

**Universidad Nacional Autónoma de Nicaragua, UNAN – León**  
**Facultad de Ciencias y Tecnología**  
**Ingeniería en Telemática**



**Prototipo de aplicación web alojada en una Raspberry Pi 3  
modelo B para el acceso gratuito a video bajo demanda por  
Internet**

**Monografía para optar al título de Ingeniero en Telemática**

Autores:

Br. Modesto Francisco Herrera Rivera.

Br. Katherine Taniuska Saldaña Mendoza.

Br. José Levy Salinas Pérez.

Tutor: M.Sc. Denis Espinoza Hernández

León, Nicaragua

Diciembre de 2018

---



---

## DEDICATORIA

El presente trabajo está dedicado a:

Mis padres Modesto Francisco Herrera Zeledón y Merling del Socorro Rivera Hernández por su apoyo incondicional y motivación para seguir adelante.

*Modesto Francisco Herrera Rivera*

Mis padres Pedro Ariel Saldaña Medina y María Esperanza Mendoza, por su amor, trabajo y sacrificio.

*Katherine Taniuska Saldaña Mendoza*

Marcelino Jesús Salinas González y Rosario Alina Pérez Mendoza, mis padres, por creer en mí y ser el soporte en mis metas.

*José Levy Salinas Pérez*

Gracias a nuestros padres ha sido posible llegar hasta la culminación de nuestras carreras, el comienzo de una nueva etapa y oportunidades en el mundo laboral.

Nuestros hermanos, por estar presentes y brindarnos su apoyo incondicional ante cualquier dificultad que enfrentemos.

Todas las personas que nos brindaron su apoyo para la culminación de este trabajo.

---



---

## AGRADECIMIENTO

Agradecemos este trabajo a:

Jehová Dios, por bendecir nuestro camino, guiarnos a lo largo de nuestra vida, ser apoyo y fortaleza en todo momento.

Nuestro tutor, Máster Denis Leopoldo Espinoza Hernández, de forma especial, por su paciencia, motivación y ayuda para lograr la finalización de este proyecto.

Nuestros docentes, por compartir sus conocimientos con responsabilidad y entusiasmo.

Máster Eduardo Santiago Molina Poveda, por brindarnos siempre su apoyo y su tiempo de forma generosa y desinteresada.

---



---

## Índice general

<b>1</b>	<b>INTRODUCCIÓN.....</b>	<b>1</b>
1.1	ANTECEDENTES.....	3
1.2	PLANTEAMIENTO DEL PROBLEMA .....	5
1.3	JUSTIFICACIÓN .....	7
	<i>Originalidad</i> .....	7
	<i>Alcance</i> .....	7
	<i>Impacto</i> .....	7
	<i>Producto</i> .....	7
1.4	OBJETIVOS.....	8
	<i>Objetivo general</i> .....	8
	<i>Objetivos específicos</i> .....	8
<b>2</b>	<b>MARCO TEÓRICO.....</b>	<b>9</b>
2.1	INTERNET VS TELEVISIÓN CONVENCIONAL.....	9
2.2	USABILIDAD DE LOS SITIOS .....	10
2.3	RASPBERRY .....	12
	<i>Raspberry Model B</i> .....	12
2.4	TECNOLOGÍAS .....	13
	<i>Desarrollo web</i> .....	13
	<i>API Web</i> .....	18
	<i>Frameworks</i> .....	19

---



---

<b>3</b>	<b>DISEÑO METODOLÓGICO.....</b>	<b>20</b>
3.1	DESCRIPCIÓN GENERAL DEL PROYECTO.....	20
3.2	ETAPAS DE TRABAJO.....	22
	<i>Obtención de los contenidos.....</i>	<i>22</i>
	<i>Aplicación web de selección de contenidos.....</i>	<i>22</i>
	<i>Desarrollo del control remoto para la aplicación.....</i>	<i>22</i>
	<i>Requisitos.....</i>	<i>23</i>
<b>4</b>	<b>RESULTADOS.....</b>	<b>24</b>
4.1	WEB CRAWLER.....	24
4.2	DISEÑO DE LA INTERFAZ Y FUNCIONES PRINCIPALES .....	26
4.3	CONTROL REMOTO.....	29
4.4	RESULTADOS FINALES.....	31
<b>5</b>	<b>CONCLUSIONES.....</b>	<b>33</b>
<b>6</b>	<b>RECOMENDACIONES .....</b>	<b>34</b>
<b>7</b>	<b>BIBLIOGRAFÍA.....</b>	<b>35</b>
<b>8</b>	<b>ANEXOS .....</b>	<b>39</b>
8.1	COSTOS DEL PROYECTO .....	39
8.2	EJEMPLO DE WEB CRAWLER.....	39
8.3	MANUAL DE USUARIO .....	42

---



---

## Índice de ilustraciones

ILUSTRACIÓN 1: ESQUEMA DE FUNCIONAMIENTO .....	21
ILUSTRACIÓN 2: UBICACIÓN DEL ENLACE AL VIDEO.....	24
ILUSTRACIÓN 3: CÓDIGO PARA OBTENER ENLACES DINÁMICAMENTE .....	25
ILUSTRACIÓN 4: ARCHIVO JSON GENERADO DINÁMICAMENTE.....	26
ILUSTRACIÓN 5: VALIDACIÓN DE CONTENIDO A MOSTRAR .....	27
ILUSTRACIÓN 6: WEB SOCKET .....	27
ILUSTRACIÓN 7: CONTROLADOR Y ÓRDENES QUE VIAJAN A TRAVÉS DEL WEB SOCKET .....	28
ILUSTRACIÓN 8: CONTROLADOR DE EVENTOS ANGULARJS .....	29
ILUSTRACIÓN 9: CÓDIGO HTML DEL CONTROL .....	30
ILUSTRACIÓN 10: FUNCIONES DEFINIDAS PARA EL CONTROL.....	30
ILUSTRACIÓN 11: ACTIVACIÓN DEL INPUT BÚSQUEDA .....	30
ILUSTRACIÓN 12: CÓDIGO QR DINÁMICO PARA OBTENCIÓN DE CONTROL REMOTO .....	31
ILUSTRACIÓN 13: CONTROL.....	32
ILUSTRACIÓN 14: PÁGINA PRINCIPAL.....	32
ILUSTRACIÓN 15: EJEMPLO DE CRAWLER.....	39
ILUSTRACIÓN 16: EJEMPLO DE ARCHIVO JSON GENERADO AUTOMÁTICAMENTE.....	40

---



---

## 1 Introducción

El método de transferencia del contenido audiovisual ha venido evolucionando constantemente [1], pasando desde la televisión, sitios web, hasta aplicaciones para Smartphones [2], permitiendo al usuario disfrutar cada vez de una manera más cómoda este tipo de servicios. Actualmente se ofertan varias soluciones de contenido audiovisual al público a través de Internet, por ejemplo: Netflix, Youtube Red y Hulu, por los cuales debe pagarse una suscripción. Además, existen dispositivos que permiten la retransmisión de contenidos hacia la televisión como Chromecast, AppleTV y Roku 3, que también tienen un precio de venta. Una alternativa a estas soluciones de paga son los sitios web o aplicaciones que brindan contenidos de manera gratuita, sin embargo, suelen mostrar excesiva publicidad haciendo que los usuarios menos diestros se desconcierten, descarguen contenido no deseado y no puedan acceder al contenido. Es bueno recalcar que, para presentar soluciones a los inconvenientes anteriormente mencionados, fue necesario investigar alternativas para lograr que el usuario pueda acceder a los contenidos gratuitos con las facilidades de los servicios de pago.

Una placa Raspberry Pi ofrece muchas facilidades para implementar una gran variedad de proyectos; por ejemplo, KANO OS [3] que se utiliza en el ámbito educativo, es un sistema para niños que fomenta la creatividad, el desarrollo musical y la programación básica. En casos industriales ofrece la posibilidad de implementarse en Robots, controlar y monitorizar una red de sensores, entre otras aplicaciones [4], del mismo modo el uso en campos del entretenimiento. Para el año 2017 una Raspberry PI con 512 MB de memoria RAM, Procesador ARM11 multitarea a 700Mhz y Memoria SD de hasta 16gb, se podía adquirir por \$30 (US dólares). Su costo accesible y su sistema basado en Linux, motiva ampliamente su uso y multiplica las opciones de aplicaciones que pueden alojarse en ella [5]. Por ejemplo, la libertad de escoger un lenguaje de programación basado en la necesidad del proyecto y la simplicidad de obtener repositorios de múltiples servidores.

Las mejoras en las comunicaciones, la sencillez de obtener un dispositivo para navegar en línea y una vasta cantidad de contribuyentes, han provocado que el desarrollo web avance a pasos agigantados. Para entrar un poco en contexto, en la década de 1990 el acceso a la información en línea era limitada, la velocidad de las comunicaciones para sectores profesionales alcanzaba los 145Mbps, a través de fibra óptica, pero el público común y consumidores en general solo podía acceder a través de su línea telefónica lo que limitaba



---

la velocidad a 56Kbps [6] lo cual presentaba dificultades para el desarrollo y diseño web. 37 años más tarde, en 2017, el acceso a las comunicaciones es para todo público, con una velocidad promedio global de 5.6Mbps [7]. Usando un navegador convencional se pasó de acceder a una simple página web de información a aplicaciones web complejas.

