puede almacenar para ser procesados por la CPU. Cuanta más memoria RAM tenga el servidor, más servicios o programas puede tener arrancados en un mismo momento sin que el rendimiento se vea afectado. La elección de la cantidad de memoria RAM depende de los servicios que se necesiten tener en un servidor.

##### **4.2.4.3 Memoria de almacenamiento.**

Dentro de los servidores se utiliza un sistema de almacenamiento denominado RAID (Redundant Array of Independent Disk o Arreglo Redundante de Discos Independientes). Un RAID se define como una agrupación de discos que actúan colectivamente como un

único sistema de almacenamiento que, en la mayoría de los casos, soporta el no funcionamiento de uno de los discos sin perder información y operar con independencia.

#### 4.2.5 Sistema de gestión de base de datos.

Todo servidor web requiere de un mecanismo que permita alojar, administrar y relacionar los diferentes tipos de datos que manejan y que son necesarios para ejecutar sus aplicaciones. Debido a esto, es necesaria la implementación de un Sistema de gestión de base de datos o SGBD. Una base de datos puede definirse como un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Los gestores de bases de datos se encargan de almacenar, administrar y gestionar información, la cual estará disponible para cualquier tipo de aplicación siempre y cuando la solicite con su sintaxis correcta y tenga los permisos respectivos para acceder a ella (ver Figura 12) (Camps Paré et al., n.d.).



Figura 12.- Sistema de gestión de bases de datos.

Fuente: "SISTEMA MANEJADOR DE BASES DE DATOS Base de Datos :," n.d..

##### 4.2.5.1 Objetivos y servicios de los SGBD.

- **Consultas no predefinidas y complejas:** Los usuarios podrán hacer consultas de cualquier tipo y complejidad directamente al SGBD. El SGBD tendrá que responder inmediatamente sin que estas consultas estén preestablecidas; es decir, sin que se tenga que escribir, compilar y ejecutar un programa específico para cada consulta (Camps Paré et al., n.d.).
- **Flexibilidad e independencia:** Para conseguir esta independencia, los usuarios deben poder desconocer las características físicas de la base de datos con que trabajan, es decir, que no necesitan saber nada sobre el soporte físico, ni estar al



corriente de qué sistema operativo se utiliza, qué índices hay, la compresión o no compresión de datos, etc. De este modo, se pueden hacer cambios de tecnología y cambios físicos para mejorar el rendimiento sin afectar a nadie. Este tipo de independencia recibe el nombre de independencia física de los datos. Por otro lado, también es necesario que los usuarios no tengan que hacer cambios cuando se modifica la descripción lógica o el esquema de la base de datos (por ejemplo, cuando se añaden o suprimen entidades o interrelaciones, atributos, etc)(Camps Paré et al., n.d.).

- **Problemas de la redundancia:** El SGBD debe permitir que el diseñador defina datos redundantes, pero entonces tendría que ser el mismo SGBD el que hiciese automáticamente la actualización de los datos en todos los lugares donde estuviesen repetidos (Camps Paré et al., n.d.).
- **Integridad de los datos:** Cuando el SGBD detecte que un programa quiere hacer una operación que va contra las reglas establecidas al definir la base de datos, no se lo deberá permitir, y le tendrá que devolver un estado de error. Al diseñar una base de datos para un sistema específico y escribir su esquema, no sólo definiremos los datos, sino también las reglas de integridad que queremos que el SGBD haga cumplir.

En ciertas ocasiones pueden ocurrir errores inesperados o de sintaxis con la base de datos que puede provocar una pérdida irremediable de los datos. Para ello, los SGBD cuentan con procesos de restauración capaces de reconstruir la base de datos y darle el estado consistente y correcto anterior al incidente. Esto se acostumbra a hacer gracias a la obtención de copias periódicas de los datos (se denominan copias de seguridad o back-up) y mediante el mantenimiento continuo de un diario (log) donde el SGBD va anotando todas las escrituras que se hacen en la base de datos (Camps Paré et al., n.d.).

- **Concurrencia de usuarios:** Un objetivo fundamental de los SGBD es permitir que varios usuarios puedan acceder concurrentemente a la misma base de datos (Camps Paré et al., n.d.).
- **Seguridad:** esta es una característica muy importante y hace referencia a los temas relativos a la confidencialidad, las autorizaciones, los derechos de acceso, etc. Los SGBD permiten definir autorizaciones o derechos de acceso a diferentes

niveles: al nivel global de toda la base de datos, al nivel entidad y al nivel atributo. Los SGBD actuales incluyen también técnicas de encriptación de la información que manejan (Camps Paré et al., n.d.).

#### 4.2.5.2 Elementos de un SGBD.

Los elementos que integra un SGBD son los siguientes:

- **El motor de la base de datos** acepta peticiones lógicas de los otros subsistemas del SGBD, las convierte en su equivalente físico y accede a la base de datos y diccionario de datos en el dispositivo de almacenamiento (Camps Paré et al., n.d.).
- **El subsistema de definición de datos** ayuda a crear y mantener el diccionario de datos y define la estructura del fichero que soporta la base de datos (Camps Paré et al., n.d.).
- **El subsistema de manipulación de datos** ayuda al usuario a añadir, cambiar y borrar información de la base de datos y la interroga para extraer información. El subsistema de manipulación de datos suele ser el interfaz principal del usuario con la base de datos. Permite al usuario especificar sus requisitos de la información desde un punto de vista lógico (Camps Paré et al., n.d.).
- **El subsistema de generación de aplicaciones** contiene utilidades para ayudar a los usuarios en el desarrollo de aplicaciones. Usualmente proporciona pantallas de entrada de datos, lenguajes de programación e interfaces (Camps Paré et al., n.d.).
- **El subsistema de administración** ayuda a gestionar la base de datos ofreciendo funcionalidades como almacenamiento y recuperación, gestión de la seguridad, optimización de preguntas, control de concurrencia y gestión de cambios (Camps Paré et al., n.d.).

#### 4.2.5.3 Administración y funciones dentro de un SGBD.

El usuario o aplicación puede realizar las siguientes acciones:

- **Buscar datos:** Los sistemas de gestión de bases de datos permiten al usuario o aplicación obtener datos, a partir de condiciones de búsqueda establecidas (consultas).
- **Añadir datos:** El usuario puede añadir o ingresar datos en el instante que sea necesario.

- **Modificar datos:** Los sistemas de gestión de bases de datos permiten al usuario modificar los datos y los campos que definen los tipos de datos almacenados, cuando sea necesario.
- **Eliminar datos:** El usuario puede eliminar los datos que no están siendo utilizados en cualquier instante de tiempo.

#### 4.2.5.4 Bases de datos relacionales.

Las bases de datos relacionales son un tipo de bases de datos basado en relaciones o tablas (Sánchez, n.d.). Las características principales de una base de datos relacional son las siguientes:

- Una base de datos relacional se compone de varias tablas o relaciones (interconexiones).
- Las tablas que componen una base de datos relacional son un conjunto de registros, filas o tuplas.
- No puede existir dos tablas con el mismo nombre (no permite la interconexión).
- El lenguaje estándar para generar consultas a bases de datos relacionales es SQL (*Structured Query Language* o Lenguaje de consulta estructurado).
- Garantiza una integridad referencial. Si se elimina un registro, se eliminan todos los registros dependientes.
- La estructura referencial permite generar sistemas escalables.

Existen una gran variedad de SGBD relacionales y algunos de los más utilizados se detallan en la Tabla 6:

Tabla 6.- Comparativa de SGBD.  
Fuente: Iruela, n.d.

| Nombre del SGBD             | S.O. para su ejecución. | Lenguajes de programación soportados  | Licencia          | Características   |
|-----------------------------|-------------------------|---|-------------------|---|
| <b>Microsoft Access</b>     | Microsoft Windows       | - C<br>- C++  | Propietaria       | - Access es de interfaz gráfica, lo que permite aprovechar la potencia gráfica de Windows.<br>- Facilita la administración de datos.<br>- Automatiza fácilmente muchas tareas sin necesidad de programar.<br>- Es posible producir informes y formularios efectivos en un solo documento. |
| <b>MySQL</b>                | Multiplataforma         | - C<br>- C++<br>- Pascal<br>- Delphi<br>- Java<br>- PHP<br>- Python<br>- Ruby   | Libre/Propietaria | - Velocidad al realizar las operaciones.<br>- Bajo costo en requerimientos para la elaboración de bases de datos.<br>- Facilidad de configuración e instalación.  |
| <b>Oracle Database</b>      | Multiplataforma         | - Java<br>- PSL/SQL   | Propietaria       | - Soporte de transacciones.<br>- Estabilidad.<br>- Escalabilidad.<br>- Es multiplataforma.  |
| <b>Microsoft SQL Server</b> | Microsoft Windows       | - SQL<br>- C<br>- C++   | Propietaria       | - Soporte de transacciones.<br>- Escalabilidad, estabilidad y seguridad.<br>- Soporta procedimientos almacenados.<br>- Incluye también un potente entorno gráfico de administración.<br>- Permite administrar información de otros servidores de datos.                                   |
| <b>PostgreSQL</b>           | Multiplataforma         | - PL/PgSQL<br>- C<br>- C++<br>- Java PL/Java web<br>- Python<br>- Ruby<br>- PHP | Libre/Propietaria | - Alta concurrencia: mediante un sistema denominado MVCC (Acceso concurrente multiversión).<br>- Amplia variedad de tipos nativos: provee nativamente varios soportes.<br>- Ahorros considerables de costos de operación.<br>- Estabilidad y confiabilidad.                               |

#### 4.2.5.5 MySQL.

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto que, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario. El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. En las últimas versiones de este SGBD se pueden destacar las siguientes características:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos. El servidor soporta mensajes de error en distintas lenguas.

Las ventajas y desventajas de este SGBD se indican en la Tabla 7:

*Tabla 7.- Ventajas y desventajas de MySQL*

*Fuente: Enríquez, Maldonado, Nakamura, & Nogueron, 2017 .*

|                    |  |
|--------------------|--|
| <b>VENTAJAS</b>    | <ul style="list-style-type: none"><li>- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.</li><li>- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.</li><li>- Facilidad de configuración e instalación.</li><li>- Soporta gran variedad de Sistemas Operativos Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.</li><li>- Conectividad y seguridad.</li></ul> |
| <b>DESVENTAJAS</b> | <ul style="list-style-type: none"><li>- Un gran porcentaje de las utilidades de MySQL no están documentadas.</li><li>- No es intuitivo, como otros programas (<i>ACCESS</i>).</li></ul>  |

### 4.3 COMPRESIÓN DE VIDEO.

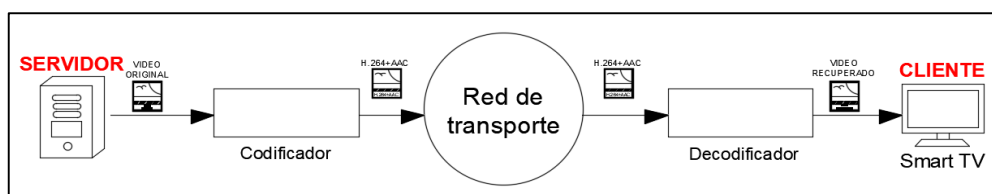
A la hora de instalar una determinada aplicación sobre una red IP, es muy importante tener en cuenta la carga que supondría el funcionamiento de dicha aplicación sobre la red, pudiendo así determinar si sería factible su implementación. En el caso de aplicaciones multimedia, concretamente las de reproducción y transmisión de video, se cuenta con las conocidas como técnicas de compresión de video digital, las cuales permiten reducir la carga que este tipo de datos representa sobre la red en la cual funcionan.

La compresión de vídeo consiste en reducir el número de datos usado para representar las imágenes y eliminar datos redundantes del vídeo para que el archivo digital se pueda enviar a través de la red y almacenar de una manera más eficiente en discos informáticos. Con técnicas eficaces se puede reducir el tamaño del fichero sin que apenas afecte a la calidad de la imagen (Cho, Kim, Shin, & Choi, 2010).

La compresión de vídeo es una compensación entre el espacio de almacenamiento, la calidad del vídeo y el costo del hardware requerido para descomprimir el vídeo en un tiempo razonable. La mayoría de las compresiones de vídeo son con pérdida, es decir, se realizan con la premisa de que muchos de los datos presentes antes de la compresión no son necesarios para percibir una buena calidad de vídeo. Si se pierde calidad durante la compresión, es imposible recuperar la imagen a su calidad original (Cho et al., 2010).

#### 4.3.1 Códecs de video.

En el proceso de compresión se aplica un algoritmo al vídeo original (codificador) creando así un archivo comprimido, el cual se halla listo para ser almacenado o transmitido. Al momento de ser transferido, es necesario aplicar su respectivo algoritmo inverso (decodificador) en el destino, de manera que pueda recuperarse la información comprimida de modo que se genera un nuevo archivo de vídeo que incluye prácticamente el mismo contenido que el vídeo original (Yáñez Izquierdo, 2011).



*Figura 13.- Proceso básico de codificación y decodificación de video.  
Fuente: Elaboración propia.*

El tiempo que se tarda en comprimir, enviar, descomprimir y mostrar un archivo es lo que se denomina latencia y cuanto más avanzado sea el algoritmo de compresión, mayor será su latencia. El par de algoritmos que funcionan conjuntamente se denomina códec de video (codificador/decodificador) (Yáñez Izquierdo, 2011).

Los códecs de vídeo de estándares diferentes no suelen ser compatibles entre sí, es decir, el contenido de vídeo comprimido con un estándar no se puede descomprimir con otro estándar diferente (Yáñez Izquierdo, 2011).

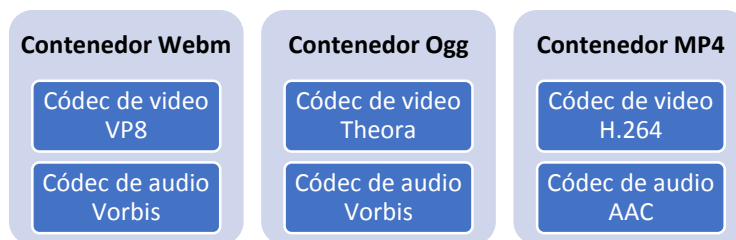
Para que un archivo de video pueda ser audible, requiere también su respectivo códec de audio. Es por ello que los códecs de video suelen ir acompañados de un códec de audio, y estos a su vez se los comprime dentro de un archivo denominado “contenedor” o “formato contenedor”. Un formato contenedor es un tipo de formato de archivo que almacena información de vídeo, audio, metadatos e información de sincronización y corrección de errores siguiendo un formato preestablecido en su especificación técnica. En la Tabla 8 se puede ver ejemplos de formatos contenedores de video.

*Tabla 8.- Formatos contenedor de video.  
Fuente: Elaboración propia.*

| FORMATO CONTENEDOR | VIDEO  |        |        |        |       |        | AUDIO  |        |        |       |      |      |
|--------------------|--------|--------|--------|--------|-------|--------|--------|--------|--------|-------|------|------|
|                    |        |        |        |        |       |        |        |        |        |       |      |      |
| <b>MPEG-TS</b>     | MPEG-2 | H.264  | -      | -      | -     | -      | AAC    | PCM    | MP3    | AC3   | -    | -    |
| <b>MPEG-PS</b>     | MPEG-2 | -      | -      | -      | -     | -      | PCM    | MP3    | AC3    | -     | -    | -    |
| <b>MPEG 1</b>      | MPEG-1 | -      | -      | -      | -     | -      | PCM    | MP3    | -      | -     | -    | -    |
| <b>OGG</b>         | THEORA | DIRAC  | OGGUVS | -      | -     | -      | PCM    | VORBIS | FLAC   | SPEEX | OPUS | -    |
| <b>WMV</b>         | H.264  | WMV    | -      | -      | -     | -      | WMA    | -      | -      | -     | -    | -    |
| <b>MP4</b>         | H.264  | H.265  | MPEG-4 | MPEG-2 | H.263 | DIRAC  | AAC    | VORBIS | MP3    | AC3   | WMA  | -    |
| <b>MOV</b>         | MPEG-1 | MPEG-2 | MPEG-4 | H.263  | H.264 | M-JPEG | AAC    | PCM    | MP3    | AC3   | -    | -    |
| <b>FLV</b>         | H.264  | -      | -      | -      | -     | -      | AAC    | PCM    | MP3    | -     | -    | -    |
| <b>WEBM</b>        | VP8    | VP9    | -      | -      | -     | -      | VORBIS | OPUS   | -      | -     | -    | -    |
| <b>AVI</b>         | H.263  | H.264  | -      | -      | -     | -      | AAC    | PCM    | VORBIS | MP3   | AC3  | FLAC |
| <b>3GP</b>         | MPEG4  | H.263  | -      | -      | -     | -      | AAC    | -      | -      | -     | -    | -    |

En los sistemas web, bajo el estándar HTML5, actualmente existen 3 formatos de video soportados de manera predeterminada, si se desea agregar más formatos, es necesario instalar nuevos complementos que permitan su compatibilidad. Los formatos predeterminados son:

- **Ogg/ogv:** los archivos bajo el contenedor ogg/ogv pueden utilizar como códec de video Theora junto a un códec de audio Vorbis (Yan, Yang, Lan, & Tong, 2012).
- **Webm:** los archivos webm utilizan el códec de video VP8 o VP9, junto al códec de audio Vorbis u Opus (Yan et al., 2012).
- **Mp4:** los archivos bajo el encapsulado mp4 utilizan el códec de video H.264 o H.265 junto al códec de audio AAC (Yan et al., 2012).



*Figura 14.- Formatos contenedor de video para HTML5.  
Fuente: Elaboración propia.*

#### 4.3.1.1 Formato Mp4 (H.264/AAC).

Este formato contenedor utiliza el códec de video H.264 junto al códec de audio AAC. Este códec de video fue desarrollado por la Organización Internacional de Normalización (ISO), la Comisión Electrotécnica Internacional (IEC) y el Grupo de Expertos de Imágenes en Movimiento (MPEG, por sus siglas en inglés) en el año 2003 (Yoon et al., 2012). Se lo puede encontrar bajo el estándar ISO/IEC 14496-14 como MPEG-4 parte 14.

La fama de este códec, gracias a la implementación de HTML5 en las plataformas web, no sólo se limita a los servicios web sino también a las tecnologías de video digital en general. Cada vez más se tienen dispositivos que soporten y manejen este códec de video, incluso existen dispositivos los cuales están optimizados específicamente para este formato (Yoon et al., 2012). Se puede reducir sus características a las siguientes:

- Tiene baja latencia sin sacrificar en gran medida su calidad de imagen.
- Tiene una tasa de compresión de hasta el 80% comparado con el formato MPEG y hasta un 50% en comparación con MPEG-4 parte 2.
- Es un códec libre, aunque en sus inicios fue de pago.
- Al ser muy popular tiene un alto nivel de operabilidad y disponibilidad en una gran variedad de dispositivos.



- Presenta diferentes niveles de funcionamiento según la aplicación que se le dé (Andrews, 2015).

#### **4.3.1.2 Formato Ogv/Ogg (Theora/Vorbis).**

Se trata de un contenedor de video creado en el año 2004 a partir del código fuente de VP3 y que fue liberado por la empresa On2 en 2001. En aquel entonces, era el único competidor del códec H.264 en la web. Se caracteriza por usar el códec de video Theora junto al códec de audio Vorbis. Presenta grandes desventajas frente al H.264 como por ejemplo que su acogida en dispositivos digitales es mucho menor, por lo que los dispositivos digitales que permiten crear contenidos nativamente en Theora son prácticamente nulos.

La principal ventaja de Ogv/Ogg es que es libre, por lo que no hay que pagar por su uso, lo cual lo hace fácilmente de implementar en sistemas destinados a cualquier tipo de navegador web, aunque se trate de un navegador web nuevo como por ejemplo Chromium.

Como principales desventajas de este códec es que consume al menos un 25% más de recursos hardware que H.264 (tiempo, procesador, memoria, etc). Además, ofrece una calidad de imagen menor y los ficheros generados son algo más grandes, lo que indica que su tasa de compresión no está tan optimizada.

Dado que sus características de rendimiento resultan inferiores a las del códec H.264 y por su poca acogida, resulta difícil encontrar un navegador web compatible con este códec.

#### **4.3.1.3 Formato Webm (VP8/Vorbis).**

Webm es un formato multimedia abierto y desarrollado por Google en el año 2010. Su uso principalmente está orientado para la web junto con HTML5. Se conforma del códec de vídeo VP8 (desarrollado originalmente por On2 Technologies) y el códec de audio Vorbis dentro de un contenedor Matroska (WebM, n.d.-a).

Entre las características que nos ofrece VP8 como códec se tiene:

- Es de código abierto y popular entre los desarrolladores de aplicaciones de video en la web.

- No es un códec ampliamente adoptado, por lo que la interoperabilidad con la mayoría de sistemas de video actuales resulta un problema.
- Es lento en comparación con H.264, es decir, su tiempo de codificación es elevado.
- Promete utilizar la mitad de ancho de banda que H.264 pero con muchas desventajas (WebM, n.d.-b).

En la Tabla 9 se indica una comparativa entre las características de los 3 tipos de formatos:

*Tabla 9.- Características de formatos MP4, WEBM Y OGG*

*Fuente: López Herreros, 2014.*

| FORMATO     | DESARROLLADOR   | LICENCIA | FORMATO DE CODIFICACIÓN DE VIDEO  | FORMATO DE CODIFICACIÓN DE AUDIO                    | STREAMMING |
|-------------|-----------------|----------|---|---|------------|
| <b>MP4</b>  | ISO, IEC y MPEG | Libre    | - H.265<br>- MPEG-2 Part 2<br>- MPEG-4 ASP<br>- H.264/MPEG-4 AVC<br>- H.263<br>- VC-1 | - AAC<br>- MP3<br>- Vorbis<br>- AC-3<br>- WMA       | Si         |
| <b>WEBM</b> | Google          | Libre    | - VP8<br>- VP9  | - Vorbis<br>- Opus                                  | Si         |
| <b>OGG</b>  | Xiph.org        | Libre    | - Theora<br>- Dirac<br>- OggUVS   | - Vorbis<br>- FLAC<br>- Speex<br>- Opus<br>- OggPCM | Si         |

#### 4.4 ENTORNO DE DESARROLLO WEB.

La creación de un sitio web o una página de internet en general implica dos factores, el front-end y el back-end. El *back-end* de un sistema de *software* es la parte que procesa la entrada de datos que se efectuó desde el *front-end* es decir, son los procesos que utiliza el administrador del sitio con sus respectivos sistemas para resolver las peticiones de los usuarios. Esta parte del sistema se halla compuesto por bases de datos, servidores y la programación para procesar los datos, entre otras cosas. Por otro lado, el *front-end* hace referencia a la visualización del usuario navegante o, dicho de otra manera, es la parte que interactúa con los usuarios (interfaz de usuario). De una manera práctica, el *front-end* sería todos esos formularios que se presentan en las webs, las aplicaciones que manejamos y todas esas cosas que tengan que realizar los usuarios (Pascual Torres, 2016). En la Figura 15 se ilustra la diferenciación entre el back-end y el front-end de un sistema.

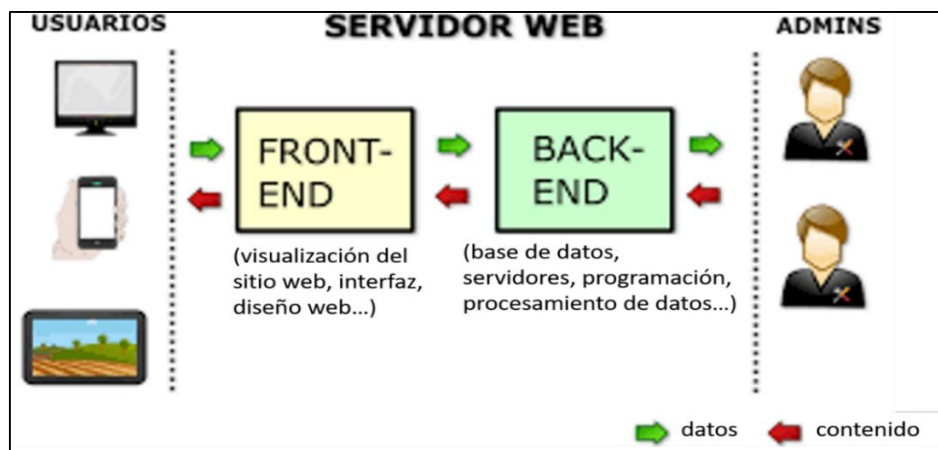


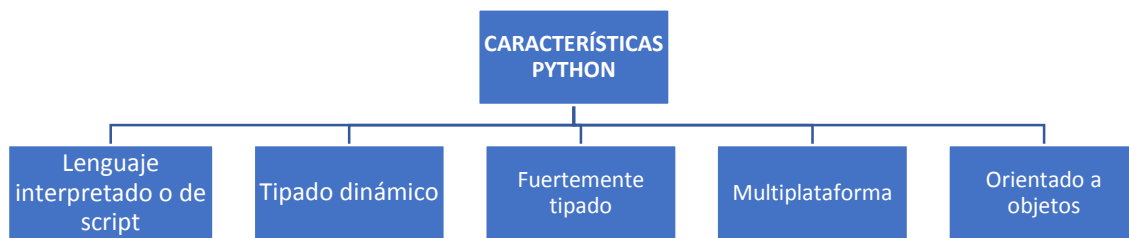
Figura 15.- Diferenciación del back-end y front-end de un sistema.

Fuente: ALEGSA, n.d..

Para el *back-end* del Infochannel se utilizó Python como lenguaje de programación, en conjunto con el *framework* Django, cuyas características y funcionalidades se detallan a continuación.

##### 4.4.1 Python.

Python es un lenguaje de programación creado por Guido Van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible (González Duque, 2017). Se puede definir Python, de acuerdo a las características que se muestran en la Figura 16:



*Figura 16.- Características Python.  
Fuente: González Duque, 2017.*

- **Lenguaje interpretado o de script:** es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (González Duque, 2017).
- **Tipado dinámico:** el tipo de dato se determina en tiempo de ejecución según el tipo del valor que se le asigne a la variable (González Duque, 2017).
- **Fuertemente tipado:** No se permite tratar a una variable como si fuera de un tipo distinto al que tiene, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente (González Duque, 2017).
- **Multiplataforma:** El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.) por lo que si no utilizamos librerías específicas de cada plataforma nuestro programa podrá correr en todos estos sistemas sin grandes cambios (González Duque, 2017).
- **Orientado a objetos:** La orientación a objetos es un paradigma de programación en el que los conceptos del mundo real relevantes para nuestro problema se trasladan a clases y objetos en nuestro programa. La ejecución del programa consiste en una serie de interacciones entre los objetos (González Duque, 2017).

En cuanto a sus ventajas y desventajas se enuncian en la Tabla 10:

Tabla 10.- Ventajas y desventajas de Python.  
Fuente: González Duque, 2017 .

| VENTAJAS   | DESVENTAJAS  |
|--|--|
| Accesible para todo el mundo ( <i>Open Source</i> ).   | No es adecuado para la programación de bajo nivel o para aplicaciones en las que el rendimiento sea crítico. |
| Sintaxis simple, clara y sencilla.   |  |
| Es muy compacto, lo que se traduce en ahorro de código a gran escala.  |  |
| Es muy popular, lo que significa que tiene una gran comunidad de desarrolladores detrás, los cuales resultan útiles al momento de solventar errores en el código que se esté trabajando. |  |

#### 4.4.2 Framework.

El término *framework* o marco de referencia, hoy en día, es bastante mencionado no sólo en el desarrollo de sistemas de *software*, sino también en el campo de las aplicaciones web. Actualmente podemos encontrar *frameworks* especializados en el desarrollo de aplicaciones, ya sea médicas, para juegos, para servidores, entre otras (Gutiérrez, n.d.).

Dicho de forma general, un *framework* se define como un entorno de desarrollo incompleto, modular y altamente personalizable, el cual, según se configure, permite crear de manera rápida aplicaciones específicas (Gutiérrez, n.d.).

En la Figura 17 se enumeran los objetivos de un framework:

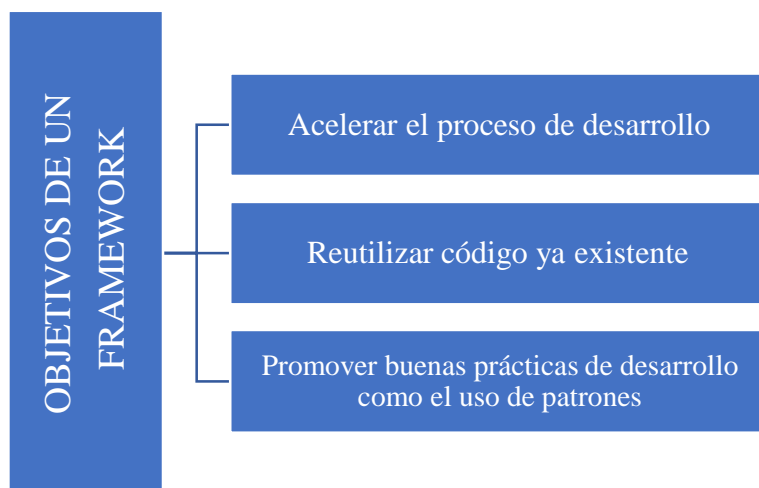


Figura 17.- Objetivos de un Framework.  
Fuente: Gutiérrez, n.d..

En la Tabla 11, se enumera las características comunes para todo *framework* de desarrollo web:

**Tabla 11.- Características generales de un Framework.**  
**Fuente:** Mendoza Vázquez, n.d..

|  |  |
|--|--|
| <b>Abstracción de URLs y sesiones</b>      | No es necesario manipular directamente las URLs ni las sesiones, el <i>framework</i> ya se encarga de hacerlo.   |
| <b>Acceso a bases de datos</b>             | Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos.  |
| <b>Controladores</b>                       | La mayoría de <i>frameworks</i> implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.  |
| <b>Autenticación y control de acceso</b>   | Incluyen mecanismos para la identificación de usuarios mediante <i>login</i> y <i>password</i> y permiten restringir el acceso a determinadas páginas a determinados usuarios.   |
| <b>Internacionalización</b>                | Se refiere al proceso de diseño de una aplicación de <i>software</i> para que pueda adaptarse a varios idiomas y regiones sin cambios de ingeniería. Esto garantiza el uso de potenciales usuarios de todo el mundo.   |
| <b>Separación entre diseño y contenido</b> | Mediante la arquitectura MVC (Modelo-Vista-Controlador), los <i>frameworks</i> separan su estructura en 3 partes, que son: el modelo de datos, las entradas del usuario y la interfaz gráfica. Estos 3 componentes pueden evolucionar de manera independiente.   |
| <b>Caché</b>                               | Permite tener una copia de documentos web a la mano con el propósito de reducir el uso del ancho de banda, y peticiones al servidor.   |
| <b>Utilización de plantillas</b>           | Aplicando plantillas y el uso de variables para aquellas partes dinámicas donde se insertan los datos extraídos de una base de datos, se puede reducir en gran medida el número de páginas en un sitio. Además, con el uso de etiquetas especiales para las variables, se puede automatizar la actualización de ciertas zonas y reducir la programación. |
| <b>Ajax</b>                                | Permite realizar intercambios de menor información con el servidor de forma asíncrona, de manera que no se tiene que recargar la página web cada vez que el usuario pide un cambio. Esto aumenta la interactividad con la página, la velocidad y la usabilidad.  |

#### 4.4.3 Framework Django.

Django nació de forma natural de aplicaciones de la vida real escritas por un equipo de desarrolladores web del diario *Lawrence Journal-World* en el año 2003. Este *framework* surgió cuando los programadores de este periódico estadounidense, comenzaron a usar Python para crear sus aplicaciones (Holovaty & Kaplan Moss, 2015).

Este *framework* nació gracias a la imperiosa necesidad de la creación de aplicaciones a una velocidad vertiginosa, para mantener sus sitios web actualizados y siempre a la vanguardia (Holovaty & Kaplan Moss, 2015).

En el año 2005 pasó a ser software libre y adquirió el nombre de Django en honor al guitarrista de Jazz “Django Reinhardt”. Hoy en día, Django es un proyecto estable y maduro, de código abierto con cientos de miles de colaboradores y usuarios de todo el mundo (Holovaty & Kaplan Moss, 2015).

Como se ve en la Figura 18, Django forma parte de la tercera generación del desarrollo web:

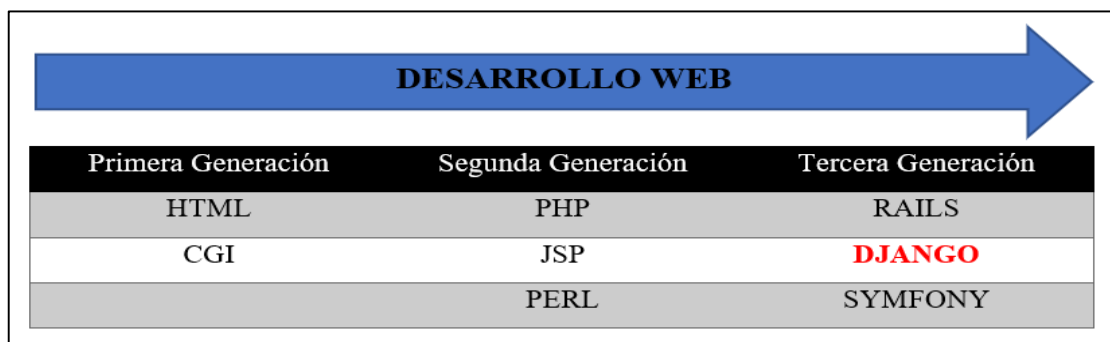


Figura 18.- Generaciones de desarrollo web.  
Fuente: Holovaty & Kaplan Moss, 2015.

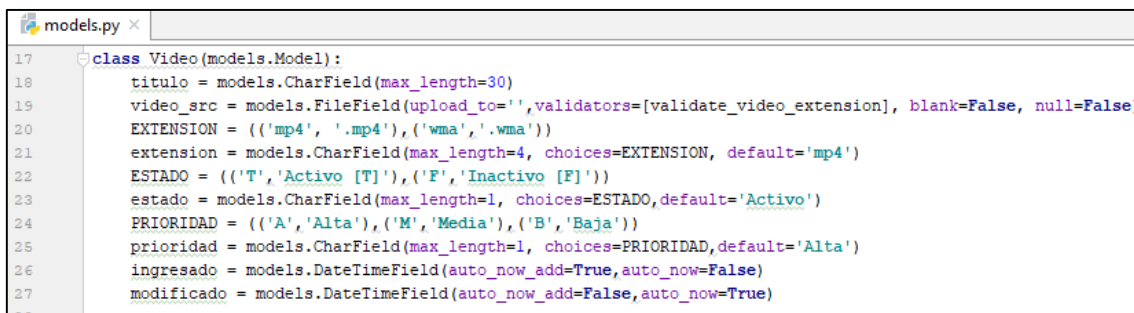
##### 4.4.3.1 Componentes de Django.

Además de las ventajas de un *framework*, Django promueve el desarrollo rápido e impulsa el desarrollo de código limpio al promover buenas prácticas de desarrollo web, siguiendo el principio DRY (*Don't Repeat Yourself* o No te repitas). Por ello usa una modificación de la arquitectura Modelo-Vista-Controlador (MVC), llamada MTV (*Model – Template – View*), que sería Modelo-Plantilla-Vista, esta forma de trabajar permite que sea pragmático (Holovaty & Kaplan Moss, 2015).

Su arquitectura MTV al basarse en MVC consta de los siguientes componentes:

- **El modelo (Model).**

En Django, al igual que la arquitectura MVC, este componente mantiene su propio nombre, y es el encargado de definir los datos almacenados, los cuales se encuentran en forma de clases de Python. Cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros y métodos que los configuran (ver Figura 19). Todo esto permite indicar y controlar el comportamiento de los datos (Holovaty & Kaplan Moss, 2015).

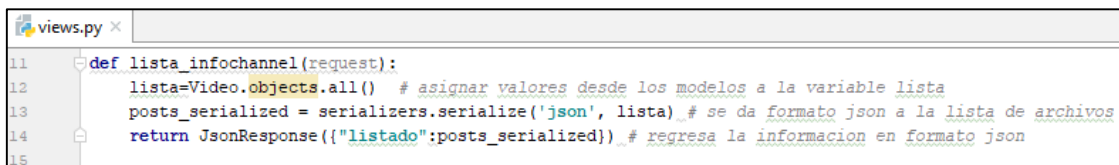


```
17 class Video(models.Model):
18     titulo = models.CharField(max_length=30)
19     video_src = models.FileField(upload_to='', validators=[validate_video_extension], blank=False, null=False)
20     EXTENSION = (('mp4', '.mp4'), ('wma', '.wma'))
21     extension = models.CharField(max_length=4, choices=EXTENSION, default='mp4')
22     ESTADO = (('T', 'Activo [T]'), ('F', 'Inactivo [F]'))
23     estado = models.CharField(max_length=1, choices=ESTADO, default='Activo')
24     PRIORIDAD = (('A', 'Alta'), ('M', 'Media'), ('B', 'Baja'))
25     prioridad = models.CharField(max_length=1, choices=PRIORIDAD, default='Alta')
26     ingresado = models.DateTimeField(auto_now_add=True, auto_now=False)
27     modificado = models.DateTimeField(auto_now_add=False, auto_now=True)
```

Figura 19.- Ejemplo de modelo en Django.  
Fuente: Elaboración propia.

- **La vista (View).**

En Django, el componente “Controlador” pasa a llamarse “Vista”, a diferencia de la arquitectura MVC, y esta se presenta en forma de funciones en Python (ver Figura 20), cuyo propósito es principalmente determinar qué datos serán visualizados. El ORM (*Object-Relational mapping* o Mapeo objeto-relacional) de Django permite escribir código Python en lugar de SQL para hacer las consultas que necesita la vista. La vista también se encarga de tareas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. Hay que recalcar que la vista no tiene nada que ver con el estilo de presentación de los datos, sólo se encarga de los datos, la presentación es tarea de la plantilla (Holovaty & Kaplan Moss, 2015).



```
11 def lista_infochannel(request):
12     lista=Video.objects.all() # asignar valores desde los modelos a la variable lista
13     posts_serialized = serializers.serialize('json', lista) # se da formato json a la lista de archivos
14     return JsonResponse({"listado":posts_serialized}) # regresa la informacion en formato json
15
```

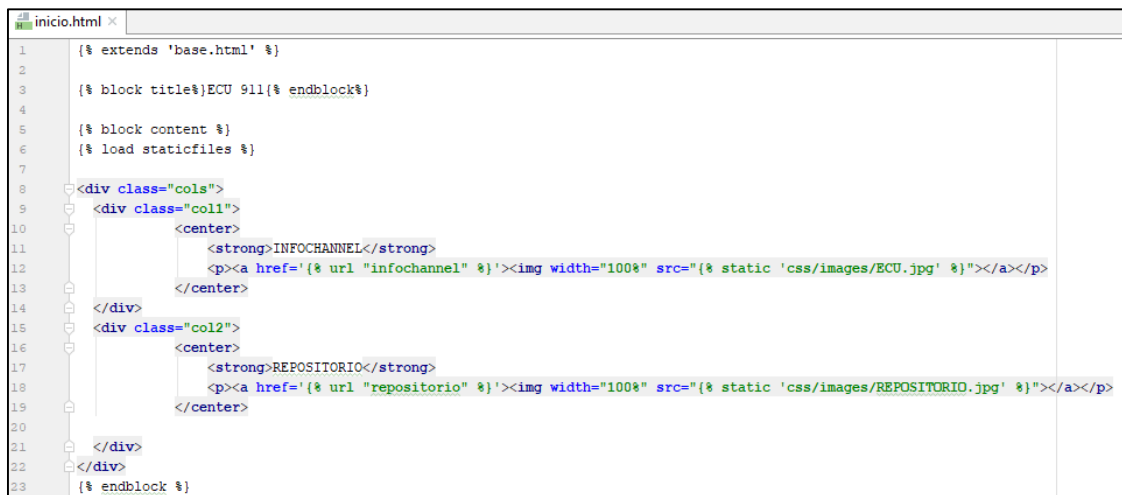
Figura 20.- Ejemplo de Vista en Django.  
Fuente: Elaboración propia.



La Vista al igual que los Modelos de Django, forman parte de la programación que interactúa con la base de datos del sitio web, lo que indica que forma parte del *Back-end* del sistema.

- **La plantilla (*Template*).**

Al igual que en la Vista, en Django el componente Vista del MVC toma el nombre de Plantilla, y esta es básicamente una página HTML con algunas etiquetas extras propias de Django, y es dónde se programa la estructura visual del sitio web desde el punto de vista del cliente (*Frontend*). Una plantilla además de renderizar código HTML también puede manejar otros lenguajes como CSS, JavaScript, XML, CSV, entre otros (Holovaty & Kaplan Moss, 2015).



```
1  {% extends 'base.html' %}
2
3  {% block title%}ECU 911{% endblock%}
4
5  {% block content %}
6  {% load staticfiles %}
7
8  <div class="cols">
9    <div class="col1">
10     <center>
11       <strong>INFOCHANNEL</strong>
12       <p><a href='{% url "infochannel" %}'></a></p>
13     </center>
14   </div>
15   <div class="col2">
16     <center>
17       <strong>REPOSITORIO</strong>
18       <p><a href='{% url "repositorio" %}'></a></p>
19     </center>
20   </div>
21 </div>
22 </div>
23  {% endblock %}
```

Figura 21.- Ejemplo de Plantilla HTML en Django.  
Fuente: Elaboración propia.

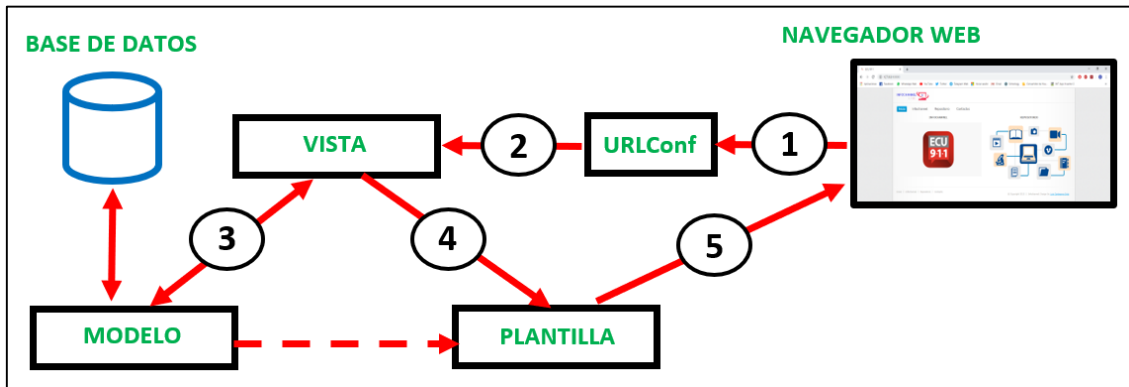
La plantilla recibe los datos de la vista y luego los organiza para la presentación en el navegador web. Las etiquetas que Django usa para las plantillas permiten que sea flexible para los diseñadores del *Front-end*, incluso tiene estructuras de datos como “*if*”, por si es necesaria una presentación lógica de los datos, estas estructuras son limitadas para evitar un desorden poniendo cualquier tipo de código Python. Esto permite que la lógica del sistema siga permaneciendo en la Vista (Holovaty & Kaplan Moss, 2015).