Gracias a las facilidades y novedades del desarrollo web y de las capacidades de una Raspberry Pi fue posible dar una solución alternativa a la problemática anteriormente descrita, desarrollando dos aplicaciones web ambas alojadas en la Raspberry; una de ellas encargada de mostrar el contenido audiovisual al cual se puede acceder, y la otra permite usar un Smartphone o una Tablet como control remoto, para garantizar una sencilla navegación en la primera aplicación.





## 1.1 Antecedentes

Los inicios del streaming datan de la década de 1980 cuando la informática empezó a llegar a una mayor cantidad de gente, en aquella época los ordenadores eran poco potentes, por ejemplo la velocidad de computo llegaba a 33 MHz con el Intel 80386 [8] e intentar reproducir archivos de video MPEG-1 [9] de esa época con resolución 352x240 a 30 frames por segundo y audio comprimido en mp3 [10] era para pocos ordenadores ya que estos eran de un elevado costo, limitando así dicho servicio [2].

En los años noventa la tecnología vio una gran mejora, tanto en el ancho de banda que se ofrecía como las características de los ordenadores, abriendo paso a la implementación del primer software orientado al streaming conocido como RealPlayer creado por RealNetworks el cual marcó el inicio de una nueva manera de ver contenido sin tener que almacenarlo de manera local, el cual también eliminaba la espera de descargar un archivo antes de visualizarlo. Luego surgieron soluciones como QuickTime, ActivePlayer, Adobe, entre otros, quienes ofrecían el mismo servicio, no obstante, la limitante era mejorar el internet a nivel global [2].

Hoy en día existen técnicas usadas por sitios para streaming entre ellas la descarga progresiva que consiste en reproducir audio/vídeo comprimido que es descargado mientras se reproduce [11] esto se logra mediante el envío secuencial de fragmentos a través de la red. Por otra parte, la transmisión por secuencias se produce en servidores multimedia que disponen de un software especial para gestionar óptimamente el streaming de audio y vídeo. [12]

En la actualidad existen servicios que ofrecen streaming, entre ellos se encuentran:

1. Netflix un servicio de streaming que permite a los clientes una gran variedad de programas televisivos, películas, documentales y más sobre miles de dispositivos conectados a internet. [13].
2. Hulu servicio de suscripción de vídeo bajo demanda propiedad de Hulu LLC, Disney-ABC, Fox Entertainment, NBCUniversal y Turner Broadcasting System. Está orientado principalmente a las series y programas de entretenimiento como realities, talk shows o concursos, pero también incluye películas y documentales. [14].



---

Como dispositivo físico que opera entre la televisión y el servicio, sobresalen:

1. AppleTV: Con el Apple TV se puede ver programas y películas con buena calidad y también disfrutar del contenido de apps como iTunes, HBO GO, FOX, Netflix y muchas más. [15].
2. Chromecast: es un dispositivo HDMI que se conecta a la televisión para recibir la señal de los dispositivos multimedia conectados en una misma red wifi. Basta con emparejar los dispositivos, pulsar la opción Enviar y disfrutar de tus contenidos en pantalla grande. [16].

Como trabajo monográfico:

1. Se han realizado estudios en los cuales se hace uso de crawlers para la extracción de datos, como el trabajo “Desarrollo de herramientas constructivistas para la publicación y personalización de contenidos de cursos en línea”, Este trabajo busca innovar en los cursos en línea. Se ha construido un conjunto de herramientas que dan al alumno un papel activo en la construcción de su aprendizaje y al formador le permiten crear cursos más dinámicos y atractivos. [17]
2. Investigaciones para tesis de grado relacionados al contenido audiovisual como oportunidad para empresas, con el título “El consumo de contenido audiovisual en las generaciones Baby Boomers y X en Bogotá. Una oportunidad para la industria”. [18]



---

## 1.2 Planteamiento del problema

A lo largo de los años la televisión ha venido cambiando: desde la típica televisión análoga hasta la televisión digital, y más recientemente la televisión a través de Internet.

La televisión por Internet ha ganado popularidad en la comunidad hispana [19]. Sin embargo, en países en vías de desarrollo como Nicaragua, la penetración de este tipo de tecnologías y plataformas es un poco más lenta debido a los costos que el acceso a Internet supone, por ejemplo, del año 2010 al 2015 ha habido un aumento en las conexiones de banda ancha de apenas el 30% [20].

En la actualidad, se ofrecen servicios que brindan acceso a contenido audiovisual por internet, como iTunes, Netflix, Hulu, Youtube Red, entre otros. Por otro parte, se han desarrollado múltiples dispositivos que funcionan como intermediarios entre la televisión y la Internet como Apple TV, Chromecast, Roku TV y Amazon Fire TV-. No obstante, estos ofrecen contenido exclusivo de paga que, además, no está disponible en algunas regiones [21].

Aunque existen programas, apps y webs que son alternativas libres de cargos monetarios -como Kodi, Rosadin TV, Youtube, Dailymotion, entre otros-, lamentablemente son accesibles solo para aquellas personas que tienen un buen dominio de las tecnologías actuales. Esto provoca que, por falta de destreza tecnológica, una parte de la población no pueda gozar del contenido que ofrecen estos servicios. A este problema se suma el hecho de que, en algunas webs, para acceder al contenido deseado, debe llevarse a cabo muchos pasos, entre ellos: ingresar a la página web, buscar el contenido, enfrentarse a molestos anuncios, tediosos registros y enlaces que nos redirigen a sitios web que comprometen tanto la seguridad de la persona como la del dispositivo.

Al analizar este problema, se observó la necesidad de crear un software implantado en un dispositivo que permita a todas las personas que cuentan con Internet residencial de banda ancha y un televisor con una entrada HDMI, el acceso de manera rápida y sencilla a una buena cantidad de programas gratuitos en línea.



---

Por tal razón nos surgen las siguientes preguntas:

¿Qué metodología se debería emplear para desarrollar una aplicación, para el acceso gratuito a video bajo demanda por Internet sin estar sujeto a pagos y que disminuya la publicidad?

¿De qué manera se obtendría el contenido y qué criterios debía cumplir para ser mostrado al usuario?

¿Qué método se utilizaría para controlar desde un dispositivo móvil la interfaz mostrada en el televisor?

¿Cómo debían organizarse e interactuar los elementos que componen la solución para garantizar su correcto funcionamiento?



---

## 1.3 Justificación

Con base en la problemática encontrada en los servicios de streaming que se ofrecen actualmente y descrita arriba, se buscó brindar una solución que unifique contenido audiovisual gratuito que se encuentra en Internet.

### **Originalidad**

Aunque en la actualidad existen distintas alternativas que proporcionan video bajo demanda, este proyecto perseguía disminuir los anuncios, evitar los registros, y las membresías.

### **Alcance**

Se espera que personas de cualquier edad y con un nivel mínimo de experiencia en tecnologías digitales tengan la posibilidad de utilizar este dispositivo.

### **Impacto**

El desarrollo de este servicio ofrece beneficios en los ámbitos económicos y del entretenimiento pues facilita el acceso gratuito a contenido multimedia.

### **Producto**

El producto final es una alternativa libre de membresías y registros que facilita la reproducción de video bajo demanda.



---

## 1.4 Objetivos

### Objetivo general

Desarrollar aplicaciones Web que faciliten el acceso al servicio de video bajo demanda por Internet sin estar sujeto a pagos, disminuya la publicidad y sea fácil de usar.

### Objetivos específicos

- Obtener acceso a las URL de los videos en los sitios web fuentes, haciendo uso de web crawlers, que permitan almacenar la información de los videos para su posterior reproducción.
- Desarrollar una aplicación web que visualice en un televisor una interfaz con el contenido de archivos JSON.
- Crear una aplicación web secundaria para ser cargada en dispositivos móviles, que muestre un control remoto que permita interactuar con la interfaz mostrada en el televisor.



---

## 2 Marco teórico

### 2.1 Internet vs televisión convencional

Si bien la llegada de la radio y posteriormente de la televisión revolucionó el mundo de la comunicación y eliminó fronteras al poder conocer en tiempo real todo tipo de noticias de cualquier continente, lo cierto es que internet lo ha hecho con más fuerza, obligando a los dos medios mencionados a evolucionar, sobre todo en el caso de la televisión. [22]

Las nuevas tecnologías han traído avances formidables. Internet ha desplazado en mucha medida a la televisión por su capacidad para interactuar con los usuarios. Hoy se pueden hacer preguntas, contestar trivias, opinar o comentar directamente un contenido que se transmite vía Internet.

Además, con Internet se tiene acceso a “streaming” que es cargar directamente desde Internet el contenido que queremos ver en nuestros dispositivos, y es algo cada vez más común. No importa pues si se tiene una Tablet, una computadora de escritorio o un teléfono, el contenido puede visualizarse en cualquiera de esos elementos. Pensemos en Netflix o Hulu. Antes, para ver películas de estreno había que ir al cine. La realidad es que ya no es necesario esa práctica.

Las empresas de medios han tenido entonces que tomar cartas en el asunto. Prácticamente todas las cadenas tradicionales de TV se están ajustando a los tiempos modernos y ya tienen la posibilidad de transmitir sus contenidos a través de sus portales, o bien, en apps dedicadas y escritas con ese propósito. Quien ignore estos nuevos medios pronto quedará obsoleto sin duda.

Hasta hace poco, casi todas las apps de streaming usaban lo que se puede llamar el modelo Netflix/Hulu. Se paga una cuota modesta y se tiene acceso a una enorme biblioteca de shows de TV, de películas, de contenidos de video en general, los cuales se pueden ver en cualquier momento, tantas veces como se quiera, en el orden que se prefiera y con la frecuencia que se quiera. No hay incluso nada que grabar (por ello las grabadoras de video se están volviendo prácticamente obsoletas) porque todo está en la “nube”, esperando.



---

Realmente el hacer streaming de video no es diferente a lo que se hace cuando se manda una señal de video por uno de los proveedores de TV de paga por cable. Lo que es distinto es que en el modelo Netflix/Hulu no se tienen numerosas opciones de canales. [23]

## 2.2 Usabilidad de los sitios

La Usabilidad es la medida de la calidad de la experiencia que tiene un usuario cuando interactúa con un producto o sistema. Esto se mide a través del estudio de la relación que se produce entre las herramientas (entendidas en un Sitio Web el conjunto integrado por el sistema de navegación, las funcionalidades y los contenidos ofrecidos) y quienes las utilizan, para determinar la eficiencia en el uso de los diferentes elementos ofrecidos en las pantallas y la efectividad en el cumplimiento de las tareas que se pueden llevar a cabo a través de ellas.

Adicionalmente se debe anotar que la palabra Usabilidad deriva del inglés Usability, cuya traducción más acertada es "facilidad y simplicidad de uso de un artículo u objeto". Se ha considerado adecuado utilizar la palabra usabilidad con dicha traducción debido a que en idioma español no existen palabras que describan con tanta precisión este concepto. En este sentido, lo más cercano sería Utilidad, palabra que, de acuerdo al Diccionario de la Real Academia Española, significa "que trae o produce provecho, comodidad, fruto o interés". Sin embargo, debido a que dicho término no engloba la idea de facilidad o simplicidad en el uso de un objeto o espacio por parte de quien lo emplea, se ha preferido utilizar la derivación de la palabra originalmente en idioma inglés.

Dadas las definiciones anteriores es claro que abordar la Usabilidad implica también revisar una serie de aspectos relacionados con el uso y la manera en que las personas se relacionan con los sistemas que se les ofrecen.

- **Facilidad de aprendizaje:** define en cuánto tiempo un usuario, que nunca ha visto una interfaz, puede aprender a usarla bien y realizar operaciones básicas.
- **Facilidad y Eficiencia de uso:** determina la rapidez con que se pueden desarrollar las tareas, una vez que se ha aprendido a usar el sistema.
- **Facilidad de recordar cómo funciona:** se refiere a la capacidad de recordar las características y forma de uso de un sistema para volver a utilizarlo a futuro.





- **Frecuencia y gravedad de errores:** plantea la ayuda que se le entrega a los usuarios para apoyarlos cuando deban enfrentar los errores que cometen al usar el sistema.
- **Satisfacción subjetiva:** indica lo satisfechos que quedan los usuarios cuando han empleado el sistema, gracias a la facilidad y simplicidad de uso de sus pantallas.

Un Sitio Web puede ser mejorado desde diferentes acercamientos, quedando la Usabilidad como uno más de ellos, permitiendo la aparición de otros con similar importancia que deben ser tenidos en cuenta al momento de hacer el rediseño de un Sitio Web.

A estos acercamientos simultáneos al de la Usabilidad el autor los denomina las Facetas de la Experiencia del Usuario, y corresponden a los siguientes:

- **Útil:** es necesario preguntarnos si nuestros productos y sistemas son útiles, y aplicar nuestro conocimiento para definir soluciones innovadoras que apoyan la utilidad.
- **Usable:** corresponde a la facilidad de uso o Usabilidad sigue siendo un aspecto fundamental necesario, pero no suficiente, por lo que se debe complementar con las demás facetas.
- **Deseable:** si bien los sitios deben ser eficientes, en particular con el uso de medios más complejos (imágenes, sonidos, animaciones), esto se debe equilibrar con los demás valores del diseño emocional.
- **Accesible:** los sitios Web deben ser accesibles a las personas con discapacidades (más de 10% de la población). Para los Sitios Web de Gobierno ya es un requisito normativo.
- **Creíble:** la credibilidad es uno de los factores más importantes de tener en cuenta y por ello se deben revisar los elementos de diseño que afectan la confianza que nos tienen los usuarios.
- **Valioso:** las facetas ayudan a determinar los aspectos que llevan a que nuestros sitios ofrezcan valor para nuestros usuarios. [24]



---

## 2.3 Raspberry

Raspberry Pi es un miniordenador de pequeño tamaño, bajo coste y bajo consumo cuyos primeros modelos fueron lanzados en abril de 2012. Generalmente este tipo de miniordenadores ejecutan sistemas operativos basados en Linux y están íntimamente relacionados con el Open Software. No obstante, el desarrollo en sí de Raspberry Pi no es Open Hardware.

El proyecto Raspberry Pi empezó en 2006, y estaba inspirado en el BBC Micro, un ordenador que permitió popularizar la informática en 1981. Este ordenador tuvo dos modelos A y B y, por este motivo, la primera versión de Raspberry Pi tuvo también dos variantes A y B. Según sus creadores, el objetivo era crear un dispositivo de bajo coste que permitiera acercar a todos los públicos un ordenador con el cual mejorar las habilidades en informática, programación y electrónica.

Aunque la potencia de un Raspberry Pi es más reducida que un ordenador convencional, sigue siendo un ordenador considerablemente práctico. A esto hay que sumar que cuenta con funciones propias de electrónica. Algunos de los proyectos más “convencionales” y habituales son, por ejemplo, aprender Linux o programación, conectarlo a una televisión y usarlo como centro multimedia, conectarle discos duros y emplearlo como servidor NAS, o incluso como pequeño servidor doméstico.

Las aplicaciones menos habituales (y más interesantes) incluyen todo tipo de proyectos de electrónica y robótica, aplicaciones de visión por ordenador, aplicaciones en IoT (Internet de las cosas). Hasta hay proyectos que conectan varios Raspberry Pi en paralelo para hacer un mini-cluster de computación.

Existen dos familias de Raspberry Pi, la “tradicional” y una más reciente de tamaño muy reducido llamada Zero. Cada modelo de ambas familias tiene sus propias características de CPU, RAM, número de puertos, conectividad, entre otros. [25]

### **Raspberry Model B**

El Model A fue diseñado con características básicas, pero a la vez muy completas, cuenta con salida de video y audio, un procesador capaz de realizar tareas interesantes como la reproducción de video de alta definición, un tamaño pequeño por unos 25 dólares, pero la



gente empezó a pedir conexión a internet, y por lo tanto la fundación Raspberry Pi decidió fabricar el Model B, mejorando el producto y añadiendo un puerto Ethernet, sin embargo, con un precio de unos 35 dólares.

El Raspberry Pi Model B es un dispositivo de dimensiones 85,6 x 53,98 x 17 mm con la posibilidad de mostrar video 1080p y conectarse a internet, puerto HDMI y USB, slot para tarjetas MicroSD y sistema operativo basado en Linux.

Existen tres principales distribuciones de Linux promocionadas por la fundación Raspberry Pi: Raspbian "wheezy" (basada en Debian), Arch Linux ARM y QtonPi. La primera ha sido clasificada como "para principiantes", y es que se trata de una distribución preconfigurada con administrador de ventanas como LXDE y con algunos navegadores de internet como Midori. La segunda, Arch Linux, tiene como meta proporcionar grandes niveles de control al usuario. Por último, QtonPi es una plataforma embebida para el uso de aplicaciones Qt.

Las distribuciones vienen con algunas aplicaciones preinstaladas, entre las que se encuentran los navegadores de internet Midori, Dillo y NetSurf y herramientas de programación como IDLE para Python, Scratch y Squeak.

Este modelo brinda las características necesarias para el presente proyecto, entre ellas facilidad de implementación, portabilidad, conexiones necesarias, soporte y software libre. [25]

## **2.4 Tecnologías**

### **Desarrollo web**

#### **HTML5**

Es la última versión de HTML. Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos:

- Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5 y amigos, a menudo reducido a HTML5.



- Diseñado para ser utilizable por todos los desarrolladores de Open Web, esta página referencia numerosos recursos sobre las tecnologías de HTML5, clasificados en varios grupos según su función. [26]
- HTML5 ofrece todos los elementos para la creación de la interfaz gráfica de la aplicación y control remoto.