#### 4.4.3.2 Configuración de las rutas en Django.

Django posee un mapeo de URLs que permite controlar el despliegue de las vistas, esta configuración es conocida como URLConf. El trabajo del URLConf es leer la URL que el usuario solicitó, encontrar la vista apropiada para la solicitud y pasar cualquier variable

que la vista necesite para completar su trabajo. Además permite que las rutas manejadas por Django sean agradables y entendibles para el usuario (Holovaty & Kaplan Moss, 2015).

La Figura 22 resume el funcionamiento de Django:



*Figura 22.- Funcionamiento del MTV de Django.  
Fuente: Holovaty & Kaplan Moss, 2015.*

De la Figura 22, se define el siguiente proceso:

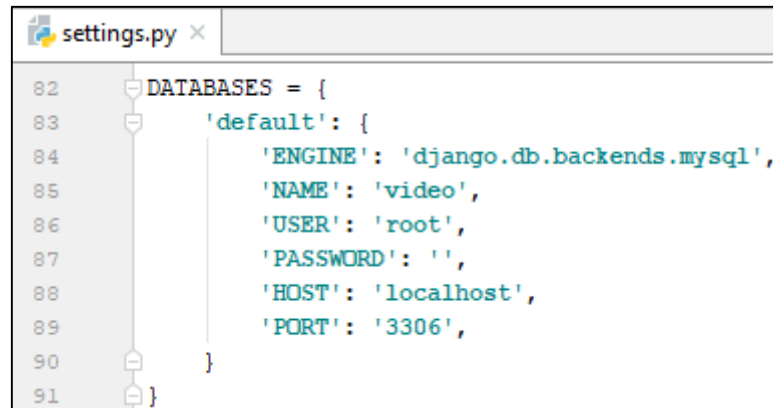
1. El navegador web envía una solicitud.
2. El URLConf interpreta la solicitud y ubica la vista apropiada.
3. La vista interactúa con el modelo para obtener datos.
4. La vista llama a la plantilla.
5. La plantilla renderiza la respuesta a la solicitud del navegador (Holovaty & Kaplan Moss, 2015).

#### **4.4.3.3 Integración de Django con la base de datos.**

En las aplicaciones web modernas, la lógica arbitraria a menudo implica interactuar con una base de datos. Detrás de escena, un sitio web impulsado por una base de datos se conecta a un servidor de base de datos, recupera algunos datos de este, y los muestra con un formato agradable en una página web, del mismo modo el sitio puede proporcionar funcionalidad que permita a los visitantes del sitio ingresar datos a la base de datos; por su propia cuenta. Django es apropiado para crear sitios web que manejen una base de datos, ya que incluye una manera fácil pero poderosa de realizar consultas a bases de datos utilizando Python (Holovaty & Kaplan Moss, 2015).

Gracias al modelo MTV y a la separación de los procesos explicada en el apartado anterior (ver sección 4.3.3.1), Django ofrece una manera cómoda de establecer la conexión con

base de datos y para ello se requiere algunas configuraciones iniciales: se necesita indicarle a Django qué servidor de base de datos usar y cómo conectarse al mismo (Holovaty & Kaplan Moss, 2015). Estas configuraciones se muestran en la Figura 23:

A screenshot of a code editor window titled 'settings.py'. The code defines a 'DATABASES' dictionary with a 'default' configuration for a MySQL database. The configuration includes engine, name, user, password, host, and port.

```
82 DATABASES = {
83     'default': {
84         'ENGINE': 'django.db.backends.mysql',
85         'NAME': 'video',
86         'USER': 'root',
87         'PASSWORD': '',
88         'HOST': 'localhost',
89         'PORT': '3306',
90     }
91 }
```

Figura 23.- Ejemplo de las configuraciones iniciales de la conexión a la base de datos.  
Fuente: Elaboración propia.

Gracias a estas simples líneas de códigos, la conexión a la base de datos se establece de manera global, de forma que podemos acceder a la información necesaria sin necesidad de escribir nuevamente las líneas de acceso a la base de datos.

#### 4.4.3.4 Interfaz de administración de Django.

Para cierto tipo de sitios Web, una interfaz de administración es una parte esencial de su infraestructura. Ya que se trata de una interfaz basada en HTML, limitada a los administradores autorizados, y les permite agregar, editar y eliminar el contenido del sitio.

Una de las partes más poderosas y atractivas de Django es la interfaz administrativa. Que se encarga de leer automáticamente los metadatos de los modelos configurados, para así proveer una interfaz poderosa y lista para la producción, de manera que inmediatamente se pueda comenzar a añadirle contenido al sitio Web. Sin embargo, la interfaz de administración es enteramente opcional, porque únicamente un cierto tipo de sitios necesitan esta funcionalidad. Esto significa que se puede activar o desactivar según las necesidades específicas del sistema (Holovaty & Kaplan Moss, 2015).

#### 4.4.4 Diseño de la interfaz de usuario (front-end).

Internet es la tecnología que involucra casi cualquier actividad que realiza el ser humano, hoy en día la mayoría de la información se traslada utilizándolo. Si se quisiera cuantificar las tecnologías que utiliza internet se podría llegar a varios miles de ejemplos, pero existe

una tecnología que es la base de todos los proyectos, avances e innovación que suceden en internet: HTML (Dimas & Castillo, 2016).

HTML es una tecnología que no funciona por sí sola, se complementa con otras formando así el conjunto que consolida la base sólida de cualquier proyecto web: HTML, CSS y JavaScript (Dimas & Castillo, 2016).

Para la interfaz de usuario o *front-end* de este proyecto se usó las tecnologías de programación web HTML5, CSS3 y JavaScript, en colaboración con el *framework* Django. Como editor de código se usó el IDE (*Integrated Development Environment o Entorno de Desarrollo Integrado*) JetBrains PyCharm Community. A continuación, se describe cada una de estas herramientas:

#### **4.4.4.1 HTML5.**

HTML5 es un lenguaje usado para estructurar y presentar el contenido en la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, junto a CSS para los estilos y JavaScript para agregar dinamismo y funcionalidad. En un principio, HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos (como fotografías, animaciones, etc). Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos (Francés López & Palau Arnau, 2012).

HTML5 es la quinta revisión del estándar que fue creado en 1990. Esta nueva versión del estándar HTML hereda las ventajas de sus versiones pretéritas, y por supuesto, agregando nuevas funcionalidades entre las cuales se tiene (Francés López & Palau Arnau, 2012):

- Inclusión de nuevas etiquetas a las ya existentes (por ejemplo, la etiqueta “<section>”, “<audio>”, “<video>”, entre otras) que hacen posible el uso de nuevas funcionalidades sin la necesidad de instalar nuevos plugins.
- Inserción de multimedia en los sitios web, que ahora contarán con etiquetas HTML especiales para poder ser incluidos.
- Ahorro de código, debido a una sintaxis mucho más simple.

#### **4.4.4.2 CSS3.**

El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML. El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. CSS3 es la tercera versión de este lenguaje de programación (Francés López & Palau Arnau, 2012).

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento (etiquetas html y otros lenguajes) de su presentación (interfaz visual, tamaños, colores, márgenes, etc.). La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso, podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "<style>" (Francés López & Palau Arnau, 2012).

#### **4.4.4.3 JavaScript.**

JavaScript es un lenguaje de programación interpretado, orientado a objetos y de uso dinámico no estático. Se utiliza principalmente como parte de un navegador web y que permite mejorar la interfaz de usuario, las páginas web dinámicas, el acceso local a bases de datos, etc. JavaScript fue diseñado con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Tradicionalmente se ha utilizado en páginas HTML para realizar operaciones y únicamente en el lado del cliente, sin acceso a funciones del servidor. JavaScript es interpretado por el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML. Se puede incluir JavaScript tanto de manera interna como externa, es decir, dentro del propio fichero web como en un fichero externo que es como el W3C define que es lo correcto (Francés López & Palau Arnau, 2012).

#### **4.4.4.4 JetBrains PyCharm Community.**

PyCharm es un IDE dedicado de Python y Django que proporciona una amplia gama de herramientas esenciales para los desarrolladores de Python y que están estrechamente integradas en el programa de forma que se pueda crear un ambiente adecuado para el

desarrollo productivo de Python y el desarrollo web. Este es un IDE multiplataforma, por lo que puede trabajar sin problemas en Windows, Linux y MacOS (PyCharm documentation, 2017).

Con PyCharm es posible desarrollar aplicaciones no sólo en Python, sino que también se pueden desarrollar aplicaciones Django, Flask, Pyramid y WebToPy. También, es totalmente compatible con HTML (incluido HTML5), CSS, JavaScript y XML. Estos lenguajes se incluyen en el IDE a través de complementos y se activan de forma predeterminada. El soporte para otros idiomas también se puede agregar a través de complementos (PyCharm documentation, 2017).

## 5. MATERIALES Y MÉTODOS.






### 5.1 MATERIALES.

Para el desarrollo de este proyecto se utilizaron los siguientes elementos:

#### 5.1.1 Herramientas físicas.

Tabla 12.- Herramientas físicas.

Fuente: Elaboración propia.




| Herramienta   | Descripción  |
|---|--|
| <b>Modem Huawei HG531S</b><br>                       | Utilizado para crear una red local en la cual testear el sistema.  |
| <b>Patch Cord UTP Cat. 5e</b><br>                   | Utilizado para interconectar los equipos dentro de la red de pruebas.  |
| <b>Computadora portátil ASUS F541U</b><br>         | Se usaron como servidores de pruebas y como herramientas para el desarrollo del proyecto.  |
| <b>Computadora portátil DELL Inspiron 3421</b><br> |  |
| <b>Smart TV LG 42LN5700-SH</b><br>                 | Dispositivo de presentación y visualización del sistema dentro del edificio de la institución pública SIS ECU 911 de la ciudad de Loja. Se utilizó también para realizar las pruebas finales de rendimiento. |

|   |   |
|---|---|
| <p><b>Smart TV LG 42LB5800-SB</b></p>            | <p>Dispositivos utilizados para probar localmente (en la red de pruebas) el funcionamiento del sistema. En ellos se puede observar la funcionalidad y rendimiento del sistema, así como posibles errores en la presentación, utilizando sus navegadores web como clientes del servidor en desarrollo.</p> |
| <p><b>Smart TV TCL L32D2730</b></p>              |   |
| <p><b>Servidor Lenovo ThinkServer RD350</b></p>  | <p>Servidor del SIS ECU 911 de la ciudad de Loja. En él se halla instalado el sistema ya finalizado y en perfecto funcionamiento.</p>   |



### 5.1.2 Herramientas de software.

Tabla 13.- Herramientas de software.

Fuente: Elaboración propia.

| Herramienta  | Descripción  |
|--|--|
| <p><b>Entorno Python 3.7.2.</b></p>   | <p>Lenguaje de programación usado para la programación del back-end del proyecto.</p>  |
| <p><b>IDE JetBrains PyCharm Community (versión 183.5912.18).</b></p>          | <p>IDE desarrollado principalmente para Python, aunque también soporta HTML5, CSS3 y JavaScript. También es compatible con el Framework Django. Gracias a este software se pudo realizar la programación de la aplicación, depuración del código fuente y la integración de los diferentes elementos que conforman el sistema.</p> |
| <p><b>Framework Django (versión 2.1.7).</b></p>                               | <p>Es el Framework y base principal de este proyecto. Gracias a sus funcionalidades y estructura de trabajo, se logró crear el presente proyecto.</p>  |
| <p><b>Librerías Python utilizadas:</b></p> <ul style="list-style-type: none"> <li>- pysql (versión 0.9.3).</li> <li>- django-cleanup (versión 3.2.0).</li> </ul> | <p>Librerías utilizadas para la conexión entre Django y la base de datos en MySQL, y para la actualización de archivos que hayan sido reemplazados o eliminados de la base de datos.</p>   |



|   |  |
|---|--|
| <p><b>Lenguajes de programación: Python, HTML5, CSS3 y JavaScript.</b></p>   | <p>Son los lenguajes base para la programación del front-end del proyecto.</p>   |
| <p><b>Herramientas JavaScript:</b></p> <ul style="list-style-type: none"> <li>- Ajax.</li> <li>- jQuery.</li> </ul>  | <p>Herramientas JavaScript utilizadas para dar funcionalidad a la interfaz de usuario.</p>   |
| <p><b>Paquete de software libre XAMPP (X, Apache, MySQL, PHP y Perl) (versión 7.3.2-2).</b></p>                      | <p>Es el paquete de software que engloba el servidor web Apache y el SGBD MySQL.</p>   |
| <p><b>Navegadores Google Chrome, Mozilla Firefox, Opera y navegador de WebOS.</b></p>                              | <p>Navegadores sobre los cuales se probó el sistema.</p>   |
| <p><b>HTML5 Test</b></p>   | <p>Se trata de una funcionalidad en línea, la cual detalla las características de un navegador web. Se utilizó para averiguar qué códecs de video son compatibles con los diferentes navegadores usados.</p> |
| <p><b>Programa de codificación de video WinX HD Video Converter Deluxe (versión 5.12.1).</b></p>                   | <p>Software usado para la conversión y codificación de videos a los formatos soportados por un navegador con HTML5.</p>  |

## **5.2 MÉTODOS:**

La metodología utilizada para desarrollar el proyecto es la siguiente:

### **5.2.1 Método experimental.**

La experimentación tiene como objetivo el estudio de un determinado suceso o factor y sus cualidades intrínsecas. Este método se utilizó en la selección del códec de video para el sistema, y para su comparación con las demás alternativas disponibles, de esta manera se pudo determinar las ventajas y desventajas de los mismos, y cuál de ellos produce mejores resultados. Los resultados de este método se muestran más adelante en el inciso 6 de este documento.

### **5.2.2 Método de programación extrema.**

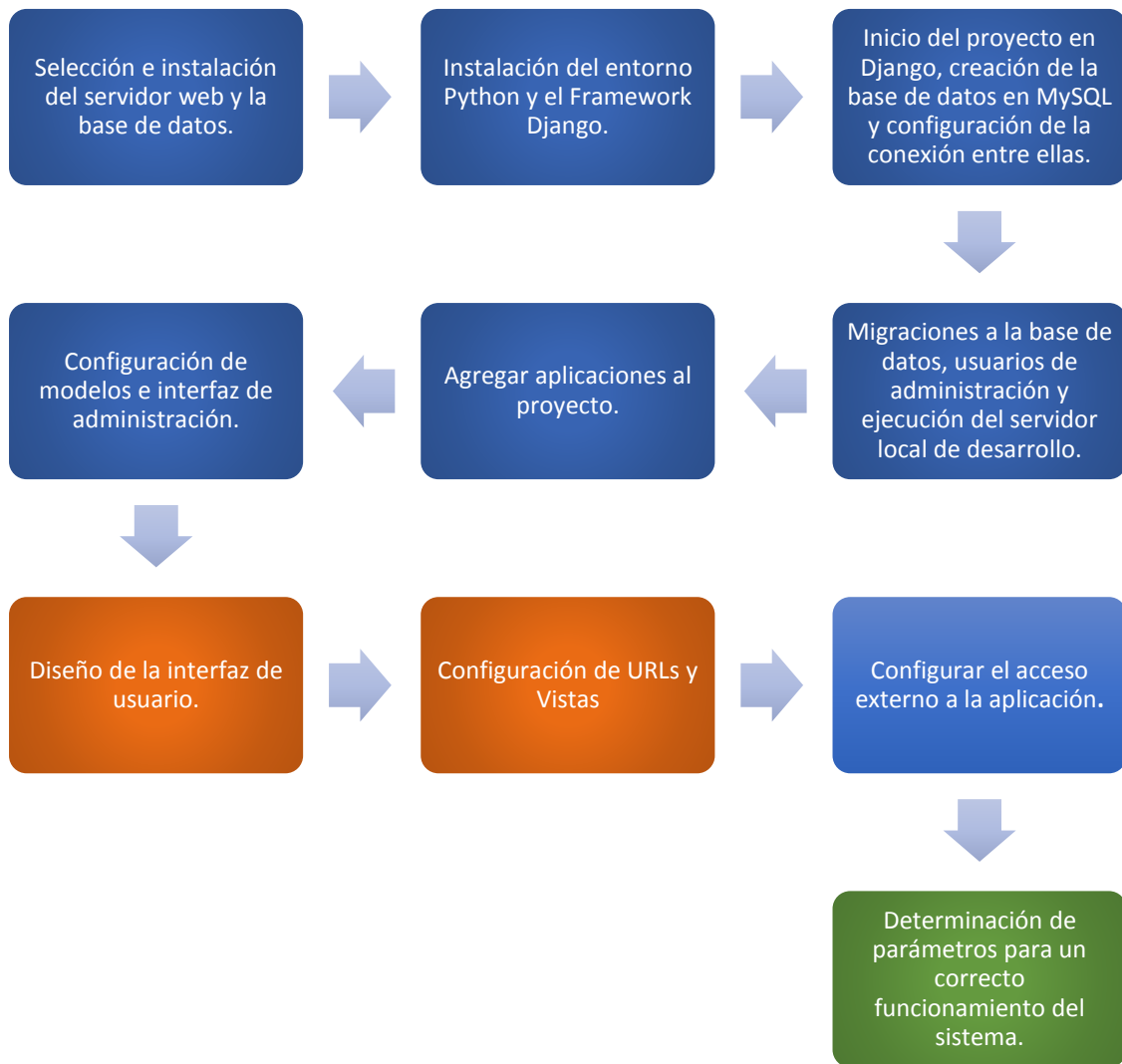
Para el desarrollo del software del Infochannel se utilizó una metodología ágil denominada XP (eXtreme Programming o programación extrema). Este método se pudo ejecutar gracias al uso de un framework, y particularmente a su servidor local de desarrollo, el cual agilizó y facilitó en gran medida, el desarrollo del sistema y el poder realizar pruebas rápidas de funcionamiento.

### **5.2.3 Método deductivo.**

Con este método se logró analizar la información recopilada para poder aclarar los conceptos y poder encausar este trabajo por el mejor camino. De esta manera se pudo seleccionar las tecnologías sobre las cuales implementar el sistema y los diferentes tipos de códecs de video compatibles, todo ello en base a la investigación previamente realizada.

## **5.3 PROCESO DE DISEÑO DEL SISTEMA:**

El proceso de diseño del sistema de “Infochannel” como aplicación web y para ser implementado en la red LAN del SIS ECU911 de la ciudad de Loja tuvo distintas etapas, desde la investigación de las tecnologías a utilizar, hasta la implementación de una red de pruebas que permita determinar las configuraciones óptimas para su correcto funcionamiento. El ambiente de desarrollo, así como los programas usados durante su diseño, se ejecutaron sobre el sistema operativo Windows 10, pero esto no hace que el producto final funcione únicamente sobre esta plataforma, sino que, al haber elegido herramientas de *software* libre y multiplataforma, se aseguró que este proyecto pueda ser migrado a múltiples sistemas. A continuación, se detalla el proceso de diseño:



*Figura 24.- Proceso de diseño del sistema.  
Fuente: Elaboración propia.*

### 5.3.1 Selección e instalación del servidor web y la base de datos.

De entre las tecnologías de servidores web explicadas en este documento, se seleccionó el *web server* Apache, debido principalmente a que es de software libre, a su característica de ser multiplataforma, a su fácil adaptabilidad con lenguajes de programación como Python, PHP o Ruby, y a sus otras características detalladas en la Tabla 5 de la literatura de este documento. Esta herramienta nos servirá principalmente para atender las peticiones desde los diferentes dispositivos conectados a nuestro sistema, ya sea los *Smart TV*, computadoras o inclusive *Smartphones*. También es importante por su rendimiento sobre sistemas web que no demanden una gran cantidad de solicitudes por parte de los clientes hacia el servidor.

Por otra parte, como alternativa SGBD se seleccionó MySQL principalmente por su adaptación al *web server* Apache, por ser igualmente de licencia libre y por su interfaz fácil de manejar al momento de crear y trabajar con bases de datos. Otro factor importante, es su velocidad de respuesta al trabajar con aplicaciones web.

Teniendo en cuenta las características de ambas herramientas, se recurrió al paquete de *software* XAMPP, el cual incluye en un único archivo ejecutable (.exe), los *softwares* Apache y MySQL, junto a otras herramientas útiles. Los procesos de instalación del programa pueden variar en cierta medida, de acuerdo al sistema operativo en el cual se vaya a instalar. El *software* XAMPP se lo puede obtener gratuitamente de su página oficial “[www.apachefriends.org/es/index.html](http://www.apachefriends.org/es/index.html)”.

Tras haber descargado XAMPP desde su página oficial, se ejecuta el archivo y se sigue los pasos que indica su asistente de instalación:

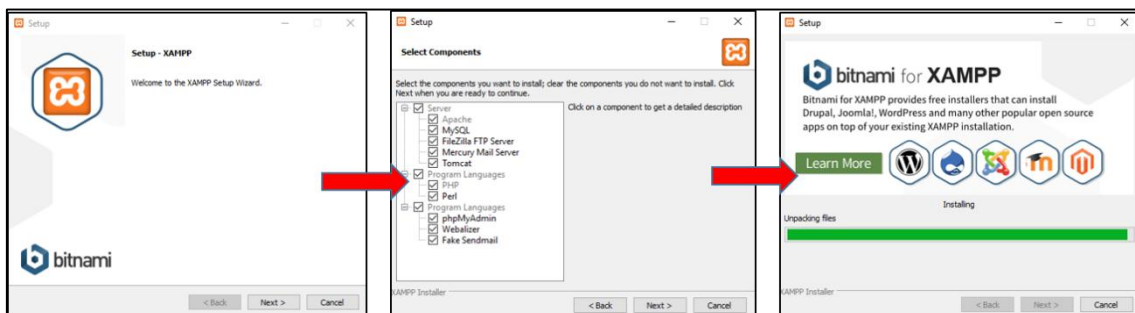


Figura 25.- Instalación de XAMPP 3.2.3.  
Fuente: Elaboración propia.

En la ventana central de la Figura 25 se debe seleccionar los elementos del paquete XAMPP que se vayan a usar. Para este proyecto se requiere únicamente de las herramientas Apache, MySQL, PHP y phpMyAdmin.

### 5.3.2 Instalación del entorno Python y el Framework Django.

Python en primera instancia fue elegido para ese proyecto por su facilidad de comprensión y su sintaxis sencilla, ya que se trata de un lenguaje de programación de muy alto nivel. También cabe recalcar su versatilidad y su gran comunidad de desarrolladores, los cuales proveen de una excelente bibliografía que sirve de apoyo para solventar posibles problemas durante el desarrollo de cualquier aplicación.

Para su instalación, obtenemos gratuitamente el archivo ejecutable desde su sitio oficial “[www.python.org/downloads/](http://www.python.org/downloads/)”.

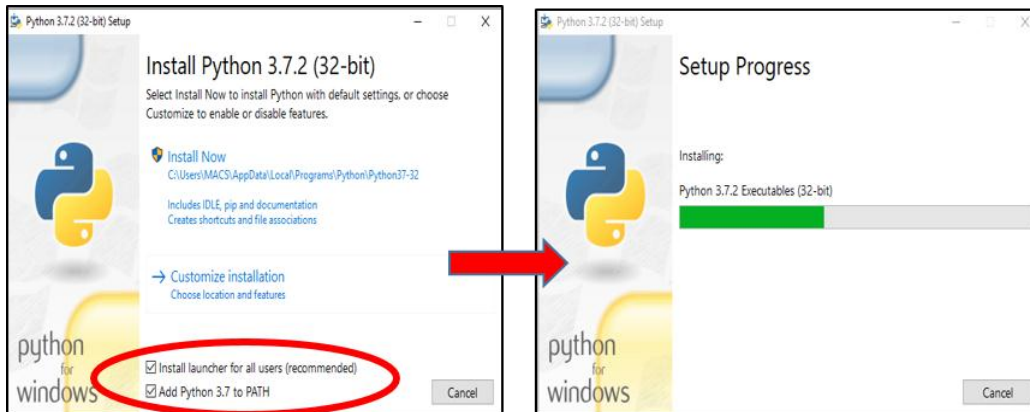


Figura 26.- Instalación del entorno Python 3.7.2.  
Fuente: Elaboración propia.

Durante el proceso de instalación es importante marcar la casilla “Add Python 3.7 to PATH” (ver Figura 26), para asegurarse que posteriormente puedan ejecutarse comandos Python desde el “cmd” o “Command Prompt” (es el CLI o Interfaz de Línea de Comandos de Windows).

Ahora, una vez instalado el entorno Python, se procede a la instalación del *framework* Django, el cual se instala únicamente desde la línea de comandos de Windows (ver Figura 27). Para ello, se abre el “cmd” y se ejecuta el siguiente comando:

```
pip install Django
```

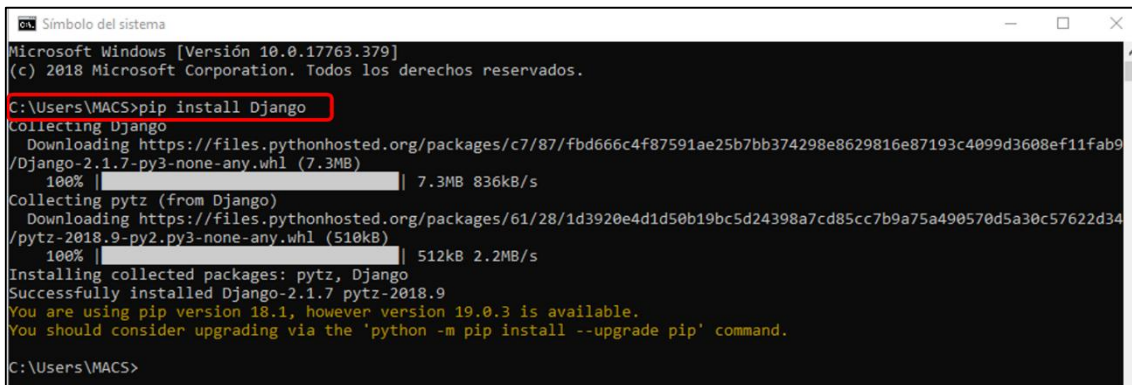


Figura 27.- Instalación del Framework Django 2.1.7.  
Fuente: Elaboración propia.

### 5.3.3 Inicio del proyecto en Django, creación de la base de datos en MySQL y configuración de la conexión entre ellas.

Una vez seleccionadas las herramientas a utilizar, se prosigue con la creación del proyecto sobre el *framework* Django. Para ello, al igual que su instalación, se requiere del “cmd” de Windows en donde se seguirá los siguientes pasos:

Primero se necesita ingresar al directorio donde se instaló el *software* XAMPP, concretamente a la carpeta “htdocs” que es dónde se alojan las aplicaciones que se desee subir al servidor:

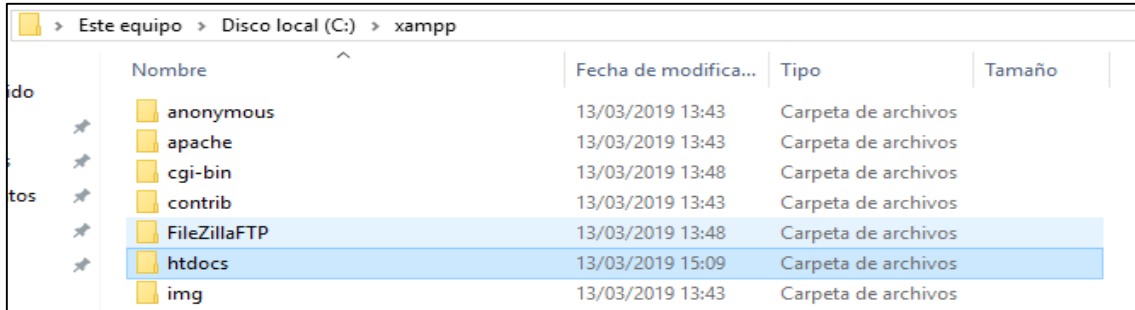


Figura 28.- Directorio "htdocs" del servidor XAMPP.  
Fuente: Elaboración propia.

Para acceder a dicho directorio desde el “cmd” utilizamos el comando de navegación “cd”, a continuación, en la Figura 29 se muestra un ejemplo:

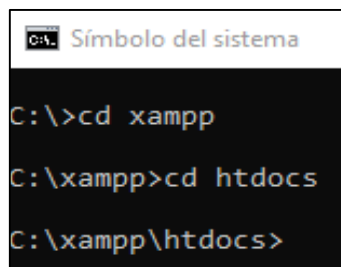


Figura 29.- Acceso al directorio "htdocs" desde el cmd de Windows.  
Fuente: Elaboración propia.

Estando ya en el directorio “htdocs”, se ingresa los siguientes comandos para la creación del proyecto en Django:

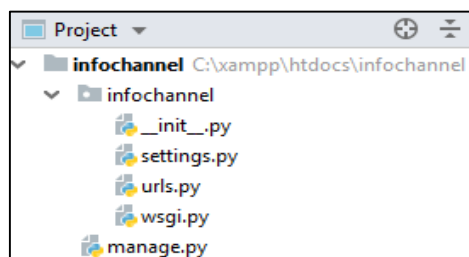
```
django-admin.py startproject infochannel
```



Figura 30.- Creación del proyecto en Django.  
Fuente: Elaboración propia.

Como se puede observar en la Figura 30, al ejecutar el comando desde el cmd se crea inmediatamente un directorio nuevo con el nombre especificado y los archivos Python necesarios para el desarrollo del proyecto. Estos archivos son:

- **infochannel/:** es el directorio de trabajo externo denominado “infochannel/”. Es solo un contenedor, es decir una carpeta que contiene nuestro proyecto, por lo que se le puede cambiar el nombre en cualquier momento sin afectar el proyecto en sí.
- **manage.py:** es una utilidad de línea de comandos que permite interactuar con un proyecto Django de varias formas. Con el comando “manage.py help” desde el CLI se puede visualizar las opciones que nos da esta utilidad. No se recomienda editar este archivo, ya que este es creado en el directorio convenientemente para manejar el proyecto.
- **infochannel/infochannel/:** es el directorio interno de “infochannel/” y este contiene el paquete Python para tu proyecto.
- **\_\_init\_\_.py:** Un archivo requerido para que Python trate a este directorio como un paquete, es decir, para que lo reconozca como parte del proyecto en desarrollo.
- **settings.py:** permite detallar las opciones o configuraciones para el proyecto de Django.
- **urls.py:** es donde se declara las URLs para el proyecto de Django, otras palabras, es una “tabla de contenidos” del sitio hecho con Django.
- **wsgi.py:** es el punto de entrada WSGI (Web Server Gate Interface) para el servidor Web, encargado de servir nuestro proyecto.



*Figura 31.- Archivos de Python base para el proyecto con Django.  
Fuente: Elaboración propia.*

En cuanto se haya creado correctamente el proyecto, el siguiente paso es crear la base de datos con la cual se trabajará. En este caso se hará desde el SGBD MySQL, al cual podemos acceder desde el panel de control XAMPP.

Lo primero que se debe hacer es iniciar el servidor Apache y el SGBD MySQL con la opción “Start” del panel de control. Tras ello, desde la opción “Admin” de MySQL se puede acceder a su interfaz web de administración.

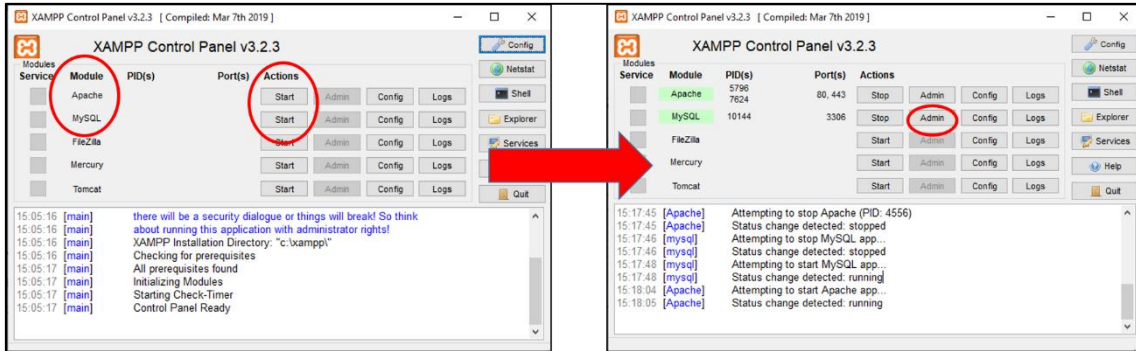


Figura 32.- Acceso a MySQL para la creación de la base de datos.  
Fuente: Elaboración propia.

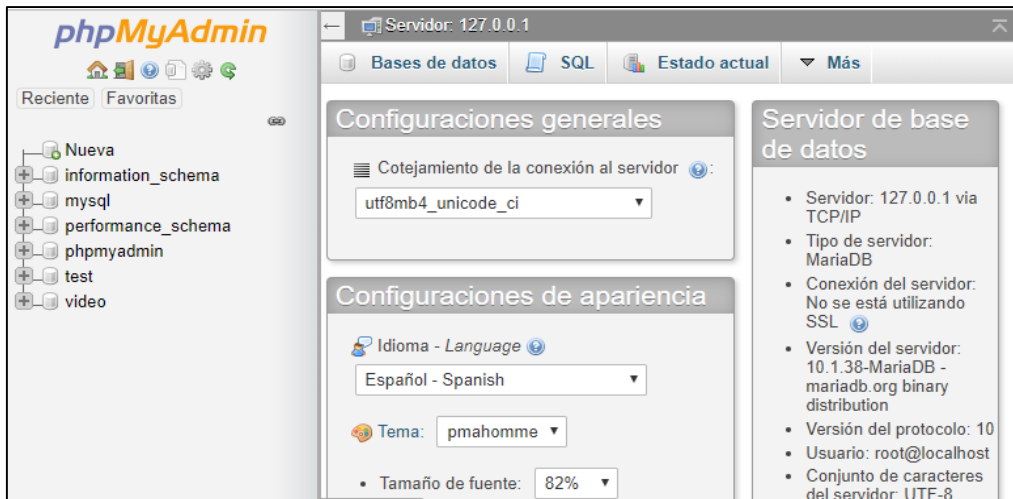
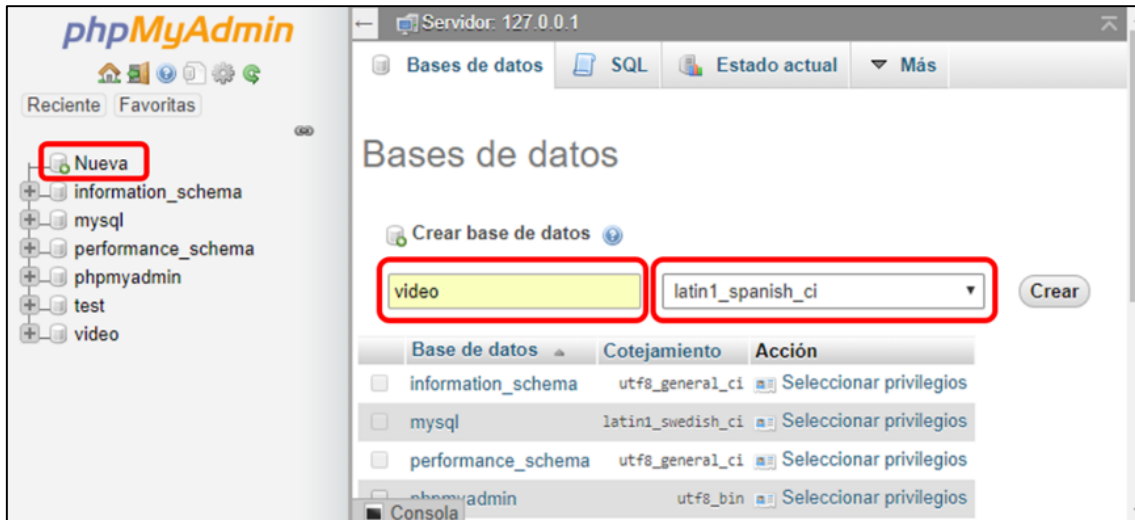


Figura 33.- Interfaz web de administración de MySQL.  
Fuente: Elaboración propia.



Desde la interfaz web de MySQL, en el panel de la izquierda se hace clic en la opción “Nueva” para crear la base de datos a la cual se le asigna un nombre y el idioma deseado (ver Figura 34). Con esto, el SGBD crea de forma rápida la base de datos que se usará a lo largo de todo el desarrollo.



*Figura 34.- Creación de la base de datos en phpMyAdmin.  
Fuente: Elaboración propia.*

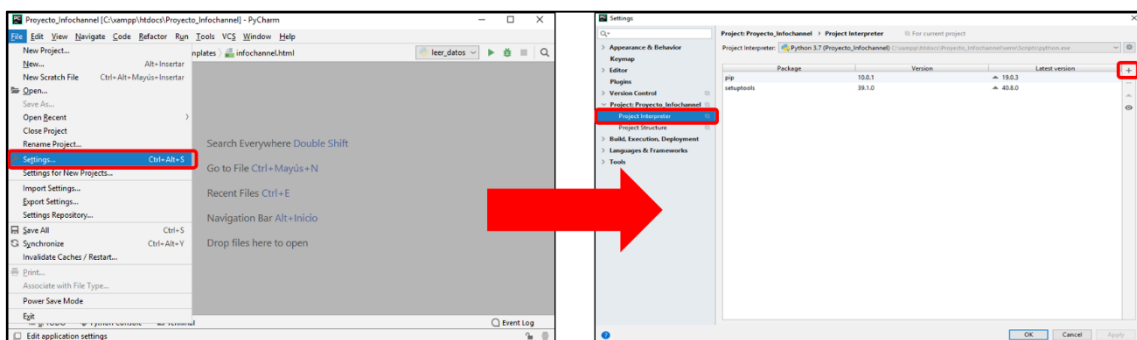
Una vez creada la base de datos que manejará el proyecto, el siguiente paso es configurar Django de manera que pueda conectarse a dicha base de datos.

Para la conexión entre la aplicación Django con la base de datos en MySQL se requiere de una librería especial de Python, la cual se denomina “pymysql”. Hay que destacar que de acuerdo al SGBD con el que trabaje, se requerirá de una librería diferente.

Para la instalación de la librería “pymysql” o de cualquier otra librería, existen dos métodos, desde el IDE Pycharm o desde el cmd de Windows. Su proceso de instalación es el siguiente:

- **Desde Pycharm:**

Desde la pestaña “File” se accede a la opción “Settings” y se mostrará la ventana de configuraciones. En esta ventana en el panel izquierdo se escoge la opción “Project interpreter” para acceder al intérprete del proyecto y visualizar las librerías Python que se están usando actualmente, y de ser necesario, instalar nuevas librerías desde el gestor de librerías de Pycharm (ver Figura 35).



*Figura 35.- Acceso al gestor de librerías de Pycharm.  
Fuente: Elaboración propia.*

A continuación, buscamos la librería “pymysql” e instalamos su paquete. En la ventana de la derecha de la Figura 36 podemos ver cómo la librería ha sido agregada al proyecto correctamente.

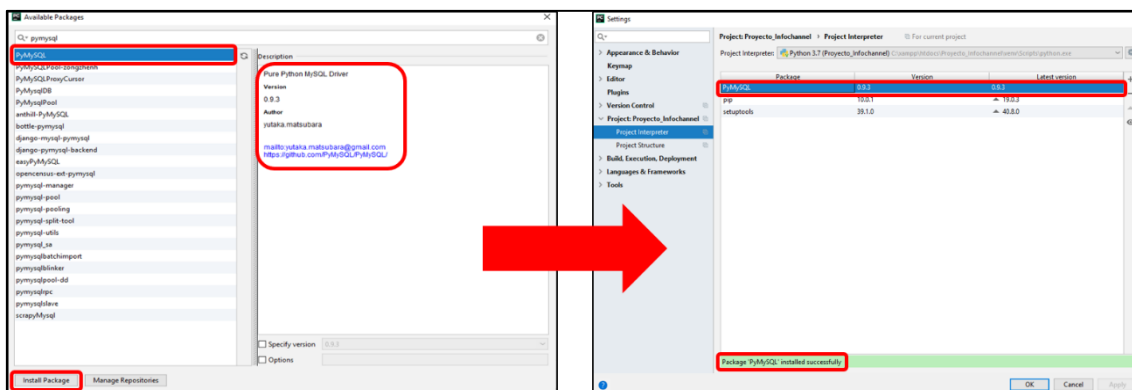


Figura 36.- Integración de la librería "pymysql" desde el gestor de librerías de Pycharm.  
Fuente: Elaboración propia.

- Desde el cmd:

Desde la interfaz de línea de comandos el proceso es más rápido, siempre y cuando conozcamos la sintaxis correcta para instalar la librería, por ello es que usar el gestor de Pycharm suele ser la opción más sencilla para usuarios nuevos. Para la instalación de la librería "pymysql" desde el CLI de Windows se utiliza el siguiente comando:

```
pip install pymysql
```

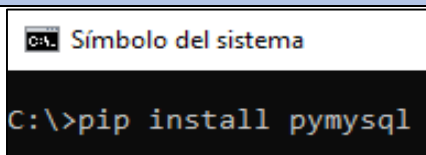


Figura 37.- Instalación de la librería "pymysql" desde el CLI de Windows.  
Fuente: Elaboración propia.

Para finalizar, se debe agregar al archivo "\_\_init\_\_.py" (ver Figura 31) la siguiente línea de código:

```
import pymysql
pymysql.install_as_MySQLdb()
```

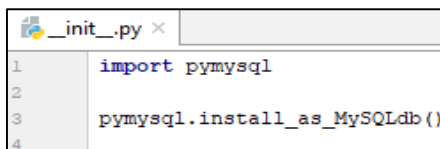
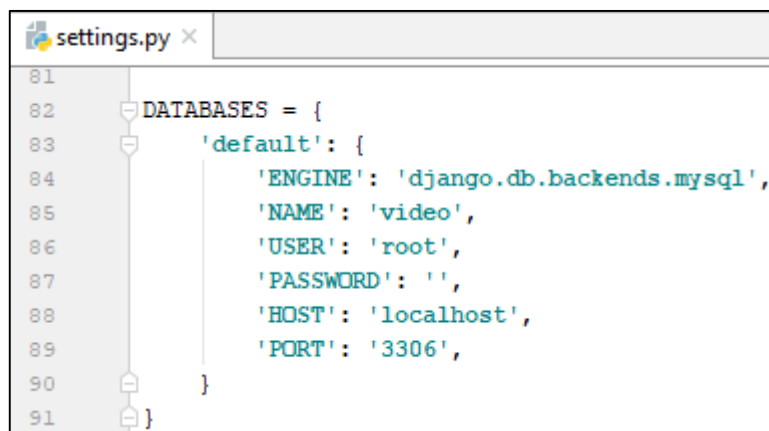


Figura 38.- Adhesión de la librería "pymysql" al proyecto.  
Fuente: Elaboración propia.

El proceso de mostrado en la Figura 38, se debe realizar ya sea si se ha instalado la librería “pymysql” desde el gestor de librerías de Pycharm o desde el CLI de Windows.