### 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 o XML (y por extensión en XHTML). 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.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

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>". [27]

El CSS sirve para definir la estética de un sitio web en un documento externo y eso mismo permite que modificando ese documento (la hoja CSS) podamos cambiar la estética entera de un sitio web, el mismo sitio web puede variar totalmente de estética cambiando solo la CSS, sin tocar para nada los documentos HTML o jsp o asp que lo componen.

CSS es un lenguaje utilizado para dar estética a un documento HTML. Podemos indicar propiedades como el color, el tamaño de la letra, el tipo de letra, entre otras propiedades, también se puede dar forma a otros elementos que no sean letras, como colores de fondo de una página, tamaños de un elemento (por ejemplo, el alto y el ancho de una tabla). [28]

CSS3 permite una vista amigable, atractiva, usable y sencilla para la aplicación.



## JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras. [29]

## AJAX

“Ajax no es una tecnología. En realidad, son varias tecnologías, cada una de las cuales florece por derecho propio, uniéndose de formas nuevas y poderosas”. Ajax incorpora: presentación basada en estándares usando XHTML y CSS; visualización dinámica e interacción utilizando el modelo de objetos de documento; intercambio de datos y manipulación utilizando XML y XSLT; recuperación asíncrona de datos utilizando XMLHttpRequest; y JavaScript enlazando todo junto.

El modelo de aplicación web clásico funciona así: la mayoría de las acciones de los usuarios en la interfaz desencadenan una solicitud HTTP a un servidor web. El servidor realiza algunos procesos: recupera datos, procesa números, habla con varios sistemas heredados y luego devuelve una página HTML al cliente. Es un modelo adaptado del uso original de la



---

Web como un medio de hipertexto, pero lo que hace que la Web sea buena para el hipertexto no lo hace necesariamente bueno para las aplicaciones de software.

Una aplicación Ajax elimina la naturaleza de inicio-parada-inicio-parada de la interacción en la Web al introducir un intermediario, un motor Ajax, entre el usuario y el servidor.

En lugar de cargar una página web, al inicio de la sesión, el navegador carga un motor Ajax, escrito en JavaScript y generalmente escondido en un marco oculto. Este motor es responsable de representar la interfaz que el usuario ve y de comunicarse con el servidor en su nombre. El motor Ajax permite que la interacción del usuario con la aplicación suceda de forma asíncrona, independientemente de la comunicación con el servidor. Por lo tanto, el usuario nunca está mirando una ventana de navegador en blanco y un icono de reloj de arena, esperando que el servidor haga algo. [30]

## JavaScript

JavaScript (JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js o Apache CouchDB. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa.

El estándar de JavaScript es ECMAScript. Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición, conocida inicialmente como ECMAScript 6 o ES6, se liberó el 17 de Julio de 2017. Desde entonces, los estándares ECMAScript están en ciclos de lanzamiento anuales. Esta documentación se refiere a la última versión del borrador, que actualmente es ECMAScript 2018.

JavaScript no debe ser confundido con el lenguaje de programación Java. Java es una marca registrada de Oracle en Estados Unidos y otros países. Sin embargo, los dos lenguajes de programación tienen muchas diferencias en las sintaxis, semántica y usos. [31]

Javascript posibilita la funcionalidad de la aplicación, obtener y filtrar los enlaces, editar el árbol DOM con los nuevos elementos.



## PHP

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Este mismo sitio web de Wikipedia está desarrollado en PHP. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. La versión más reciente de PHP es la 5.3.2 (for Windows) del 4 de marzo de 2010. [32]

## Python

Python es un lenguaje de programación interpretado, interactivo y orientado a objetos. Incorpora módulos, excepciones, tipificación dinámica, tipos de datos dinámicos de muy alto nivel y clases. Python combina un poder notable con una sintaxis muy clara. Tiene interfaces para muchas bibliotecas y llamadas de sistema, así como para varios sistemas de ventanas, y es extensible en C o C ++. También se puede usar como un lenguaje de extensión para aplicaciones que necesitan una interfaz programable. Finalmente, Python es portátil: se ejecuta en muchas variantes de Unix, en Mac y en Windows 2000 y posteriores.

Python es un lenguaje de programación de propósito general de alto nivel que se puede aplicar a muchas clases diferentes de problemas.

El lenguaje viene con una gran biblioteca estándar que cubre áreas como el procesamiento de cadenas (expresiones regulares, Unicode, cálculo de diferencias entre archivos), protocolos de Internet (HTTP, FTP, SMTP, XML-RPC, POP, IMAP, programación CGI), ingeniería de software (pruebas de unidad, registro, creación de perfiles, análisis del código Python) e interfaces del sistema operativo (llamadas al sistema, sistemas de archivos, sockets TCP / IP). Mire la tabla de contenido de la biblioteca estándar de Python para tener



---

una idea de lo que está disponible. Una amplia variedad de extensiones de terceros también está disponible. [33]

## **API Web**

### **Selenium WebDriver**

Selenium es un conjunto de herramientas para automatizar los navegadores web en muchas plataformas.

Selenium se ejecuta en muchos navegadores y sistemas operativos. Puede ser controlado por muchos lenguajes de programación y frameworks. [34]

El cambio más grande en Selenium recientemente ha sido la inclusión de la API WebDriver. Conducir un navegador de forma nativa como usuario, ya sea localmente o en una máquina remota utilizando el servidor Selenium, marca un gran avance en términos de automatización del navegador.

Selenium WebDriver encaja en el mismo papel que Remote Control, y ha incorporado los enlaces 1.x originales. Se refiere tanto a los enlaces de idioma como a las implementaciones del código de control del navegador individual. Esto se conoce comúnmente como "WebDriver" o, a veces, como Selenium 2.

Selenium 1.0 + WebDriver = Selenium 2.0

- WebDriver está diseñado en una interfaz de programación más simple y concisa, además de abordar algunas limitaciones en la API de Selenium Remote Control.
- WebDriver es una API orientada a objetos compacta en comparación con Selenium1.0
- Impulsa el navegador de manera mucho más efectiva y supera las limitaciones de Selenium 1.x que afectaron nuestra cobertura de prueba funcional, como la carga o descarga de archivos, las ventanas emergentes y la barrera de diálogos.
- WebDriver supera la limitación de la política de origen de host único de Selenium Remote Control.
- WebDriver es el nombre de la interfaz clave contra la cual se deben escribir las pruebas en Java, las clases de implementación que se deben usar se enumeran a





---

continuación: ChromeDriver, EventFiringWebDriver, FirefoxDriver, HtmlUnitDriver, InternetExplorerDriver, PhantomJSdriver, RemoteWebDriver, SafariDriver. [35]

## Frameworks

### AngularJS

AngularJS es un marco estructural para aplicaciones web dinámicas. Le permite utilizar HTML como idioma de su plantilla y le permite extender la sintaxis de HTML para expresar los componentes de su aplicación de manera clara y sucinta. El enlace de datos de AngularJS y la inyección de dependencia eliminan gran parte del código que de otro modo tendría que escribir. Y todo sucede dentro del navegador, lo que lo convierte en un socio ideal con cualquier tecnología de servidor.

AngularJS no es una pieza única en el rompecabezas global de la construcción del lado del cliente de una aplicación web. Maneja todo el código de pegamento DOM y AJAX que una vez escribiste a mano y lo coloca en una estructura bien definida. Esto hace que AngularJS tenga una opinión acerca de cómo se debe construir una aplicación CRUD (Crear, Leer, Actualizar, Eliminar).

AngularJS simplifica el desarrollo de aplicaciones al presentar un mayor nivel de abstracción al desarrollador. Como cualquier abstracción, tiene un costo de flexibilidad. En otras palabras, no todas las aplicaciones son adecuadas para AngularJS. AngularJS fue construido con la aplicación CRUD en mente. [36]

### Materialize

Materialize es un moderno framework front-end CSS basado en Material Design que utiliza elementos y principios de Material Design, para crear un marco que incorpora componentes y animaciones que proporcionan más información a los usuarios. Además, un solo sistema de respuesta subyacente en todas las plataformas permite una experiencia de usuario más unificada. [37]



---

## 3 Diseño Metodológico

### 3.1 Descripción general del proyecto

El presente proyecto tiene como objetivo unificar contenido audiovisual gratuito que se encuentra en los sitios AnimeFLV, SerieSyPelis24 y PelisPedia; mediante una aplicación donde los usuarios, de cualquier edad y con un nivel mínimo de experiencia en manejo de tecnología, puedan acceder a contenido audiovisual de una manera cómoda, sin registros y sin excesos de publicidad irritante y peligrosa, que puede exponer al dispositivo a contenido malicioso.

Para cumplir con las expectativas se pensó en materiales Hardware y Software a utilizar. Se evaluó las alternativas Hardware, entre ellas, Laptops, Smartphones, Raspberry Pi y Arduino; luego de investigar acerca de precios, capacidades y facilidad de uso, se tomó la decisión de utilizar la **Raspberry Pi 3 modelo B**, debido a su diseño compacto y accesibilidad, además brinda un sistema Linux que se adapta a las necesidades del proyecto.