Con la librería ya instalada correctamente, se accede desde Pycharm al archivo “Settings.py”, el cual se creó con anterioridad al momento de iniciar el proyecto Django desde la línea de comandos (ver Figura 30). Dentro del archivo se debe buscar la sección “DATABASES”, en la cual se configuran las bases de dato necesarias para el proyecto. De acuerdo a las configuraciones previamente establecidas en el diseño del proyecto, la configuración de la conexión con la base de datos queda de la siguiente manera:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'video',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

A screenshot of a code editor window titled 'settings.py'. The code is displayed with line numbers from 81 to 91. The configuration for the database is as follows:

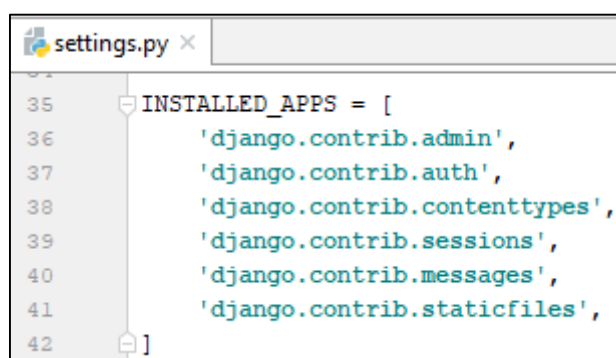
```
81
82 DATABASES = {
83     'default': {
84         'ENGINE': 'django.db.backends.mysql',
85         'NAME': 'video',
86         'USER': 'root',
87         'PASSWORD': '',
88         'HOST': 'localhost',
89         'PORT': '3306',
90     }
91 }
```

Figura 39.- Configuración de la conexión entre Django con la base de datos.  
Fuente: Elaboración propia.

- **ENGINE:** en esta opción se especifica el SGBD usado para el proyecto.
- **NAME:** esto indica el nombre de la base de datos a la que se desea acceder.
- **USER/PASSWORD:** son el usuario y contraseña de acceso a la base de datos.
- **HOST:** es la dirección IP del equipo donde se encuentra ubicada la base de datos.
- **PORT:** es el puerto lógico para acceder a la base de datos.

### 5.3.4 Migraciones a la base de datos, usuarios de administración y ejecución del servidor local de desarrollo.

Con la comunicación entre Django y la base de datos configurada correctamente, se procede a crear las primeras tablas dentro de la base de datos. Django tiene un conjunto de tablas por defecto, las cuales alojan la información necesaria para sus aplicaciones base, y se pueden observar dentro del archivo “Settings.py” en la sección de “INSTALLED\_APPS” (ver Figura 40).

A screenshot of a code editor window titled 'settings.py'. The code shows a list of Django core applications under the variable 'INSTALLED\_APPS'. The list includes: 'django.contrib.admin', 'django.contrib.auth', 'django.contrib.contenttypes', 'django.contrib.sessions', 'django.contrib.messages', and 'django.contrib.staticfiles'. The line numbers 35 through 42 are visible on the left side of the editor.

```
35 INSTALLED_APPS = [  
36     'django.contrib.admin',  
37     'django.contrib.auth',  
38     'django.contrib.contenttypes',  
39     'django.contrib.sessions',  
40     'django.contrib.messages',  
41     'django.contrib.staticfiles',  
42 ]
```

Figura 40.- Aplicaciones base de Django.  
Fuente: Elaboración propia.

Estas aplicaciones base se incluyen por defecto al crear cualquier proyecto, como conveniencia para los casos más comunes:

- **django.contrib.admin:** Es la interfaz administrativa.
- **django.contrib.auth:** Es el sistema de autenticación.
- **django.contrib.contenttypes:** Sirve para tratar los tipos de contenidos.
- **django.contrib.sessions:** Se encarga de manejar sesiones.
- **django.contrib.messages:** Se ocupa del manejo de mensajes.
- **django.contrib.staticfiles:** Sirve para el control de archivos estáticos.

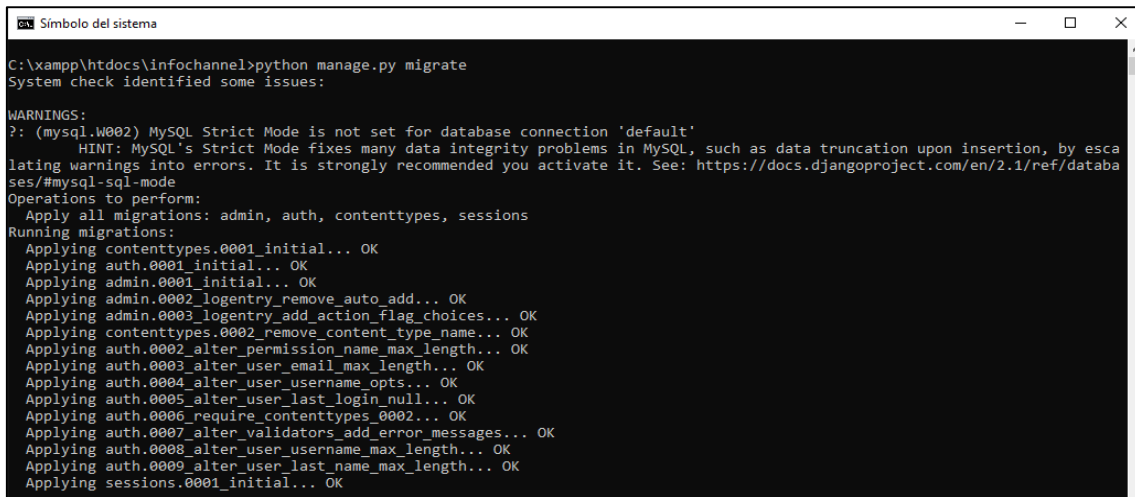
En la sección “INSTALLED\_APPS” es obligatorio especificar todas y cada una de las aplicaciones que se vaya creando o adjuntando al proyecto.

Las tablas que administran la información de estas aplicaciones se cargan a la base de datos ejecutando el siguiente comando:

```
python manage.py migrate
```

El comando “migrate” busca la variable “INSTALLED\_APPS” y crea las tablas necesarias de cada una de las aplicaciones registradas en el archivo “settings.py”, que

contiene todas las aplicaciones. Al ejecutar este comando, en el cmd se puede observar un mensaje por cada migración aplicada (ver Figura 41). Todo comando que contenga la sintaxis “manage.py”, puede ser ejecutado desde el CLI de Windows siempre y cuando se esté ubicado en el directorio donde se encuentre el archivo “manage.py” (ver Figura 30).



```
Símbolo del sistema
C:\xampp\htdocs\infochannel>python manage.py migrate
System check identified some issues:

WARNINGS:
?: (mysql.W002) MySQL Strict Mode is not set for database connection 'default'
   HINT: MySQL's Strict Mode fixes many data integrity problems in MySQL, such as data truncation upon insertion, by escalating warnings into errors. It is strongly recommended you activate it. See: https://docs.djangoproject.com/en/2.1/ref/databases/#mysql-sql-mode
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
```

*Figura 41.- Primeras migraciones del proyecto.  
Fuente: Elaboración propia.*

Por cada nueva aplicación que se cree o que se adjunte al proyecto, es necesario realizar las migraciones a la base de datos. Para ello, como se ha indicado previamente, se debe usar el comando “python manage.py migrate”.

Revisando nuevamente el contenido de la base de datos se puede observar que las dependencias (tablas) de las aplicaciones por defecto ya han sido migradas a la base de datos (ver Figura 42).

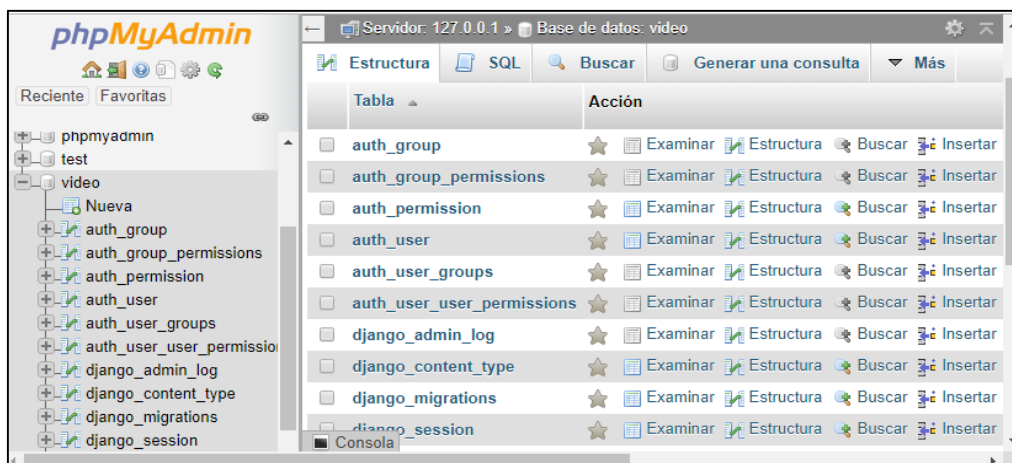


Figura 42.- Vista de las migraciones realizadas desde phpMyAdmin.  
Fuente: Elaboración propia.

Otro tema importante es la administración de la aplicación. Para ello, Django incorpora una interfaz de administración propia y personalizable, la cual permite agregar, editar y eliminar contenido del sitio web. Para acceder a esta interfaz es necesario que como mínimo exista un usuario administrador del sistema, y para crearlo, desde el cmd se ejecuta el comando:

```
python manage.py createsuperuser
```

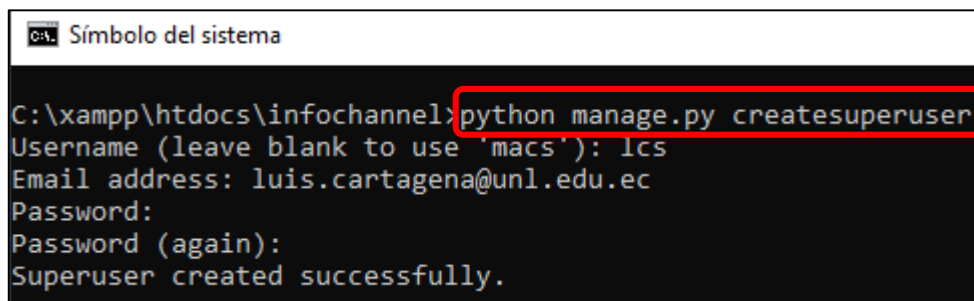
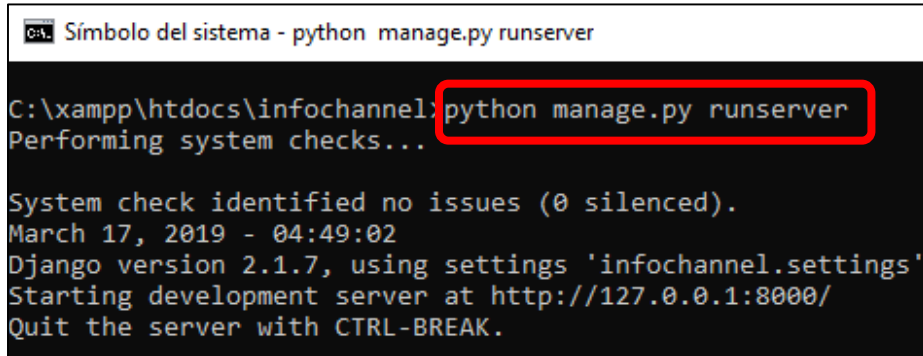


Figura 43.- Creación de un usuario administrador del sistema.  
Fuente: Elaboración propia.

Como se observa en la Figura 43, al ejecutar el comando se despliega una serie de preguntas para su configuración (Username o nombre de usuario, e-mail address o dirección de correo electrónico y password o contraseña). Este al ser un comando con la sintaxis de “manage.py”, debe ser ejecutado desde el cmd en el mismo directorio donde se encuentre el archivo “manage.py” (ver Figura 31).

A continuación, para observar el desarrollo del proyecto, se procede a levantar el servidor local de desarrollo que incorpora Django. Para ello se utiliza el siguiente comando desde el cmd:

```
python manage.py runserver
```



```
C:\xampp\htdocs\infochannel>python manage.py runserver
Performing system checks...

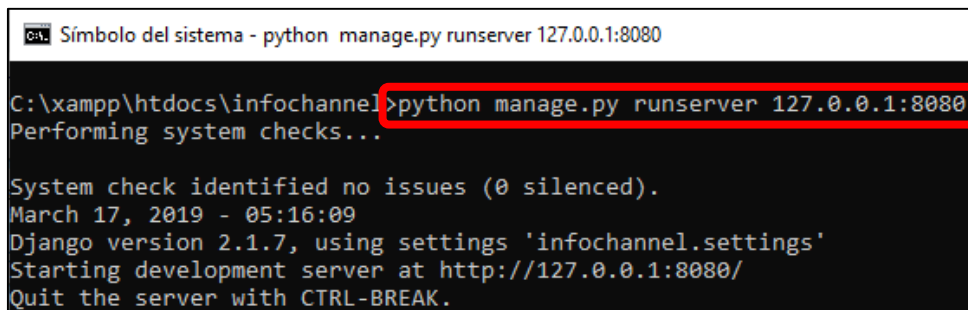
System check identified no issues (0 silenced).
March 17, 2019 - 04:49:02
Django version 2.1.7, using settings 'infochannel.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

*Figura 44.- Activación del servidor de desarrollo de Django.  
Fuente: Elaboración propia.*

El comando “python manage.py runserver” debe ser ejecutado desde el cmd en el mismo directorio donde se encuentre el archivo “manage.py” (ver Figura 31).

En la Figura 44, el propio Django nos da la información de cómo podemos ver el funcionamiento actual del sistema. Para ello necesitamos un navegador en el cual ingresar la dirección IP del servidor de desarrollo. En este caso la dirección es “127.0.0.1:8000”. Como se puede apreciar se trata de una dirección loopback, que está usando el puerto 8000. Esta es la configuración que viene por defecto al activar el servidor de desarrollo, pero de ser necesario, se puede configurar una dirección diferente de la siguiente manera:

```
python manage.py runserver 127.0.0.1:8080
```



```
C:\xampp\htdocs\infochannel>python manage.py runserver 127.0.0.1:8080
Performing system checks...

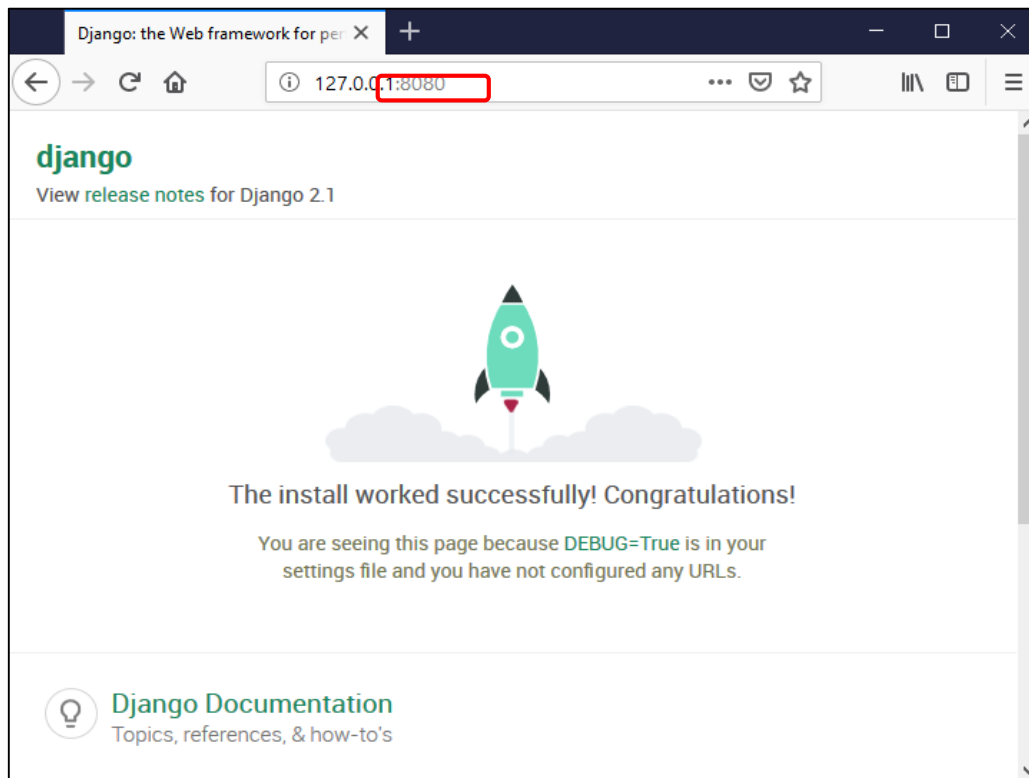
System check identified no issues (0 silenced).
March 17, 2019 - 05:16:09
Django version 2.1.7, using settings 'infochannel.settings'
Starting development server at http://127.0.0.1:8080/
Quit the server with CTRL-BREAK.
```

*Figura 45.- Activación del servidor de desarrollo con parámetros cambiados.  
Fuente: Elaboración propia.*



Como se observa en la Figura 45, se ha modificado el puerto de acceso al “8080”, aunque también es posible modificar la IP. Para ello se debe configurar la nueva dirección de red sobre la que vaya a funcionar el servidor.

Ahora para ver el progreso de la aplicación, usando la IP del servidor de desarrollo, desde un navegador web se puede visualizar el estado actual del proyecto.



*Figura 46.- Interfaz inicial de la aplicación web en Django.  
Fuente: Elaboración propia.*

Como se puede observar en la Figura 46, la dirección IP configurada para el servidor es la “127.0.0.1:8080”, la cual resulta ser la misma que fue configurada con anterioridad (ver Figura 45).

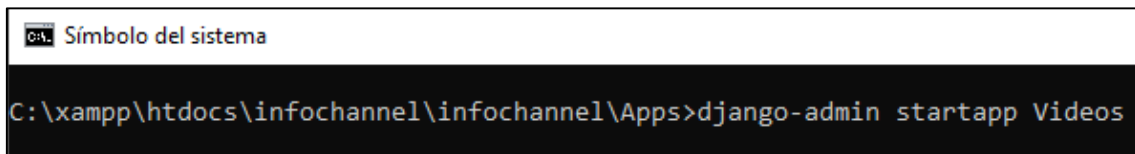
### **5.3.5 Agregar aplicaciones al proyecto.**

Dentro de un proyecto desarrollado con Django, se puede alojar diversas aplicaciones orientadas cada una a los diferentes servicios que se desee ofrecer dentro del servidor. De esta forma cada una de las aplicaciones puede ser independientes y, al ser portables, podrían reutilizarse para futuros proyectos.

Para este proyecto, se ha optado por crear la aplicación de nombre “Videos”, la cual se encargará de cargar videos al servidor, editarlos, eliminarlos y visualizarlos desde la

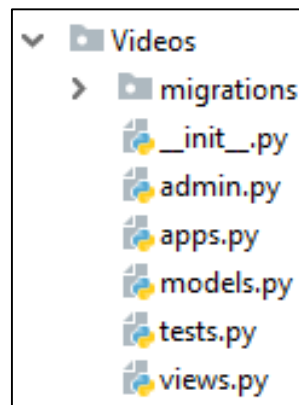
interfaz de administración. Para crear una aplicación dentro del proyecto se recurre al siguiente comando:

```
django-admin startapp Videos
```



*Figura 47.- Creación de una nueva aplicación dentro de un proyecto Django.  
Fuente: Elaboración propia.*

Con este comando se genera la aplicación con el nombre especificado, en forma de un paquete Python. De igual forma, se crea la estructura base de la aplicación conformada por un conjunto de archivos. Estos son:

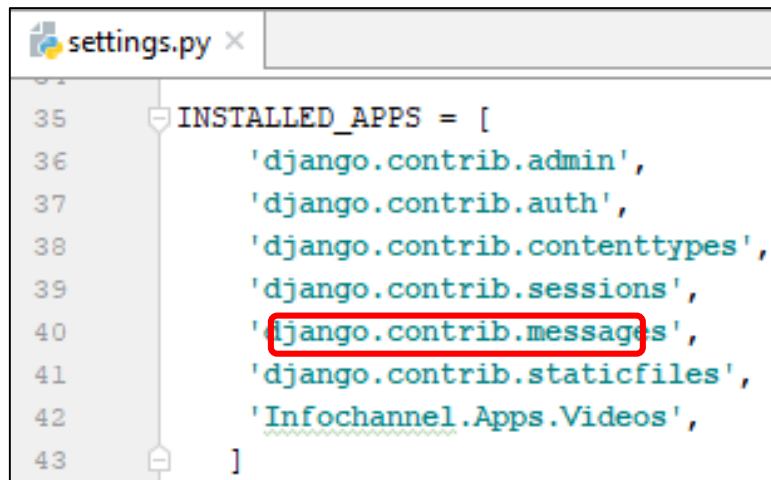


*Figura 48.- Archivos base de una aplicación en Django.  
Fuente: Elaboración propia.*

- **Videos/:** es el directorio general de la aplicación en Django.
- **Videos/migrations/:** dentro de este directorio es donde se alojan los archivos de configuración específicos de la aplicación.
- **\_\_init\_\_.py:** es un archivo requerido para que Python trate a este directorio como un paquete, es decir, para que lo reconozca como parte de la aplicación.
- **admin.py:** sirve para determinar qué modelos podrán ser visualizados desde la interfaz web administrativa.
- **apps.py:** este archivo se crea para ayudar al usuario a incluir cualquier configuración de aplicación para la aplicación. Usando esto, se puede configurar algunos de los atributos de la aplicación.

- **models.py:** en este archivo se configura qué tipo de información se va a almacenar en la base de datos, es decir, en la aplicación Django. Aquí se crean formularios o tablas, en las cuales se especifica qué tipo de datos es y su relación de dependencia con otros modelos o tablas.
- **test.py:** dentro de este archivo se configuran pruebas de rendimiento automatizadas. Se utilizan para detectar bugs o errores dentro de la aplicación.
- **views.py:** en este archivo se programa el manejo de datos, es decir, permite escribir código las consultas a la base de datos, que sean necesarias la vista. La vista también se encarga de tareas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios.

Como se ha mencionado anteriormente, al crear o adjuntar una nueva aplicación dentro del proyecto, esta debe ser declarada en las configuraciones generales del proyecto (“Settings.py”).



```

35 INSTALLED_APPS = [
36     'django.contrib.admin',
37     'django.contrib.auth',
38     'django.contrib.contenttypes',
39     'django.contrib.sessions',
40     'django.contrib.messages',
41     'django.contrib.staticfiles',
42     'Infochannel.Apps.Videos',
43 ]

```

*Figura 49.- Configuración del archivo "Settings.py" para agregar nueva aplicación.  
Fuente: Elaboración propia.*

### 5.3.6 Configuración de modelos e interfaz de administración.

Los modelos son uno de los 3 componentes básicos de Django según su modelo MTV. Estos determinan qué información es almacenada en la base de datos, lo que significa que aquí es donde se define qué datos serán necesarios para estructurar el sistema. Estos se configuran en el archivo “models.py” (ver Figura 48).

Los modelos nos permiten elaborar tablas personalizadas con parámetros específicos de acuerdo a nuestras necesidades. Por cada modelo creado, se debe configurar como

mínimo un parámetro, especificando el tipo de dato (*char*, *string*, etc). Por medio de estos, se diseñó el formulario de subida de archivos hacia la base de datos (ver Figura 55 y Figura 56).

De acuerdo a la aplicación planteada en esta tesis y a los requerimientos por parte del SIS ECU 911 de la ciudad de Loja, se establecieron dos clases de modelos: el primero dedicado exclusivamente al denominado “Infochannel” (modelo “Video”) y el segundo serviría como base para un repositorio multimedia (modelo “Repositorio”). En la Figura 50, se observan los modelos declarados para este proyecto.



```
models.py x
19 class Video(models.Model):
20     titulo = models.CharField(max_length=30)
21     video_src = models.FileField(upload_to='', validators=[validate_video_extension], blank=False, null=False)
22
23     ingresado = models.DateTimeField(auto_now_add=True, auto_now=False)
24     modificado = models.DateTimeField(auto_now_add=False, auto_now=True)
25
26     def __str__(self):
27         return self.titulo
28     class Meta:
29         ordering = ["ingresado"]
30
31 class Repositorio(models.Model):
32     titulo = models.CharField(max_length=30)
33     video_src = models.FileField(upload_to='repositorio/videos/', validators=[validate_video_extension], blank=False, null=False)
34     imagen_src = models.FileField(upload_to='repositorio/imagenes/', validators=[validate_file_extension], blank=False, null=False)
35     # ESTADO = (('T', 'Activo [T]'), ('F', 'Inactivo [F]'))
36     # estado = models.CharField(max_length=1, choices=ESTADO, default='Activo')
37     ingresado = models.DateTimeField(auto_now_add=True, auto_now=False)
38     modificado = models.DateTimeField(auto_now_add=False, auto_now=True)
39
40     def __str__(self):
41         return self.titulo
42     class Meta:
43         ordering = ["ingresado"]
```

Figura 50.- Declaración de modelos.  
Fuente: Elaboración propia.

Con los modelos ya configurados, lo siguiente es validarlos, exportarlos y determinar cómo serán mostrados en la interfaz de administración.

Para validar los modelos creados se utiliza el siguiente comando:

```
python manage.py makemigrations
```

```
C:\xampp\htdocs\infochannel>python manage.py makemigrations
Migrations for 'Videos':
  infochannel\Apps\Videos\migrations\0001_initial.py
  - Create model Repositorio
  - Create model Video
```

Figura 51.- Validación de modelos.  
Fuente: Elaboración propia.

Como se observa en la Figura 51, los modelos creados han sido validados y están listos para ser migrados a la base de datos. Este proceso debe realizarse siempre y cuando se creen nuevos modelos en el proyecto.

Para migrar los modelos a la base de datos se usa el siguiente comando:

```
python manage.py migrate
```

```
C:\xampp\htdocs\infochannel>python manage.py migrate
System check identified some issues:

WARNINGS:
?: (mysql.W002) MySQL Strict Mode is not set for database connection 'default'
   HINT: MySQL's Strict Mode fixes many data integrity problems in MySQL, such as data truncation upon insertion, b
y escalating warnings into errors. It is strongly recommended you activate it. See: https://docs.djangoproject.com/en/2.
1/ref/databases/#mysql-sql-mode
Operations to perform:
  Apply all migrations: Videos, admin, auth, contenttypes, sessions
Running migrations:
  Applying Videos.0001_initial... OK
```

Figura 52.- Migración de modelos a la base de datos.  
Fuente: Elaboración propia.

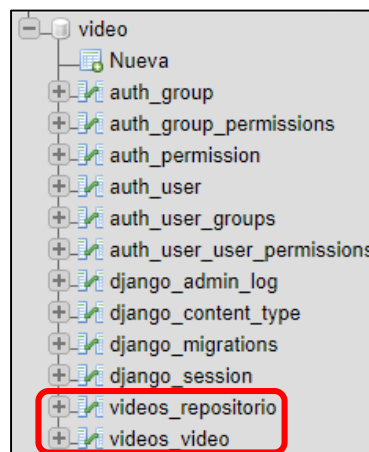


Figura 53.- Vista de modelos agregados desde el administrador de la base de datos.  
Fuente: Elaboración propia.

En la Figura 53 se puede observar que los modelos ya han sido migrados a la base de datos, y esto puede verificarse desde el administrador web de la base de datos (phpMyAdmin).

Con los modelos ya validados y migrados a la base de datos, ahora se procede a configurar cómo se van a visualizar los mismos en el administrador web de Django. Para ello accedemos al archivo Python “admin.py” (ver Figura 48).

Como se ha especificado en la literatura de esta tesis, la interfaz de administración es uno de los componentes más importantes de Django, a la hora de plantear el diseño de un sitio web, ya que mediante esta es posible manipular de forma fácil la información que se usará en el sitio web. Esta interfaz es completamente configurable de acuerdo a las necesidades del proyecto.

En la Figura 54 se puede visualizar la configuración de los modelos para la interfaz de administración de la aplicación, mientras que en las Figuras 55 y 56 e observa el resultado desde la interfaz web de administración:

```
admin.py x
1  from django.contrib import admin
2  from Infochannel.Apps.Videos.models import Video, Repositorio
3
4  # Register your models here.
5
6  class VideoRegistrado (admin.ModelAdmin):
7      list_display = ["titulo", "video_src", "ingresado", "modificado"]
8      list_filter = ["ingresado", "modificado"]
9      search_fields = ["titulo"]
10     list_editable = ["video_src"]
11     class Meta:
12         model = Video
13         ordering = ["ingresado"]
14
15     class RepositorioVideos (admin.ModelAdmin):
16         list_display = ["titulo", "video_src", "ingresado", "modificado"]
17         list_filter = ["ingresado", "modificado"]
18         search_fields = ["titulo"]
19         list_editable = ["video_src"]
20
21         class Meta:
22             model = Repositorio
23             ordering = ["ingresado"]
24
25     admin.site.register(Video, VideoRegistrado)
26     admin.site.register(Repositorio, RepositorioVideos)
```

Figura 54.- Configuración de la visualización de modelos en la interfaz de administración.  
Fuente: Elaboración propia.

The screenshot shows the 'Añadir repositorio' (Add repository) form. At the top, there is a header with the application name 'INFOCHANNEL - SIS ECU 911' and user navigation links: 'BIENVENIDO/A. LCS. VER EL SITIO / CAMBIAR CONTRASEÑA / TERMINAR SESIÓN'. Below the header is a breadcrumb trail: 'Inicio > Videos > Repositorios > Añadir repositorio'. The main form area is titled 'Añadir repositorio' and contains three input fields: 'Titulo:' with a text box, 'Video src:' with a 'Seleccionar archivo' button and the text 'Ningun archivo seleccionado', and 'Imagen src:' with a 'Seleccionar archivo' button and the text 'Ningun archivo seleccionado'. At the bottom right, there are three buttons: 'Grabar y añadir otro', 'Grabar y continuar editando', and 'GRABAR'.

*Figura 55.- Modelo de subida de videos al Repositorio (vista desde el administrador web).  
Fuente: Elaboración propia.*

The screenshot shows the 'Añadir video' (Add video) form. At the top, there is a header with the application name 'INFOCHANNEL - SIS ECU 911' and user navigation links: 'BIENVENIDO/A. LCS. VER EL SITIO / CAMBIAR CONTRASEÑA / TERMINAR SESIÓN'. Below the header is a breadcrumb trail: 'Inicio > Videos > Videos > Añadir video'. The main form area is titled 'Añadir video' and contains two input fields: 'Titulo:' with a text box and 'Video src:' with a 'Seleccionar archivo' button and the text 'Ningun archivo seleccionado'. At the bottom right, there are three buttons: 'Grabar y añadir otro', 'Grabar y continuar editando', and 'GRABAR'.

*Figura 56.- Modelo de subida de videos al Infochannel (vista desde el administrador web).  
Fuente: Elaboración propia.*

### 5.3.7 Diseño de la interfaz de usuario.

Hasta este punto, se ha llevado a cabo la configuración del back-end de la aplicación. A continuación, se describirá el proceso del diseño de la interfaz de usuario.

La interfaz de usuario está basada en programación de interfaces web, lo que significa el uso de lenguajes como HTML, CSS y JavaScript.

Antes de programar las plantillas de usuario, es necesario configurar su conectividad con el back-end. Para ello, en el archivo "Settings.py", en la sección de "TEMPLATES", es necesario agregar el directorio de alojamiento para las plantillas HTML. Este directorio debe crearse igualmente en el directorio especificado (ver Figura 57).

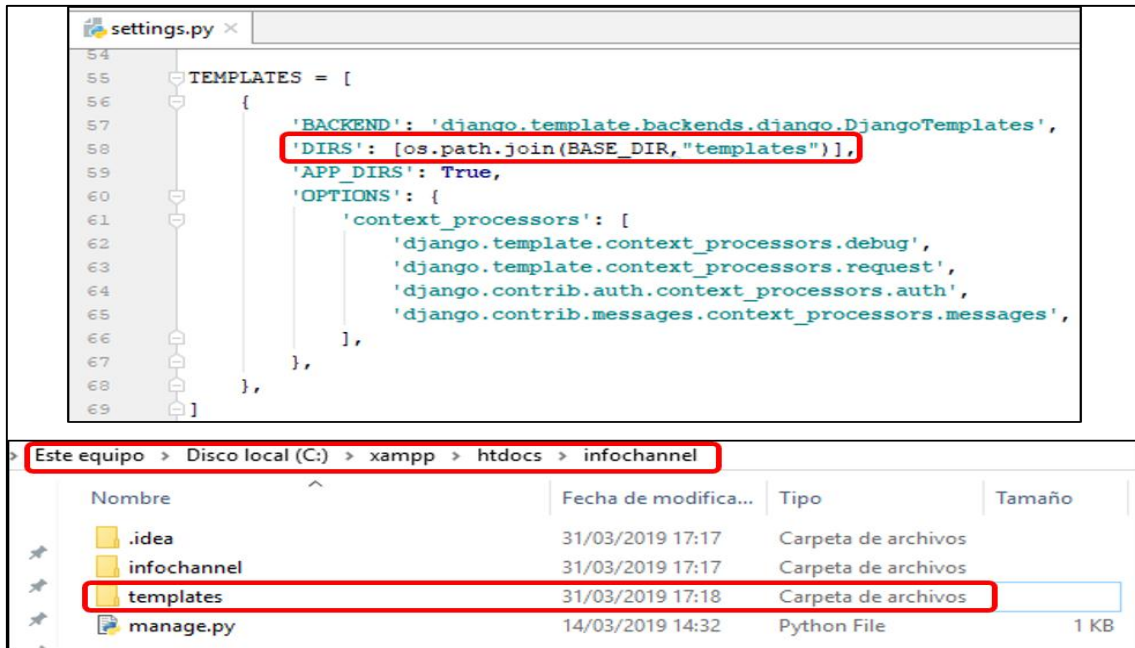


Figura 57.- Configuración del directorio de plantillas.  
Fuente: Elaboración propia.

Además de las configuraciones de la Figura 57, también es necesario indicar los directorios de alojamiento de los archivos estáticos y archivos multimedia que se usarán en el sistema. En la Figura 58, se indica esta configuración, la cual se hace de igual forma en el archivo “Settings.py”.

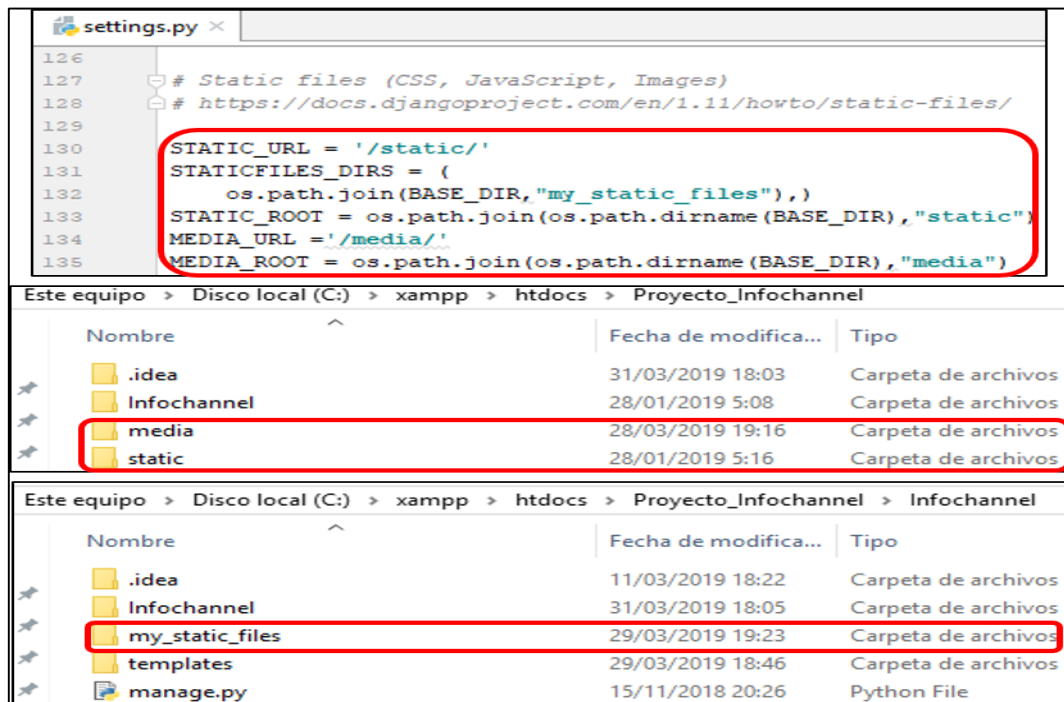
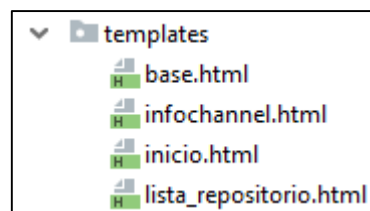


Figura 58.- Configuración de directorios para archivos estáticos y multimedia.  
Fuente: Elaboración propia.



Estos directorios son importantes para la visualización de los archivos tanto estáticos como multimedia dentro del sitio web, sin ellos los usuarios que accedan a la aplicación únicamente podrán visualizar los archivos HTML como texto plano, es decir, sin ningún formato. Dentro de los directorios para archivos estáticos se alojarán los archivos de diseño (archivos CSS) y de funcionalidad (archivos JavaScript), además de las imágenes que dan apariencia a la interfaz de usuario. Estos archivos en conjunción con las plantillas HTML conforman la estructura base de la interfaz de usuario.

Con los directorios ya configurados, ya es posible comenzar con el diseño de las diferentes instancias de la aplicación. Estas instancias se han dividido en 4: una plantilla base, la cual heredará sus características a las demás; la pantalla de inicio o de presentación del sistema, la del Infochannel y una tercera como repositorio de videos (ver Figura 59).



*Figura 59.- Plantillas HTML utilizadas.  
Fuente: Elaboración propia.*

En la plantilla base, usando una de las características de los Frameworks conocida como “herencia de plantillas”, se determinó qué elementos serán comunes para todas, de modo que, al navegar entre las diferentes plantillas, estos elementos permanecerían estáticos, lo que reduce el tiempo de carga de la plantilla y mejora su eficiencia.

Para extender las características de la plantilla base se debe utilizar unas etiquetas especiales que permiten cargar los elementos del archivo “base.html” a las demás plantillas. Esta etiqueta debe ser insertada al inicio de cada una de las plantillas (ver Figura 60).

```
inicio.html x
1 {% extends 'base.html' %}
2
3 {% block title%}ECU 911{% endblock%}
4
5 {% block content %}
6 {% load staticfiles %}

infochannel.html x
1 {% extends 'base.html' %}
2 {% block title%}
3 INFOCHANNEL
4 {% endblock%}
5 {% block javascript %}
6 <script language="javascript">
7     n_video=0;

lista_repositorio.html x
1 {% extends 'base.html' %}
2 {% block title%}
3 REPOSITORIO
4 {% endblock%}
5 {% block javascript %}
6 <script language="javascript">
7     (function($) {
```

Figura 60.- Etiquetas para la herencia de plantillas en Django.  
Fuente: Elaboración propia.

Una de las funcionalidades importantes de JavaScript usadas en el diseño de la interfaz de usuario, es la librería Ajax, la cual permite actualizar los valores de la interfaz de forma localizada, es decir, permite dinamizar ciertos parámetros sin la necesidad de recargar completamente la página. Esta librería fue una importante herramienta a la hora de programar un sistema completamente automático, en lo referente a la actualización de datos y a la interacción con la base de datos.

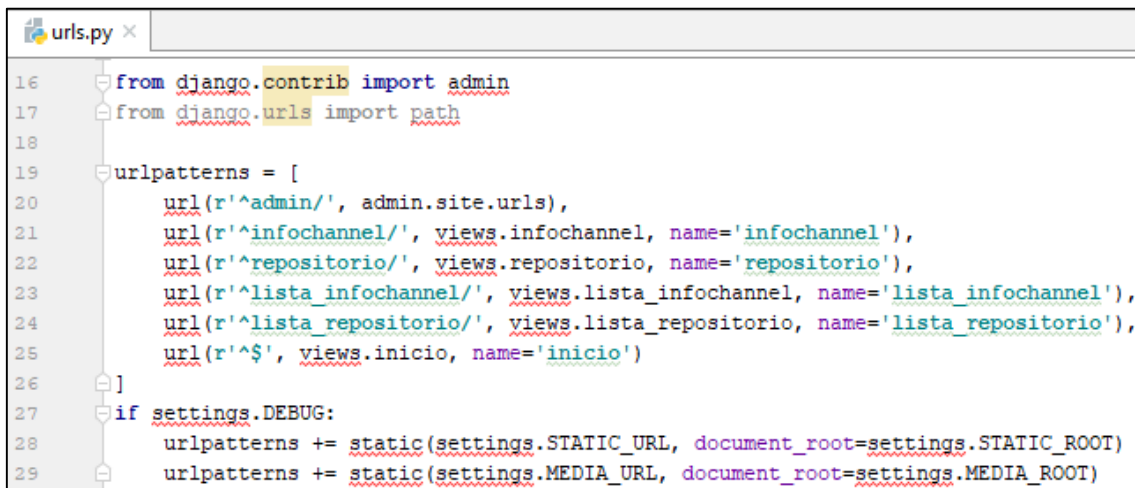
La programación de cada una de las plantillas de este sistema se adjunta en el Anexo 9 de esta tesis.

### 5.3.8 Configuración de las URLs y Vistas.

Las Vistas y las URLs tienen una estrecha relación, ya que estas suponen el nexo entre el usuario o cliente del sistema, y la información almacenada en la base de datos gracias a los modelos. La configuración de estas dos a igual que la de los modelos se relaciona entre sí (ver Figura 22).

Una URL sirve para denominar y localizar recursos dentro de un sitio web. Por medio de las URLs, se podrá navegar entre las diferentes instancias o plantillas del sistema.

Las URLs en Django tienen una sintaxis y fácil de entender. Estas deben ser declaradas en el archivo “urls.py” (ver Figura 31) en base a las Vistas programadas. En la Figura 61 se detalla la configuración de las URLs usadas en el sistema:



```
16 from django.contrib import admin
17 from django.urls import path
18
19 urlpatterns = [
20     url(r'^admin/', admin.site.urls),
21     url(r'^infochannel/', views.infochannel, name='infochannel'),
22     url(r'^repositorio/', views.repositorio, name='repositorio'),
23     url(r'^lista_infochannel/', views.lista_infochannel, name='lista_infochannel'),
24     url(r'^lista_repositorio/', views.lista_repositorio, name='lista_repositorio'),
25     url(r'^$', views.inicio, name='inicio')
26 ]
27 if settings.DEBUG:
28     urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
29     urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Figura 61.- Configuración de las URLs del sistema.  
Fuente: Elaboración propia.

Con las URLs configuradas, lo siguiente es configurar las vistas de forma correcta en base a las URLs. La configuración de las Vistas se realiza en el archivo “views.py” (ver Figura 48). Como se menciona en la literatura de esta tesis, el propósito de las Vistas es principalmente determinar qué datos serán visualizados, aunque también se encarga de tareas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. Hay que recalcar que la vista no tiene nada que ver con el estilo de presentación de los datos, sólo se encarga de los datos, la presentación es tarea de las plantillas. En la Figura 62 se detalla la configuración de las Vistas del sistema:

```
views.py x
1 from django.shortcuts import render
2 from .models import Video, Repositorio
3 from django.http import JsonResponse
4 from django.core import serializers
5
6 # Create your views here.
7
8 def infochannel(request):
9     return render_(request, "infochannel.html")
10
11 def lista_infochannel(request):
12     lista=Video.objects.all() # asignar valores desde los modelos a la variable lista
13     posts_serialized = serializers.serialize('json', lista) # se da formato json a la lista de archivos
14     return JsonResponse({"listado":posts_serialized}) # regresa la informacion en formato json
15
16 def repositorio(request):
17     return render_(request, "lista_repositorio.html")
18
19 def lista_repositorio(request):
20     lista=Repositorio.objects.all() # asignar valores desde los modelos a la variable lista
21     posts_serialized = serializers.serialize('json', lista) # se da formato json a la lista de archivos
22     return JsonResponse({"listado":posts_serialized}) # regresa la informacion en formato json
23
24 def inicio(request):
25     return render (request, "inicio.html")
```

Figura 62.- Configuración de las Vistas del sistema.  
Fuente: Elaboración propia.