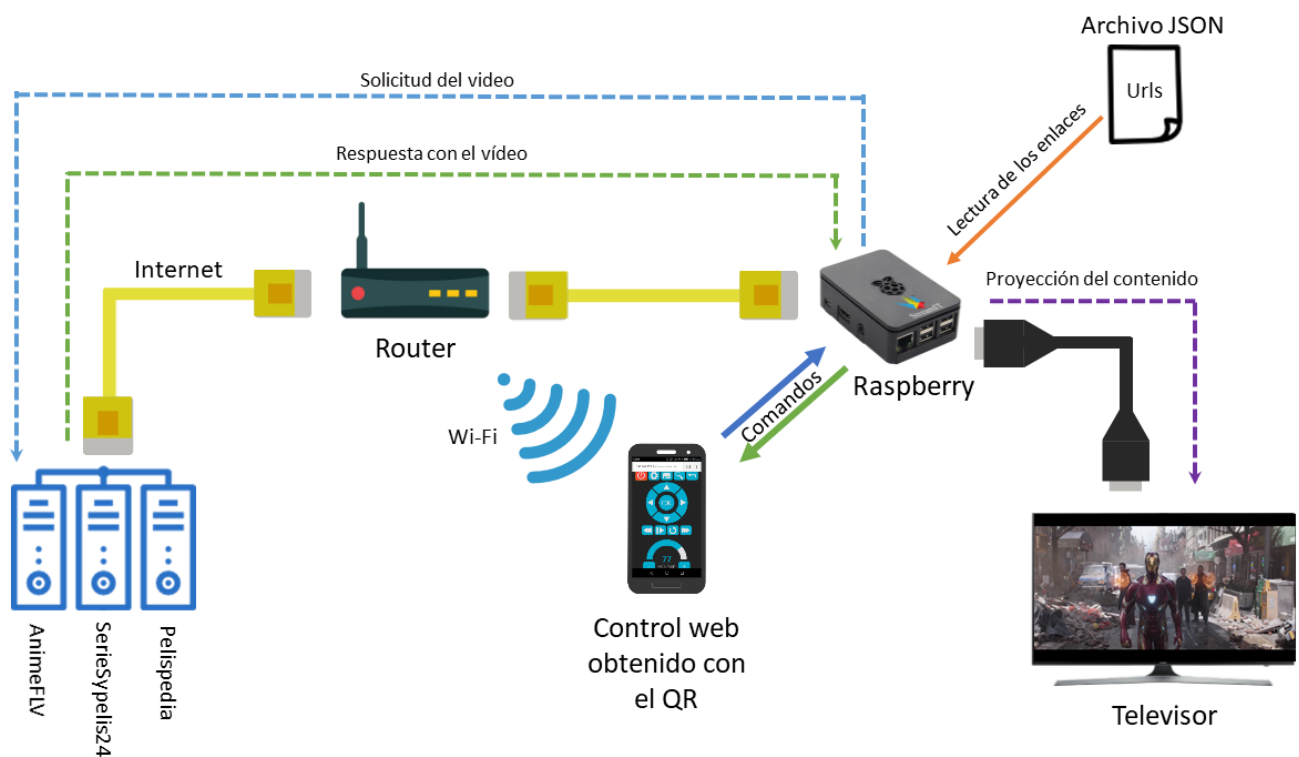
Por otra parte, se analizó qué tecnologías Software era más conveniente utilizar, ya que se pretendía desarrollar una aplicación donde se muestre el contenido audiovisual y un control remoto para facilitar la navegación y la reproducción de contenido; como primera opción se valoró el uso de tecnología web (PHP, JQuery, CSS3, HTML5) para la aplicación, programación en Android para el control remoto y Python para la comunicación entre ambas. Después de analizar la situación se determinó que ambas aplicaciones podían ser desarrolladas con **tecnología web**, debido a que esta es muy flexible, solo se escribe un archivo y funciona igual en cualquier plataforma, su único requisito es un navegador web; y era posible establecer la comunicación entre ambas aplicaciones utilizando **PHP**, así que se decidió el uso de tecnologías web en absoluto.

Para poner en funcionamiento esta aplicación se debe conectar la Raspberry al televisor con un cable HDMI, al internet y al tomacorriente. Una vez encendida, automáticamente se muestra la interfaz de la aplicación web en el televisor, en donde inicialmente se presenta un código QR que debe ser escaneado en un dispositivo móvil con un lector de código QR para acceder al control remoto y poder interactuar con la aplicación principal de manera



sencilla. Internamente, al iniciar y en segundo plano, se ejecuta un web socket server que sirve para mantener la comunicación entre el control remoto y la aplicación principal.

Cuando la aplicación inicia, esta solicita los archivos JSON que contienen el catálogo de contenido a mostrar lo que incluye el título, descripción, imagen de portada y lo más importante, la URL al video. Al terminar la carga del archivo JSON, AngularJS se encarga de leerlo y crear el árbol DOM con la información de este y por último, al dar clic sobre un elemento, éste accede a la URL de servidores externos y comienza a reproducir el contenido audiovisual.



**Ilustración 1: Esquema de funcionamiento**



---

## 3.2 Etapas de trabajo

### Obtención de los contenidos

#### Descripción

En esta etapa se exploró diversas alternativas para obtener las fuentes de contenido que serían provistas al usuario final. En la primera versión los enlaces fueron agregados de forma manual, y posteriormente se desarrollaron Web Crawlers para obtener el contenido de forma dinámica.

#### Resultados

Archivos en formato JSON en los cuales se encuentra almacenada la información de los videos (descripción, URL de la imagen de portada, título, género, URL del video).

### Aplicación web de selección de contenidos

#### Descripción

En esta etapa se creó una interfaz web para mostrar el contenido a los usuarios finales, se desarrolló diferentes versiones en los que se probó diversas opciones para obtener un diseño elegante, rápido y atractivo para los usuarios.

#### Resultados

Interfaz web utilizando AngularJS por la rapidez y sencillez que provee al desarrollador, obteniendo una interfaz amigable para el usuario final.

### Desarrollo del control remoto para la aplicación

#### Descripción

En esta etapa se desarrolló un control remoto multiplataforma que permite la comunicación entre el usuario y la aplicación para brindar comodidad al seleccionar el contenido deseado.

#### Resultados

Control remoto, utilizando tecnologías web y Web Sockets en PHP, para el cual se necesita solamente un navegador web.



---

## Requisitos

Para poder llevar a cabo el proyecto se utilizó lo siguiente:

### Hardware

Laptop Samsung NP300E:

- Procesador Intel Celeron B820
- RAM 8 GB
- Windows 7

Laptop Dell 5567:

- Procesador I7-7700u
- RAM 16 GB
- Windows 10

Raspberry PI 3 modelo B

- Procesador Broadcom BCM2837 Quadcore
- SDRAM 1 GB
- Puerto HDMI
- Puerto Ethernet
- WI-FI

### Software

- Python v3.6
- Selenium v3.14.0
- AngularJS v1.5.8
- JQuery v3.1.1
- Materialize v1.0.0
- PyCharm Community Edition PC-183.4284.139
- PHPStorm PS-182.4129.45
- Chrome v70.0.3538.110
- Raspbian GNU/Linux 8.0 (Jessie)
- Chromium v56.0.2924.84



## 4 Resultados

### 4.1 Web Crawler

Para el correcto funcionamiento del proyecto fue necesario obtener enlaces a los videos de las categorías películas, animes y series. Se determinó que era necesario desarrollar un crawler para poder obtener los enlaces que se encontraban embebidos en etiquetas iframe.

```

▼<div role="tabpanel" class="tab-pane active" id="video_box" style="background: #000; color: #fff;">
  ▼<iframe width="560" height="315" frameborder="0" src="https://s1.animeflv.net/embed.php?s=natsuki&v=Z21OK3VFUW5JZW42YzVMQUlqW...dKN01aYTQ1Mk5WQnhENC9YcnZnbWpVQnBtTjd1TDR4eThHOUhUWjIxVVhmTEdsOGtkQXc9PQ==" scrolling="no" allowfullscreen>
    ▼#document
      <!doctype html>
      ▼<html>
        ▶<head>...</head>
        ▼<body style="background: #000;">
          ▼<div id="container" style="z-index: 80;">
            ▼<div id="player" class="jwplayer jw-reset jw-state-playing jw-skin-jw-skin-glow jw-stretch-uniform jw-breakpoint-1 jw-flag-small-player jw-flag-time-slider-above jw-flag-user-inactive" tabindex="0" aria-label="Video Player" style="width: 382px; height: 280px;">
              <div class="jw-aspect jw-reset"></div>
              ▼<div class="jw-media jw-reset">
                <video class="jw-video jw-reset" disableremoteplayback webkit-playsinline playsinline jw-loaded="data" src="https://storage.googleapis.com/atomic-drake-210708.appspot.com/3025_9.mp4" jw-played></video> == $0
              </div>
              <div class="jw-preview jw-reset"></div>
              ▶<div class="jw-captions jw-reset jw-captions-enabled">...</div>
              ▶<div class="jw-title jw-reset" style="display: none;">...</div>
              ▶<div class="jw-overlays jw-reset">...</div>
              <div class="afs_ads" style="width: 1px; height: 1px; position: absolute; background: transparent;">&nbsp;</div>
            </div>
          </div>
        </body>
      </html>
    </div>
  </div>

```

#### Ilustración 2: Ubicación del enlace al video

Se llevó a cabo un sin número de pruebas luego de las cuales definimos que Python brindaba las herramientas y librerías necesarias para la programación del Crawler, gracias a la librería Selenium que permitió la extracción de enlaces del contenido multimedia para cada una de las categorías.

Para lograrlo se accede al iframe contenedor del video como se muestra en la ilustración 3. En la primera línea del código se inicializa el constructor del webdriver y se abre la url del video a obtener. Como las URLs están embebidas en iframes, se localizan las etiquetas <iframe>, se accede a cada una con la función switch\_to.frame() que permite leer el contenido de cada iframe y para finalizar se localiza la etiqueta video, se lee al atributo SRC,



se guarda la URL y se devuelve el control de la página principal con la función `switch_to.default_content()` para poder continuar con la extracción de los demás enlaces.

```
driver = webdriver.Firefox()
driver.implicitly_wait(30)
driver.get(url)

try:
    iframe = driver.find_elements_by_tag_name('iframe')
    for i in range(1, len(iframe)):
        driver.switch_to.frame(iframe[i])
        url = driver.find_element_by_tag_name('video').get_attribute('src')
        driver.switch_to.default_content()
finally:
    driver.quit();
```

### Ilustración 3: Código para obtener enlaces dinámicamente

Se hizo necesario modificar el Crawler para cada sitio web, pues los enlaces se cargan de diferentes formas en cada uno. Por tanto, para agregar nuevas fuentes de contenido debe modificarse el Crawler para cada una de las páginas web que se deseen agregar como fuentes.

El resultado final que se obtiene luego de recorrer los sitios web con el Crawler, son archivos en formato JSON en los cuales se almacenan los enlaces a los videos, además de información general como el tipo de contenido, título, imagen de portada, descripción, género y si el contenido se divide en temporadas y episodios. Dichos archivos se almacenan de manera local.



```
Animes:
[
  {
    ID_A: '0',
    Titulo: 'Tsurune: Kazemai Koukou Kyuudoubu',
    Imagen: 'https://animeflv.nethttps://animeflv.net/uploads/animes/cove',
    Descripcion: 'Cuando Minato Narumiya comienza como estudiante en la P',
    Genero: 'Deportes,Drama,Escolares',
    Episodios: [
      {
        URL: 'https://fex.net/play/387597123478/894158039'
      },
      {
        URL: 'https://fex.net/play/387597123478/917937585'
      },
      {
        URL: 'https://fex.net/play/387597123478/938092754'
      },
      {
        URL: 'https://fex.net/play/387597123478/954789827'
      },
      {
        URL: 'https://fex.net/play/387597123478/970464711'
      },
      {
        URL: 'https://fex.net/play/387597123478/987730760'
      }
    ]
  }
],
```

Ilustración 4: Archivo JSON generado dinámicamente

## 4.2 Diseño de la interfaz y funciones principales

La interfaz web mostrada en la Raspberry fue construida con tecnologías web. Para la construcción del esqueleto se utiliza principalmente HTML5 y PHP7 en algunos casos, ya que se requiere la interacción con comandos del sistema, un servidor web y otras tareas a las cuales PHP7 ya brinda soporte.

Para el diseño del proyecto se utilizó el framework Materialize, que incluye clases para una maquetación sencilla, íconos, funciones y elementos pre construidos como ventanas modales (MODALS), tarjetas (CARDS), etiquetas (TABS), entre otros, sin embargo, también se debió hacer uso de CSS3 puro para realizar ajustes con el fin de adaptarlo a las necesidades de las aplicaciones que componen el proyecto.





Para contar con una aplicación dinámica y fluida, se decidió desarrollar una aplicación web de una sola página (en inglés Single Page Application SPA) empleando Angular. Para mantener la conexión del web socket y desarrollar un controlador capaz de traducir eficientemente las señales de comunicación entre dispositivos se utilizó JQuery.

En la línea 11 de la ilustración 5, se valida la categoría del contenido a mostrar, si este es un anime o una serie se genera una ventana modal que ocupa aproximadamente la mitad de la pantalla para permitir mostrar los capítulos como se puede ver en las líneas 19 y 24. Si el contenido a mostrar es una película se genera una ventana modal que ocupa toda la pantalla.

```

10 <section class="modal-series white-text animated scrollable">
11 <section class="col {{(type != 'pelicula')}? 'm5': 'm12'}} s12">
12 <i class="fa fa-close material-icons red" id="close_menu">close</i>
13 <h3>{{videoData.Titulo}}</h3>
14 <h5>{{videoData.Genero}}</h5>
15 <p>{{videoData.Resumen || videoData.Descripcion }}</p>
16 </section>
17 <section class="row scrollable col m7 s12" id="series_list" ng-show="(type != 'pelicula')">
18 <ul class="row" style="margin: 0 auto 10px;">
19 <li ng-if="type == 'series'" class="col s12" ng-repeat="temp in seriesCaps">
20 <div class="select li_caps col s12" ng-repeat="capitulos in temp.capitulos">
21 <span class="blue-grey darken-0 col s12" ng-click="cargarVideo(capitulos.URL)"> {{ capitulos.Titulo_Cap ||
("Sin Titulo "+($index+1))}}</span>
22 </div>
23 </li>
24 <li ng-if="type == 'anime'" class="col s12">
25 <div class="select li_caps col s12" ng-repeat="capitulos in videoData.Episodios">
26 <span class="blue-grey darken-0 col s12" ng-click="cargarVideo(capitulos.URL)"> {{ capitulos.Titulo_Cap ||
("Capitulo "+($index+1))}}</span>
27 </div>

```

**Ilustración 5: Validación de contenido a mostrar**

Se hizo uso de un web socket PHP para enviar los mensajes de abrir o cerrar la conexión, y para enviar los códigos correspondientes a cada acción que se desea realizar desde el control remoto.

```

1 <?php
2     $Server = new PHPWebSocket();
3     $Server->bind('message', 'wsOnMessage');
4     $Server->bind('open', 'wsOnOpen');
5     $Server->bind('close', 'wsOnClose');
6     $Server->wsStartServer('0.0.0.0', 9300);
7 ?>

```

**Ilustración 6: Web Socket**



Se desarrolló un controlador para establecer e interpretar los mensajes que se envían desde el control hacia la interfaz web que se muestra en la Raspberry, el cual contiene dos funciones principales: bind y connect. A la función bind se le pasan tres señales como parámetros: open, close y message. La señal open se utiliza para abrir la conexión del socket. La señal close, se emplea para cerrar la conexión del socket. Y el parámetro message se usa para indicar al socket, mientras está abierto, que lo que recibe son instrucciones remotas del control web que se carga en el dispositivo móvil.

```
1 ▼ var Server = new FancyWebSocket(), actual = 0, temp_pos = 0;
2   Server.bind( 'open' , function() { console.log( "Connected." ); });
3   Server.bind( 'close' , function() { console.log( "Disconnected." );});
4 ▼   Server.bind('message' , function( data ) {
5     let modalState = $('<div class="modal"></div>').is(':visible'), videotag = $('<div class="video"></div>'),
6 ▼     switch (datos[0] + ':' + datos[1]) {
7       case '1:1': //Desplazar Arriba
8       case '1:2': //Desplazar Derecha
9       case '1:3': //Desplazar Abajo
10      case '1:4': //Desplazar Izquierda
11      case '1:5': //Para Toda Misión xD
12      case '1:6': //Regresar
13      case '1:7': //Regresar a la pagina principal
14      case '1:8': //Apagar
15      case '2:1': //Play/Pause
16      case '2:2': //Avanzar
17      case '2:3': //Retroceder
18      case '2:4': //Reiniciar
19      case '2:5': //Volumen
20      case '3:1': //Buscar
21    }
22  });
23  Server.connect();
```

#### Ilustración 7: Controlador y órdenes que viajan a través del Web Socket

Se creó funciones para cargar la lista de series, animes, películas, los videos, y en el caso de los animes y series cargar la lista de los episodios.



```
1  var app = angular.module('app', []);
2  function streamitctrl ($scope){
3      $scope.login = true;
4      $scope.seriesList = Series;
5      $scope.animesList = Animes;
6      $scope.moviesList = [];
7      $scope.login = () => {};
8      $scope.cargarVideo = (val) => {};
9      $scope.loadVideoData = (model, type) => {};
10 }
11 app.controller('streamitctrl',streamitctrl);
12 // JQuery + Materialize + fix functions
13 $(document).ready(function () {
14     //Inicialización de modals y tabs de Materialize
15     $('.modal').modal(); $('.tabs').tabs();
16 });
```

Ilustración 8: Controlador de eventos AngularJS

### 4.3 Control Remoto

La necesidad de controlar de manera sencilla los elementos de la interfaz web nos llevaron a desarrollar un control que aprovecha los dispositivos móviles para el manejo de la interfaz.