Las configuraciones de las URLs y las Vistas suponen la finalización del diseño y configuración del sistema. Por supuesto este es un sistema probado únicamente en un entorno local, lo que no permite visualizar posibles fallas en su funcionamiento. Para solventar esto es que se requiere de una última etapa en la cual se ponga a prueba el sistema en un entorno real.

### 5.3.9 Configurar el acceso externo a la aplicación.

Hasta este punto la aplicación únicamente ha podido probarse localmente desde la computadora de desarrollo, mediante una dirección loopback. Para realizar pruebas desde un dispositivo externo, es necesario realizar ciertos ajustes a la configuración del servidor Apache.

En primer lugar, desde la interfaz de administración de XAMPP, accedemos al archivo “httpd-xampp.conf” (ver Figura 63)

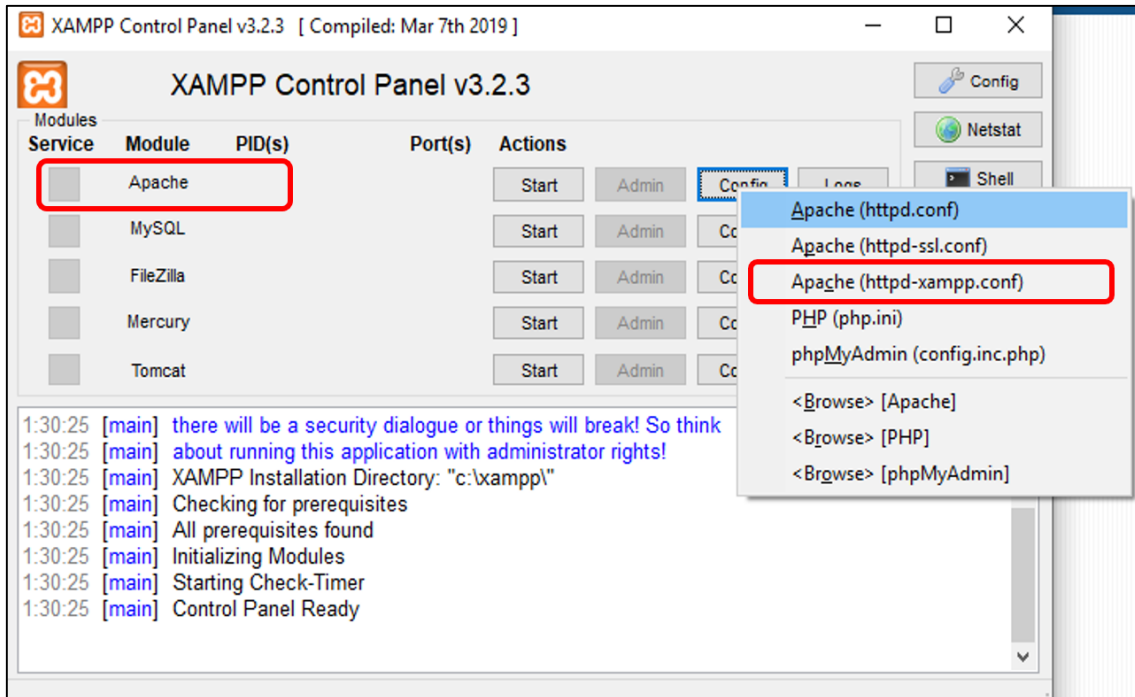


Figura 63.- Acceso al archivo de configuración del servidor Apache.  
Fuente: Elaboración propia.

Dentro de este archivo es necesario agregar ciertas líneas de código que permitirán el acceso externo a la aplicación. El código que debe agregarse se indica a continuación:

```
#
# New XAMPP security concept
#

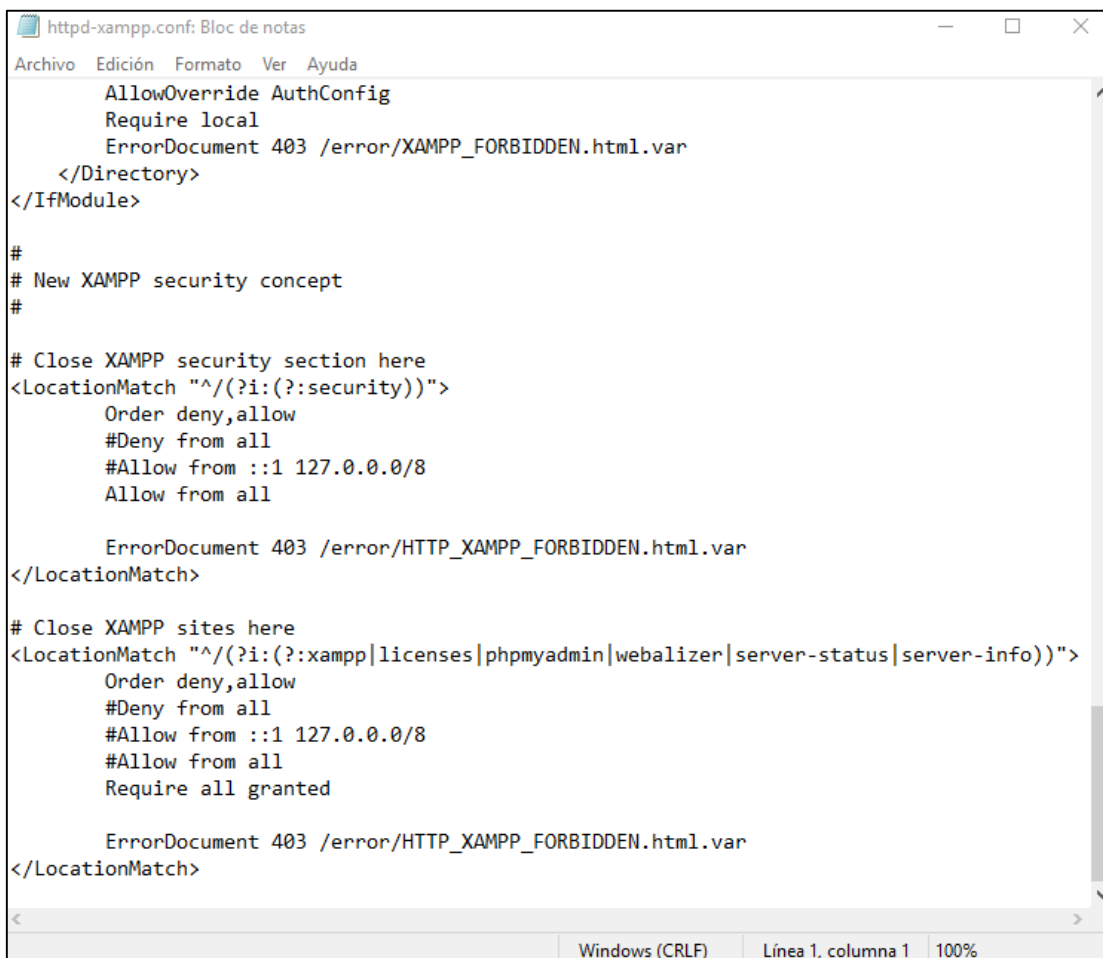
# Close XAMPP security section here
<LocationMatch "^/(?i:(?:security))">
Order deny,allow
#Deny from all
#Allow from ::1 127.0.0.0/8
Allow from all

ErrorDocument 403 /error/HTTP_XAMPP_FORBIDDEN.html.var
</LocationMatch>

# Close XAMPP sites here
<LocationMatch "^/(?i:(?:xampp|licenses|phpmyadmin|webalizer|server-status|server-info))">
Order deny,allow
#Deny from all
#Allow from ::1 127.0.0.0/8
#Allow from all
Require all granted

ErrorDocument 403 /error/HTTP_XAMPP_FORBIDDEN.html.var
</LocationMatch>
```

Estas líneas de código deben agregarse al final del archivo quedando de la siguiente manera:



```
httpd-xampp.conf: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
AllowOverride AuthConfig
Require local
ErrorDocument 403 /error/XAMPP_FORBIDDEN.html.var
</Directory>
</IfModule>
#
# New XAMPP security concept
#
# Close XAMPP security section here
<LocationMatch '^/(?i:(?:security))'>
    Order deny,allow
    #Deny from all
    #Allow from ::1 127.0.0.0/8
    Allow from all

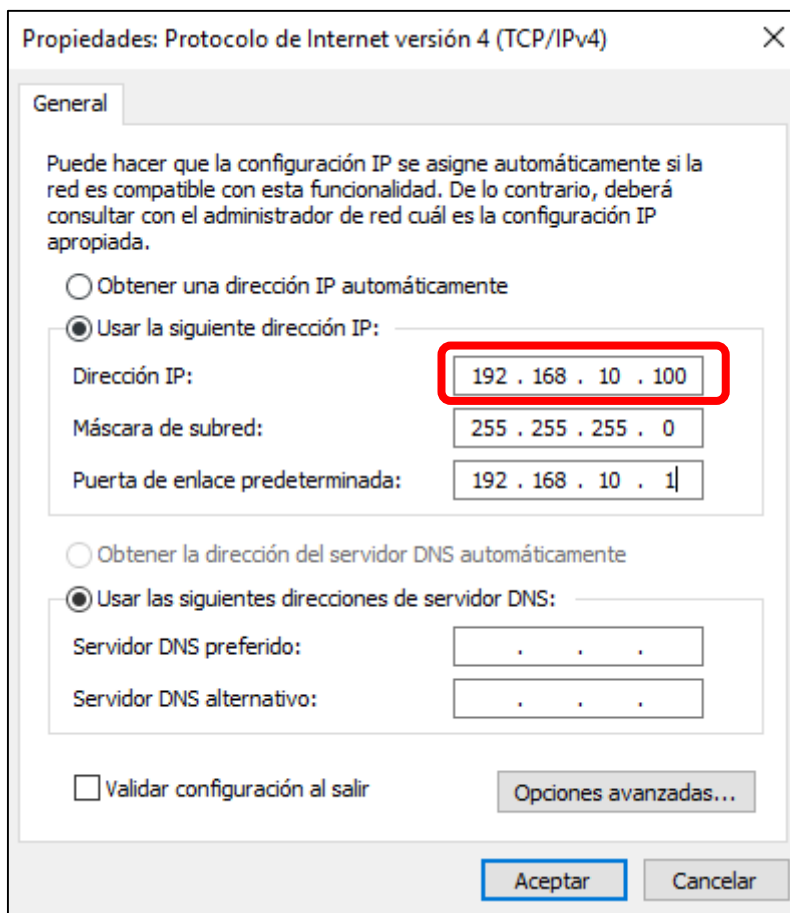
    ErrorDocument 403 /error/HTTP_XAMPP_FORBIDDEN.html.var
</LocationMatch>
# Close XAMPP sites here
<LocationMatch '^/(?i:(?:xampp|licenses|phpmyadmin|webalizer|server-status|server-info))'>
    Order deny,allow
    #Deny from all
    #Allow from ::1 127.0.0.0/8
    #Allow from all
    Require all granted

    ErrorDocument 403 /error/HTTP_XAMPP_FORBIDDEN.html.var
</LocationMatch>
Windows (CRLF)  Línea 1, columna 1  100%
```

*Figura 64.- Archivo de configuración de Apache modificado para el acceso remoto al servidor.  
Fuente: Elaboración propia.*

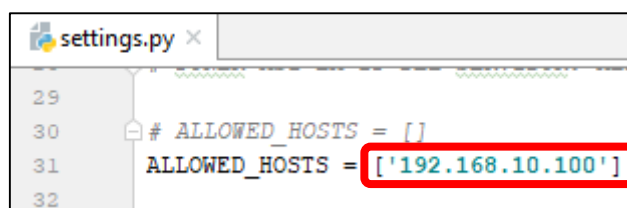
Con la configuración indicada en la Figura 64 ya se ha habilitado el acceso remoto al servicio. Sin embargo, es necesario realizar otras configuraciones adicionales.

Al tratarse de un servidor, su dirección IP debe ser configurada estáticamente de modo que cualquier dispositivo sepa siempre a dónde apuntar en caso de solicitar sus servicios. Para ello es necesario configurar una IP estática al equipo sobre el cual se desplegará el sistema (ver Figura 65).



*Figura 65.- Configuración de la dirección IP del sistema en el dispositivo servidor.  
Fuente: Elaboración propia.*

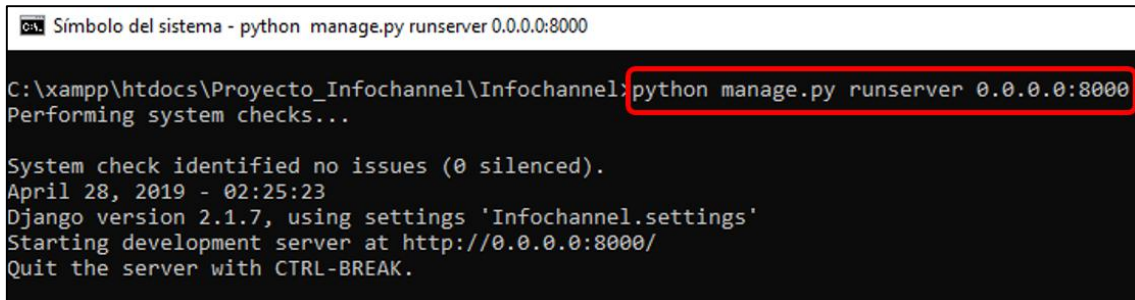
Otro de los requisitos necesarios es configurar la IP del servidor dentro del código fuente de la aplicación. Para ello se accede al archivo de configuración del proyecto Django “Settings.py”, en el cual se debe realizar los siguientes cambios:



*Figura 66.- Configuración de la dirección IP del servidor en la aplicación.  
Fuente: Elaboración propia.*

La dirección IP estática del servidor debe ser agregada tal y como se muestra en la Figura 66, de esta manera la aplicación diseñada sería accesible desde cualquier dispositivo de la red sobre la que se despliegue.

Con las configuraciones antes indicadas, ahora es necesario arrancar el servidor Apache tal, el SGBD MySQL y el servidor de la aplicación. Para ello desde el cmd accedemos al directorio de la aplicación en donde se encuentre el archivo “manage.py” y ejecutamos el servidor como se indica en la Figura 67.

A screenshot of a Windows command prompt window. The title bar reads "Símbolo del sistema - python manage.py runserver 0.0.0.0:8000". The command prompt shows the directory path "C:\xampp\htdocs\Proyecto\_Infochannel\Infochannel>" followed by the command "python manage.py runserver 0.0.0.0:8000", which is highlighted with a red rectangular box. The output of the command is displayed below: "Performing system checks...", "System check identified no issues (0 silenced).", "April 28, 2019 - 02:25:23", "Django version 2.1.7, using settings 'Infochannel.settings'", "Starting development server at http://0.0.0.0:8000/", and "Quit the server with CTRL-BREAK." data-bbox="141 208 855 341"/>

*Figura 67.- Arranque del servidor Django sobre red real.  
Fuente: Elaboración propia.*

### **5.3.10 Determinación de parámetros para el correcto funcionamiento del sistema.**

Hasta este punto se ha concluido con el desarrollo del sistema y este está listo para ser probado en un entorno real. Como todo sistema previo a ser implementado, este debe ser sometido a múltiples pruebas que aseguren y validen su correcto funcionamiento. A lo largo del diseño del presente sistema, se han podido realizar pruebas de escritorio y pruebas locales sobre el funcionamiento del sistema, de modo que gracias a ello se ha podido detectar y corregir errores en su estructura. Las pruebas locales se realizaron en el mismo equipo en el que se desarrolló y programó el sistema, gracias al servidor de desarrollo que incorpora el Framework Django. El despliegue y uso de este servidor ya se mencionó en el apartado 5.3.4 del presente documento.

Sin embargo, las pruebas locales no son suficientes para corroborar el correcto funcionamiento del sistema, sino que requiere ser ejecutado sobre un entorno real que pueda dar datos más exactos acerca del funcionamiento del mismo.

Para las pruebas de funcionamiento se ha recurrido a una red doméstica descrita en la Figura 68.



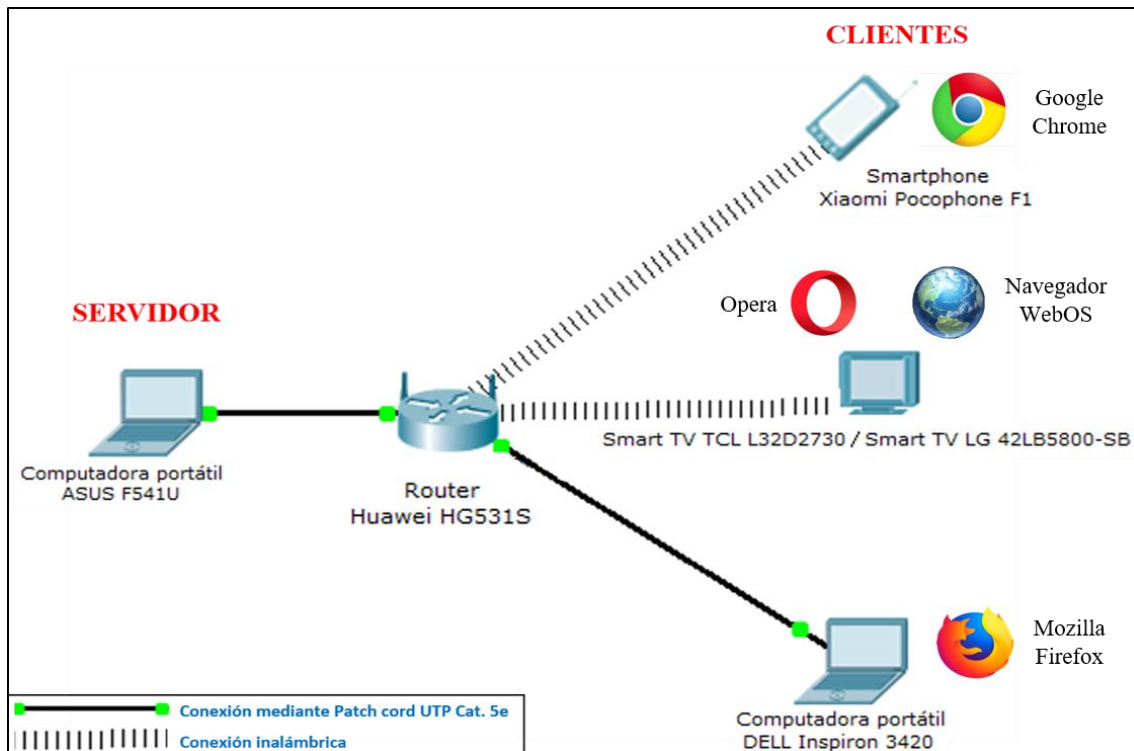


Figura 68.- Red de pruebas del sistema.  
Fuente: Elaboración propia.

Al tratarse de una arquitectura cliente-servidor, para acceder al servicio de *streaming*, los dispositivos cliente deben conectarse desde el navegador a la dirección IP del servidor, la cual debe ser configurada de forma estática dentro de la red.

Sobre la red mostrada en la Figura 68, se llevó a cabo las diferentes pruebas de calidad y compatibilidad de los códecs, así como las pruebas de funcionalidad de la interfaz de usuario. Como se especifica en la misma figura, se utilizaron como dispositivos de prueba para el sistema, un smartphone, dos Smart Tvs, y una computadora portátil, cada uno de ellos con sus respectivos navegadores web.

Para comprobar la compatibilidad y disponibilidad de los códecs de audio y video bajo el estándar HTML5 con un navegador web, se utilizó la herramienta en línea “HTML5 TEST”, la cual hace un análisis detallado del navegador en cuestión y devuelve como resultado la calidad con la cual dicho navegador implementa el estándar HTML5. A continuación, en la Figura 69 se puede evidenciar un ejemplo del test HTML5 de los navegadores Google Chrome, Mozilla Firefox, Opera y el Navegador WebOs, junto a los dispositivos usados para las pruebas de desarrollo del sistema.

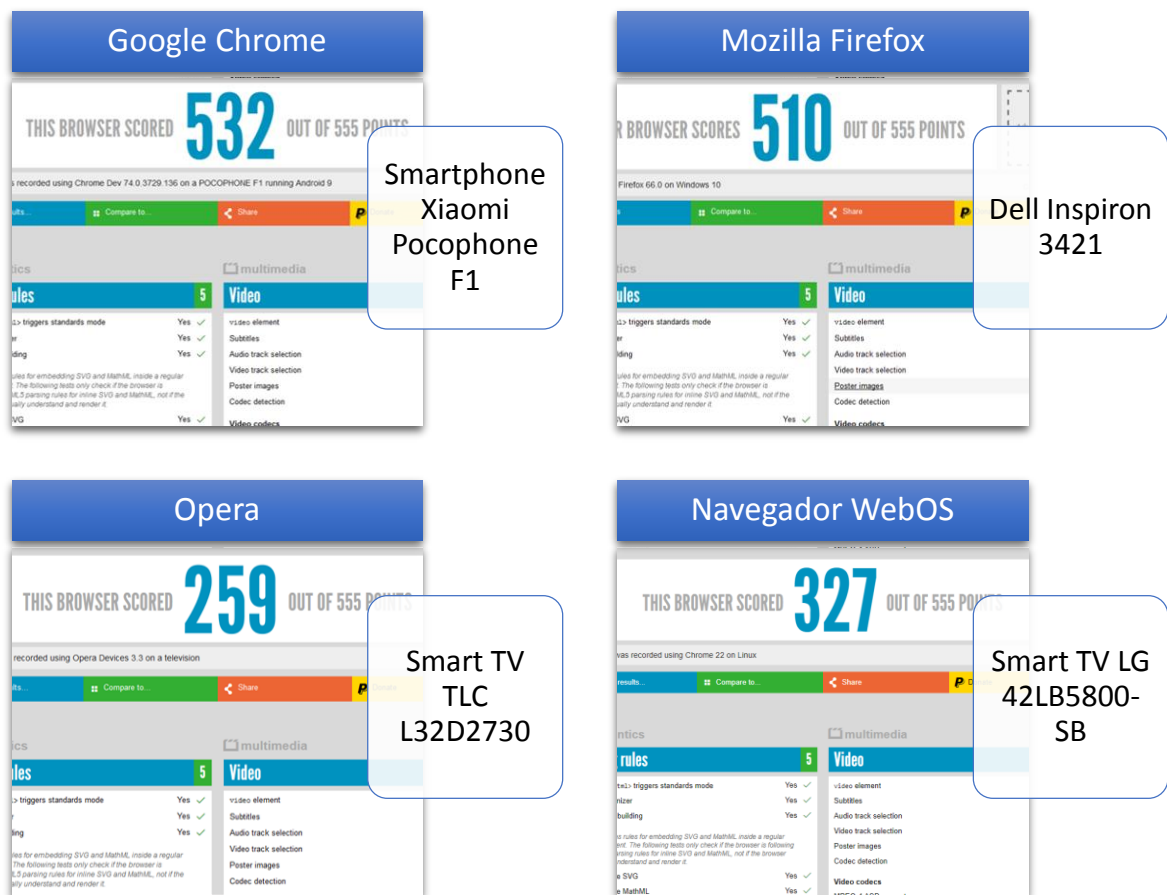
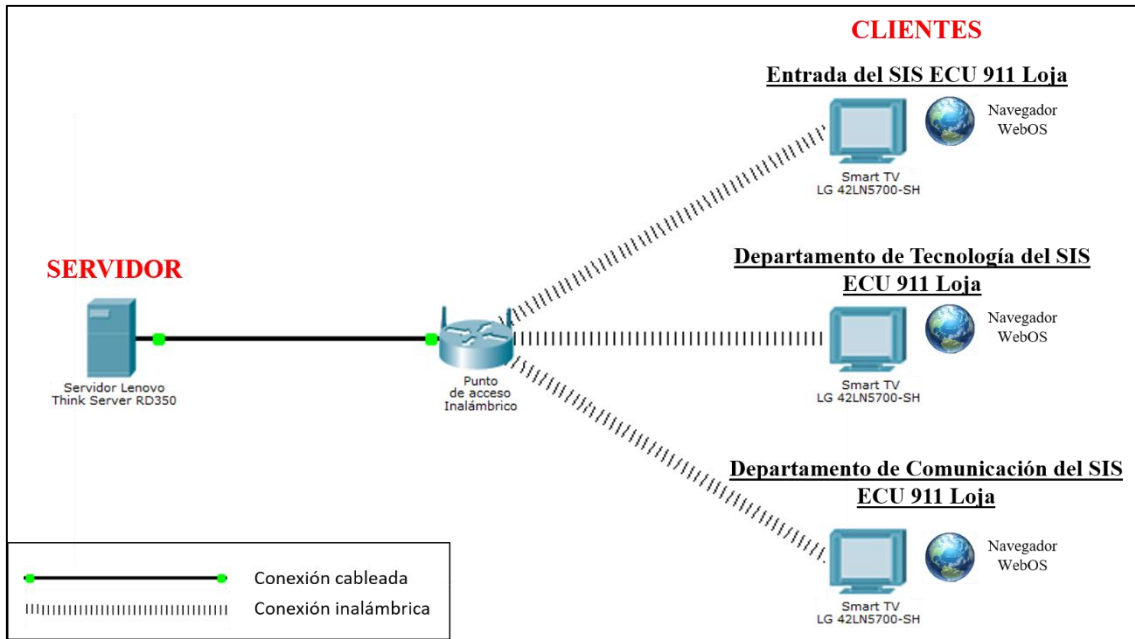


Figura 69.- Calificación de parámetros HTML5 de los navegadores utilizados.  
Fuente: Elaboración propia.

Otro de los parámetros a tener en cuenta respecto a los formatos de video es la tasa de compresión de sus códecs. Para ello se hizo uso de la herramienta de software *Winx HD Video Converter Deluxe* como codificador de video. Las tasas de compresión promedio se obtuvieron en base a la compresión de un conjunto muestral de 50 videos con una duración de 5 minutos cada uno.

Los resultados obtenidos de las diferentes pruebas mencionadas posteriormente servirían para implementar el sistema de forma ágil sobre la red del SIS ECU 911 de la ciudad de Loja. Dentro de dicha red, el sistema tendría como dispositivos finales los Smart Tvs LG modelo 42LN5700-SH (ver Tabla 12) con su navegador propio del sistema operativo WebOS. La red de despliegue del sistema en la institución ECU 911 de la ciudad de Loja puede evidenciarse en la Figura 70.



*Figura 70.- Red del SIS ECU 911 de la ciudad de Loja sobre la cual se instaló el sistema de Infochannel.  
Fuente: Elaboración propia.*

Los parámetros planteados para el correcto funcionamiento del sistema de Infochannel, fueron determinados junto con el personal técnico a cargo de la administración de la red de la institución pública SIS ECU 911 de la ciudad de Loja.

## 6. RESULTADOS.

Los diferentes resultados del presente proyecto de tesis se detallan a continuación:

### 6.1 Características del formato de video.

Para garantizar una buena calidad de servicio, es necesario determinar qué formato de video será el más adecuado para el funcionamiento del sistema. Para determinar esto existen diversos que se han considerado y medido. A continuación, se presentan los resultados de algunos de los criterios tomados en cuenta para la selección del formato de video adecuado a las necesidades de la problemática de este proyecto de investigación.

#### 6.1.1 Interoperabilidad de los códecs HTML5.

Utilizando la aplicación HTML5 TEST se obtuvo que los códecs, tanto de video como de audio, soportados por los navegadores utilizados son:

##### 6.1.1.1 Códecs de video.

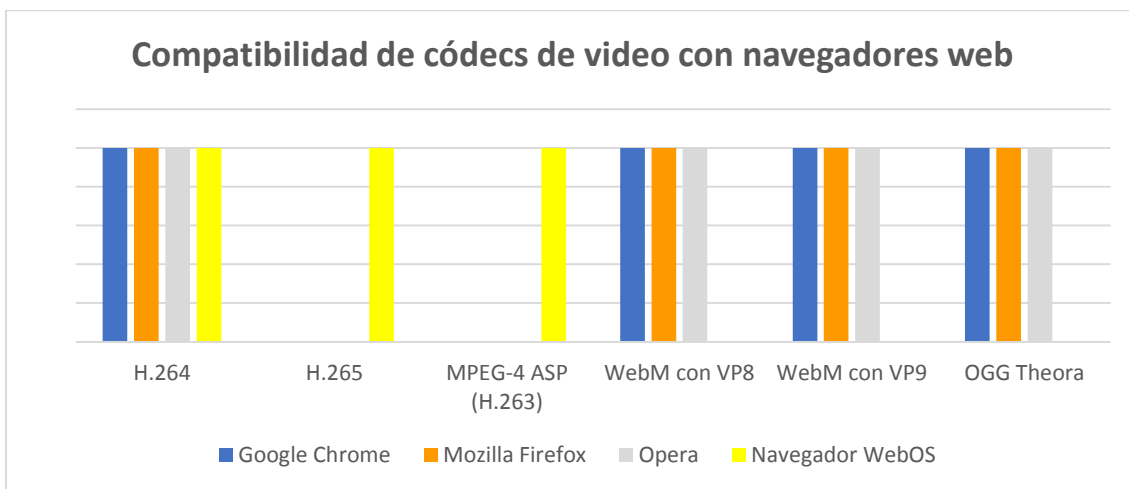
| GOOGLE CHROME         |              | MOZILLA FIREFOX       |              |
|-----------------------|--------------|-----------------------|--------------|
| <b>Video</b>          | <b>29/33</b> | <b>Video</b>          | <b>29/33</b> |
| video element         | Yes ✓        | video element         | Yes ✓        |
| Subtitles             | Yes ✓        | Subtitles             | Yes ✓        |
| Audio track selection | No ✗         | Audio track selection | No ✗         |
| Video track selection | No ✗         | Video track selection | No ✗         |
| Poster images         | Yes ✓        | Poster images         | Yes ✓        |
| Codec detection       | Yes ✓        | Codec detection       | Yes ✓        |
| <b>Video codecs</b>   |              | <b>Video codecs</b>   |              |
| MPEG-4 ASP support    | No ✗         | MPEG-4 ASP support    | No ✗         |
| H.264 support         | Yes ✓        | H.264 support         | Yes ✓        |
| H.265 support         | No ✗         | H.265 support         | No ✗         |
| Ogg Theora support    | Yes ✓        | Ogg Theora support    | Yes ✓        |
| WebM with VP8 support | Yes ✓        | WebM with VP8 support | Yes ✓        |
| WebM with VP9 support | Yes ✓        | WebM with VP9 support | Yes ✓        |
| OPERA                 |              | NAVEGADOR WEBOS       |              |
| <b>Video</b>          | <b>29/33</b> | <b>Video</b>          | <b>29/33</b> |
| video element         | Yes ✓        | video element         | Yes ✓        |
| Subtitles             | Yes ✓        | Subtitles             | Yes ✓        |
| Audio track selection | No ✗         | Audio track selection | No ✗         |
| Video track selection | No ✗         | Video track selection | No ✗         |
| Poster images         | Yes ✓        | Poster images         | Yes ✓        |
| Codec detection       | Yes ✓        | Codec detection       | Yes ✓        |
| <b>Video codecs</b>   |              | <b>Video codecs</b>   |              |
| MPEG-4 ASP support    | No ✗         | MPEG-4 ASP support    | Yes ✓        |
| H.264 support         | Yes ✓        | H.264 support         | Yes ✓        |
| H.265 support         | No ✗         | H.265 support         | Yes ✓        |
| Ogg Theora support    | Yes ✓        | Ogg Theora support    | No ✗         |
| WebM with VP8 support | Yes ✓        | WebM with VP8 support | No ✗         |
| WebM with VP9 support | Yes ✓        | WebM with VP9 support | No ✗         |

Figura 71.- Códecs de video determinados con la herramienta "HTML5 TEST".

Fuente: Elaboración propia.

**Tabla 14.-** Compatibilidad de códecs de video con navegadores web.  
**Fuente:** Elaboración propia.

| CÓDECS          | H.264 | H.265 | MPEG-4 ASP (H.263) | WebM con VP8 | WebM con VP9 | OGG Theora |
|-----------------|-------|-------|--------------------|--------------|--------------|------------|
| Google Chrome   | SI    | NO    | NO                 | SI           | SI           | SI         |
| Mozilla Firefox | SI    | NO    | NO                 | SI           | SI           | SI         |
| Opera           | SI    | NO    | NO                 | SI           | SI           | SI         |
| Navegador WebOS | SI    | Si    | SI                 | NO           | NO           | NO         |



**Figura 72.-** Compatibilidad de códecs de video con navegadores web.  
**Fuente:** Elaboración propia.

De acuerdo con los datos obtenidos de la aplicación “HTML5 TEST”, el códec de video H.264 es compatible con el 100% de los navegadores seleccionados para el estudio, los códecs H.265 y MPEG-4 ASP (H.263) son compatibles con un 25% de los navegadores y, por último, los códecs de video VP8, VP9 y Theora son compatibles con un 75% del total.

### 6.1.1.2 Códecs de audio.

| GOOGLE CHROME              |              | MOZILLA FIREFOX            |              |
|----------------------------|--------------|----------------------------|--------------|
| <b>Audio</b>               | <b>29/30</b> | <b>Audio</b>               | <b>27/30</b> |
| audio element              | Yes ✓        | audio element              | Yes ✓        |
| Loop audio                 | Yes ✓        | Loop audio                 | Yes ✓        |
| Preload in the background  | Yes ✓        | Preload in the background  | Yes ✓        |
| <b>Advanced</b>            |              | <b>Advanced</b>            |              |
| Web Audio API              | Yes ✓        | Web Audio API              | Yes ✓        |
| Speech Recognition         | Prefixed ✓   | Speech Recognition         | No ✗         |
| Speech Synthesis           | Yes ✓        | Speech Synthesis           | Yes ✓        |
| <b>Audio codecs</b>        |              | <b>Audio codecs</b>        |              |
| PCM audio support          | Yes ✓        | PCM audio support          | Yes ✓        |
| MP3 support                | Yes ✓        | MP3 support                | Yes ✓        |
| AAC support                | Yes ✓        | AAC support                | Yes ✓        |
| Dolby Digital support      | No ✗         | Dolby Digital support      | No ✗         |
| Dolby Digital Plus support | No ✗         | Dolby Digital Plus support | No ✗         |
| Ogg Vorbis support         | Yes ✓        | Ogg Vorbis support         | Yes ✓        |
| Ogg Opus support           | Yes ✓        | Ogg Opus support           | Yes ✓        |
| WebM with Vorbis support   | Yes ✓        | WebM with Vorbis support   | Yes ✓        |
| WebM with Opus support     | Yes ✓        | WebM with Opus support     | Yes ✓        |
| OPERA                      |              | NAVEGADOR WEBOS            |              |
| <b>Audio</b>               | <b>29/30</b> | <b>Audio</b>               | <b>20/30</b> |
| audio element              | Yes ✓        | audio element              | Yes ✓        |
| Loop audio                 | Yes ✓        | Loop audio                 | Yes ✓        |
| Preload in the background  | Yes ✓        | Preload in the background  | Yes ✓        |
| <b>Advanced</b>            |              | <b>Advanced</b>            |              |
| Web Audio API              | Yes ✓        | Web Audio API              | No ✗         |
| Speech Recognition         | Prefixed ✓   | Speech Recognition         | No ✗         |
| Speech Synthesis           | Yes ✓        | Speech Synthesis           | No ✗         |
| <b>Audio codecs</b>        |              | <b>Audio codecs</b>        |              |
| PCM audio support          | Yes ✓        | PCM audio support          | No ✗         |
| MP3 support                | Yes ✓        | MP3 support                | Yes ✓        |
| AAC support                | Yes ✓        | AAC support                | Yes ✓        |
| Dolby Digital support      | No ✗         | Dolby Digital support      | Yes ✓        |
| Dolby Digital Plus support | No ✗         | Dolby Digital Plus support | Yes ✓        |
| Ogg Vorbis support         | Yes ✓        | Ogg Vorbis support         | No ✗         |
| Ogg Opus support           | Yes ✓        | Ogg Opus support           | No ✗         |
| WebM with Vorbis support   | Yes ✓        | WebM with Vorbis support   | No ✗         |
| WebM with Opus support     | Yes ✓        | WebM with Opus support     | No ✗         |

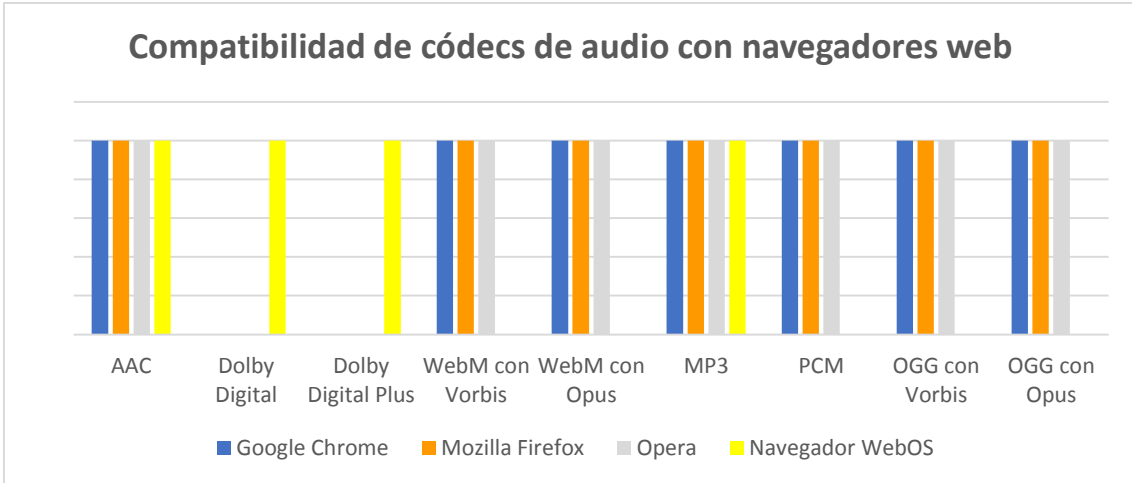
Figura 73.- Códecs de audio determinados con la herramienta "HTML5 TEST".

Fuente: Elaboración propia.

Tabla 15.- Compatibilidad de códecs de audio con navegadores web.

Fuente: Elaboración propia.

| CÓDECS          | AAC | Dolby Digital | Dolby Digital Plus | WebM con Vorbis | WebM con Opus | MP3 | PCM | OGG con Vorbis | OGG con Opus |
|-----------------|-----|---------------|--------------------|-----------------|---------------|-----|-----|----------------|--------------|
| Google Chrome   | SI  | NO            | NO                 | SI              | SI            | SI  | SI  | SI             | SI           |
| Mozilla Firefox | SI  | NO            | NO                 | SI              | SI            | SI  | SI  | SI             | SI           |
| Opera           | SI  | NO            | NO                 | SI              | SI            | SI  | SI  | SI             | SI           |
| Navegador WebOS | SI  | SI            | SI                 | NO              | NO            | SI  | NO  | NO             | NO           |

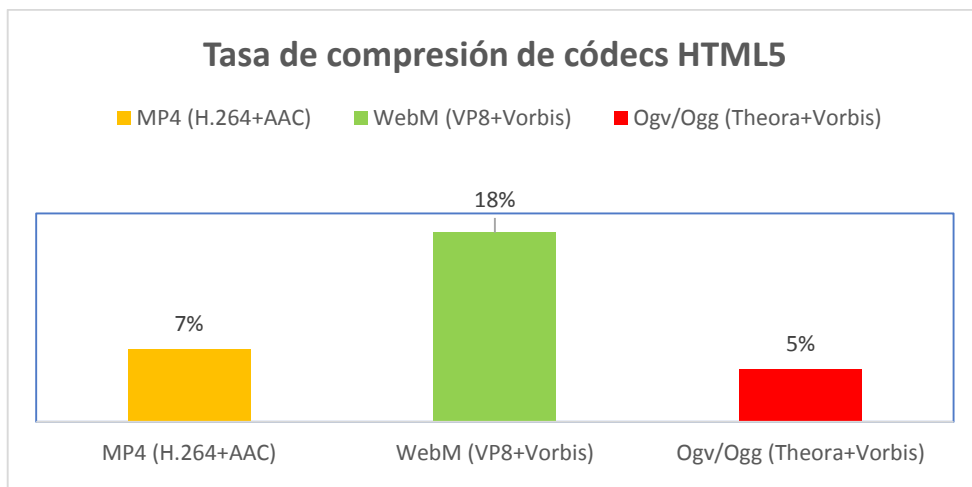


*Figura 74.-Compatibilidad de códecs de audio con navegadores web.  
Fuente: Elaboración propia.*

De acuerdo con los datos obtenidos de la aplicación “HTML5 TEST”, los códecs de audio AAC y MP3 son compatibles con el 100% de los navegadores utilizados. Los códecs Vorbis (WebM), Opus (WebM), PCM, Vorbis (OGG) y Opus (OGG) son compatibles con el 75% de los navegadores y, finalmente, los códecs Dolby Digital y Dolby Digital Plus son compatibles con un 25% de los navegadores.

### 6.1.2 Tasa de compresión de códecs.

La tasa de compresión de estos códecs es importante para disminuir el uso de memoria de almacenamiento en el servidor. Las tasas de compresión promedio para los códecs compatibles con HTML5 obtenidas con el software *Winx HD Video Converter Deluxe* son:



*Figura 75.- Tasa de compresión de códecs HTML.  
Fuente: Elaboración propia.*

De acuerdo a los resultados mostrados en la Figura 75, el formato de video WebM (VP8+Vorbis) ofrece una tasa de compresión del 18%, seguido del formato MP4 (H.264+AAC) con un 7% de compresión y finalmente el formato OGV/OGG (Theora+Vorbis) el cual ofrece una compresión del 5%.

### 6.1.3 Características generales de los formatos de video para el servicio de video streaming.

De acuerdo a la información recopilada en la literatura de este documento, a los resultados de las pruebas de compatibilidad, tasa de compresión y calidad de los códecs estándar de HTML5, se obtuvieron los siguientes resultados:

*Tabla 16.- Características de los códecs estándar para HTML5.  
Fuente: Elaboración propia.*

| Formato de video        | Basado en estándares abiertos | Compatibilidad/ Disponibilidad | Tasa de compresión | Latencia | Calidad de imagen |
|-------------------------|-------------------------------|--------------------------------|--------------------|----------|-------------------|
| MP4 (H.264+AAC)         | Si                            | Alta                           | 7%                 | Baja     | Media/Alta        |
| WebM (VP8+Vorbis)       | Si                            | Media                          | 18%                | Baja     | Alta              |
| OGV/OGG (Theora+Vorbis) | Si                            | Baja/Nula                      | 5%                 | Media    | Baja              |

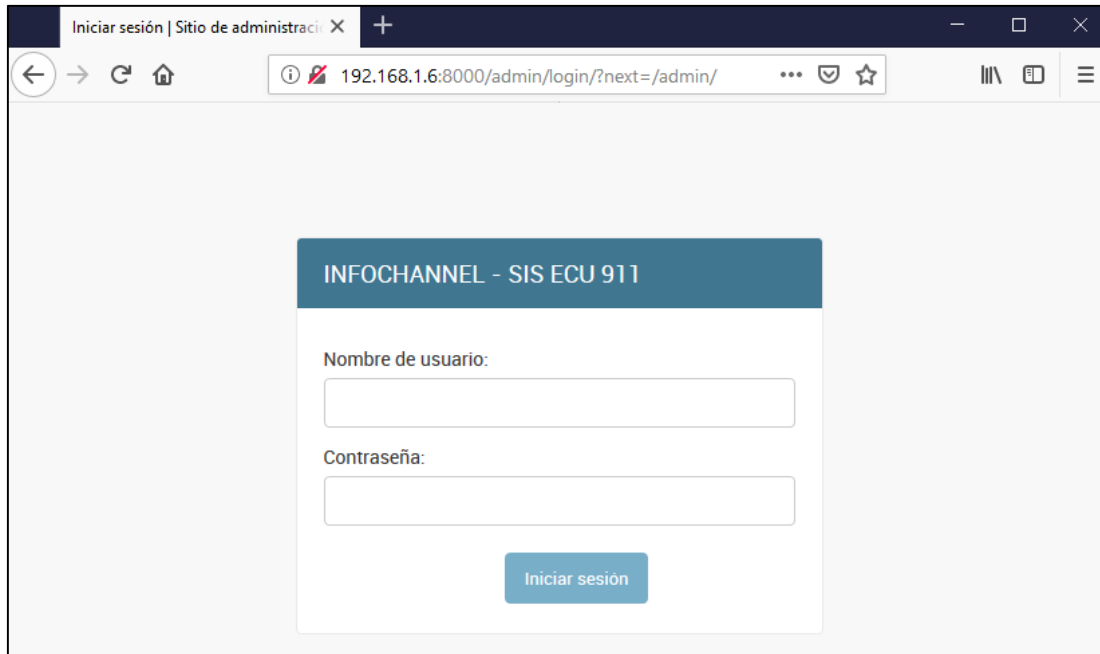
## 6.2 Sistema de “Infochannel”.

El desarrollo del sistema tuvo como resultado dos tipos de interfaz: La interfaz de administración y la interfaz de usuario.

### 6.2.1 Interfaz de administración.

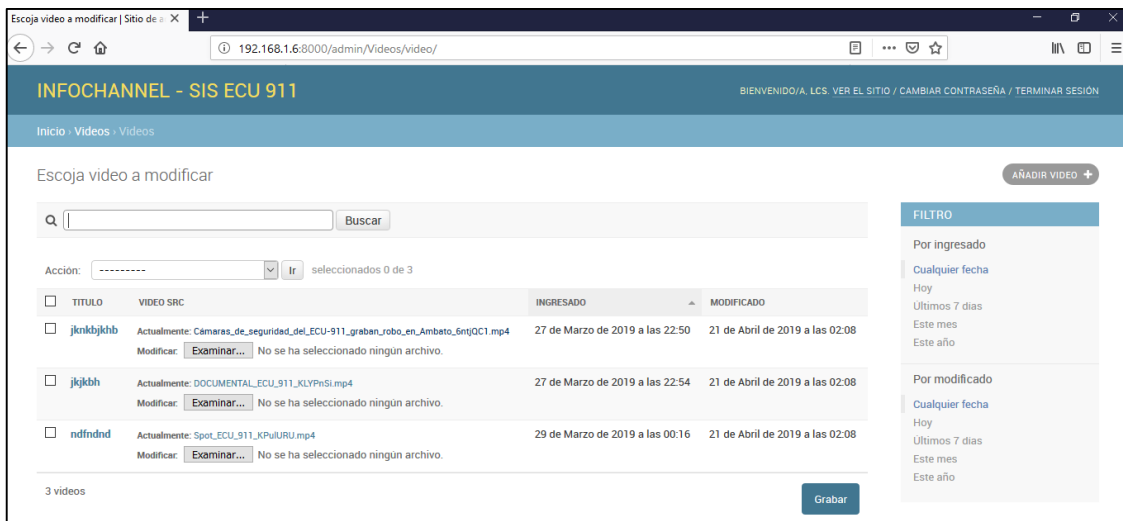
La interfaz de administración tiene como finalidad controlar el contenido multimedia que se podrá visualizar en la interfaz de usuario. El acceso a la interfaz administrativa estará limitado únicamente al administrador de red o a cualquier usuario autorizado por él, ya que consta de un sistema de acceso mediante usuario y contraseña (Figura 76).





**Figura 76.-** Interfaz de acceso al panel de administración del Infochannel.  
**Fuente:** Elaboración propia.

La interfaz de administración también posee un sistema de control sencillo y fácil de manejar, tal como se ve en la Figura 77.



**Figura 77.-** Interfaz de administración del Infochannel.  
**Fuente:** Elaboración propia.

## 6.2.2 Interfaz de usuario.

A continuación, se puede ver la interfaz de usuario del sistema de Infochannel:



Figura 78.- Pantalla de inicio del Infochannel.  
Fuente: Elaboración propia.

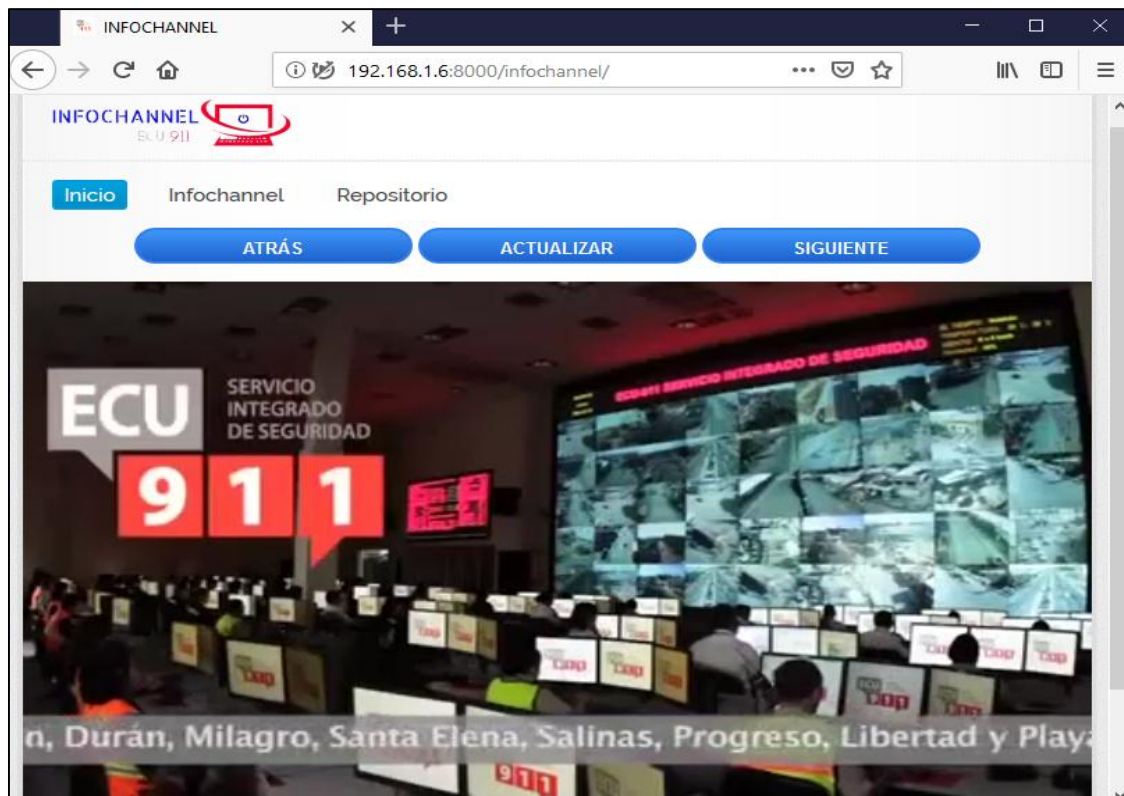
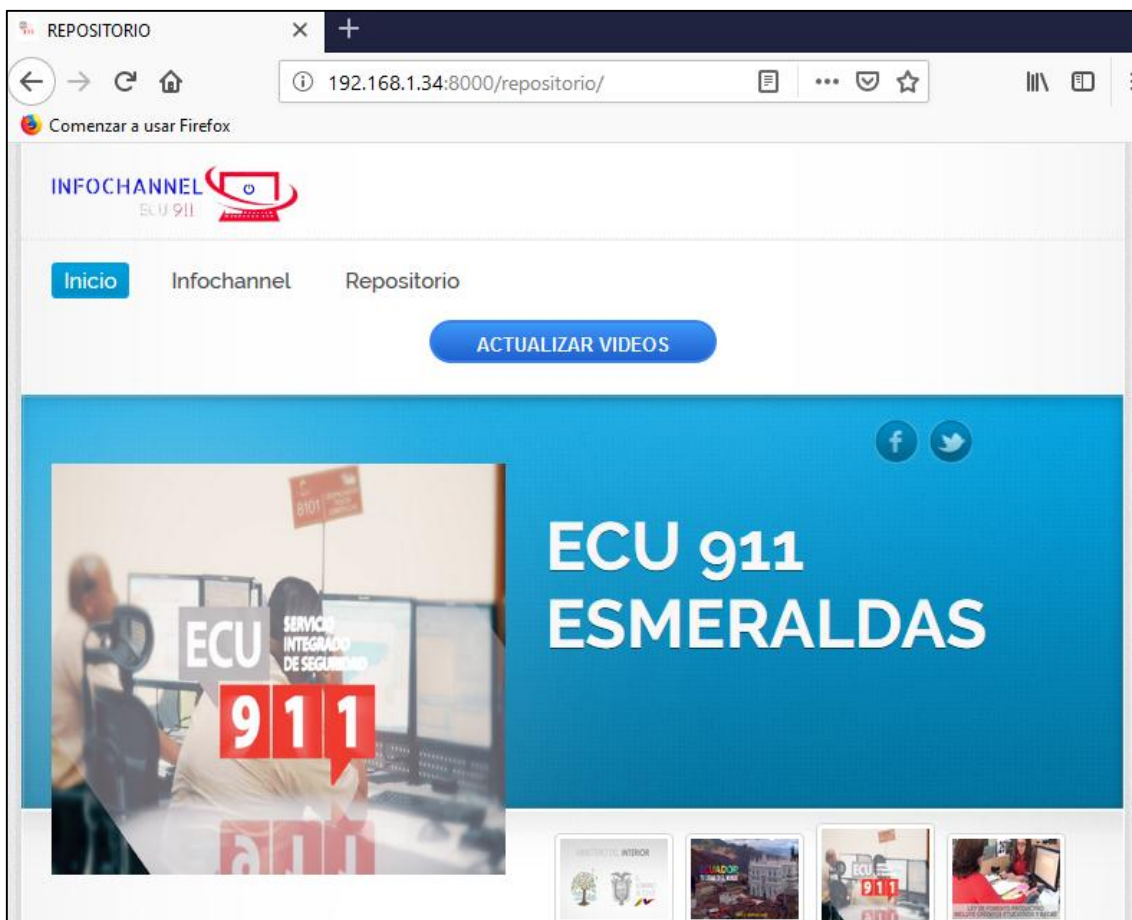


Figura 79.- Pantalla de reproducción del Infochannel.  
Fuente: Elaboración propia.



*Figura 80.- Pantalla de visualización del Repositorio.  
Fuente: Elaboración propia.*

### 6.2.3 Diagrama físico del sistema implementado.

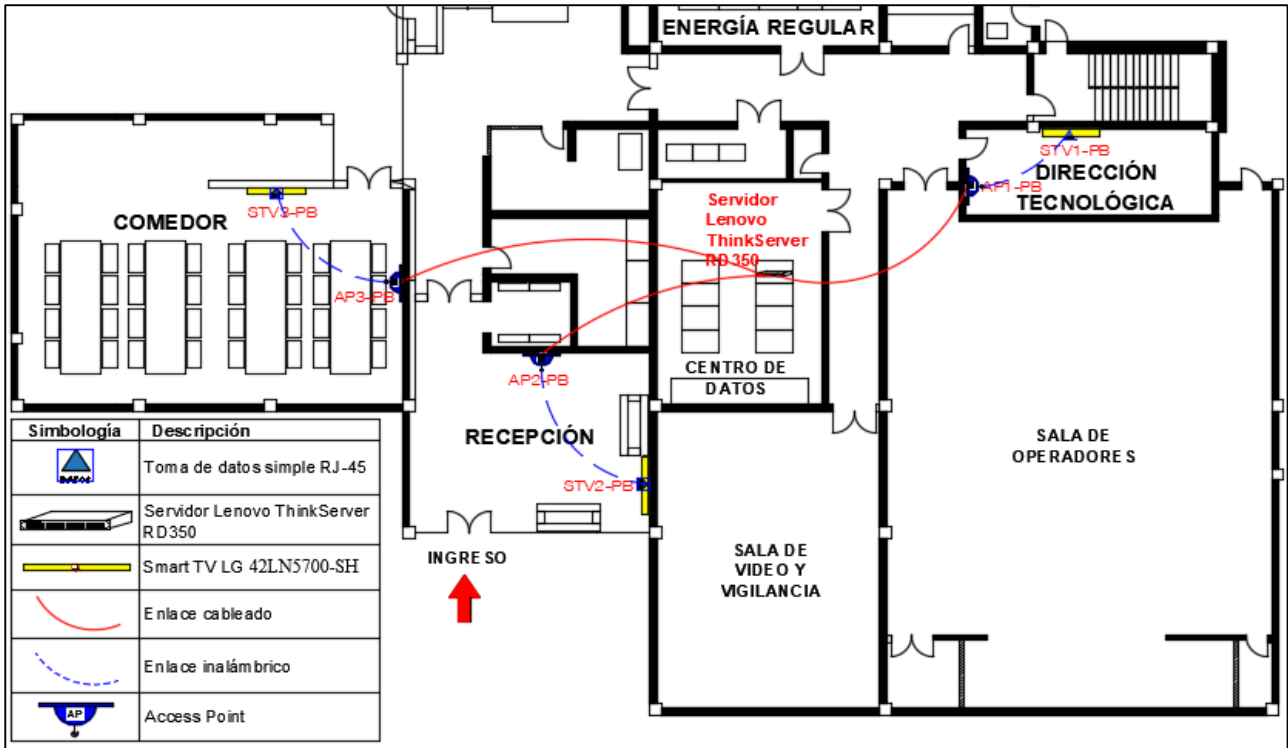


Figura 81.- Diagrama físico del sistema de Infochannel dentro del SIS ECU911 Loja.  
Fuente: Elaboración propia.

### 6.2.4 Diagrama lógico del sistema implementado.

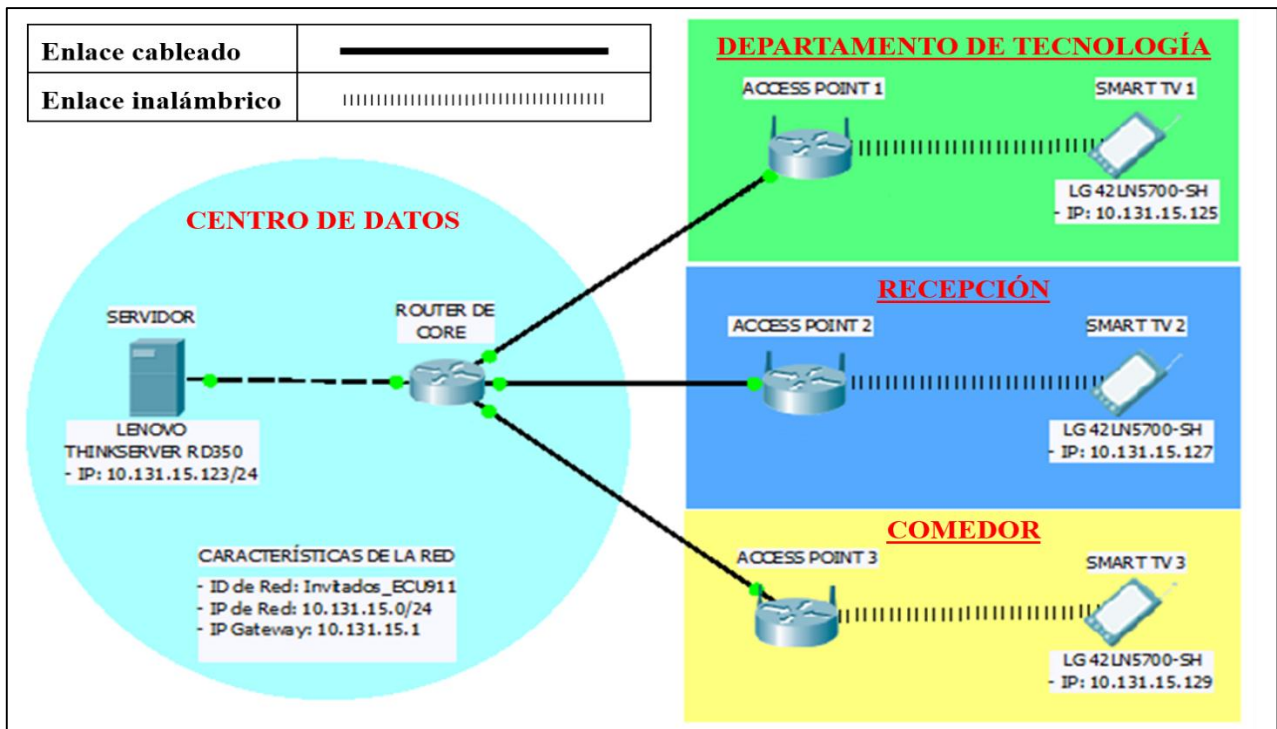
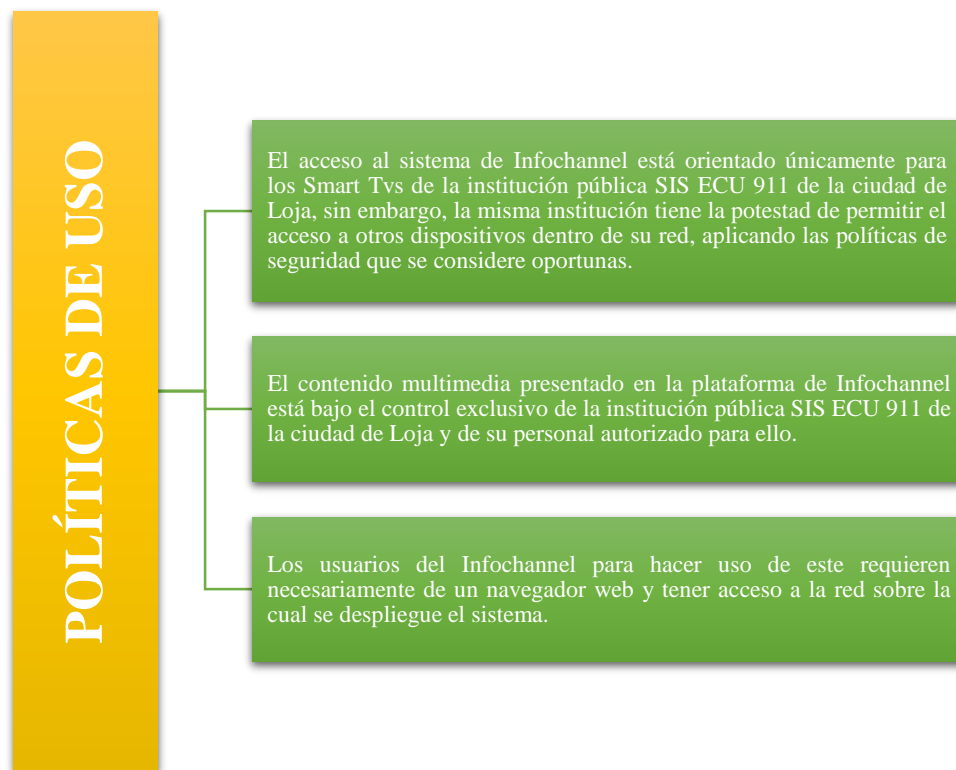


Figura 82.- Diagrama lógico del sistema de Infochannel dentro del SIS ECU911 Loja.  
Fuente: Elaboración propia.

### 6.3 Parámetros para el correcto funcionamiento del sistema.

#### 6.3.1 Políticas de uso.

A continuación, se especifican las políticas que se han establecido para el uso del sistema de Infochannel detallado en este trabajo de tesis:



*Figura 83.- Políticas de uso del sistema Infochannel  
Fuente: Elaboración propia.*

#### 6.3.2 Requisitos para su despliegue:

Para el despliegue del sistema de Infochannel, se requieren ciertos requisitos tanto en software como en hardware. Estos son:

##### 6.3.2.1 Requisitos de software:

Como requisitos del lado del servidor se tiene:

- Se requiere tener instalado el entorno Python (recomendada la versión 3.7.2).
- Instalar el framework Django con una versión compatible con el entorno Python instalado (recomendada la versión 2.7.1).
- Paquete de software XAMPP, con Apache para el despliegue del servidor web y junto a MySQL para la gestión de la base de datos. Si se recurre a un motor de

base de datos diferente, se deberá usar una librería diferente para la comunicación entre la aplicación y la base de datos.

- Librería Python “pymysql” (versión 0.9.3) para la conexión entre la aplicación y MySQL
- Librería Python “django-cleanup” (versión 3.2.0) para la depuración de archivos de la aplicación.

Como requisitos de software para el usuario se tiene:

- Navegador web compatible con los formatos de video seleccionados para la transmisión de los mismos. Se recomienda un navegador web se encuentre lo más actualizado posible.

### 6.3.2.2 Requisitos de Hardware:


Para el diseño y pruebas del correcto funcionamiento del sistema de Infochannel, este ha sido desplegado sobre dos dispositivos que, pese a que sus características no son las de un hardware especializado para servidores, estos han cumplido con las cualidades suficientes y necesarias para el correcto funcionamiento del sistema (ver Tabla 17).

*Tabla 17.- Características referenciales de hardware para el despliegue del sistema de Infochannel.  
Fuente: Elaboración propia.*

| ESPECIFICACIONES TÉCNICAS DEL DISPOSITIVO 1   |                                |
|---|--------------------------------|
|  |                                |
| <b>Tipo de dispositivo</b>  | Computadora portátil           |
| <b>Marca</b>  | DELL                           |
| <b>Modelo</b>   | Inspiron 3421                  |
| <b>Procesador</b>   | Intel Core i3-3217U 1.8 GHz    |
| <b>Memoria RAM</b>  | 4 GB DDR3                      |
| <b>Memoria de almacenamiento</b>  | 500 GB 5400 RPM HDD            |
| ESPECIFICACIONES TÉCNICAS DEL DISPOSITIVO 2   |                                |
|  |                                |
| <b>Tipo de dispositivo</b>  | Computadora portátil           |
| <b>Marca</b>  | ASUS                           |
| <b>Modelo</b>   | F541U                          |
| <b>Procesador</b>   | Intel Core i3-6006 CPU 2.00GHz |
| <b>Memoria RAM</b>  | 12 GB DDR4                     |
| <b>Memoria de almacenamiento</b>  | 1 TB 5400 RPM HDD              |

En contraste con los datos indicados en la Tabla 17, a continuación, en la Tabla 18 se exponen las características físicas del servidor sobre el cual ha sido implementado el sistema de Infochannel, dentro de la red del SIS ECU 911 de la ciudad de Loja.

*Tabla 18.- Especificaciones técnicas del servidor Lenovo Think Server RD350.  
Fuente: Elaboración propia.*

| <b>ESPECIFICACIONES TÉCNICAS</b>  |   |
|---|---|
|  |   |
| <b>Tipo de dispositivo</b>  | Servidor  |
| <b>Marca</b>  | Lenovo  |
| <b>Modelo</b>   | Think Server RD350  |
| <b>Procesador (max)/Caché (max)</b>   | Hasta dos procesadores Intel® Xeon® de 14 núcleos serie E5-2600 v4 / hasta 35 MB por procesador (hasta 105w TDP)  |
| <b>Dimensiones</b>  | 4QL - 4x4.5 pulgadas  |
| <b>Factor de forma</b>  | 1U Rack   |
| <b>Chipset</b>  | Intel C610  |
| <b>Memoria RAM</b>  | <ul style="list-style-type: none"> <li>- Hasta 16 zócalos DDR4 DIMM (8 DIMM por procesador).</li> <li>- Se admiten RDIMM y DIMM de carga reducida (LRDIMM).</li> <li>- Los tipos de memoria no pueden ser entremezclados.</li> <li>- Velocidad de memoria hasta 2400 MHz.</li> </ul>  |
| <b>Memoria RAM máxima</b>   | <ul style="list-style-type: none"> <li>- Con RDIMM: Hasta 512 GB con RDIMM de 16x32 GB y dos procesadores.</li> <li>- Con LRDIMMs: Hasta 1024 GB con 16x64 GB LRDIMMs y dos procesadores.</li> </ul>  |
| <b>Protección de memoria</b>  | - ECC, Patrol Scrubbing, Demand Scrubbing, Sparing, Mirroring, Lockstep Mode.   |
| <b>Memoria de almacenamiento máximo</b>   | <ul style="list-style-type: none"> <li>- Hasta 61 TB con 8x 7.68 TB 2.5" SSD SAS de capacidad empresarial.</li> <li>- Hasta 48 TB con 4x 12 TB 3.5" NL HDD SAS / SATA.</li> <li>- Hasta 9.6 TB con 8x 1.2 TB 2.5" SAS HDD.</li> </ul> Se admite la mezcla de SAS / SATA.  |
| <b>Interfaces de red</b>  | Dos puertos RJ-45 Gigabit Ethernet 1000BASE-T (Intel i210) integrados para el sistema operativo, el Puerto 1 se puede configurar como compartido con la administración de sistemas. Puerto Gigabit Ethernet dedicado para la gestión de sistemas.   |
| <b>Sistema de refrigeración</b>   | Hasta 6 ventiladores redundantes de intercambio fácil (sin intercambio en caliente): <ul style="list-style-type: none"> <li>- Con 1 procesador: 4 ventiladores (3+1 redundante).</li> <li>- Con 2 procesos: 6 ventiladores (5+1 redundante).</li> </ul>   |
| <b>Sistema de administración</b>  | UEFI, ThinkServer System Management (TSM) basado en ASPEED AST2400, controlador de administración de placa base compatible con IPMI 2.0 (BMC), soporte para Lenovo XClarity Administrator. Función de software TSM Premium Upgrade opcional para presencia remota. ThinkServer Deployment Manager. Lenovo Energy Manager XClarity (activado con TSM Premium o licencia por separado). |
| <b>Sistemas operativos soportados</b>   | Microsoft Windows Server, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, VMware ESXi, Citrix XenServer.  |

#### **6.4 Proceso de implementación del Infochannel.**

A continuación, se detalla el proceso para el despliegue de este servicio:

1. En primer lugar, se requiere instalar el entorno Python y el Framework Django. Se recomienda usar la versión 3.7.2 de Python, la cual está disponible en su página oficial [www.python.org](http://www.python.org). La instalación de ambos softwares se detalla en la sección 5.3.2 de esta tesis.
2. Como segundo requisito, es necesario el paquete de software XAMPP para desplegar el servidor sobre la red. Su proceso de instalación se puede consultar en la sección 5.3.1.
3. También se debe instalar la librería “pymysql” para permitir la conexión entre el servidor y la base de datos. Su proceso de instalación se detalla en la sección 5.3.3 del presente documento.
4. Para la tarea de limpieza de archivos se requiere de la librería “django-cleanup”, cuyo proceso de instalación puede consultarse en la bibliografía oficial de Python (<https://pypi.org/project/django-cleanup/0.1.9/>).
5. Dentro del directorio de instalación del software XAMPP, concretamente en la carpeta “htdocs”, se debe instalar el código fuente del programa.
6. A continuación, es necesario crear la base de datos y configurar los parámetros de acceso a ella dentro del código de la aplicación (Consultar la sección 5.3.3).
7. El siguiente paso es realizar las primeras migraciones hacia la base de datos y configurar los usuarios de administración (Revisar la sección 5.3.4).
8. Ahora es necesario configurar el acceso externo a la aplicación, de modo que esta esté disponible para cualquier usuario dentro de la red sobre la cual se implemente. Estas configuraciones pueden consultarse en el apartado 5.3.9 del presente documento.
9. Una vez habilitado el acceso externo, finalmente es momento de inicializar el servidor. Para ello se ha de seguir los pasos indicados en la sección 5.3.9.
10. Con el programa ya funcionando, el siguiente paso es acceder a la interfaz de administración y desde ahí, cargar los videos que se desee mostrar.



## **7. DISCUSIÓN.**

Luego de sintetizar los resultados del diseño e implementación del sistema de video streaming, combinando las diferentes tecnologías web seleccionadas y teniendo como base los diferentes estándares de información consultados, se puede determinar lo siguiente:

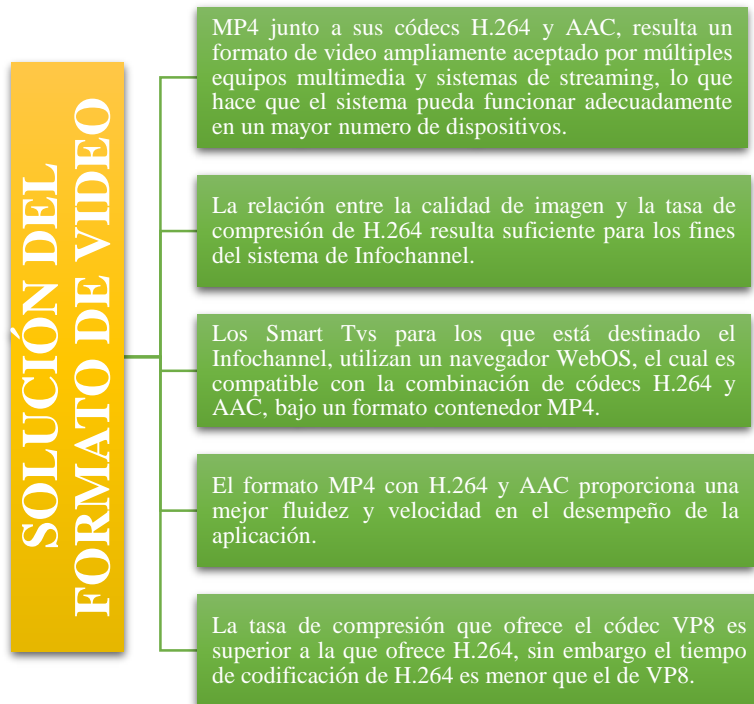
Es cierto que actualmente existen numerosas herramientas a disposición del público que permiten establecer de manera fácil y rápida un servicio de video streaming. Sin embargo, es importante tener en cuenta el tipo de dispositivo al que se pretende orientar dicho sistema o, en otras palabras, el nivel de compatibilidad del servicio. Es por ello que comparando el alcance que pueda tener una aplicación móvil frente a un sistema web, se tuvo como resultado que la opción web es una alternativa que ofrece una mayor compatibilidad con los dispositivos de usuario debido a que su requisito mínimo es tener instalado un cliente o navegador web sin importar el sistema operativo que maneje. En contraste con esto, las aplicaciones móviles requieren un nivel de compatibilidad con el sistema operativo que las aloje. Por otra parte, el uso de un NAS para el diseño del sistema también resultó una alternativa bastante atractiva dada su fácil implementación sobre cualquier red de datos. Sin embargo, uno de los objetivos del proyecto planteado, es el de implementar el sistema de Infochannel aprovechando los recursos de red ya existentes dentro de la institución, por lo que la alternativa de usar el NAS fue descartada de forma inmediata, teniendo en cuenta también su alto costo de adquisición. Otra desventaja de los sistemas de difusión ya creados es que poseen funcionalidades ya predeterminadas por el fabricante o empresa desarrolladora del producto, lo que hace que el sistema no pueda ser personalizado visualmente o de manera funcional; es por ello que se recurrió a un sistema web modular y completamente personalizable, dado que se tiene pleno acceso al código fuente del mismo. Tomando en consideración todo esto, es que se recurrió a un sistema de difusión basado en servicios web el cual pueda ser fácilmente accesible desde un navegador web y que permita la implementación de nuevas funcionalidades a criterio de la institución ECU 911 Loja.

El uso de un servidor web Apache, resultó bastante conveniente para la implementación del Infochannel, ya que, en un inicio, se desconocía por completo sobre qué lenguaje de programación se llevaría a cabo el sistema, y al tener integrado dentro de sus funciones un intérprete de lenguajes, resultó fácil probar diferentes tipos de programación durante

el desarrollo del backend del sistema. Si bien, de acuerdo a sus características investigadas, este tipo de servidor web no tiene un buen desempeño frente a un gran número de solicitudes simultáneas, sin embargo, para el propósito de este tema, resultaron suficientes, ya que su funcionamiento principal está orientado únicamente a 3 Smart Tv dentro de la institución pública SIS ECU 911 de la ciudad de Loja. A futuro, para una implementación del Infochannel dentro de un entorno con mayor demanda, perfectamente podría ser sustituido por el servidor web Nginx.

Todas las herramientas utilizadas para el diseño del Infochannel, debido a que son de software libre y multiplataforma, hacen que el sistema diseñado, teóricamente, pueda ser implementado sobre sistemas operativos como Windows, Mac OS y las diferentes distribuciones de Linux, y por consiguiente sobre diferentes tipos de dispositivos. A lo largo del diseño del Infochannel, este fue validado correctamente sobre los sistemas operativos Windows y las distribuciones Linux de Ubuntu y Debian.

El sistema de Infochannel está orientado a ser desplegado mediante el uso de un navegador web como medio de acceso al servicio de streaming, por lo que la compatibilidad del navegador web utilizado, con los diferentes códecs de video compatibles con el estándar HTML5, es sumamente importante. De acuerdo a los datos obtenidos y especificados en el apartado 6.1.1.1 y 6.1.1.2 de los resultados, los códecs más óptimos, en cuanto a compatibilidad con navegadores web, son el códec H.264 para video, y para audio los códecs AAC y MP3, dado que estos son compatibles con el 100% de los navegadores seleccionados para este estudio. Estos parámetros son importantes a tener en cuenta al momento de establecer hacia qué tipo de dispositivos está orientado el sistema de Infochannel y, puesto que es un sistema ideado para los Smart TV de la institución pública SIS ECU 911 de la ciudad de Loja, se seleccionó un códec que necesariamente debe ser compatible con el Navegador WebOS y que al mismo tiempo resulte compatible con la mayor cantidad de equipos posible. Siendo este el caso y, tomando en cuenta los resultados de la Tabla 16 y en base a los requerimientos exigidos por la institución pública SIS ECU 911 de la ciudad de Loja, se determinó que el códec de video H.264 junto al códec de audio AAC y bajo el encapsulado MP4, resulta el formato de video recomendado y más adecuado para el sistema implementado. A continuación, se exponen los motivos de la elección de este formato de video:



*Figura 84.- Formato de video óptimo para el sistema.  
Fuente: Elaboración propia.*

Por otra parte, el sistema de Infochannel fue desplegado sobre varios dispositivos de usuario para comprobar la compatibilidad de los códecs seleccionados y el funcionamiento de su interfaz de usuario (ver Anexo 1). Con estos dispositivos se comprobó que el sistema puede funcionar sobre cualquier dispositivo que tenga acceso al servicio desde un navegador web que cumpla con los requisitos mínimos expuestos en la sección 6.3.2.1 de este documento.

Más allá del códec de video, también resulta importante la capacidad del navegador web sobre el cual se utiliza el sistema. Unos de los principales inconvenientes que se presentaron durante el despliegue y pruebas del sistema, principalmente sobre los Smart TVs, es que los navegadores web de los mismos poseen características demasiado básicas y poco eficientes para un servicio de streaming de alta calidad. Esto puede verse claramente en la Figura 68, donde se exponen las bajas puntuaciones que reciben los navegadores Opera y el Navegador WebOS de los Smart TVs frente a las altas puntuaciones que reciben los navegadores Google Chrome y Mozilla Firefox de otros dispositivos como un Smartphone y una Computadora. Esto se debe precisamente a que los navegadores nativos de los Smart TVs no vienen debidamente optimizados y de acuerdo al sistema operativo que manejan estos dispositivos, resulta imposible instalar

nuevos y mejores navegadores web. Estos navegadores poco eficientes suponen desventajas y limitaciones para el correcto funcionamiento del sistema, como por ejemplo una baja velocidad de carga de los videos y de transición entre video y video, e incluso la subida de archivos de más de 50 Mb. Como solución a esto se dejó recomendada la instalación de un STB (Set-Top Box) que tenga ya instalado los navegadores Google Chrome o Mozilla Firefox, y que por supuesto puedan actualizarse a nuevas y mejores versiones de dichos navegadores. En el Anexo 4 se encuentra las especificaciones técnicas del STB recomendado y con el cual también se hizo pruebas del sistema.

Dentro de los parámetros para el correcto funcionamiento del sistema, se elaboraron unas políticas de uso, las cuales fueron concebidas junto con el personal técnico encargado de la institución pública SIS ECU 911 de la ciudad de Loja, estableciendo de qué manera se manejará el sistema dentro de la institución.

En cuanto a los requisitos de software, quedan recomendados exclusivamente los programas especificados en la sección 6.3.2.1 de los resultados, puesto que han sabido dar un resultado satisfactorio de acuerdo a la evaluación final del sistema hecha por el personal del SIS ECU 911 Loja y, concretamente por el departamento de Comunicación Social, dado que ellos serán los encargados de administrar el Infochannel.

Los requisitos de hardware necesarios para el despliegue de este sistema no han sido cuantificados mediante alguna herramienta matemática o de software, por lo que, con los datos expuestos en la tabla 17, se pretende exponer un modelo referencial de hardware para el correcto funcionamiento del sistema de Infochannel, no obstante, el hardware usado para la implementación final de este proyecto se expone en la Tabla 18.

El sistema de Infochannel posee dos interfaces bien diferenciadas y programadas de acuerdo a los objetivos planteados al principio de este proyecto. Una de estas interfaces es la de control del sistema, la cual puede verse en la Figura 76 de los resultados. Esta interfaz posee un sistema de acceso por usuario y contraseña, lo cual agrega seguridad administrativa al sistema. Desde la interfaz de control es posible realizar la carga, modificación y eliminación del contenido multimedia que podrá visualizarse dentro del Infochannel. También es posible realizar la adhesión y eliminación de los usuarios administrativos del sistema. Como se puede ver en la Figura 77, se ha agregado la función

de búsqueda por nombre y un filtro de archivos para encontrar de forma más rápida los archivos deseados.

En cuanto a la interfaz de usuario, la cual se puede visualizar en la Figura 78, está completamente automatizada para actualizar su contenido multimedia acorde a los cambios que se realice desde la interfaz de control, aunque también posee botones para el control de la reproducción. Los eventos de actualización del contenido, control de la reproducción y lectura asíncrona de la base de datos, se configuraron gracias a Ajax, mejorando así la fluidez, velocidad e interactividad de la interfaz de usuario. La funcionalidad de esta interfaz como canal de difusión de los videos institucionales internos dentro de la institución SIS ECU 911 de la ciudad de Loja quedó demostrada y completamente funcional dentro de sus instalaciones.

En la Figura 78 se puede ver la pantalla de “Inicio” del sistema, en el cual se ve el enlace de acceso al servicio de “Infochannel”. También se puede ver un segundo servicio agregado a petición de la institución denominado “Repositorio”. Esta interfaz está orientada a la interacción con el usuario y sirve para presentar y reproducir una lista de videos desde una base de datos mostrando una visualización previa y los títulos de los videos.

También como parte de los resultados del sistema de Infochannel, en la Figura 81 se puede ver el Diagrama Físico del sistema, en el cual se indica la localización exacta de los Smart TVs y el servidor físico del sistema dentro de la institución pública ECU 911 Loja. En la Figura 82, se presenta el diagrama lógico de la red sobre la que funciona el sistema, indicando información como el nombre de la red para acceder al Infochannel, la dirección de red, la puerta de enlace o Gateway, la máscara de subred, y las direcciones IP de los diferentes dispositivos conectados al servicio (servidor y clientes).

El manejo del sistema de Infochannel, tanto para su interfaz de usuario como para la interfaz de administración, ha sido diseñado para que sea lo más intuitiva posible, asegurando así su correcto uso. Sin embargo, a petición de la institución pública ECU 911 de la ciudad de Loja, en los anexos 6, 7 y 8 se adjuntan un conjunto de Guías de funcionamiento del sistema, para el usuario final, para el administrador del sistema y también para personal técnico encargado del mantenimiento y configuración del sistema, de modo que se entienda cada una de las opciones que ofrece el sistema diseñado.

Si bien el sistema diseñado posee las características funcionales planteadas al inicio del presente proyecto de tesis, se ha caído en cuenta acerca del potencial de uso que pudiera tener no sólo como un Infochannel dentro de la institución pública SIS ECU 911 de la ciudad de Loja, si no que al ser un sistema abierto y basado en software libre, puede ser implementado sobre cualquier red corporativa o del hogar, prestando el servicio de streaming de video para los clientes de una determinada empresa o institución, o cómo repositorio multimedia para el hogar. Por supuesto que estos usos requerirían de ciertas modificaciones del código fuente, sin embargo, gracias a la versatilidad de las herramientas empleadas en el sistema y a la gran cantidad de información que se tiene acerca de estas, no es algo que se considere imposible.

Teniendo en cuenta los criterios expuestos en esta discusión y los resultados obtenidos del funcionamiento del propio servicio, se debe mencionar que se cumplió con todos los requerimientos y metas establecidas por la institución pública SIS ECU 911 de la ciudad de Loja, tal y como se demuestra con la certificación extendida por la propia institución y que se adjunta el Anexo 3.

## 8. CONCLUSIONES.

- Las herramientas base investigadas y utilizadas para el desarrollo del backend del sistema de Infochannel, tales como Python, Apache, MySQL y Django agregaron eficiencia al sistema, dado los resultados obtenidos, y versatilidad, debido a sus características de software libre y multiplataforma, lo que hacen que el sistema de Infochannel pueda ser desplegado fácilmente y sin problemas sobre sistemas operativos como Windows y distribuciones Linux como Ubuntu o Debian.
- De entre los formatos libres compatibles con el estándar HTML, el formato WEBM junto a los códecs VP8 y Vorbis, ofrece mejores resultados en cuanto a calidad de imagen, tasa de compresión, rendimiento y latencia, frente a los demás formatos que ofrece el estándar. Sin embargo, su baja disponibilidad y compatibilidad con diferentes tipos de dispositivos hacen que, el formato MP4 junto a los códecs H.264 y AAC, sea la mejor opción posible, ya que posee características de calidad, compresión, latencia y rendimiento aceptables y suficientes para el funcionamiento del sistema y, además, asegura su compatibilidad con una gran cantidad de equipos y, por ende, la compatibilidad de la propia aplicación web.
- Las interfaces de administración y presentación del Infochannel, así como sus funcionalidades, fueron desarrolladas y diseñadas exitosamente en base a los requerimientos emitidos por parte de la institución pública SIS ECU 911 de la ciudad de Loja, facilitándoles así la administración y uso del sistema.
- Los parámetros recomendados para el correcto funcionamiento del sistema han sido determinados en base al software utilizado para el desarrollo del Infochannel y a las pruebas experimentales realizadas dentro de la red interna del edificio de la institución pública SIS ECU 911 de la ciudad de Loja y con los diferentes dispositivos de usuario expuestos en el Anexo 1.
- Con el sistema en funcionamiento se ha cumplido con los objetivos propuestos en la investigación y la aprobación emitida mediante certificación de validez otorgado por la institución en la que se intervino.