Para que este control funcionara en cualquier dispositivo sin importar el sistema operativo, decidimos programarlo utilizando tecnologías web, de esta manera solamente se requiere un navegador web y un lector de QR para poder usarlo.

Para lograr la comunicación entre ambos dispositivos (Raspberry-dispositivo móvil) se hizo uso de un web socket PHP que procesa los mensajes.

Al pulsar un botón en el control se envía el mensaje al web socket PHP, éste lo recibe, lo procesa, se lo pasa mediante JQuery a la aplicación que se muestra en la televisión y se realiza la acción.



```

1 <section class="OPTIONS">
2   <input type="button" class="shutdown"      ng-click="flag_power=!flag_power">
3   <input type="button" class="btn_small home"  ng-click="send('1:7')">
4   <input type="button" class="buscar"        ng-click="flag_search=!flag_search">
5   <input type="button" class="btn_small regresar" ng-click="send('1:6')">
6 </section>
7 <section class="PAD">
8   <svg width="230" height="230">
9     <g class="btn_PAD" ng-click="send('1:1')">
10      <path fill="#00ADCE" d="M73.536,69.731c22.912-21.09,58.002-21.482,81.376-1.193134.475-34.474 C147.003-4.699,
11      <path fill="#E2E2E2" d="M97.7 45L114.2 16.4 130.7 45"/>
12    </g>
13    <g class="btn_PAD" ng-click="send('1:2')"> </g>
14    <g class="btn_PAD" ng-click="send('1:4')"> </g>
15    <g class="btn_PAD" ng-click="send('1:3')"> </g>
16    <g class="btn_PAD" ng-click="send('1:5')"> </g>
17  </svg>
18 </section>
19 <section class="VIDEO">
20   <input type="button" class="btn_small retroceder" ng-click="send('2:3')">
21   <input type="button" class="btn_small play-pause" ng-click="send('2:1')">
22   <input type="button" class="btn_small reiniciar" ng-click="send('2:4')">
23   <input type="button" class="btn_small avanzar" ng-click="send('2:2')">
24 </section>
25 <section class="Vol">
26   <input id="volumen" data-width="210" data-height="120" data-min=0 ng-model="volumen">
27   <section>
28     <input type="button" ng-click="btnVolumen('restar') value="-" class="btn_small vol_down"><label>Volumen</label>
29     <input type="button" ng-click="btnVolumen('sumar') value="+" class="btn_small vol_up">
30   </section>
31 </section>

```

**Ilustración 9: Código HTML del control**

Las funciones que se definieron para el control permiten la interacción con los elementos, ya sea moverse entre ellos, seleccionar, buscar, reproducir, subir el volumen, adelantar, retroceder y pausar.

```

1 let app = angular.module('app', []);
2 app.controller('streamitctrl', ($scope) => {
3   let Server = new FancyWebSocket();
4   $scope.busqueda = '';
5   $scope.volumen = 100;
6   $scope.flag_search = false;
7   $scope.flag_power = false;
8   $('#volumen').val($scope.volumen).trigger('change').dial({ });
9   $scope.btnVolumen = (tipo) => { };
10  $scope.send = ( text )=> { Server.send( 'message', text );};
11  $scope.clean = () => { $scope.busqueda = ''; Server.send( 'message', '3:1:'+ $scope.busqueda );};
12  Server.bind( 'open' , function() { console.log( "Connected." ); });
13  Server.bind( 'close' , function() { console.log( "Disconnected." ); });
14  Server.bind( 'message', function( payload ){ console.log( payload ); });
15  Server.connect();
16  //keyboard support
17  $('Control').keydown(function(event) {
18    if (event.keyCode === 38 ) { $scope.send('1:1'); }
19    else if (event.keyCode === 39 ) { $scope.send('1:2'); }
20    else if (event.keyCode === 40 ) { $scope.send('1:3'); }
21    else if (event.keyCode === 37 ) { $scope.send('1:4'); }
22    else if (event.keyCode === 13 ) { $scope.send('1:5'); }
23  });
24 });

```

**Ilustración 10: Funciones definidas para el control**

```

1 <input type="button" class="buscar" ng-click="flag_search=!flag_search">

```

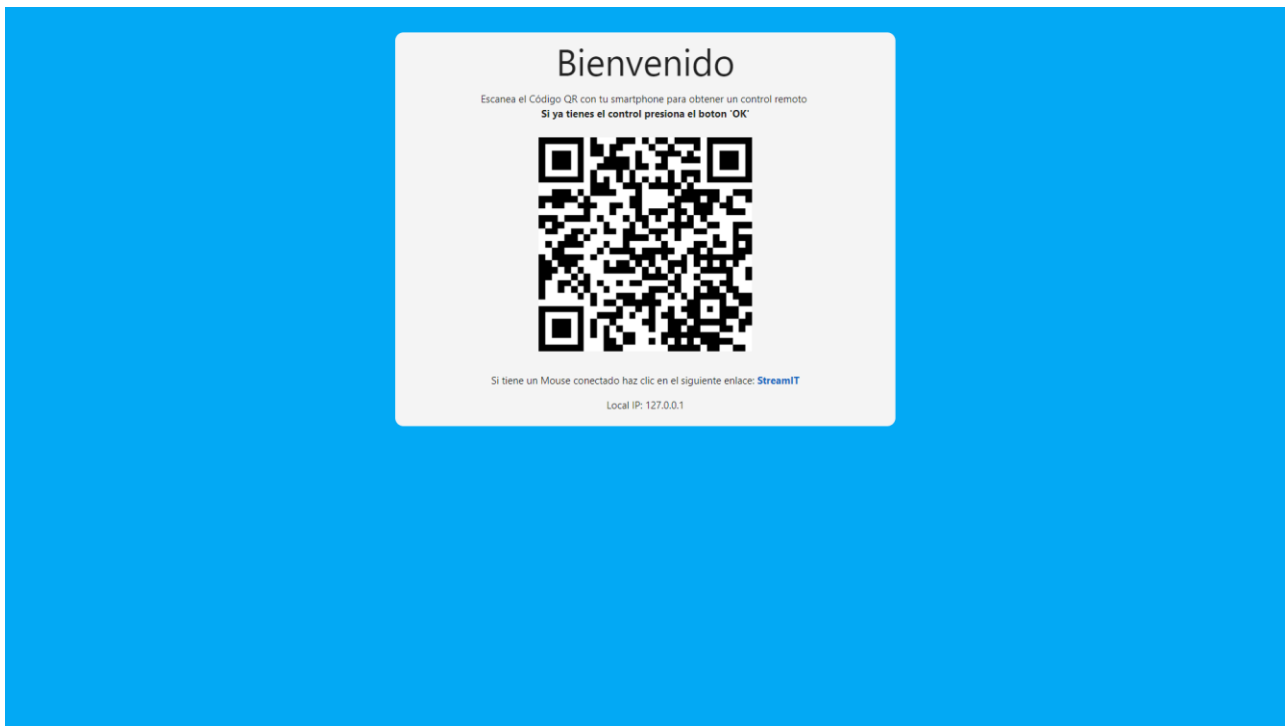
**Ilustración 11: Activación del input búsqueda**



En la ilustración 9 se aprecia la estructura que se definió para el control. Además, en cada botón se incorporó la función ng-click con códigos predefinidos, que se encarga de enviar los mensajes al web socket PHP.

#### 4.4 Resultados finales

El resultado final que se obtiene luego de la ejecución del proyecto es una página en la que se muestra un código QR dinámico mediante el cual se obtiene el control remoto, una interfaz amigable y atractiva; y un control remoto sencillo que permite interactuar con los elementos de la aplicación principal