## 9. RECOMENDACIONES.

- Se recomienda el uso de un Framework que supla las exigencias y necesidades del sistema requerido, de manera que se pueda agilizar el proceso de diseño y desarrollo de la aplicación web.
- Los scripts desarrollados en los diferentes entornos de programación y utilizados en el desarrollo del sistema de Infochannel requieren de un nivel de conocimiento medio de programación en Python, HTML5, CSS y JavaScript, por lo que se recomienda incrementar las habilidades en programación para un mejor entendimiento de esta propuesta.
- Para una mejor experiencia de usuario, se recomienda el uso de los navegadores web Google Chrome o Mozilla Firefox para acceder al servicio del Infochannel, y que estén actualizados a sus versiones más recientes.
- Para un correcto despliegue y funcionamiento del sistema de Infochannel se recomienda cumplir con los requisitos mínimos expuestos en la sección 6.3.2 de los resultados.
- Como trabajo a futuro, para conocer el grado de satisfacción de los usuarios del Infochannel implementado dentro de la institución SIS ECU 911 de la ciudad de Loja, se recomienda realizar un pequeño análisis de la experiencia de usuario, pudiendo así determinar posibles modificaciones y mejoras para el sistema.



## 10. BIBLIOGRAFÍA.

- ALEGSA. (n.d.). Definición de Front-end. Retrieved from <http://www.alegsa.com.ar/Dic/front-end.php>
- Andrews, P. (2015). *Magic Quadrant for Enterprise Video Content Management*.
- Bammidi, R. (2017). *Remote Monitoring and Control by Embedded Database Design and Web Server Implementation Using SQLite database and Boa Web Server*. 17, 1–5.
- Bárcena, A., Prado, A., Cimoli, M., & Pérez, R. (2015). *La nueva revolución digital*.
- Bazán, P. (2017). *Aplicaciones, servicios y procesos distribuidos*. 125.
- Cajamarca Sacta, S. E., & Medina Pesántes, E. A. (2011). *Estudio técnico sobre la implementación de una estación de radiodifusión por internet para la Universidad Politécnica Salesiana sede Cuenca*.
- Camps Paré, R., Pérez Mora, Ó., Costal Costa, D., Carme Martín, E., Gibert Ginestà, M., & Casillas Santillán, L. A. (n.d.). *Bases de datos*. Retrieved from [www.glo.org.mx](http://www.glo.org.mx)
- Camuñas, M. (n.d.). Plataformas y herramientas para emitir en streaming. Retrieved August 18, 2019, from <https://www.maxcf.es/emitir-en-streaming/>
- Cho, C. S., Kim, Y., Shin, H. S., & Choi, B. H. (2010). *Implementation of a fixed-point MPEG-4 scalable lossless coding encoder*. 1–5.
- Colores Vargas, J. M. (2010). *ESTUDIO COMPARATIVO DE SISTEMAS DE DIFUSION DE VIDEO AFLUENTE*.
- de la Torre, A. (2012). Lenguajes del lado servido o cliente. Retrieved from [http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html)
- Dimas, J., & Castillo, L. (2016). *HTML5, CSS Y JAVASCRIPT Crea tu web y apps con el estándar de desarrollo*. Retrieved from [www.conlicencia.com](http://www.conlicencia.com);
- ECU 911. (2017). *Informe de gestión del servicio integrado de seguridad ECU 911. Quito: Servicio Integrado de Seguridad ECU 911*.
- Enríquez, A., Maldonado, J., Nakamura, Y., & Nogueron, G. (2017). *MySQL ¿Qué es MySQL?* 2. Retrieved from <http://www.gridmorelos.uaem.mx/~mcruz/cursos/miic/MySQL.pdf>
- Fernández Moreno, D. (2006). *Diseño e implementación de una aplicación web para la gestión y control de visitas a centros educativos*. Universidad Politécnica de Madrid.
- Fitzpatrick, B. (n.d.). Google Chicago Job Openings. Retrieved February 19, 2019, from 2018 website: <https://web.archive.org/web/20120415021232/http://www.google.com/intl/en/jobs/>

uslocations/chicago/index.html

Francés López, J. M., & Palau Arnau, V. (2012). *Portal Web para la gestión de Ayuntamientos*. Retrieved from <https://riunet.upv.es/handle/10251/17587>

González Duque, R. (2017). Python para todos. *Web Book*, 160. <https://doi.org/10.1017/CBO9781107415324.004>

Growing, H. E., For, M., Storage, N., Result, I. S. a, The, O. F., Demand, E., ... Hardware, T. N. (2012). Network Attached Storage Architecture. *Communications of the ACM*, 43(11), 37–45. <https://doi.org/10.1145/353360.353362>

Gutiérrez, J. J. (n.d.). *¿Qué es un framework web?* Retrieved from [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf)

Holovaty, A., & Kaplan Mosss, J. (2015). *La guía definitiva de Django: Desarrolla aplicaciones web de forma rápida y sencilla*. 598. Retrieved from <http://github.com/saulgm/djangobook.com>

Hosting, O. (2015). Servidores Dedicados. Retrieved February 18, 2019, from [https://okhosting.com/blog/servidores-dedicados/#Servidores\\_Paginas\\_web](https://okhosting.com/blog/servidores-dedicados/#Servidores_Paginas_web)

Huang, Z., Xia, C., Sun, B., & Xue, H. (2015). Analyzing and summarizing the web server detection technology based on HTTP. *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, 2015–Novem*, 1042–1045. <https://doi.org/10.1109/ICSESS.2015.7339231>

Incencio Piñeiro, G. S. (2014). *Sistema informático para la evaluación de atributos de calidad en componentes biométricos*. 3, 19–35.

Iruela, J. (n.d.). Los gestores de bases de datos más usados. Retrieved from <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>

Junta de Andalucía. (n.d.). 2 . *MS Internet Information Services*. Retrieved from [http://www.juntadeandalucia.es/empleo/recursos/material\\_didactico/especialidades/materialdidactico\\_administrador\\_servidores/Content/3-servicios\\_web/2-IIS.pdf](http://www.juntadeandalucia.es/empleo/recursos/material_didactico/especialidades/materialdidactico_administrador_servidores/Content/3-servicios_web/2-IIS.pdf)

Kashyap, A., Member, G. S., Bing, B., & Member, S. (2010). *Efficient HD Video Streaming Over the Internet*. 272–275.

López Herreros, A. (2014). “ *Estudio de emisión de vídeo sobre HTML5* .” Retrieved from <https://riunet.upv.es/bitstream/handle/10251/44895/memoria.pdf?sequence=1>

Mendoza Vázquez, I. I. (n.d.). *Definición de un Framework para aplicaciones Web con navegación sensible a concerns*. Retrieved from [http://sedici.unlp.edu.ar/bitstream/handle/10915/4192/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/4192/Documento_completo.pdf?sequence=1)

NETCRAFT. (n.d.). October 2018 Web Server Survey. Retrieved February 10, 2019,

from 2018 website: <https://news.netcraft.com/archives/2018/10/29/october-2018-web-server-survey.html>

Nginx, A. (2010). *Nginx.HTTP.Server.pdf*.

Pascual Torres, J. (2016). *Reconstrucción del front-end y back-end de un juez automático de teoría de la computación*. Retrieved from [http://img03.en25.com/Web/FCBusinessIntelligenceLtd/%7B3935e417-7a97-4941-b40f-388e37c4ac95%7D\\_2792\\_21SEP15\\_brochure.pdf](http://img03.en25.com/Web/FCBusinessIntelligenceLtd/%7B3935e417-7a97-4941-b40f-388e37c4ac95%7D_2792_21SEP15_brochure.pdf)

Prendes, I. M. (2011). *Tecnologías Multimedia-Tecnología Streaming*.

PyCharm documentation. (2017). *PyCharm Help. Meet PyCharm*.

Sánchez, J. (n.d.). Apuntes de Sistemas de Gestión de Base de Datos. Retrieved March 10, 2019, from <http://www.jorgesanchez.net/>

SISTEMA MANEJADOR DE BASES DE DATOS Base de Datos : (n.d.). Retrieved from <https://homeworkdatabase.wordpress.com/2015/06/27/sghbd-o-sghbd-sistema-manejador-de-base-de-datos/>

Systems, C. (2010). *HTTP Apache Web Server*. 1–9. <https://doi.org/10.1021/es001616r>

Tanenbaum, A. S., & Wetherall, D. J. (2011). Redes de Computadoras. In *Redes de computadoras*. <https://doi.org/10.1017/CBO9781107415324.004>

Tecnología&Informática. (n.d.). ¿Qué es un servidor NAS? ¿Cómo construir uno? - Tecnología + Informática. Retrieved August 18, 2019, from <https://tecnologia-informatica.com/que-es-servidor-nas-construir/>

Uribe Sosa, D. S. (2015). *Caracterización de arquitecturas de sistemas de computo servidores*. Universidad Autónoma de Ciudad de Juárez.

WebM, P. (n.d.-a). The WebM Open Media Project Blog: Introducing WebM, an open web media project. Retrieved February 25, 2019, from <http://blog.webmproject.org/2010/05/introducing-webm-open-web-media-project.html>

WebM, P. (n.d.-b). The WebM Project | About WebM. Retrieved February 25, 2019, from <https://www.webmproject.org/about/>

Western Digital Corporation. (n.d.). Network Attached Storage (NAS). Retrieved August 18, 2019, from <https://www.wd.com/es-es/solutions/what-is-network-attached-storage-nas.html>

Yan, X., Yang, L., Lan, S., & Tong, X. (2012). Application of HTML5 multimedia. *Proceedings - 2012 International Conference on Computer Science and Information Processing, CSIP 2012*, 871–874. <https://doi.org/10.1109/CSIP.2012.6308992>

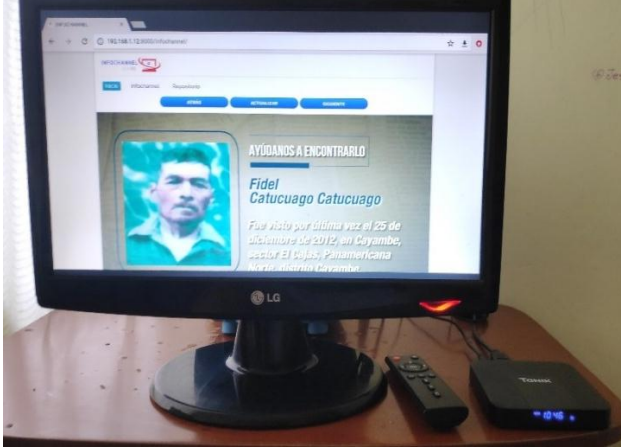
Yáñez Izquierdo, A. (2011). *Formatos de audio y video: códecs*. 1–23.

Yoon, Y., Kim, M., Lee, S., Lee, B., Hyun, S. J., & Lee, K. (2012). Performance analysis of H.264/AVC, H.264/SVC, and VP8 over IEEE 802.11 wireless networks. *2012 IEEE Symposium on Computers and Communications (ISCC)*, 000151–000156. <https://doi.org/10.1109/ISCC.2012.6249284>

## **11.ANEXOS.**

## ANEXO 1: FUNCIONAMIENTO DEL INFOCHANNEL SOBRE DIFERENTES PLATAFORMAS DE USUARIO.

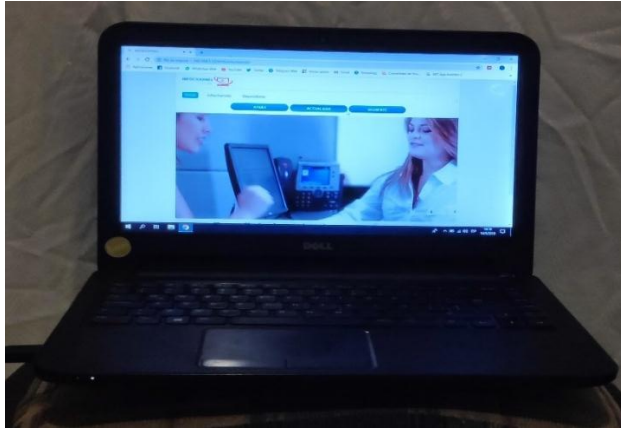
**STB Tanix TX3 Mini**



**Smartwatch LEMFO LEM 1**



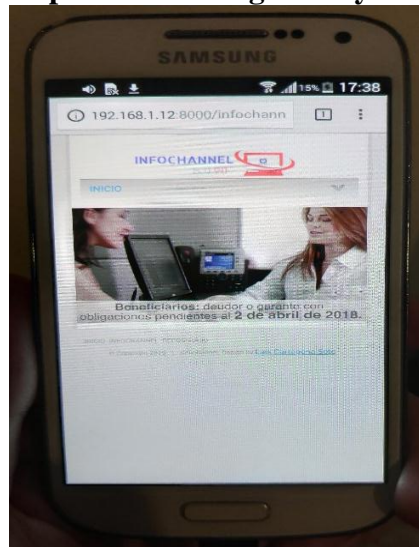
**Computadora portátil DELL Inspiron 3421**



**Smart TV TCL L32D2730**

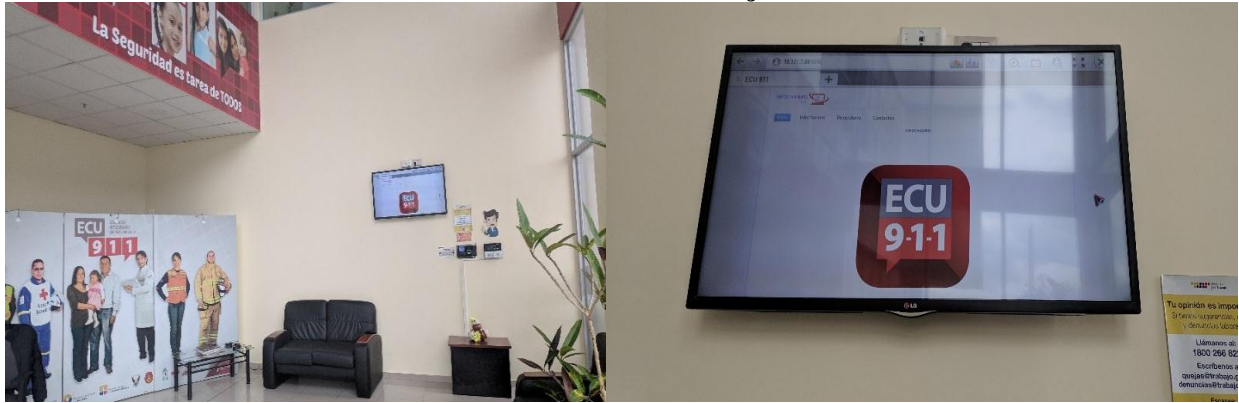


**Smartphone Samsung Galaxy S3 mini**

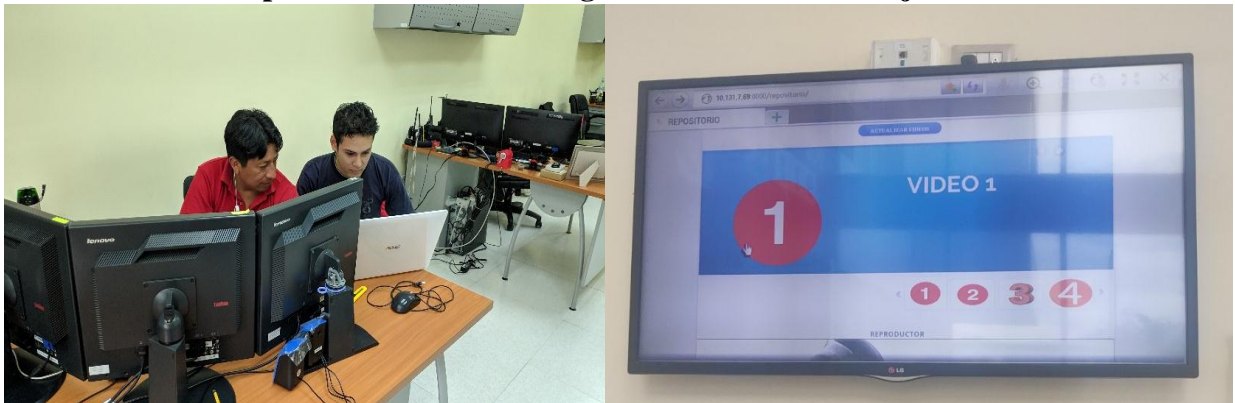


## ANEXO 2: IMPLEMENTACIÓN DEL INFOCHANNEL SOBRE LA RED LOCAL DEL SIS ECU 911 DE LA CIUDAD DE LOJA.

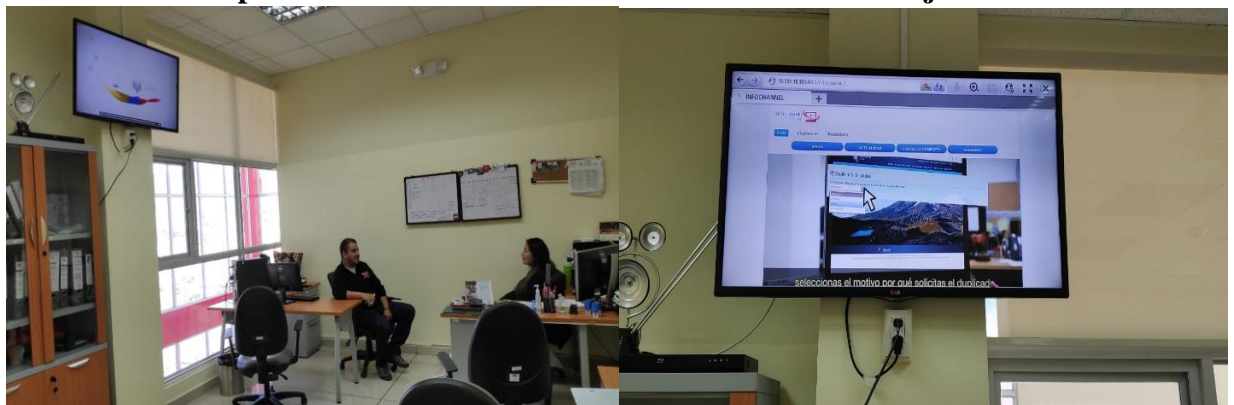
**Entrada del SIS ECU 911 Loja**



**Departamento de Tecnología del SIS ECU 911 Loja**



**Departamento de Comunicación del SIS ECU 911 Loja**



**ANEXO 3: CERTIFICACIÓN DEL SISTEMA DE INFOCHANNEL POR  
PARTE DEL SIS ECU 911 DE LA CIUDAD DE LOJA.**



Loja, 28 de mayo del 2019

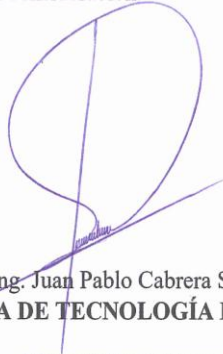
Ing.  
Juan Pablo Cabrera Samaniego  
**ESPECIALISTA DE TECNOLOGÍA LOCAL ECU-911 LOJA**

**CERTIFICA:**

Que el Sr. **Luis Alberto Cartagena Soto**, con cédula **1900523117** egresado de la carrera de Ingeniería en Electrónica y Telecomunicaciones de la Universidad Nacional de Loja, realizó el desarrollo del proyecto de tesis denominado “IMPLEMENTACIÓN DE UN INFOCHANNEL UTILIZANDO SERVICIOS WEB PARA LA VISUALIZACIÓN DE CONTENIDO MULTIMEDIA PARA EL SIS ECU 911 DE LA CIUDAD DE LOJA”.

Las actividades fueron ejecutadas de manera satisfactoria cumpliendo con los objetivos específicos propuestos para el proyecto de tesis y dentro de los lineamientos establecidos, bajo la supervisión del personal técnico encargado, desde el 8 de marzo del 2018 al 20 de mayo del 2019, como se detalla en el anexo.

Para fines pertinentes de mi consideración.




Ing. Juan Pablo Cabrera Samaniego  
**ESPECIALISTA DE TECNOLOGÍA LOCAL ECU-911 LOJA**



**Anexo.**

| <b>Semanas</b>                               | <b>Actividades</b>  |
|--|---|
| 8 de marzo / 30 de abril del 2018            | <ul style="list-style-type: none"> <li>- Selección del sistema operativo para la implementación del servidor multimedia.</li> <li>- Estudio y comparación de servidores de aplicaciones de software y códecs de video.</li> <li>- Selección del servidor de aplicaciones y códecs de video para el sistema multimedia.</li> </ul> |
| 31 de abril / 25 de mayo del 2018            | <ul style="list-style-type: none"> <li>- Recopilación de información acerca del uso de servicios web necesarios para la implementación del sistema.</li> <li>- Selección de los servicios web requeridos para el sistema.</li> </ul>  |
| 28 de mayo/ 6 de septiembre del 2018         | <ul style="list-style-type: none"> <li>- Capacitación personal en los lenguajes de programación Python, HTML5, CSS y JavaScript.</li> <li>- Capacitación personal en el Framework Django.</li> <li>- Primeras configuraciones del backend del sistema y comunicación entre la base de datos y el sistema en proceso.</li> </ul>   |
| 7 de septiembre / 2 de noviembre del 2018    | <ul style="list-style-type: none"> <li>- Desarrollo de la interfaz del backend del sistema y la interfaz de administración.</li> </ul>  |
| 5 de noviembre de 2018 / 4 de enero del 2019 | <ul style="list-style-type: none"> <li>- Desarrollo y primeras pruebas de la interfaz de usuario y estructura de la misma.</li> </ul>   |
| 7 de enero / 14 de febrero del 2019          | <ul style="list-style-type: none"> <li>- Desarrollo y corrección de errores de la interfaz de usuario.</li> </ul>   |
| 18 de febrero / 28 de marzo del 2019         | <ul style="list-style-type: none"> <li>- Pruebas de funcionamiento sobre la red local del SIS ECU 911 de la ciudad de Loja.</li> </ul>  |
| 29 de marzo / 5 de abril del 2019            | <ul style="list-style-type: none"> <li>- Selección de los códecs compatibles para la visualización de videos sobre los Smart TVs del SIS ECU 911 de la ciudad de Loja.</li> </ul>   |
| 8 de abril / 1 de mayo del 2019              | <ul style="list-style-type: none"> <li>- Pruebas de funcionamiento para la corrección y optimización del sistema desarrollado.</li> <li>- Pruebas de Campo y validación de los resultados.</li> </ul>   |
| 17 de mayo del 2019                          | <ul style="list-style-type: none"> <li>- Presentación del sistema al personal encargado del SIS ECU 911 de la ciudad de Loja.</li> <li>- Validación del Infochannel sobre un sistema operativo Linux.</li> </ul>  |
| 20 de mayo del 2019                          | <ul style="list-style-type: none"> <li>- Entrega del proyecto denominado: "IMPLEMENTACIÓN DE UN INFOCHANNEL UTILIZANDO SERVICIOS WEB PARA LA VISUALIZACIÓN DE CONTENIDO MULTIMEDIA PARA EL SIS ECU 911 DE LA CIUDAD DE LOJA".</li> </ul>  |

**ANEXO 4: CARACTERÍSTICAS TÉCNICAS DEL STB RECOMENDADO  
PARA SU INTALACIÓN EN EL SISTEMA DE INFOCHANNEL.**

| <b>ESPECIFICACIONES TÉCNICAS</b>   |   |
|--|---|
|  |   |
| <b>Características principales</b>   |   |
| <b>Marca</b>   | Tanix   |
| <b>Modelo</b>  | TX3 Mini  |
| <b>Sistema Operativo</b>   | Android 7.1   |
| <b>CPU</b>   | S905W up to 2.0 GHz, Quad core ARM Cortex-A53   |
| <b>GPU</b>   | Mali-450 penta-core, up to 750MHz+(DVFS)  |
| <b>RAM</b>   | 1GB/2GB DDR3  |
| <b>Almacenamiento</b>  | 8GB/16GB eMMC   |
| <b>Wifi</b>  | IEEE 802.11 b/g/n   |
| <b>SPDIF</b>   | Coaxial (Optical)   |
| <b>Especificaciones de interfaz</b>  |   |
| <b>Salida HDMI</b>   | HDMI2.0 to support maximum 4K x 2K display  |
| <b>Puerto USB</b>  | 2 High speed USB 2.0, support U DISK and USB HDD  |
| <b>Lector de tarjetas</b>  | TF Card reader  |
| <b>Salida de datos</b>   | SPDIF   |
| <b>Salida AV</b>   | AV (3 in 1) port  |
| <b>LAN</b>   | RJ45 10M/100M   |
| <b>Suministro eléctrico</b>  | DC Jack 5V/2A   |
| <b>Especificaciones de software</b>  |   |
| <b>KODI</b>  | Kodi 17.3   |
| <b>Formatos decodificador</b>  | 4K, MPEG1/2/4, H.264, HD AVC/VC-1, RM/RMVB, Xvid/DivX3/4/5/6, RealVideo8/9/10                                       |
| <b>Formatos de video</b>   | 4K@30fps, AVI, H.264, VC-1, MPEG-2, MPEG-4, DIVD/DIVX, Real8/9/10, RM, RMVB, PMP, FLV, MP4, M4V, VOB, WMV, 3GP, MKV |
| <b>Formatos de audio</b>   | MP1, MP2, MP3, WMA, OGG, AAC, M4A, FLAC, APE, AMR, RA, WAV  |
| <b>Formatos de imagen</b>  | JPEG/BMP/GIF/PNG/TIFF   |
| <b>DLNA</b>  | DLNA files sharing  |
| <b>Idioma</b>  | English/French/German/Spanish/Italian etc. 24 languages   |
| <b>Funciones de red</b>  | Skype, Picasa, Youtube, Flickr, Facebook, Online movies, Google Play Store, etc.                                    |

## ANEXO 5: GUÍA DE USUARIO DEL INFOCHANNEL.

Las interfaces de usuario del sistema de Infochannel diseñado son 3, la interfaz de inicio, la del Infochannel y la del Repositorio de archivos, y presentan las siguientes características:

- **Interfaz de inicio:**



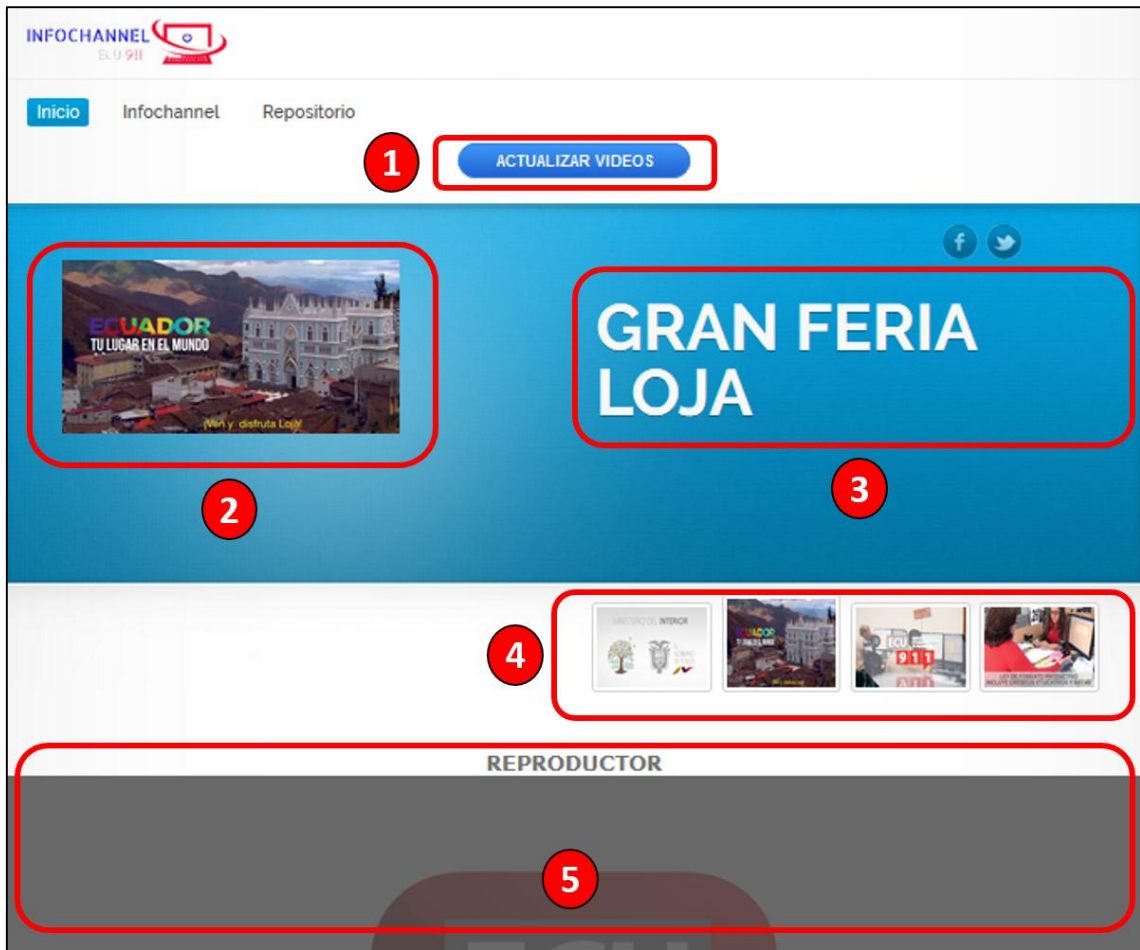
1. Panel de navegación del Infochannel. Permite moverse entre las diferentes interfaces de usuario. Este encabezado se encuentra presente de la misma forma en todas las interfaces de usuario.
2. Es un segundo panel de navegación que, al igual que el panel superior, permite navegar entre las diferentes interfaces de usuario y se encuentra presente en todas ellas.
3. Enlaces de acceso al servicio de Infochannel y al Repositorio de archivos. El Infochannel se trata de un canal automatizado de sólo visualización de videos institucionales diseñado para dispositivos que no están destinados para la interacción con usuarios, como por ejemplo un Smart TV. El Repositorio, por otro lado, está diseñado para alojar una lista de reproducción de videos, los cuales pueden ser visualizados a gusto del usuario, ya que dispone de un panel de navegación y pre-visualización de los videos.
4. Se trata de una rúbrica por parte del diseñador del sistema, que da acceso a su información de contacto.

- **Interfaz del Infochannel:**



1. Botones de control de la reproducción del video. La reproducción del Infochannel se encuentra automatizada para su actualización y reproducción automática para un dispositivo que no requiere de la interacción de usuarios, sin embargo, de querer controlar la reproducción desde algún dispositivo, como por ejemplo desde una computadora o un Smartphone, se agregaron los botones de control de los eventos como son:
  - “**ATRÁS**”: Retrocede al video previo dentro de la lista de reproducción.
  - “**ACTUALIZAR**”: actualiza la lista de reproducción si hubiere cambios en la lista de archivos a reproducirse.
  - “**PANTALLA COMPLETA**”: Maximiza la vista del video a pantalla completa.
  - “**SIGUIENTE**”: adelanta al siguiente video en la lista de reproducción.
2. Cuadro de reproducción del video. Sus dimensiones son fijas dentro del navegador usado.

- **Interfaz del Repositorio:**

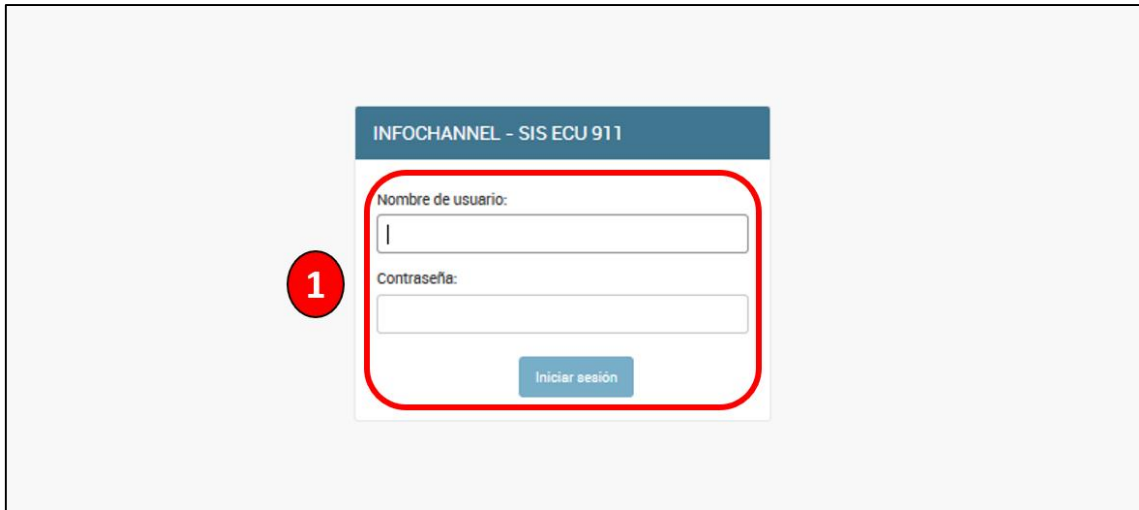


1. Botón de actualizar: sirve para actualizar la lista de reproducción de videos, si se hubiere agregado archivos nuevos. El botón esta puesto si el usuario desea realizar esta actualización, sin embargo, también existe la función de actualización automática al finalizar cualquier video.
2. Foto del video seleccionado para su reproducción. Para visualizar el video deseado se debe hacer clic sobre la foto indicada.
3. Título del video seleccionado.
4. Panel de navegación entre los videos dentro de la lista de reproducción. Contienen una pequeña imagen de previsualización de los videos.
5. Reproductor HTML5 para el video seleccionado.

## ANEXO 6: GUÍA DE ADMINISTRACIÓN DEL INFOCHANNEL.

La interfaz de administración del sistema de Infochannel presenta las siguientes interfaces, cada una con sus respectivas características:

- **Interfaz de acceso al Sistema de administración:**



1. Es la interfaz de acceso como administrador del Sistema. Para acceder se requiere de un usuario y contraseña previamente registrados por línea de comandos o desde dentro de la interfaz de administración.

Para acceder a la interfaz de administración de debe acceder desde la misma dirección de inicio del Infochannel, agregando “/admin”, a continuación se muestra un ejemplo:



- **Interfaz de inicio del Sistema de administración del Infochannel:**

The screenshot shows the administrative interface for 'INFOCHANNEL - SIS ECU 911'. At the top, there is a header with the title and a welcome message: 'BIENVENIDO/A, LCS. VER EL SITIO / CAMBIAR CONTRASEÑA / TERMINAR SESIÓN'. Below this, the main content area is titled 'Sitio administrativo' and is divided into two main sections. The left section, 'AUTENTICACIÓN Y AUTORIZACIÓN', contains two rows: 'Grupos' and 'Usuarios', each with '+ Añadir' and 'Modificar' buttons. The right section, 'VIDEOS', contains two rows: 'Repositorios' and 'Videos', each with '+ Añadir' and 'Modificar' buttons. On the right side, there is a sidebar titled 'Acciones recientes' with a sub-section 'Mis acciones' listing recent actions with status indicators (red 'x' for deletion, green '+' for addition) and the word 'Video' below each action name.

1. Encabezado de bienvenida y cierre de la sesión de administración.
2. Acceso a la vista para agregar nuevos usuarios de administración.
3. Acceso a las vistas para agregar videos al Infochannel o al Repositorio de videos.
4. Panel de visualización de cambios hechos por el administrador.



- **Interfaz de control y adhesión de videos al Infochannel:**

The screenshot shows the 'INFOCHANNEL - SIS ECU 911' interface. At the top, there is a navigation bar with 'Inicio' and 'Videos' (1). Below it, the main content area is titled 'Escoja video a modificar'. It features a search bar (2) with a 'Buscar' button, an action dropdown menu (3) with 'Ir' and 'seleccionados 0 de 3', and a table of videos (5). The table has columns for 'TITULO', 'VIDEO SRC', 'INGRESADO', and 'MODIFICADO'. It lists three videos with their respective titles, sources, and dates. To the right, there is a 'FILTRO' sidebar (6) with options for 'Por ingresado' and 'Por modificado', each with sub-options like 'Cualquier fecha', 'Hoy', 'Últimos 7 días', 'Este mes', and 'Este año'. A 'Añadir Video' button (4) is located at the top right, and a 'Grabar' button is at the bottom right.

1. Panel de navegación entre las interfaces de administración. Este es común a todas las interfaces dentro del sistema de administración.
2. Barra de búsqueda de archivos.
3. Opción para eliminar archivos de video del sistema.
4. Opción para agregar nuevos archivos al Infochannel.
5. Videos ya subidos al sistema.
6. Filtro de archivos agregados al sistema. Pueden filtrarse por fecha de ingreso y por fecha de modificación de los archivos. Su finalidad es agilizar la búsqueda de archivos dentro del sistema.

Los paneles de control, tanto del Infochannel como del Repositorio son exactamente los mismos, sin embargo pueden cambiar los campos que se deben rellenar al momento de agregar un nuevo archivo al sistema.

- **Interfaces de subida de archivos al sistema de Infochannel.**

INFOCHANNEL - SIS ECU 911  
BIENVENIDO/A, LCS. VER EL SITIO / CAMBIAR CONTRASEÑA / TERMINAR SESIÓN

Inicio > Vídeos > Vídeos > Añadir video

Añadir video

Titulo:

Video src:  No se ha seleccionado ningun archivo.

1. Campos de información del archivo que se desea agregar al sistema. Todos los campos mostrados son obligatorios y deben ser debidamente rellenos, de lo contrario, el sistema no permitirá agregar nuevos archivos. El campo del título del archivo está fijado a un máximo de 30 caracteres y el formato del video que se desea agregar está limitado a los formatos “.mp4”, “.webm” y “.ogg”.
2. Botones para agregar el video.

En la interfaz de adhesión de archivos del Repositorio se agrega un campo extra para subir una imagen de pre-visualización. Esta imagen, al igual que los videos, está limitada a los formatos “.jpg” y “.png”. A continuación puede verse la interfaz de subida de archivos al Repositorio.

INFOCHANNEL - SIS ECU 911  
BIENVENIDO/A, LCS. VER EL SITIO / CAMBIAR CONTRASEÑA / TERMINAR SESIÓN

Inicio > Vídeos > Repositorios > Añadir repositorio

Añadir repositorio

Titulo:

Video src:  No se ha seleccionado ningun archivo.

Imagen src:  No se ha seleccionado ningun archivo.

## **ANEXO 7: GUÍA DEL PROGRAMADOR DEL INFOCHANNEL.**

### **GUÍA DEL PROGRAMADOR**

#### **1. INTRODUCCIÓN.**

El presente manual describe los aspectos técnicos referentes al sistema denominado “Infochannel”. El manual técnico hace referencia a la información necesaria con el fin de orientar al personal en la concepción, planteamiento, análisis, programación e instalación del sistema. Es de notar que la redacción propia del manual técnico está orientada a personal con conocimientos en sistemas y tecnologías de información, conocimientos de programación avanzada sobre entorno web, administración de bases de datos, responsables del mantenimiento e instalación del sistema en los servidores.

#### **2. OBJETIVOS.**

Instruir el uso del Sistema de Infochannel, para el acceso oportuno y adecuado en la instalación del mismo, mostrando los pasos a seguir en el proceso de instalación, así como la descripción de los archivos relevantes del sistema los cuales nos orienten en la configuración y soporte del mismo.

#### **3. REQUERIMIENTOS.**

##### **3.1. Sistema operativo.**

El sistema de Infochannel ha sido validado y desplegado con éxito sobre sistemas operativos como:

- Windows 10 Home versión 1809.
- Linux Ubuntu versión 18.04.
- Linux Debian “Stretch” versión 9.0.

Esto no hace que el sistema sea compatible únicamente con los sistemas operativos antes mencionados, dado que el sistema está basado en herramientas de software libre multiplataforma, lo que también la hace compatible con sistemas operativos como MacOS, Windows 7 o superior, y las diferentes distribuciones Linux.

### **3.2. Requisitos para su despliegue.**

Con la aplicación ya desarrollada y completamente programada, como requisitos para el despliegue de la misma sobre el sistema operativo seleccionado, se tiene:

- Entorno de programación Python 2.7 o superior (recomendada la versión 3.7.2).
- Framework de desarrollo web Django con una versión compatible con el entorno Python instalado (recomendada la versión 2.7.1 o superior).
- Servidor web Apache versión 2.4 o superior.
- Sistema de Gestión de Bases de Datos MySQL versión 10.3 o superior.
- Librería Python “pymysql” (versión 0.9.3) para la conexión entre la aplicación y MySQL
- Librería Python “django-cleanup” (versión 3.2.0) para la depuración de archivos de la aplicación.
- Navegador web para la visualización y prueba del sistema (recomendados Google Chrome versión 30 o superior, o Mozilla Firefox versión 35 o superior).

### **4. INSTALACIÓN Y CONFIGURACIÓN PARA SU DESPLIEGUE.**

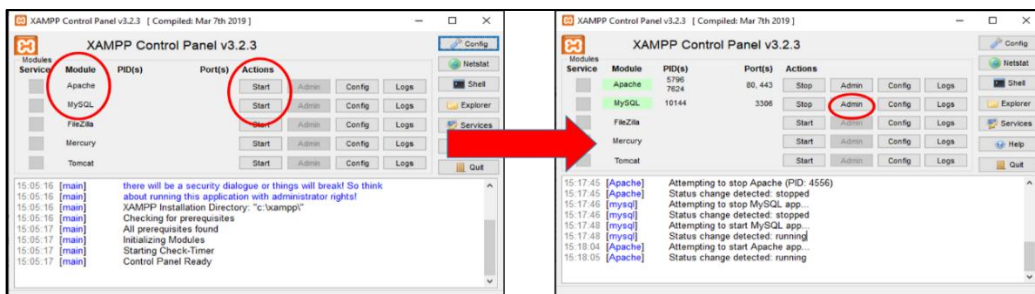
A continuación, se detalla el proceso para el despliegue de este servicio:

1. En primer lugar, se requiere instalar el entorno Python y el Framework Django. Se recomienda usar la versión 3.7.2 de Python, la cual está disponible en su página oficial [www.python.org](http://www.python.org). La instalación de Python sobre sistemas operativos como Windows y MacOS puede llevarse a cabo ejecutando directamente el gestor de instalación del programa. Para las distribuciones Linux el entorno Python suele venir incluido dentro de sus herramientas predeterminadas, sin embargo, también puede ser instalado directamente desde la línea de comandos. Los comandos de instalación variarán según la distribución elegida. Para mayor información consultar la página oficial del sistema operativo seleccionado.
2. Instalar el framework Django. El proceso de instalación para cualquier sistema operativo ha de ejecutarse desde la línea de comandos y, al ser una extensión de Python, la sintaxis para su instalación desde la línea de comandos es “*pip install Django*”.

```
Simbolo del sistema
Microsoft Windows [Versión 10.0.17763.379]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

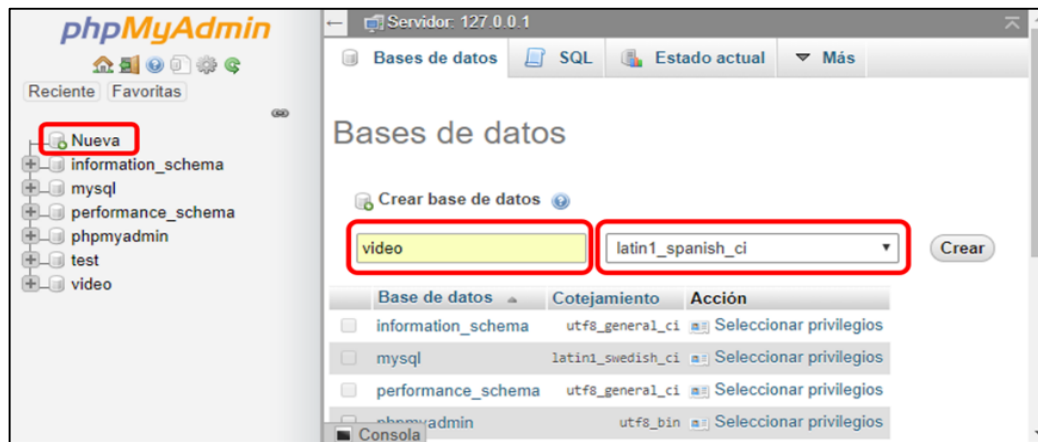
C:\Users\MACS>pip install Django
Collecting Django
  Downloading https://files.pythonhosted.org/packages/c7/87/fbd666c4f87591ae25b7bb374298e8629816e87193c4099d3608ef11fab9/Django-2.1.7-py3-none-any.whl (7.3MB)
    100% |#####| 7.3MB 836kB/s
```

3. Luego es necesario instalar el paquete de software XAMPP el cual contiene el servidor web Apache junto al Gestor de Bases de Datos MySQL. Su método de instalación es similar al del entorno Python para los sistemas operativos Windows y MacOS, mientras que para las diferentes distribuciones Linux es mediante la línea de comandos.
4. Instalar la librería “pymysql” en su versión 0.9.3 para permitir la conexión entre el servidor y la base de datos, y la librería “django-cleanup” en su versión 3.2.0 para la tarea de limpieza de archivos subidos a la aplicación. Su proceso de instalación puede ejecutarse mediante línea de comandos o por medio del editor de código PyCharm, el cual permite agregar librerías Python a la aplicación directamente desde la interfaz gráfica. Para la librería “django-cleanup” es necesario agregar la línea “django\_cleanup” dentro del archivo “settings.py” en la sección de aplicaciones del proyecto (“INSTALLED\_APPS”).
5. Dentro del directorio de instalación del software XAMPP, concretamente en la carpeta “htdocs”, se debe alojar el código fuente del programa.
6. A continuación, es necesario crear la base de datos y configurar los parámetros de acceso a ella dentro del código de la aplicación. En este caso se hará desde el SGBD MySQL, al cual podemos acceder desde el panel de control XAMPP. Lo primero que se debe hacer es iniciar el servidor Apache y el SGBD MySQL con la opción “Start” del panel de control. Tras ello, desde la opción “Admin” de MySQL se puede acceder a su interfaz web de administración.



Desde la interfaz web de MySQL, en el panel de la izquierda se hace clic en la opción “Nueva” para crear la base de datos a la cual se le asigna un nombre y el

idioma deseado. Con esto, el SGBD crea de forma rápida la base de datos que se usará a lo largo de todo el desarrollo.



El nombre para la base de datos configurada dentro del código, así como las respectivas configuraciones de acceso son las siguientes:

```
settings.py x
81
82 DATABASES = {
83     'default': {
84         'ENGINE': 'django.db.backends.mysql',
85         'NAME': 'video',
86         'USER': 'root',
87         'PASSWORD': '',
88         'HOST': 'localhost',
89         'PORT': '3306',
90     }
91 }
```

En caso de usar valores diferentes para el acceso a la base de datos, deben ser modificados en el archivo “*settings.py*” del proyecto.

7. El siguiente paso es realizar las primeras migraciones hacia la base de datos y configurar los usuarios de administración. Con el comando “*python manage.py migrate*” Django busca la variable “*INSTALLED\_APPS*” y crea las tablas necesarias de cada una de las aplicaciones registradas en el archivo “*settings.py*”, que contiene todas las aplicaciones. Al ejecutar este comando, en el cmd se puede observar un mensaje por cada migración aplicada. Todo comando que contenga la sintaxis “*manage.py*”, puede ser ejecutado desde el CLI (*Command Line Interface* o Interfaz de Línea de Comandos) de Windows siempre y cuando se esté ubicado en el directorio donde se encuentre el archivo “*manage.py*”.

```
Simbolo del sistema
C:\xampp\htdocs\infochannel python manage.py migrate
System check identified some issues:

WARNINGS:
?: (mysql.W002) MySQL Strict Mode is not set for database connection 'default'
   HINT: MySQL's Strict Mode fixes many data integrity problems in MySQL, such as data truncation upon insertion, by escalating warnings into errors. It is strongly recommended you activate it. See: https://docs.djangoproject.com/en/2.1/ref/databases/#mysql-sql-mode
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
```

8. Tras realizar las migraciones de la información requerida para el funcionamiento del sistema hacia la base de datos, también se requiere de un usuario que controle todo el sistema por lo que mediante el comando “*python manage.py createsuperuser*” se lleva esto a cabo. Con ello se accedería a un formulario sencillo con el que configurar el nombre de usuario y contraseña, e incluso un correo electrónico, para el usuario administrador del sistema.

```
Simbolo del sistema
C:\xampp\htdocs\infochannel python manage.py createsuperuser
Username (leave blank to use 'mac'): lcs
Email address: luis.cartagena@unl.edu.ec
Password:
Password (again):
Superuser created successfully.
```

9. Ahora es necesario configurar el acceso externo a la aplicación, de modo que esta esté disponible para cualquier usuario dentro de la red sobre la cual se implemente. Para ello es necesario acceder al archivo de configuración del servidor Apache “*httpd-xampp.conf*” y agregar al final de dicho archivo, las siguientes líneas de código:

```
# New XAMPP security concept
#
# Close XAMPP security section here
<LocationMatch "^/(?:security)">
Order deny,allow
#Deny from all
#Allow from ::1 127.0.0.0/8
Allow from all #se indican las direcciones con acceso al servidor

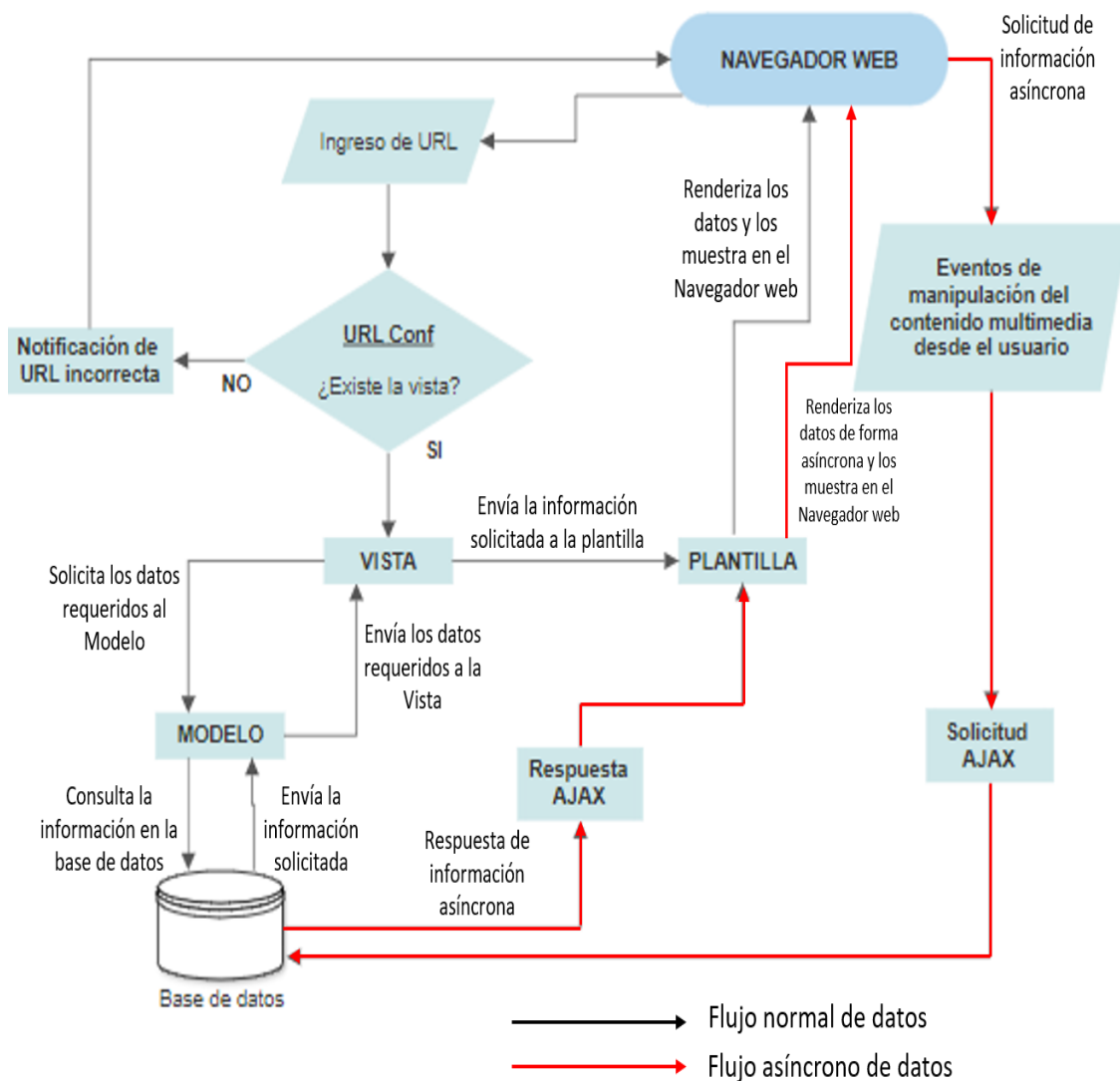
ErrorDocument 403 /error/HTTP_XAMPP_FORBIDDEN.html.var
</LocationMatch>

# Close XAMPP sites here
<LocationMatch "^/(?:xampp|licenses|phpmyadmin|webalizer|server-status|server-info)">
Order deny,allow
#Deny from all
#Allow from ::1 127.0.0.0/8
#Allow from all
Require all granted

ErrorDocument 403 /error/HTTP_XAMPP_FORBIDDEN.html.var
</LocationMatch>
```

10. Una vez habilitado el acceso externo, finalmente es momento de inicializar el servidor. Para ello se ha ejecutar el comando “*python manage.py runserver 0.0.0.0:8000*” en el directorio donde se encuentra el archivo “*manage.py*”. Esto debe realizarse desde la línea de comandos.
11. Con el programa ya funcionando, el siguiente paso es acceder a la interfaz de administración y desde ahí, cargar los videos que se desee mostrar (consultar Guía de Administración en el Anexo 6 para más información).

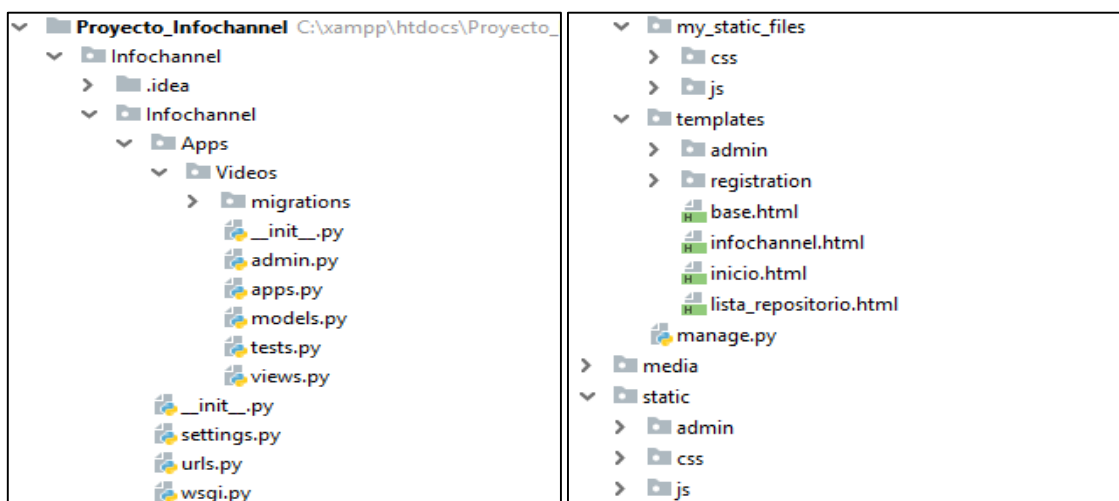
## 5. DIAGRAMA DE FLUJO DE DATOS.





## 6. LISTADO DE DIRECTORIOS Y ARCHIVOS DEL SISTEMA.

En el siguiente apartado se exponen los diferentes directorios que componen el sistema de Infochannel. Es necesario aclarar que el código fuente del sistema debe ser alojado dentro del directorio “htdocs” perteneciente al conjunto de directorios del servidor web Apache. A continuación, se puede visualizar el conjunto de directorios del sistema:



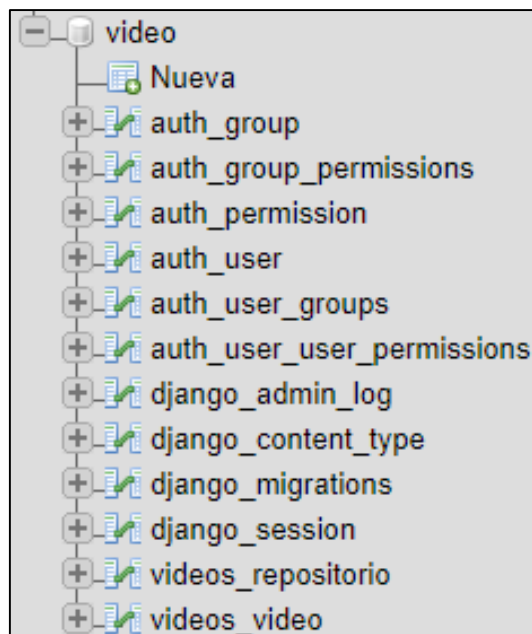
| DIRECTORIO/ARCHIVO                               | DESCRIPCIÓN   |
|--|---|
| Proyecto_Infochannel/                            | Directorio general del proyecto el cual aloja por completo el código fuente del sistema.  |
| Proyecto_Infochannel/.idea/                      | Alberga archivos XML con las direcciones específicas de cada archivo dentro del sistema. Este directorio puede encontrarse también dentro de otros subdirectorios del sistema.  |
| Proyecto_Infochannel/media/                      | Directorio de almacenamiento de los archivos que se servirán dentro de la interfaz del Infochannel.   |
| Proyecto_Infochannel/media/Repositorio/          | Directorio que almacena los archivos que se servirán dentro de la interfaz del Repositorio de archivos.   |
| Proyecto_Infochannel/media/Repositorio/Videos/   | Almacena los videos del Repositorio.  |
| Proyecto_Infochannel/media/Repositorio/Imágenes/ | Almacena las imágenes de previsualización del Repositorio.  |
| Proyecto_Infochannel/static/                     | Alberga los archivos estáticos del sistema como elementos CSS, JavaScript e incluso imágenes. Estos permiten que el sistema pueda visualizarse correctamente desde un dispositivo diferente al dispositivo que funciona como servidor de la aplicación. |
| Proyecto_Infochannel/static/admin/               | Almacena archivos referentes a la programación de la interfaz de administración del sistema.  |
| Proyecto_Infochannel/static/admin/css/           | Se almacenan archivos CSS de la interfaz de administración.   |
| Proyecto_Infochannel/static/admin/fonts/         | Se almacenan los estilos de fuente de la interfaz de administración.  |

|   |   |
|---|---|
| Proyecto_Infochannel/static/admin/img/                                  | Se almacenan las imágenes necesarias para personalizar la interfaz de administración.   |
| Proyecto_Infochannel/static/admin/js/                                   | Se almacenan archivos JavaScript de la interfaz de administración.  |
| Proyecto_Infochannel/static/css/  | Almacena archivos CSS que personalizan la interfaz de usuario del sistema.  |
| Proyecto_Infochannel/static/js/   | Almacena los archivos JavaScript que dan funcionalidad a las interfaces de usuario de la aplicación.  |
| Proyecto_Infochannel/Infochannel/                                       | Directorio que alberga las herramientas y diferentes archivos que componen el sistema y que tienen más relación con el Framework Django.  |
| Proyecto_Infochannel/Infochannel/my_static_files/                       | Alberga los archivos estáticos del sistema como elementos CSS, JavaScript e incluso imágenes. Estos permiten que el sistema pueda visualizarse correctamente únicamente desde el dispositivo que funciona como servidor de la aplicación. |
| Proyecto_Infochannel/Infochannel/my_static_files/css/                   | Aquí se almacenan los archivos CSS de las interfaces de usuario para su uso de forma local.   |
| Proyecto_Infochannel/Infochannel/my_static_files/js/                    | Aquí se almacenan los archivos JavaScript de las interfaces de usuario para su uso de forma local.  |
| Proyecto_Infochannel/Infochannel/templates/                             | Aquí se almacenan los documentos html, CSS y JavaScript para la interfaz de usuario y administración.   |
| Proyecto_Infochannel/Infochannel/templates/base.html                    | Archivo html que contiene las estructuras base para las diferentes interfaces de usuario.   |
| Proyecto_Infochannel/Infochannel/templates/inicio.html                  | Código de la interfaz de inicio del sistema.  |
| Proyecto_Infochannel/Infochannel/templates/infochannel.html             | Código de la interfaz del Infochannel.  |
| Proyecto_Infochannel/Infochannel/templates/lista_repositorio.html       | Código de la interfaz del Repositorio.  |
| Proyecto_Infochannel/Infochannel/templates/registration/logged_out.html | Código perteneciente a la interfaz de login de la interfaz de administración.   |
| Proyecto_Infochannel/Infochannel/templates/admin/ base.html             | Códigos base para la interfaz de administración.  |
| Proyecto_Infochannel/Infochannel/templates/admin/ base_site.html        |   |
| Proyecto_Infochannel/Infochannel/manage.py                              | Archivo de control e interacción entre el proyecto en desarrollo y el Framework Django. Cualquier comando de sintaxis “manage.py” debe ser ejecutado dentro del mismo directorio que este archivo.  |
| Proyecto_Infochannel/Infochannel/Infochannel/                           | Directorio que alberga las herramientas y diferentes archivos que componen el sistema y que tienen más relación con el Framework Django.  |
| Proyecto_Infochannel/Infochannel/Infochannel/ __pycache__               | Contiene archivos del sistema que agilizan la carga y renderización de la aplicación en el navegador web. Puede encontrarse un directorio similar dentro de otros subdirectorios.   |
| Proyecto_Infochannel/Infochannel/infochannel/ __init__.py               | Archivo Python que hace que el entorno Python reconozca los archivos de ese directorio como archivos de extensión “.py”. Puede encontrarse también dentro de otros subdirectorios de la aplicación.                                       |

|  |  |
|--|--|
| Proyecto_Infochannel/Infochannel/infochannel/settings.py             | Archivo de configuración global de la aplicación   |
| Proyecto_Infochannel/Infochannel/infochannel/urls.py                 | Archivo de configuración de URLs para la aplicación  |
| Proyecto_Infochannel/Infochannel/infochannel/wsgi.py                 | Archivo de configuración para la conexión entre la aplicación Django y el servidor web.  |
| Proyecto_Infochannel/Infochannel/infochannel/Apps/                   | Directorio que alberga cada una de las aplicaciones dentro del proyecto Django.  |
| Proyecto_Infochannel/Infochannel/infochannel/Apps/Videos/            | Directorio de la aplicación “Videos”.  |
| Proyecto_Infochannel/Infochannel/infochannel/Apps/Videos/migrations/ | Directorio que guarda una lista de registros de las migraciones hechas hacia la base de datos.   |
| Proyecto_Infochannel/Infochannel/infochannel/Apps/Videos/admin.py    | Archivo de configuración lógica para los datos que se manejarán dentro de la interfaz de programación.   |
| Proyecto_Infochannel/Infochannel/infochannel/Apps/Videos/apps.py     | Este archivo se crea para ayudar al usuario a incluir cualquier configuración de aplicación para la aplicación. Usando esto, se puede configurar algunos de los atributos de la aplicación |
| Proyecto_Infochannel/Infochannel/infochannel/Apps/Videos/models.py   | Archivo para la configuración de los modelos de datos que se usarán en la aplicación.  |
| Proyecto_Infochannel/Infochannel/infochannel/Apps/Videos/test.py     | Dentro de este archivo se configuran pruebas de rendimiento automatizadas. Se utilizan para detectar bugs o errores dentro de la aplicación  |
| Proyecto_Infochannel/Infochannel/infochannel/Apps/Videos/views.py    | Archivo de configuración de las diferentes vistas con las que se trabajará en la aplicación.   |

## 7. DESCRIPCIÓN DE TABLAS DENTRO DE LA BASE DE DATOS.

Para el sistema de Infochannel es necesario el uso de diversos tipos de datos, los cuales son almacenados dentro de una base de datos. Estos datos a su vez son organizados y clasificados según un sistema de tablas, las cuales se muestran y describen a continuación:



| <b>TABLA</b>               | <b>DESCRIPCIÓN</b>  |
|----------------------------|---|
| auth_group                 | Almacena información sobre los grupos registrados dentro del sistema.   |
| auth_group_permissions     | Almacena información acerca de los permisos para los grupos de usuarios.  |
| auth_permission            | Almacena cada uno de los permisos de usuario que pueden ser asignados a los administradores del sistema.  |
| auth_user                  | Almacena información acerca de cada uno de los usuarios de administración registrados dentro del sistema.   |
| auth_user_groups           | Almacena información acerca de los grupos de usuarios.  |
| auth_user_user_permissions | Almacena información acerca de los permisos de usuario.   |
| django_admin_log           | Contiene registros de los accesos al panel de administración de la aplicación.  |
| django_content_type        | Contiene información de las aplicaciones que se encuentran dentro del proyecto y los tipos de datos que maneja, así como los modelos a los que hacen referencia.          |
| django_migrations          | Almacena información de cada una de las migraciones que se realizan desde la aplicación hacia la base de datos.   |
| django_session             | Almacena información acerca de las sesiones que se dan dentro del gestor de bases de datos, es decir, cada vez que se acceda a la información dentro de la base de datos. |
| videos_repositorio         | Tabla de datos perteneciente al tipo de datos que se manejará dentro de la interfaz de Repositorio.   |
| videos_video               | Tabla de datos perteneciente al tipo de datos que se manejará dentro de la interfaz del Infochannel.  |

## 8. DEFINICIÓN DE VARIABLES.

- Archivo admin.py:

| <b>VARIABLE</b>      | <b>TIPO DE DATO</b> | <b>DESCRIPCIÓN</b>   |
|----------------------|---------------------|--|
| <b>list_display</b>  | vector              | Almacena una lista de los modelos que podrán ser visualizados en el panel de administración              |
| <b>list_filter</b>   | vector              | Almacena una lista de los modelos que aparecerán en el panel de filtros de la interfaz de administración |
| <b>search_fields</b> | vector              | Se declara los criterios para la barra de búsquedas  |
| <b>list_editable</b> | vector              | Se establecen qué modelos podrán ser modificables directamente desde la interfaz de administración.      |
| <b>model</b>         | caracter            | Especifica el modelo al que se hace referencia para ordenarlos en el panel de administración             |
| <b>ordering</b>      | vector              | Almacena el orden de los modelos para ser presentados en la interfaz de administración.                  |

- Archivo apps.py:

| VARIABLE | TIPO DE DATO | DESCRIPCIÓN   |
|----------|--------------|---|
| name     | caracter     | Almacena el nombre de la aplicación o aplicaciones que se crean dentro del proyecto Django. |

- Archivo models.py:

| VARIABLE         | TIPO DE DATO | DESCRIPCIÓN  |
|------------------|--------------|--|
| ext              | caracter     | Almacena el nombre del archivo seleccionado para subirlo al Infochannel                                      |
| valid_extensions | vector       | Almacena las extensiones de archivos de video válidas para el Infochannel                                    |
| video_src        | caracter     | Almacena el nombre del archivo original de video que se sube al Infochannel o al repositorio                 |
| imagen_src       | caracter     | Almacena el nombre del archivo original de tipo imagen que se sube como previsualización para el repositorio |
| titulo           | caracter     | Almacena el título que se desea mostrar en la interfaz de usuario para cada video subido al Infochannel.     |
| ingresado        | caracter     | Almacena la fecha y hora de subida del archivo   |
| modificado       | caracter     | Almacena fecha y hora de modificación del archivo  |

- Archivo views.py:

| VARIABLE        | TIPO DE DATO | DESCRIPCIÓN   |
|-----------------|--------------|---|
| lista           | caracter     | Variable que extrae y almacena la información desde la base de datos  |
| post_serialized | caracter     | Variable que convierte la información almacenada en la variable lista, y la convierte a formato json para su posterior manipulación dinámica desde la interfaz de usuario |

- Archivo settings.py:

| VARIABLE       | TIPO DE DATO | DESCRIPCIÓN  |
|----------------|--------------|--|
| BASE_DIR       | caracter     | Guarda el directorio raíz del proyecto   |
| SECRET_KEY     | caracter     | Código de encriptación del sistema   |
| DEBUG          | booleano     | Determina si el sistema se utiliza en modo desarrollo o en modo de producción                        |
| ALLOWED_HOST   | numerico     | Se declaran los hosts que alojan el servicio   |
| INSTALLED_APPS | caracter     | Se declaran las aplicaciones que pertenecen al proyecto  |
| MIDDLEWARE     | caracter     | Se declaran las herramientas o programas que ofrecen funcionalidades a las aplicaciones del proyecto |

|                                 |          |  |
|---------------------------------|----------|--|
| <b>ROOT_URLCONF</b>             | caracter | Especifica la ruta del archivo que contiene las URLs del sistema   |
| <b>TEMPLATES</b>                | caracter | Almacena información acerca de las plantillas html del sistema   |
| <b>WSGI_APPLICATION</b>         | caracter | Se declara el archivo que contiene la configuración del WSGI   |
| <b>DATABASE</b>                 | caracter | Se indican las bases de datos que albergan toda la información con la que funciona el sistema                        |
| <b>AUTH_PASSWORD_VALIDATORS</b> | caracter | Almacena los nombres de los diferentes procesos de validación de contraseñas del sistema Django                      |
| <b>LANGUAGE_CODE</b>            | caracter | Se indica el lenguaje del código en desarrollo para que pueda ser adaptado a múltiples idiomas soportados por Django |
| <b>TIME_ZONE</b>                | caracter | Se especifica la zona horaria del servidor o aplicación que se desarrolla  |
| <b>USE_I18N</b>                 | booleano | Se habilita para la traducción o internacionalización del código en Django   |
| <b>USE_L10N</b>                 | booleano |  |
| <b>USE_TZ</b>                   | booleano |  |
| <b>STATIC_URL</b>               | caracter | Se indica la ruta de las URLs de los archivos estáticos  |
| <b>STATICFILES_DIRS</b>         | caracter | Se declara el directorio específico de archivos estáticos  |
| <b>STATIC_ROOT</b>              | caracter | Se declara la ruta raíz de archivos estáticos  |
| <b>MEDIA_URL</b>                | caracter | Se declara la ruta de las URLs de los archivos multimedia  |
| <b>MEDIA_ROOT</b>               | caracter | Se declara la ruta raíz de archivos multimedia   |

- **Archivo urls.py:**

| VARIABLE           | TIPO DE DATO | DESCRIPCIÓN  |
|--------------------|--------------|--|
| <b>urlpatterns</b> | vector       | Se declara cada una de las URLs del sistema de Infochannel |

- **Archivo infochannel.html:**

| VARIABLE        | TIPO DE DATO | DESCRIPCIÓN   |
|-----------------|--------------|---|
| <b>n_video</b>  | numérico     | Almacena el número del archivo de video que se está reproduciendo actualmente   |
| <b>array</b>    | arreglo      | Almacena la información de cada uno de los archivos de video  |
| <b>limite</b>   | caracter     | Almacena el número total de videos que se han subido al sistema   |
| <b>fsReturn</b> | caracter     | Almacena un conjunto de condicionales para los navegadores, de modo que se pueda determinar si estos tienen compatibilidad con la función de Pantalla Completa. |

|                 |          |  |
|-----------------|----------|--|
| <b>vid</b>      | caracter | Variable que almacena el nombre del video que se está reproduciendo actualmente y lo mostrará en la interfaz de usuario                            |
| <b>form</b>     | caracter | Declaración de petición Ajax   |
| <b>post</b>     | caracter | Declaración del método de petición web mediante Ajax   |
| <b>url</b>      | caracter | Declaración de la tabla desde la cual se extraerá información para la plantilla. Pertenecen a los parámetros de la petición Ajax                   |
| <b>data</b>     | caracter | Establece el formato de los datos Ajax   |
| <b>dataType</b> | caracter | Se especifica el tipo de dato con el que se trabaja, en el caso del Infochannel es de tipo "json". Pertenecen a los parámetros de la petición Ajax |
| <b>success</b>  | caracter | Declaración final del Ajax   |

- **Archivo lista\_repositorio.html:**

| VARIABLE        | TIPO DE DATO | DESCRIPCIÓN  |
|-----------------|--------------|--|
| <b>n_video</b>  | numérico     | Almacena el número del archivo de video que se está reproduciendo actualmente  |
| <b>array</b>    | arreglo      | Almacena la información de cada uno de los archivos de video   |
| <b>limite</b>   | caracter     | Almacena el número total de videos que se han subido al sistema  |
| <b>form</b>     | caracter     | Declaración de petición Ajax   |
| <b>post</b>     | caracter     | Declaración del método de petición web mediante Ajax   |
| <b>url</b>      | caracter     | Declaración de la tabla desde la cual se extraerá información para la plantilla. Pertenecen a los parámetros de la petición Ajax                   |
| <b>data</b>     | caracter     | Establece el formato de los datos Ajax   |
| <b>dataType</b> | caracter     | Se especifica el tipo de dato con el que se trabaja, en el caso del Infochannel es de tipo "json". Pertenecen a los parámetros de la petición Ajax |
| <b>success</b>  | caracter     | Declaración final del Ajax   |

- **Archivo functions.js:**

| VARIABLE              | TIPO DE DATO | DESCRIPCIÓN  |
|-----------------------|--------------|--|
| <b>sliderDuration</b> | numérico     | Determina el tiempo de movimiento en milisegundos del panel de previsualización de la interfaz del Repositorio |

## 12. DICCIONARIO DE DATOS.

A continuación, se especifica de forma detallada los tipos de datos que se manejan dentro de cada tabla perteneciente a la base de datos del sistema:

- **Tabla auth\_group:**

|                           |   |                     |   |
|---------------------------|---|---------------------|---|
| <b>Nombre de archivo:</b> | auth_group  |                     | <b>Fecha de creación:</b>                         |
| <b>Descripción:</b>       | Almacena información sobre los grupos registrados dentro del sistema. |                     |   |
| <b>Campo</b>              | <b>Tamaño</b>   | <b>Tipo de dato</b> | <b>Descripción</b>                                |
| <b>id</b>                 | 11  | Numérico            | Identificador numérico para cada entrada de datos |
| <b>name</b>               | 80  | Caracter            | Describe el nombre para cada grupo ingresado      |

- **Tabla auth\_group\_permission:**

|                           |  |                     |   |
|---------------------------|--|---------------------|---|
| <b>Nombre de archivo:</b> | auth_group_permission  |                     | <b>Fecha de creación:</b>                         |
| <b>Descripción:</b>       | Almacena información acerca de los permisos para los grupos de usuarios. |                     |   |
| <b>Campo</b>              | <b>Tamaño</b>  | <b>Tipo de dato</b> | <b>Descripción</b>                                |
| <b>id</b>                 | 11   | Numérico            | Identificador numérico para cada entrada de datos |
| <b>group_id</b>           | 11   | Numérico            | Identificador para cada tipo de grupo             |
| <b>permission_id</b>      | 11   | Caracter            | Identificador para cada tipo de permiso           |

- **Tabla auth\_permission:**

|                           |  |                     |   |
|---------------------------|--|---------------------|---|
| <b>Nombre de archivo:</b> | auth_permission  |                     | <b>Fecha de creación:</b>                         |
| <b>Descripción:</b>       | Almacena cada uno de los permisos de usuario que pueden ser asignados a los administradores del sistema. |                     |   |
| <b>Campo</b>              | <b>Tamaño</b>  | <b>Tipo de dato</b> | <b>Descripción</b>                                |
| <b>id</b>                 | 11   | Numérico            | Identificador numérico para cada entrada de datos |
| <b>name</b>               | 255  | Caracter            | Nombre del permiso                                |
| <b>conten_type_id</b>     | 11   | Numérico            | Identificador del tipo de contenido               |
| <b>codename</b>           | 100  | Caracter            | Codificación para el tipo de permiso              |

- **Tabla auth\_user:**

|                           |   |                     |   |
|---------------------------|---|---------------------|---|
| <b>Nombre de archivo:</b> | auth_user   |                     | <b>Fecha de creación:</b>                         |
| <b>Descripción:</b>       | Almacena información acerca de cada uno de los usuarios de administración registrados dentro del sistema. |                     |   |
| <b>Campo</b>              | <b>Tamaño</b>   | <b>Tipo de dato</b> | <b>Descripción</b>                                |
| <b>id</b>                 | 11  | Numérico            | Identificador numérico para cada entrada de datos |
| <b>password</b>           | 128   | Caracter            | Contraseña de acceso encriptada                   |
| <b>last_login</b>         | 6   | Dato de tiempo      | Fecha de último acceso de usuario                 |
| <b>is_superuser</b>       | 1   | Numérico            | Indicador de super usuario (1=verdadero; 0=falso) |
| <b>username</b>           | 150   | Caracter            | Nombre de usuario de acceso                       |
| <b>first_name</b>         | 30  | Caracter            | Nombre del usuario                                |
| <b>last_name</b>          | 150   | Caracter            | Apellido del usuario                              |
| <b>email</b>              | 254   | Caracter            | Correo electrónico del usuario                    |



|                    |   |                |  |
|--------------------|---|----------------|--|
| <b>is_staff</b>    | 1 | Numérico       | Estado de personal o no (1=verdadero; 0=falso) |
| <b>is_active</b>   | 1 | Numérico       | Estado de actividad (1=verdadero; 0=falso)     |
| <b>date_joined</b> | 6 | Dato de tiempo | Fecha y hora de la creación del usuario        |

- **Tabla auth\_user\_groups:**

|                           |  |                     |   |
|---------------------------|--|---------------------|---|
| <b>Nombre de archivo:</b> | auth_user_groups                                       |                     | <b>Fecha de creación:</b>                         |
| <b>Descripción:</b>       | Almacena información acerca de los grupos de usuarios. |                     |   |
| <b>Campo</b>              | <b>Tamaño</b>  | <b>Tipo de dato</b> | <b>Descripción</b>                                |
| <b>id</b>                 | 11   | Numérico            | Identificador numérico para cada entrada de datos |
| <b>user_id</b>            | 11   | Numérico            | Identificador de usuario                          |
| <b>group_id</b>           | 11   | Caracter            | Identificador del grupo al que pertenece          |

- **Tabla auth\_user\_user\_permissions:**

|                           |   |                     |   |
|---------------------------|---|---------------------|---|
| <b>Nombre de archivo:</b> | auth_user_user_permissions                              |                     | <b>Fecha de creación:</b>                         |
| <b>Descripción:</b>       | Almacena información acerca de los permisos de usuario. |                     |   |
| <b>Campo</b>              | <b>Tamaño</b>   | <b>Tipo de dato</b> | <b>Descripción</b>                                |
| <b>id</b>                 | 11  | Numérico            | Identificador numérico para cada entrada de datos |
| <b>user_id</b>            | 11  | Numérico            | Identificador de usuario                          |
| <b>permission_id</b>      | 11  | Caracter            | Identificador del permiso de usuario asignado     |

- **Tabla django\_admin\_log:**

|                           |  |                     |   |
|---------------------------|--|---------------------|---|
| <b>Nombre de archivo:</b> | django_admin_log   |                     | <b>Fecha de creación:</b>                         |
| <b>Descripción:</b>       | Contiene registros de los accesos al panel de administración de la aplicación. |                     |   |
| <b>Campo</b>              | <b>Tamaño</b>  | <b>Tipo de dato</b> | <b>Descripción</b>                                |
| <b>id</b>                 | 11   | Numérico            | Identificador numérico para cada entrada de datos |
| <b>action_time</b>        | 6  | Dato de tiempo      | Fecha y hora de la modificación                   |
| <b>object_id</b>          | -  | Caracter            | Identificador del objeto                          |
| <b>object_repr</b>        | 200  | Caracter            | Objeto modificado                                 |
| <b>action_flag</b>        | 5  | Numérico            | Indicador de cambio                               |
| <b>change_message</b>     | -  | Caracter            | Mensaje de tipo de cambio                         |
| <b>content_type_id</b>    | 11   | Numérico            | Identificador del tipo de contenido               |
| <b>user_id</b>            | 11   | Numérico            | Identificador de usuario que realiza el cambio    |

- **Tabla django\_content\_type:**

|                           |  |                     |   |
|---------------------------|--|---------------------|---|
| <b>Nombre de archivo:</b> | django_content_type  |                     | <b>Fecha de creación:</b>                         |
| <b>Descripción:</b>       | Contiene información de las aplicaciones que se encuentran dentro del proyecto y los tipos de datos que maneja, así como los modelos a los que hacen referencia. |                     |   |
| <b>Campo</b>              | <b>Tamaño</b>  | <b>Tipo de dato</b> | <b>Descripción</b>                                |
| <b>id</b>                 | 11   | Numérico            | Identificador numérico para cada entrada de datos |
| <b>app_label</b>          | 100  | Caracter            | Etiqueta de la aplicación                         |
| <b>model</b>              | 100  | Caracter            | Nombre del modelo de datos                        |

- **Tabla django\_migrations:**

|                           |   |                           |   |
|---------------------------|---|---------------------------|---|
| <b>Nombre de archivo:</b> | django_migrations   | <b>Fecha de creación:</b> |   |
| <b>Descripción:</b>       | Almacena información de cada una de las migraciones que se realizan desde la aplicación hacia la base de datos. |                           |   |
| <b>Campo</b>              | <b>Tamaño</b>   | <b>Tipo de dato</b>       | <b>Descripción</b>                                |
| id                        | 11  | Numérico                  | Identificador numérico para cada entrada de datos |
| app                       | 255   | Caracter                  | Nombre de la aplicación que migra datos           |
| name                      | 255   | Caracter                  | Nombre del tipo de migración                      |
| applied                   | 6   | Dato de tiempo            | Fecha y hora de la migración                      |

- **Tabla django\_session:**

|                           |   |                           |  |
|---------------------------|---|---------------------------|--|
| <b>Nombre de archivo:</b> | django_session  | <b>Fecha de creación:</b> |  |
| <b>Descripción:</b>       | Almacena información acerca de las sesiones que se dan dentro del gestor de bases de datos, es decir, cada vez que se acceda a la información dentro de la base de datos. |                           |  |
| <b>Campo</b>              | <b>Tamaño</b>   | <b>Tipo de dato</b>       | <b>Descripción</b>                       |
| session_key               | 40  | Caracter                  | Clave de encriptación de la sesión       |
| session_data              | -   | Caracter                  | Clave de encriptación de datos de sesión |
| expire_date               | 6   | Dato de tiempo            | Fecha y hora de expiración de la sesión  |

- **Tabla videos\_repositorio:**

|                           |   |                           |   |
|---------------------------|---|---------------------------|---|
| <b>Nombre de archivo:</b> | videos_repositorio  | <b>Fecha de creación:</b> |   |
| <b>Descripción:</b>       | Tabla de datos perteneciente al tipo de datos que se manejará dentro de la interfaz de Repositorio. |                           |   |
| <b>Campo</b>              | <b>Tamaño</b>   | <b>Tipo de dato</b>       | <b>Descripción</b>  |
| id                        | 11  | Numérico                  | Identificador numérico para cada entrada de datos                             |
| titulo                    | 30  | Caracter                  | Título del archivo subido al repositorio y que se mostrará en la interfaz web |
| video_src                 | 100   | Caracter                  | Título original del archivo subido al repositorio                             |
| imagen_src                | 100   | Caracter                  | Título de la imagen de previsualización del archivo                           |
| ingresado                 | 6   | Dato de tiempo            | Fecha y hora de subida del archivo  |
| modificado                | 6   | Dato de tiempo            | Fecha y hora de modificación del archivo                                      |

- **Tabla videos\_video:**

|                           |  |                           |   |
|---------------------------|--|---------------------------|---|
| <b>Nombre de archivo:</b> | videos_video   | <b>Fecha de creación:</b> |   |
| <b>Descripción:</b>       | Tabla de datos perteneciente al tipo de datos que se manejará dentro de la interfaz del Infochannel. |                           |   |
| <b>Campo</b>              | <b>Tamaño</b>  | <b>Tipo de dato</b>       | <b>Descripción</b>                                |
| id                        | 11   | Numérico                  | Identificador numérico para cada entrada de datos |
| titulo                    | 30   | Caracter                  | Título del archivo subido al Infochannel          |
| video_src                 | 100  | Caracter                  | Título original del archivo subido al Infochannel |
| ingresado                 | 6  | Dato de tiempo            | Fecha y hora de subida del archivo                |
| modificado                | 6  | Dato de tiempo            | Fecha y hora de modificación del archivo          |

## 9. LIMITACIONES DEL SISTEMA.

- Para garantizar un correcto funcionamiento del sistema, así como su compatibilidad con una mayor cantidad de usuarios, los formatos de video permitidos por el sistema son los que se encuentran de forma predeterminada dentro del estándar HTML, y estos son:
  - Formato de video “mp4”.
  - Formato de video “ogg/ogv”.
  - Formato de video “webm”.
- Para determinar una limitación en cuanto al tamaño del archivo que el sistema es capaz de soportar se realizaron pruebas experimentales con diferentes dispositivos de usuario. Los resultados obtenidos de dichas pruebas reflejan una limitación en cuanto al tamaño máximo del archivo multimedia que puede ser reproducido sin una demora considerable en el dispositivo de usuario. Por lo tanto, se estableció una tabla con los límites de tamaño recomendados para el archivo de video dependiendo del dispositivo de usuario al que vaya dirigido el contenido.

| <b>Tipo de dispositivo</b> | <b>Tamaño recomendado (Megabytes)</b> |
|----------------------------|---------------------------------------|
| Computadora                | 700 Mb                                |
| Smartphone                 | 200 Mb                                |
| Smart TV                   | 50 Mb                                 |
| Smartwatch                 | 100                                   |

## 10. CÓDIGO FUENTE DEL SISTEMA.

El código fuente del sistema de Infochannel se encuentra detallado en los Anexos 8 y 9 del presente documento.

## ANEXO 8: CÓDIGO DEL BACKEND DEL SISTEMA DESARROLLADO EN PYTHON.

- Archivo “settings.py”.

```
import os
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
SECRET_KEY = '@3r#j(wn@3n*13viokvm8j7wvkp)mk3vn+u+b$!qlvazt@od*'
DEBUG = True
ALLOWED_HOSTS = ['192.168.1.6', '192.168.1.11', '192.168.10.100']
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'Infochannel.Apps.Videos',
    'django_cleanup.apps.CleanupConfig',]
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',]
ROOT_URLCONF = 'Infochannel.urls'
TEMPLATES = [ {
    'BACKEND': 'django.template.backends.django.DjangoTemplates',
    'DIRS': [os.path.join(BASE_DIR, "templates")],
    'APP_DIRS': True,
    'OPTIONS': { 'context_processors': [
        'django.template.context_processors.debug',
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
        'django.template.context_processors.media',
    ]},},],]
WSGI_APPLICATION = 'Infochannel.wsgi.application'
DATABASES = { 'default': {
    'ENGINE': 'django.db.backends.mysql',
    'NAME': 'video',
    'USER': 'root',
    'PASSWORD': '',
    'HOST': 'localhost',
    'PORT': '3306',}}
AUTH_PASSWORD_VALIDATORS = [
    { 'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator', },
    { 'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator', },
    { 'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator', },
    { 'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator', },
],]
LANGUAGE_CODE = 'es-ec'
TIME_ZONE = 'America/Guayaquil'
USE_I18N = True
USE_L10N = True
USE_TZ = True
```

```

STATIC_URL = '/static/'
STATICFILES_DIRS = (
    os.path.join(BASE_DIR, "my_static_files"),)
STATIC_ROOT = os.path.join(os.path.dirname(BASE_DIR), "static")
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(os.path.dirname(BASE_DIR), "media")

```

- Archivo “urls.py”.

```

from django.conf.urls import url
from django.contrib import admin
from django.conf import settings
from django.conf.urls.static import static
from Infochannel.Apps.Videos import views
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^infochannel/', views.infochannel, name='infochannel'),
    url(r'^repositorio/', views.repositorio, name='repositorio'),
    url(r'^lista_infochannel/', views.lista_infochannel,
name='lista_infochannel'),
    url(r'^lista_repositorio/', views.lista_repositorio,
name='lista_repositorio'),
    url(r'^$', views.inicio, name='inicio')]
if settings.DEBUG:
    urlpatterns += static(settings.STATIC_URL,
document_root=settings.STATIC_ROOT)
    urlpatterns += static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)

```

- Archivo “models.py”.

```

from django.db import models
from django.core.exceptions import ValidationError
def validate_video_extension(video_src):
    import os
    ext = os.path.splitext(video_src.name)[1]
    valid_extensions = ['.mp4', '.webm', '.ogv']
    if not ext in valid_extensions:
        raise ValidationError(u'Formato de video no válido. Seleccione un archivo
de extensión ".mp4", ".webm" u ".ogv"')
def validate_file_extension(imagen_src):
    import os
    ext = os.path.splitext(imagen_src.name)[1]
    valid_extensions = ['.jpg', '.png', '.gif']
    if not ext in valid_extensions:
        raise ValidationError(u'Formato de imagen no válida. Seleccione un archivo
de extensión ".jpg", ".png" o ".gif"')
class Video(models.Model):
    titulo = models.CharField(max_length=30)
    video_src =
models.FileField(upload_to='', validators=[validate_video_extension],
blank=False, null=False)
    ingresado = models.DateTimeField(auto_now_add=True, auto_now=False)
    modificado = models.DateTimeField(auto_now_add=False, auto_now=True)
    def __str__(self):
        return self.titulo
    class Meta:
        ordering = ["ingresado"]

```

```

class Repositorio(models.Model):
    titulo = models.CharField(max_length=30)
    video_src =
models.FileField(upload_to='repositorio/videos/',validators=[validate_video_ex
tension], blank=False, null=False,default='')
    imagen_src = models.FileField(upload_to='repositorio/imagenes/',
validators=[validate_file_extension], blank=False, null=False,default='')
    ingresado = models.DateTimeField(auto_now_add=True,auto_now=False)
    modificado = models.DateTimeField(auto_now_add=False,auto_now=True)
    def __str__(self):
        return self.titulo
    class Meta:
        ordering = ["ingresado"]

```

- Archivo “admin.py”.

```

from django.contrib import admin
from Infochannel.Apps.Videos.models import Video,Repositorio
class VideoRegistrado (admin.ModelAdmin):
    list_display = ["titulo","video_src","ingresado","modificado"]
    list_filter = ["ingresado","modificado"]
    search_fields = ["titulo"]
    list_editable = ["video_src"]
    class Meta:
        model = Video
        ordering = ["ingresado"]
class RepositorioVideos (admin.ModelAdmin):
    list_display = ["titulo","video_src","ingresado","modificado"]
    list_filter = ["ingresado","modificado"]
    search_fields = ["titulo"]
    list_editable = ["video_src"]
    class Meta:
        model = Repositorio
        ordering = ["ingresado"]
admin.site.register(Video,VideoRegistrado)
admin.site.register(Repositorio,RepositorioVideos)

```

- Archivo “views.py”.

```

from django.shortcuts import render
from .models import Video, Repositorio
from django.http import JsonResponse
from django.core import serializers
def infochannel(request):
    return render (request,"infochannel.html")
def lista_infochannel(request):
    lista=Video.objects.all()
    posts_serialized = serializers.serialize('json', lista)
    return JsonResponse({"listado":posts_serialized})
def repositorio(request):
    return render (request,"lista_repositorio.html")
def lista_repositorio(request):
    lista=Repositorio.objects.all()
    posts_serialized = serializers.serialize('json', lista)
    return JsonResponse({"listado":posts_serialized})
def inicio(request):
    return render (request,'inicio.html')

```

- Archivo “apps.py”.

```
from django.apps import AppConfig
class VideosConfig(AppConfig):
    name = 'Videos'
```

- Archivo “wsgi.py”.

```
import os
from django.core.wsgi import get_wsgi_application
os.environ.setdefault("DJANGO_SETTINGS_MODULE", "Infochannel.settings")
application = get_wsgi_application()
```

## ANEXO 9: CÓDIGO DEL FRONTEND DEL SISTEMA DESARROLLADO EN HTML5, CCS3 Y JAVASCRIPT.

- Archivo “base.html”.

```
{% load staticfiles %}
<!DOCTYPE html>
<html lang="es">
  <head>
    {% block css %}
      <script src="https://code.jquery.com/jquery-3.1.0.min.js"></script>
    {% endblock %}
    <meta charset="utf-8">
    <title>{% block title%}TITULO{% endblock%}</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1,
minimum-scale=1, maximum-scale=1, user-scalable=0" />
    <link rel="shortcut icon" type="image/x-icon" href="{% static
'css/images/favicon.ico' %}" />
    <link rel="stylesheet" href="{% static 'css/style.css' %}" type="text/css"
media="all" />
    <link
href='https://fonts.googleapis.com/css?family=Raleway:400,900,800,700,600,500,
300,200,100' rel='stylesheet' type='text/css'>
    <script src="{% static 'js/jquery-1.8.0.min.js' %}"
type="text/javascript"></script>
    <script src="{% static 'js/jquery.carouFredSel-5.5.0-packed.js' %}"
type="text/javascript"></script>
    <script src="{% static 'js/functions.js' %}"
type="text/javascript"></script>
  </head>
  <body>
    {% block header %}
<div id="wrapper">
  <div class="shell">
    <div class="container">
      <header id="header">
        <h1 id="logo" ><strong><a href="/"></a></strong></h1>
        <div class="c1">&nbsp;</div>
      </header>
      <nav id="navigation">
        <a href="/" class="nav-btn">INICIO<span></span></a>
        <ul>
          <li class="active"><a href="/">Inicio</a></li>
          <li><a href='{% url "infochannel" %}'>Infochannel</a></li>
          <li><a href='{% url "repositorio" %}'>Repositorio</a></li>
        </ul>
        <div class="c1">&nbsp;</div>
      </nav>
    </div>
    {% endblock%}
    {% block content%}
    {% endblock%}
    {% block javascript %}
    {% endblock %}
    {% block footer%}
      <div id="footer">
        <div class="footer-nav">
          <ul>
            <li><a href="/">Inicio</a></li>
```



```

        <li><a href='{% url "infochannel" %}'>Infochannel</a></li>
        <li><a href='{% url "repositorio" %}'>Repositorio</a></li>
    </ul>
    <div class="cl">&nbsp;</div>
</div>
<p class="copy">&copy; Copyright 2019<span>|</span>Infochannel.
Design by <a href="https://www.facebook.com/dark.sldk" target="_blank">Luis
Cartagena Soto</a></p>
    <div class="cl">&nbsp;</div>
</div>
    {% endblock%}
</body>
</html>

```

- Archivo “inicio.html”.

```

{% extends 'base.html' %}
{% block title%}ECU 911{% endblock%}
{% block content %}
{% load staticfiles %}
<div class="cols">
    <div class="col1">
        <center>
            <strong>INFOCHANNEL</strong>
            <p><a href='{% url "infochannel" %}'></a></p>
        </center>
    </div>
    <div class="col2">
        <center>
            <strong>REPOSITORIO</strong>
            <p><a href='{% url "repositorio" %}'></a></p>
        </center>
    </div>
</div>
{% endblock %}

```

- Archivo “infochannel.html”.

```

{% extends 'base.html' %}
{% block title%}
INFOCHANNEL
{% endblock%}
{% block javascript %}
<script language="javascript">
    n_video=0;
    (function($) {
        var array=[];
        function actualizar(){
            $.ajax({
                url: '{% url "lista_infochannel" %}',
                data: form.serialize(),
                dataType: 'json',
                success: function (data) {
                    array =$.parseJSON(data.listado);
                    limite=array.length-1;
                }
            });
        }
    });

```

```

    }
    $( "#Atras" ).click(function() {
        actualizar();
        if (n_video>0){
            n_video--;
            <!--$("#video").attr("src","{{ MEDIA_URL
}}"+array[n_video].fields.video_src);-->
        }
        else {
            n_video=limite;
            <!--$("#video").attr("src","{{ MEDIA_URL
}}"+array[n_video].fields.video_src);-->
        }
        $("#video").attr("src","{{ MEDIA_URL
}}"+array[n_video].fields.video_src);
    });
    $( "#Siguiente" ).click(function() {
        actualizar();
        if (n_video<limite){
            n_video++;
            <!--$("#video").attr("src","{{ MEDIA_URL
}}"+array[n_video].fields.video_src);-->
        }
        else if (n_video>=limite){
            n_video=0;
            <!--$("#video").attr("src","{{ MEDIA_URL
}}"+array[n_video].fields.video_src);-->
        }
        $("#video").attr("src","{{ MEDIA_URL
}}"+array[n_video].fields.video_src);
    });
    $( "#Actualizar" ).click(function() {
        $.ajax({
            url: '% url "lista_infochannel" %',
            data: form.serialize(),
            dataType: 'json',
            success: function (data) {
                array = $.parseJSON( data.listado);
                limite=array.length-1;
            }
        });
        n_video=(-1);
    });
    $( "video" ).on('ended',function() {
        if (n_video==limite){
            n_video=0;
        }
        else {
            n_video++;
        }
        actualizar();
        $("#video").attr("src","{{ MEDIA_URL
}}"+array[n_video].fields.video_src);
    });
    $( "#Completa" ).click(function(){
        var a = $('#video')[0],
            fsReturn = a.requestFullscreen ? a.requestFullscreen() :
            a.webkitRequestFullscreen ? a.webkitRequestFullscreen() :
            a.mozRequestFullscreen ? a.mozRequestFullscreen() : false;
    });

```

```

        var vid = document.getElementById("video");
        vid.onerror = function() {
            actualizar();
            $("#video").attr("src","{{ MEDIA_URL
}}"+array[n_video].fields.video_src);
        };
        var form = $(this).closest("form");
        $.ajax({
            url: '{% url "lista_infochannel" %}',
            data: form.serialize(),
            dataType: 'json',
            success: function (data) {
                array =$.parseJSON( data.listado);
                limite=array.length-1;
                $("#video").attr("src","{{ MEDIA_URL
}}"+array[0].fields.video_src);
            }
        });
    })(jQuery);
</script>
{% endblock %}
{% block content %}
{% load staticfiles %}
<nav id="navigation">
    <div class="cl">&nbsp;</div>
    <ul >
        <center>
            <strong><button id="Atras"
class="control">ATRÁS</button></strong></li>
            <strong><button id="Actualizar"
class="control">ACTUALIZAR</button></strong></li>
            <strong><button id="Completa" class="control">PANTALLA
COMPLETA</button></strong></li>
            <strong><button id="Siguiete"
class="control">SIGUIENTE</button></strong></li>

        </center>
    </ul>
    <div class="cl">&nbsp;</div>
</nav>
<div id="container">
    <div class="reproductor">
        <video id="video" width="100%" height="100%" controls autoplay
preload="auto" poster="{% static 'css/images/ECU.jpg' %}">
            <source src="" type='video/mp4'>
            <source src="" type='video/webm'>
            <source src="" type='video/ogg'>
        </video>
    </div>
</div>
</div>
{% endblock %}

```

- Archivo “lista\_repositorio.html”.

```

{% extends 'base.html' %}
{% block title%}
REPOSITORIO
{% endblock%}
{% block javascript %}

```

```

<script language="javascript">
    (function($) {
        var array=[];
        function inicializar(){
            <!--var form = $(this).closest("form");-->
            $.ajax({
                url: '{% url "lista_repositorio" %}',
                data: form.serialize(),
                dataType: 'json',
                success: function (data) {
                    array =$.parseJSON( data.listado);
                    limite=array.length;
                }
            });
            for (i=0;i<=limite;i++){
                $("#listados").append("<li id='img'+i+'><div class='slide-
cnt'><h2 id='n'+i+'></h2></div><a href='#video'><img id='gprev'+i+'
width='290px' float='left' margin-right='50px' margin-left='30px' src='
alt=' ' onClick='reproducir("+i+")' /></a></li>");
                $("#thumbs").append("<a href='#img'+i+'><img id='prev'+i+'
src=' ' /></a>");
                $("#prev"+i).attr("src","{{ MEDIA_URL
}}"+array[i].fields.imagen_src);
                $("#gprev"+i).attr("src","{{ MEDIA_URL
}}"+array[i].fields.imagen_src);
                document.querySelector('#n'+i).textContent =
array[i].fields.titulo;
            }
        }
        function actualizar(){
            $("#listados").html("");
            $("#thumbs").html("");
            $.ajax({
                url: '{% url "lista_repositorio" %}',
                data: form.serialize(),
                dataType: 'json',
                success: function (data) {
                    array =$.parseJSON( data.listado);
                    limite=array.length-1;
                    for (i=0;i<=limite;i++){
                        $("#listados").append("<li id='img'+i+'><div class='slide-
cnt'><h2 id='n'+i+'></h2></div><a href='#video'><img id='gprev'+i+'
width='290px' float='left' margin-right='50px' margin-left='30px' src='
alt=' ' onClick='reproducir("+i+")' /></a></li>");
                        $("#thumbs").append("<a href='#img'+i+'><img id='prev'+i+'
src=' ' /></a>");
                        $("#prev"+i).attr("src","{{ MEDIA_URL
}}"+array[i].fields.imagen_src);
                        $("#gprev"+i).attr("src","{{ MEDIA_URL
}}"+array[i].fields.imagen_src);
                        document.querySelector('#n'+i).textContent =
array[i].fields.titulo;
                        $("#video"+i).attr("src","{{ MEDIA_URL
}}"+array[i].fields.imagen_src);
                    }
                }
            });
        }
        $("#actualizar" ).click(function() {
            $.ajax({
                url: '{% url "lista_repositorio" %}',
                data: form.serialize(),

```

```

        dataType: 'json',
        success: function (data) {
            array = $.parseJSON( data.listado);
            limite=array.length;
            $("#listados").html("");
            $("#thumbs").html("");
            for (i=0;i<limite;i++){
                $("#listados").append("<li id='img'+i+'><div class='slide-
cnt'><h2 id='n'+i+'></h2></div><a href='#video'><img id='gprev'+i+'
width='290px' float='left' margin-right='50px' margin-left='30px' src='
alt=' ' onClick='reproducir("+i+")' /></a></li>");
                $("#thumbs").append("<a href='#img'+i+'><img id='prev'+i+'
src=' ' /></a>");
                $("#prev"+i).attr("src","{{ MEDIA_URL
}}"+array[i].fields.imagen_src);
                $("#gprev"+i).attr("src","{{ MEDIA_URL
}}"+array[i].fields.imagen_src);
                document.querySelector('#n'+i).textContent =
array[i].fields.titulo;
                $("#video"+i).attr("src","{{ MEDIA_URL
}}"+array[i].fields.imagen_src);
            }
        }
    });
});
window.reproducir = function(numero) {
    $("#video").attr("src","{{ MEDIA_URL
}}"+array[numero].fields.video_src);
    document.querySelector('#ttt').textContent =
array[numero].fields.titulo;
};
$( "#video" ).error(function() {
    actualizar();
});
$( "video" ).on('ended',function() {
    actualizar();
});
var form = $(this).closest("form");
$.ajax({
    url: '{% url "lista_repositorio" %}',
    data: form.serialize(),
    dataType: 'json',
    success: function (data) {
        array = $.parseJSON( data.listado);
        limite=array.length-1;
        inicializar();
    }
});
})(jQuery);
</script>
{% endblock %}
{% block content %}
{% load staticfiles %}
<center>
<strong><button id="actualizar" class="control">ACTUALIZAR
VIDEOS</button></strong>
</center>
<div class="cl">&nbsp;</div>
<br/>
<div class="slider-holder">
<div class="slider">
<div class="socials">

```

```

        <a target="_blank" href="https://es-la.facebook.com/ECU911/"
class="facebook-ico">facebook-ico</a>
        <a target="_blank"
href="https://twitter.com/ECU911Loja?lang=es" class="twitter-ico">twitter-
ico</a>
        <div class="cl">&nbsp;</div>
    </div>
    <div class="arrs">
        <a href="#" class="prev-arr"></a>
        <a href="#" class="next-arr"></a>
    </div>
    <ul id="listados">
    </ul>
</div>
<div><br></div>
<div id="thumbs-wrapper">
    <div id="thumbs">
    </div>
    <a id="prev" href="#"></a>
    <a id="next" href="#"></a>
</div>
</div>
<div><h2><center><strong>REPRODUCTOR</strong></center></h2></div>
<div id="container">
    <div class="reproductor">
        <video id="video" width="100%" height="100%" controls autoplay
preload="auto" poster="{% static 'css/images/ECU.jpg' %}">
            <source src="" type="video/mp4">
            <source src="" type="video/webm">
            <source src="" type="video/ogg">
        </video>
        <p id="seleccionada"></p>
    </div>
</div>
<div><p id="ttt">TITULO</p></div>
{% endblock %}

```

- Archivo “base.html” (templates/admin/base.html).

```

{% load i18n static %}<!DOCTYPE html>
{% get_current_language as LANGUAGE_CODE %}{% get_current_language_bidi as
LANGUAGE_BIDI %}
<html lang="{% LANGUAGE_CODE|default:"en-us" %}" {% if LANGUAGE_BIDI
%}dir="rtl"{% endif %}>
<head>
<title>{% block title %}{% endblock %}</title>
<link rel="stylesheet" type="text/css" href="{% block stylesheet %}{% static
"admin/css/base.css" %}{% endblock %}">
{% block extrastyle %}{% endblock %}
{% if LANGUAGE_BIDI %}<link rel="stylesheet" type="text/css" href="{% block
stylesheet_rtl %}{% static "admin/css/rtl.css" %}{% endblock %}">{% endif %}
{% block extrahead %}{% endblock %}
{% block responsive %}
    <meta name="viewport" content="user-scalable=no, width=device-width,
initial-scale=1.0, maximum-scale=1.0">
    <link rel="stylesheet" type="text/css" href="{% static
"admin/css/responsive.css" %}">
    {% if LANGUAGE_BIDI %}<link rel="stylesheet" type="text/css" href="{%
static "admin/css/responsive_rtl.css" %}">{% endif %}
{% endblock %}
{% block blockbots %}<meta name="robots" content="NONE,NOARCHIVE">{% endblock

```

```

%}
</head>
{% load i18n %}
<body class="{% if is_popup %}popup {% endif %}{% block bodyclass %}{%
endblock %}"
  data-admin-utc-offset="{% now "Z" %}">
<!-- Container -->
<div id="container">
  {% if not is_popup %}
  <!-- Header -->
  <div id="header">
    <div id="branding">
      {% block branding %}{% endblock %}
    </div>
    {% block usertools %}
    {% if has_permission %}
    <div id="user-tools">
      {% block welcome-msg %}
      {% trans 'Welcome,' %}
      <strong>{% firstof user.get_short_name user.get_username
%}</strong>.
      {% endblock %}
      {% block userlinks %}
      {% if site_url %}
      <a href="{{ site_url }}">{% trans 'View site' %}</a> /
      {% endif %}
      {% if user.is_active and user.is_staff %}
      {% url 'django-admindocs-docroot' as docsroot %}
      {% if docsroot %}
      <a href="{{ docsroot }}">{% trans 'Documentation'
%}</a> /
      {% endif %}
      {% endif %}
      {% if user.has_usable_password %}
      <a href="{% url 'admin:password_change' %}">{% trans 'Change
password' %}</a> /
      {% endif %}
      <a href="{% url 'admin:logout' %}">{% trans 'Log out' %}</a>
      {% endblock %}
    </div>
    {% endif %}
    {% endblock %}
    {% block nav-global %}{% endblock %}
  </div>
  <!-- END Header -->
  {% block breadcrumbs %}
  <div class="breadcrumbs">
  <a href="{% url 'admin:index' %}">{% trans 'Home' %}</a>
  {% if title %} &rsquo; {{ title }}{% endif %}
  </div>
  {% endblock %}
  {% endif %}
  {% block messages %}
  {% if messages %}
  <ul class="messagelist">{% for message in messages %}
    <li{% if message.tags %} class="{{ message.tags }}"{% endif %}>{{
message|capfirst }}</li>
  {% endfor %}</ul>
  {% endif %}
  {% endblock messages %}

  <!-- Content -->

```

```

<div id="content" class="{% block coltype %}colM{% endblock %}">
  {% block pretitle %}{% endblock %}
  {% block content_title %}{% if title %}<h1>{{ title }}</h1>{% endif
%}{% endblock %}
  {% block content %}
  {% block object-tools %}{% endblock %}
  {{ content }}
  {% endblock %}
  {% block sidebar %}{% endblock %}
  <br class="clear">
</div>
<!-- END Content -->
  {% block footer %}<div id="footer"></div>{% endblock %}
</div>
<!-- END Container -->
</body>
</html>

```

- Archivo “base\_site.html”.

```

{% extends "admin/base.html" %}
{% block title %}{{ title }} | {{ site_title|default:_('Django site admin')
}}{% endblock %}
{% block branding %}
<h1 id="site-name"><a href="{% url 'admin:index' %}"><strong>INFOCHANNEL - SIS
ECU 911</strong></a></h1>
{% endblock %}
{% block nav-global %}{% endblock %}

```

- Archivo “functions.js”.

```

var sliderDuration = 600;
$(function() {
  $(document).on('focusin', '.field, textarea', function() {
    if(this.title==this.value) {
      this.value = '';
    }
  }).on('focusout', '.field, textarea', function(){
    if(this.value=='') {
      this.value = this.title;
    }
  });
  $(' .main .col:last, .main .entry:last, .footer-nav ul
li:last').addClass('last');
  $(' .main .cols .col ul li:first-child, #navigation ul li:first-
child').addClass('first');
  $(' .main .cols .col ul li:odd').addClass('odd');
  // mobile-navigation
  $('#navigation a.nav-btn').click(function(){
    $(this).toggleClass('active');
    $(this).parent().find('ul').slideToggle();
  });
  $('#thumbs a:visible:first').addClass('selected');
});
$(window).load(function(){
  //slider thumb
  $('.slider ul').carouFredSel({
    align: 'center',
    circular: true,

```



```

    auto: true,
    items: {
        visible: 1,
    },
    scroll: {
        duration: sliderDuration,

        onAfter: function(data){
            data.items.old.removeClass('selected');
            $(this).trigger("currentPosition", function( pos ) {
                $('#thumbs').find('a').eq(pos).addClass('selected').siblings().removeClass('selected');
            });
        }
    },
    prev: '.prev-arr',
    next: '.next-arr'
});
$('#thumbs').carouFredSel({
    align: false,
    scroll: {
        items: 1,
        duration: sliderDuration
    },
    items: {
        visible: 4,
    },
    auto: false,
    infinite: true,
    prev: '#prev',
    next: '#next'
});
$('#thumbs a').click(function() {
    $('.slider ul').trigger('slideTo', '#' +this.href.split('#').pop());
    $('#thumbs a.selected').removeClass('selected');
    $(this).addClass('selected');
    return false;
});
});
$(window).resize(function() {
    carosel();
})
function carosel(){
    //slider thumb
    $('.slider ul').carouFredSel({
        align: 'center',
        circular: true,
        auto: false,
        items: {
            visible: 1,
        },
        prev: '.prev-arr',
        next: '.next-arr'
    });

    $('#thumbs').carouFredSel({
        align: false,
        scroll: {
            items: 1,
        },
        items: {
            visible: 4,
        }
    });
}

```

```

    },
    auto: false,
    infinite: true,
    prev: '#prev',
    next: '#next'
  });
  $('#thumbs a').click(function() {
    $('.slider ul').trigger('slideTo', '#' + this.href.split('#').pop());
    $('#thumbs a.selected').removeClass('selected');
    $(this).addClass('selected');
    return false;
  });
}

```

- Archivo “logged\_out.html”.

```

{% extends "admin/base_site.html" %}
{% load i18n %}
{% block breadcrumbs %}<div class="breadcrumbs"><a href="{% url 'admin:index' %}">{% trans 'Home' %}</a></div>{% endblock %}
{% block content %}
<p>{% trans 'Gracias por acceder a la interfaz de administración del
"Infochannel - SIS ECU 911".' %}</p>
<p><a href="{% url 'admin:index' %}">{% trans 'Log in again' %}</a></p>
{% endblock %}

```

- Archivo “style.css”.

```

* { margin: 0; padding: 0; outline: 0; }
body, html { height: 100%; }
body {
  font-size: 12px;
  line-height: 22px;
  font-family: verdana, arial, sans-serif;
  color: #727272;
  background: url(images/body-pattern.png) repeat 0 0;
  min-width: 100%;
}
/*font-family: 'Raleway', sans-serif;*/
a { color: #00acef; text-decoration: none; cursor: pointer; }
a:hover { text-decoration: underline; }
a img { border: 0; }
article, aside, details, footer, header, menu, nav, section { display: block; }
input, textarea, select { font-size: 12px; font-family: arial, sans-serif; }
textarea { overflow: auto; }
.cl { display: block; height: 0; font-size: 0; line-height: 0; text-indent: -4000px; clear: both; }
.notext { font-size: 0; line-height: 0; text-indent: -4000px; }
.left, .alignleft { float: left; display: inline; }
.right, .alignright { float: right; display: inline; }
.shell { width: 981px; margin: 0 auto; }
.container { background: url(images/container.png) repeat-y center 0; box-shadow: 0px 0px 0px 4px rgba(0,0,0,0.05); -moz-box-shadow: 0px 0px 0px 4px rgba(0,0,0,0.05); -webkit-box-shadow: 0px 0px 0px 4px rgba(0,0,0,0.05); -o-box-shadow: 0px 0px 0px 4px rgba(0,0,0,0.05); }
#header { margin-bottom: 15px; position: relative; padding-top: 25px; height: 43px; background: url(images/border-title.png) repeat-x 0 bottom; }
#logo { width: 180px; font-size: 0; line-height: 0; position: absolute; top:

```

```

15px; left: 21px; }
#logo a { height: 50px; display: block; background: url(images/logo.png) no-repeat 0 0; text-indent: -4000px; }
.search { width: 300px; margin-left: auto; padding-right: 21px; }
.search input.field { float:left; width: 166px; padding:0 8px; height: 24px; line-height: 24px; border: 1px solid #d3d3d3; font-size: 11px; color:#bebebe; background: #ededed; border-radius: 5px; -moz-border-radius: 5px; -webkit-border-radius: 5px; -o-border-radius: 5px; }
.search input.search-btn { float:left; font-size: 0; line-height: 0; text-indent: -4000px; cursor: pointer; width: 20px; height: 26px; border: 0; background: url(images/search-ico.png) no-repeat 0 0; }
#navigation { padding:0 21px; margin-bottom: 15px; }
#navigation ul { list-style:none; list-style-position: outside; }
#navigation ul li { padding: 0 10px; float: left; font-family: 'Raleway', sans-serif; font-size: 15px; font-weight: 500; }
#navigation ul li:first-child { padding-left: 0; }
#navigation ul li a { color: #4a4a4a; padding: 0 7px; display:block; height: 21px; line-height: 21px; border: 2px solid transparent; border-radius: 3px; -moz-border-radius: 3px; -webkit-border-radius: 3px; -o-border-radius: 3px; }
#navigation ul li.active a,
#navigation ul li a:hover { border: 2px solid #029cdb; background: url(images/nav-btn.png) repeat-x 0 0; color:#fff; text-decoration: none; }
#navigation a.nav-btn { display:none; }
.slider-holder { position: relative; margin-bottom:30px; }
.slider { background: url(images/slider-bg.jpg) no-repeat 0 0; width: 938px; height: 417px; position:relative; overflow: hidden; padding:0 21px; }
.slider ul { list-style:none; list-style-position: outside; }
.slider ul li { float: left; width: 938px; height: 417px; }
.slider .slide-cnt { width: 450px; height: 220px; position:relative; overflow: hidden; float: right; padding-top: 30px; }
.slider h4 { color:#fff; font-size: 16px; font-family: 'Raleway', sans-serif; font-weight: 500; text-shadow: rgba(0,0,0,0.5) 0px 1px 0px; text-transform: uppercase; }
.slider h2 { color:#f8f8f8; font-family: 'Raleway', sans-serif; font-size: 56px; line-height:56px; padding-bottom: 12px; font-weight: 700; text-shadow: rgba(0,0,0,0.5) 0px 1px 0px; }
.slider p { color:#fff; }
.slider p a { color:#fff; text-decoration: underline; }
.slider p a:hover { text-decoration: none; }
.slider img { float:left; margin-right:50px; margin-left: 30px; }
.slider .socials { width: 158px; margin-left: auto; padding-top:18px; }
.slider .socials a { width: 31px; height: 30px; float:left; margin-left: 6px; background: url(images/socials.png) no-repeat 0 0; font-size: 0; line-height: 0; text-indent: -4000px; }
.slider .socials a.facebook-ico { background-position: 0 0; }
.slider .socials a.facebook-ico:hover { background-position:0 bottom; }
.slider .socials a.twitter-ico { background-position:-37px 0; }
.slider .socials a.twitter-ico:hover { background-position:-37px bottom; }
.slider .socials a.skype-ico { background-position:-75px 0; }
.slider .socials a.skype-ico:hover { background-position:-75px bottom; }
.slider .socials a.rss-ico { background-position: -114px 0; }
.slider .socials a.rss-ico:hover { background-position: -114px bottom; }
#thumbs-wrapper { width: 434px; padding: 0 20px; position: absolute; bottom: 18px; right: 21px; overflow: hidden; }
#thumbs-wrapper a#prev { width: 12px; height: 17px; position:absolute; top: 50%; margin-top: -8px; left: 0; background: url(images/left-arr.png) no-repeat 0 0; }
#thumbs-wrapper a#next { width: 12px; height: 17px; position: absolute; top: 50%; margin-top: -8px; right: 0px; background: url(images/right-arr.png) no-repeat 0 0; }
#thumbs { height: 80px; width: 434px; position:relative; overflow: hidden; padding-top:7px; }

```

```

#thumbs a { margin-bottom:8px; width: 94px; height: 69px; margin-right: 8px;
float:left; background: #fff; padding: 3px; border-radius: 4px; -moz-border-
radius: 4px; -webkit-border-radius: 4px; -o-border-radius: 4px; border: 1px
solid #d5d5d5; }
#thumbs a img { width: 94px; height: 69px; float:left; }
#thumbs a.selected { height: 76px; position:relative; margin-top: -7px;
opacity: 1; }
#thumbs a.selected img { height: 76px; }
.main { padding:0 20px; }
.featured { background: #f1f1f1; border: 1px solid #e5e5e5; padding: 10px
213px 10px 18px; margin-bottom: 42px; position: relative; border-radius: 5px;
-moz-border-radius: 5px; -webkit-border-radius: 5px; -o-border-radius: 5px; }
.featured h4 { font-family: 'Raleway', sans-serif; line-height: 43px; font-
size: 17px; color:#4a4a4a; font-weight:400; }
.featured h4 strong { color:#00acef; font-weight: 600; }
.featured a.blue-btn { background: url(images/blue-btn.png) repeat 0 0; width:
189px; height: 39px; text-shadow: rgba(0,0,0,0.5) 0px 1px 0px; position:
absolute; top: 50%; margin-top: -20px; right: 9px; display: block; font-size:
16px; font-weight: 500; color:#fff; text-align: center; line-height: 39px;
border: 2px solid #0a7fb5; border-radius: 6px; -moz-border-radius: 6px; -
webkit-border-radius: 6px; -o-border-radius: 6px; text-decoration: none; font-
weight: 500; font-family: 'Raleway', sans-serif; text-shadow: rgba(0,0,0,0.5)
0px 1px 1px;
}
.featured a.blue-btn:hover { background-position:0 bottom; }
.main h3 { font-size: 16px; font-family: 'Raleway', sans-serif; color:#4a4a4a;
font-weight: 500; padding-bottom: 12px; margin-bottom: 8px; background:
url(images/border-title.png) repeat-x 0 bottom;}
.main h5 { font-size: 12px; }
.main h5 { padding-bottom: 10px; }
.main a.more { background: url(images/blue-arr.png) no-repeat 0 6px; text-
decoration: underline; padding-left: 8px; font-weight: 500; font-family:
'Raleway', sans-serif; }
.main a:hover { text-decoration: none; }
.main .cols { padding-bottom: 38px; }
.main .cols .col { float:left; width: 286px; padding-right: 40px; }
.main .cols .col.last { padding-right: 0; }
.main .cols .col img.left { margin-right: 24px; margin-bottom:10px; }
.main .cols .col a { text-decoration: underline; }
.main .cols .col ul { list-style:none; list-style-position: outside; }
.main .cols .col ul li { padding: 10px 10px 12px; background:
url(images/border-title.png) repeat-x 0 bottom; font-family: 'Raleway', sans-
serif; font-weight:600; }
.main .cols .col ul li.odd { background-color:#f6f6f6; }
.main .cols .col ul li a { padding: 0px 10px; display:block; background:
url(images/bullet.png) no-repeat 0px 9px; }
.main .cols .col ul li.first { padding-top:2px; }
.main .entries { background: url(images/entries.png) repeat-y 0 0;
position:relative; margin: 0 -20px; padding: 20px; margin-bottom: 1px; border-
bottom: 1px solid #e8e8e8; }
.main .entries h3 { margin-bottom: 15px; background: url(images/entry-
title.png) repeat-x 0 bottom; }
.main .entries .entry { width: 286px; float:left; padding-right: 40px; }
.main .entries .entry.last { padding-right:0; }
.main .entries .entry p.meta { color: #cbcbcb; }
.main .entries .entry p.meta a { color: #cbcbcb; }
.main .entries .entry-inner { padding-bottom: 18px; margin-bottom: 16px;
background: url(images/entry-title.png) repeat-x 0 bottom; }
.main .entries .entry .date { margin-right: 15px; position: relative;
float:left; padding: 14px 12px; width: 40px; height: 34px; background:
url(images/date-ico.png) no-repeat 0 0; }
.main .entries .entry .date strong { color:#00acef; font-size: 22px; line-

```

```

height: 22px; font-family: arial, sans-serif; font-weight: bold;
display:block; }
.main .entries .entry .date span { font-weight: bold; font-size: 9px;
color:#c6c6c6; position:relative; margin-top: 0px; display:block; line-
height:9px; }
.main .entries .entry .date em { -webkit-transform: rotate(-90deg); line-
height: 12px;
-moz-transform: rotate(-90deg); font-weight: bold; font-style: normal;
width: 12px; color:#bdbdbd; font-size: 12px; position:absolute; bottom: 20px;
left: 41px; }
.testimonials { padding-bottom: 30px; }
.testimonials p { font-style: italic; font-family: georgia ,arial, serif;
font-size: 14px; color:#898989; }
.testimonials p strong { font-size: 43px; position:relative; bottom: -19px;
font-weight: normal; padding-right: 4px; }
.testimonials p.author { font-size: 12px; text-align: right; }
.testimonials p.author strong { font-size: 12px; bottom: 0; font-weight:bold;
}
#footer { background: url(images/footer-border.png) repeat-x 0 0; padding:
18px 20px 18px 10px; }
.footer-nav { float: left; }
.footer-nav ul { list-style:none; list-style-position: outside; }
.footer-nav ul li { float: left; border-right: 1px solid #afaca9; padding:0
10px; font-size: 10px; line-height: 10px; }
.footer-nav ul li a { color: #afaca9; }
.footer-nav ul li a:hover { color: #878684; text-decoration: none; }
.footer-nav ul li.last { border-right: 0; }
#footer p.copy { font-size: 10px; color:#afaca9; line-height: 10px; float:
right; }
#footer p.copy a { font-size: 10px; color:#00acef; line-height: 10px; text-
decoration: underline; }
#footer p.copy a:hover { text-decoration: none; }
#footer p.copy span { padding:0 7px; }
/* ipad portrait */
@media only screen and ( min-width: 768px) and ( max-width: 981px ) {
body { width: auto; }
.shell { width: auto; width: 768px; }
.container { background-image:url(images/container-tablet.png); }
.slider { width: 725px; height: 366px; background-image:url(images/slider-
tablet.png); }
.slider ul li { width: 724px; }
.slider img { width: 316px; height: 286px; margin-left: 0; margin-right:
20px; }
.slider h3 { font-size:16px; }
.slider h2 { font-size: 47px; line-height: 54px; }
.slider .slide-cnt { width: 380px; height: 190px; }
#thumbs-wrapper { width: 356px; }
#thumbs a { width: 75px; height: 56px; }
#thumbs a.selected { width: 75px; height: 65px; }
#thumbs a.selected img { width: 75px; height: 65px; }
#thumbs a img { width: 75px; height: 56px; }
.featured h4 { font-size: 16px; line-height: 28px; }
.main .cols .col { width: 222px; padding-right: 30px; }
.main .cols .col img.left { float: none; display: block; margin:0 auto
12px; }
.main .entries { background-image:url(images/entries-tablet.png); }
.main .entries .entry { width: 225px; padding-right: 24px; }
.main .entries .entry img { width: 100%; }
.main .entries .entry .cnt { height:100%; overflow: hidden; }
.footer-nav { float:none; display: block; padding-bottom: 12px; }
#footer p.copy { float: none; display:block; padding-left: 10px; }
}

```

```

/* mobile*/
@media only screen and ( max-width: 640px) {
  body { width: auto; }
  .search { display:none; }
  #logo { position:relative; margin:0 auto; top: 0px; left:0; }
  #header { background: transparent; margin-bottom:0; }
  #navigation { position:relative; z-index: 100; padding:0 0px; margin:0
10px; }
  #navigation a.nav-btn { margin-bottom: 15px; text-decoration: none;
padding:0 36px 0 10px; line-height:30px; display:block; background:
url(images/navigation.png) repeat-x 0 0; height: 30px; position: relative; }
  #navigation a.nav-btn span { background: url(images/dd-nav-arrs.png) no-
repeat 0 bottom; width: 20px; height: 15px; position:absolute; top: 8px;
right: 12px; }
  #navigation a.nav-btn.active span { background-position:0 0; }
  #navigation ul { display:none; position: absolute; top: 30px; left: 0;
width: 100%; }
  #navigation ul li { float:none; height: 30px; border-top: 1px solid #fff;
display:block; padding:0 0 0 0; background: url(images/navigation.png) repeat-
x 0 0; }
  #navigation ul li a { border: 0; line-height:30px; float:none; height:
30px; display: block; font-size: 14px; padding-left: 20px; padding-right:
20px; width:100%; border-radius: 0px; -moz-border-radius: 0px; -webkit-border-
radius: 0px; -o-border-radius: 0px; box-sizing:border-box; -moz-box-
sizing:border-box; -webkit-box-sizing:border-box; }
  #navigation ul li a:hover { background-image:url(images/blue-btn.png);
border:0; }
  #navigation ul li.active span { background: transparent; border:0; }
  #navigation ul li.first { display:none; }
  .shell { width: 320px; }
  .container { background-image: url(images/container-mobile.png); width:
320px; }
  .main { padding:0 10px; }
  .slider ul li { width: 724px; }
  .slider img { width: 316px; height: 286px; float:none; display:block;
margin: 0 auto; }
  .slider h3 { font-size:16px; }
  .slider h2 { font-size: 47px; line-height: 54px; }
  .slider .slide-cnt { width: 380px; height: 190px; float:none;
display:block; }
  #thumbs-wrapper { display: none; }
  .slider-holder { width: 320px; margin:0 auto; }
  .slider { background-image:url(images/slider-mobile.png); padding:0 10px;
width:300px; margin:0 auto; height: 388px;}
  .slider .slide-cnt { width: 300px; float:none; display:block; padding-top:
12px; }
  .slider h4 { display:none; }
  .slider h2 { font-size: 38px; text-align: center; }
  .slider p { color:#cee7f1; }
  .slider ul li { width: 300px; }
  .slider .socials { display:none; }
  .slider img { float:none; display:block; margin:0 auto; width: 194px;
height: 178px; }
  .slider .arrs a { position:absolute; top: 250px; z-index: 30; background:
url(images/slider-arr.png) no-repeat 0 0; width: 26px; height: 37px; }
  .slider .arrs a.prev-arr { left: 10px; background-position:0 0; }
  .slider .arrs a.next-arr { right: 10px; background-position: right 0; }
  .main .cols .col { padding-bottom: 20px; float:none; display:block;padding-
right:0; width: auto; }
  .main .cols .col img.left { float:none; display:block; margin: 0 auto; }
  .main .entries { margin:0 -10px; }
  .main .entries .entry { width: auto; padding-right:0; float:none;

```

```

display:block; margin-bottom:14px; }
.featured h4 { line-height: 26px; padding-bottom: 6px; }
.featured a.blue-btn { float: none; display:block; right:0; left:0;
position:relative; margin: 0 auto; top:0; width: 260px; }
.featured { padding: 10px; }
#footer { padding-left: 0; padding-right:0; }
.footer-nav { float:none; display:block; height: 12px; margin:0 auto;
width: 310px; vpadding-bottom: 8px; text-transform: uppercase; }
.footer-nav ul li { font-size: 8px; border-right:0; padding:0 2px; }
#footer p.copy { font-size: 8px; text-align:center; float:none;
display:block; }
}
@media only screen and ( max-width: 767px) and (-webkit-min-device-pixel-
ratio: 1.5), only screen and (min--moz-device-pixel-ratio: 1.5), only screen
and (min-resolution: 240dpi) {
#navigation a.nav-btn { background-image:url(images/navigation@2x.png); -
webkit-background-size: 2px 30px; -moz-background-size: 2px 30px; background-
size: 2px 30px; }
}
#ttt {
font-family: georgia, serif;
font-size: 20px;
text-align: center;
}
.control {
background:-webkit-gradient( linear, left top, left bottom, color-
stop(0.05, #3d94f6), color-stop(1, #1e62d0) );
background:-moz-linear-gradient( center top, #3d94f6 5%, #1e62d0 100% );
filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#3d94f6',
endColorstr='#1e62d0');
background-color:#3d94f6;
-webkit-border-top-left-radius:42px;
-moz-border-radius-topleft:42px;
border-top-left-radius:42px;
-webkit-border-top-right-radius:42px;
-moz-border-radius-topright:42px;
border-top-right-radius:42px;
-webkit-border-bottom-right-radius:42px;
-moz-border-radius-bottomright:42px;
border-bottom-right-radius:42px;
-webkit-border-bottom-left-radius:42px;
-moz-border-radius-bottomleft:42px;
border-bottom-left-radius:42px;
text-indent:0px;
border:2px solid #337fed;
display:inline-block;
color:#ffffff;
font-family:Arial;
font-size:13px;
font-weight:bold;
font-style:normal;
height:30px;
line-height:30px;
width:200px;
text-decoration:none;
text-align:center;}
.control:hover {
background:-webkit-gradient( linear, left top, left bottom, color-
stop(0.05, #1e62d0), color-stop(1, #3d94f6) );
background:-moz-linear-gradient( center top, #1e62d0 5%, #3d94f6 100% );
filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#1e62d0',

```

```
endColorstr='#3d94f6');  
  background-color:#1e62d0;}  
.control:active {  
  position:relative;  
  top:1px;}  
.cols {display: flex;}  
.cols div {flex: 1;}  
.col1 {background-color: none;}  
.col2 {background-color: none;}  
p {margin: 1em;}
```