**Ilustración 12: Código QR dinámico para obtención de control remoto**

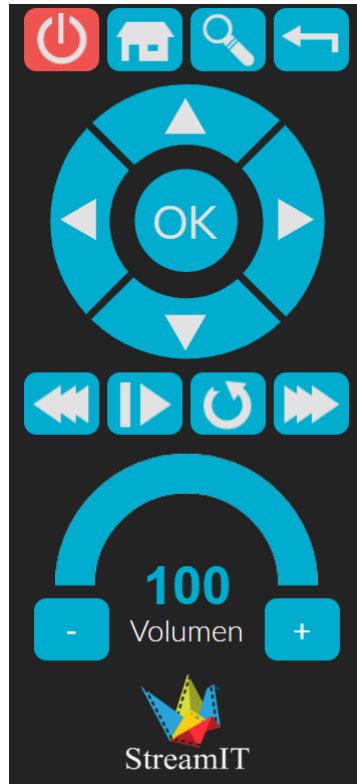


Ilustración 13: Control

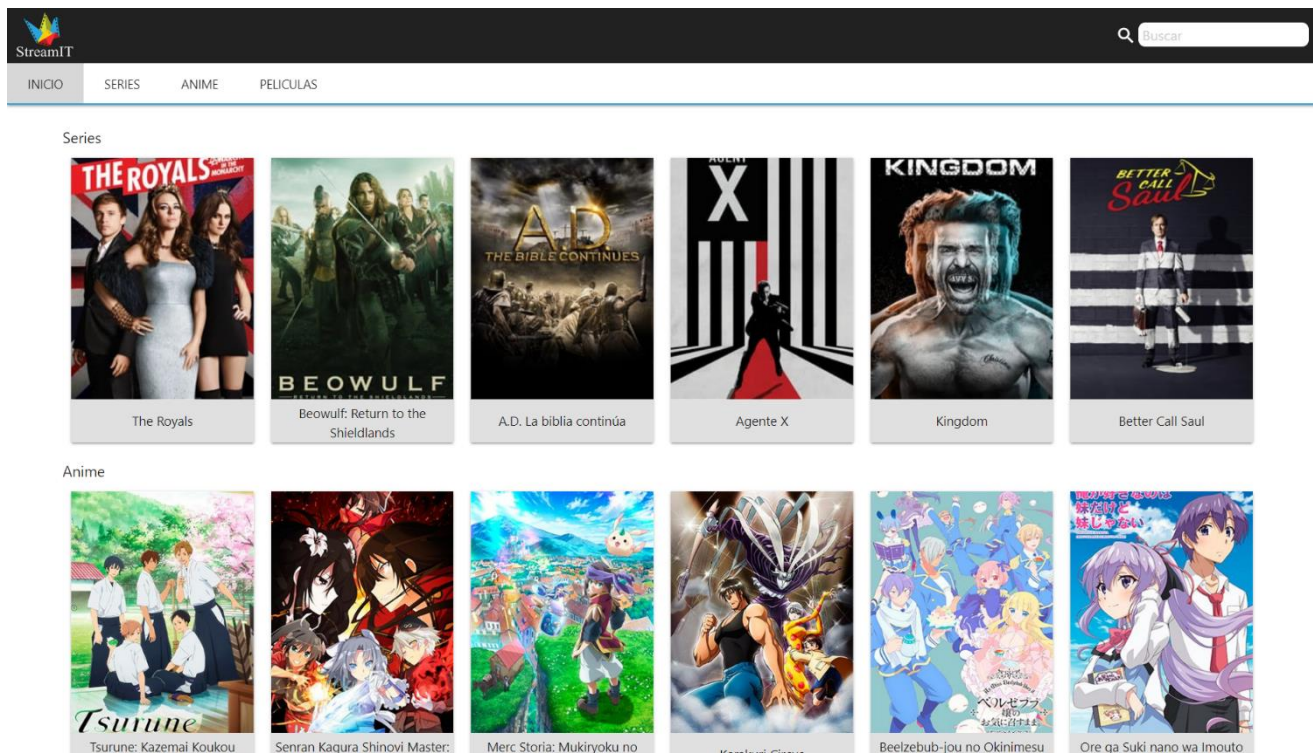


Ilustración 14: Página principal



---

## 5 Conclusiones

Se logró desarrollar aplicaciones web alojadas en una Raspberry Pi 3 modelo B que facilitan el acceso al servicio de video bajo demanda por Internet sin estar sujeto a pagos, que disminuye la publicidad y es fácil de usar.

Para ello, se obtuvo las URL de los videos en los sitios web AnimeFLV, SerieSyPelis24 y PelisPlay a través de crawlers que permitieron almacenar la información de los videos para su reproducción.

Se desarrolló una aplicación que visualiza en un televisor una interfaz con el contenido de los archivos JSON.

Finalmente, se creó una aplicación web para ser cargada en dispositivos móviles, en donde se muestra un control que permite interactuar con la interfaz mostrada en el televisor.



---

## 6 Recomendaciones

Recomendaciones para el uso del dispositivo:

- Utilizar un televisor con entrada HDMI.
- Conexión a internet mínima de 2Mbps y preferiblemente Ethernet.
- Smartphone.
- Aplicación de lector código QR.

Recomendaciones para el uso de la aplicación:

- Supervisión de un adulto.
- Asegurarse que el Smartphone esté conectado a su red doméstica.

Recomendaciones de trabajos futuros:

- Inclusión de fuentes para una mayor cantidad contenido.
- Migrar el proyecto a una aplicación para SmartTV.
- Optimizar la manera en que se obtiene el contenido.
- Investigar alternativas hardware en las cuales se pueda alojar la aplicación para un mejor desempeño.
- Emplear servidores remotos para el almacenamiento y distribución de los archivos JSON.





---

## 7 Bibliografía

- [1] A. T. Drake, «Las transformaciones de los contenidos,» 2012. [En línea]. Available: [www.raco.cat/index.php/Analisi/article/download/252606/339175](http://www.raco.cat/index.php/Analisi/article/download/252606/339175).
- [2] Hipertextual, «Hipertextual,» 9 2 2012. [En línea]. Available: <https://hipertextual.com/archivo/2012/02/historia-del-software-musica-y-video-en-streaming/>.
- [3] Kano, «Kano,» 2017. [En línea]. Available: <http://developers.kano.me>.
- [4] Gizmodo, «es.gizmodo,» 2014. [En línea]. Available: <http://es.gizmodo.com/16-geniales-proyectos-para-tu-raspberry-pi-1657920779>.
- [5] Xataka, «Xataka,» 2017. [En línea]. Available: <https://www.xataka.com/tag/raspberry-pi>.
- [6] G. Peris, «Siglo XXI,» 3 Julio 2017. [En línea]. Available: <http://www.diariosigloxxi.com/texto-diario/mostrar/764433/asi-evolucionado-velocidad-navegamos-internet>.
- [7] ADSLZone, «ADSLZone,» 2017. [En línea]. Available: <https://www.adslzone.net/2017/04/03/que-velocidad-funciona-internet-en-cada-pais/>.
- [8] Computer History, «computerhistory,» [En línea]. Available: <http://www.computerhistory.org/timeline/1987/>.
- [9] PC Doctor, «pc-doctor,» [En línea]. Available: <http://www.pc-doctor.com.mx/Radio%20Formula/temas/Historia%20del%20MP3.htm>.
- [10] Más Adelante, «masadelante,» [En línea]. Available: <http://www.masadelante.com/faqs/mpeg>.
- [11] Wokomedia, «Wokomedia,» [En línea]. Available: <https://wokomedia.com/que-es/streaming-descarga-progresiva/>.
- [12] Diseño de materiales multimedia, «Diseño de materiales multimedia \_Web 2.0,» [En línea]. Available: <http://www.ite.educacion.es/formacion/materiales/107/cd/video/video0103.html>.



- 
- [13] Netflix, «Netflix,» [En línea]. Available: <https://help.netflix.com/en/node/412>.
- [14] V. Morillo, «Fuera de series,» 26 9 2017. [En línea]. Available: <https://fueradeseries.com/que-es-hulu-diferencia-netflix-a5cc5500f5d3>.
- [15] Apple Inc., «Apple Incorporated,» [En línea]. Available: <https://www.apple.com/la/apple-tv-4k/>.
- [16] computerhoy, [En línea]. Available: <http://computerhoy.com/video/que-es-chromecast-34043>.
- [17] J. G. Fernandez, Desarrollo de herramientas constructivistas para la, 2015.
- [18] C. O. M. Carolina Ochoa Martínez, «El consumo de contenido audiovisual en las generaciones Baby Boomers y X en Bogotá. Una oportunidad para la industria.,» Bogotá, 2017.
- [19] Yahoo Research Insights, «US Hispanic Online Video Consumption,» 2016.
- [20] Comisión Económica para América Latina y el Caribe, «Estado de la banda ancha en América Latina y el Caribe,» 2016.
- [21] Google Inc., «Youtube Red,» 27 Septiembre 2017. [En línea]. Available: <https://www.youtube.com/red>.
- [22] Noticias Vigo, «Noticias Vigo,» 21 1 2017. [En línea]. Available: <https://www.noticiasvigo.es/evolucion-de-la-television-con-la-llegada-de-internet/>.
- [23] M. L. Michelone, «UNOCERO,» 13 1 2017. [En línea]. Available: <https://www.unocero.com/entretenimiento/tv-por-internet-vs-tv-por-cable-cual-es-mejor/>.
- [24] Guía Digital, «Guía Digital,» [En línea]. Available: <http://www.guiadigital.gob.cl/articulo/que-es-la-usabilidad.html>.
- [25] J. Andrade, «luisllamas,» 11 8 2012. [En línea]. Available: <https://www.luisllamas.es/que-es-raspberry-pi/>.
- [26] Mozilla Foundation, «MDN web docs,» [En línea]. Available: <https://developer.mozilla.org/es/docs/HTML/HTML5>. [Último acceso: Mayo 2018].
-



- 
- [27] Ecured, «Ecured,» [En línea]. Available: <https://www.ecured.cu/CSS3>. [Último acceso: Mayo 2018].
- [28] EcuRed, 1 Abril 2005. [En línea]. Available: [https://www.ecured.cu/Red\\_por\\_infrarrojos](https://www.ecured.cu/Red_por_infrarrojos).
- [29] JSON.org, [En línea]. Available: <https://www.json.org/json-es.html>.
- [30] Adaptative Path, «Adaptative Path,» 18 Febrero 2005. [En línea]. Available: <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/>.
- [31] Mozilla Foundation, «MDN web docs,» [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/JavaScript>. [Último acceso: Mayo 2018].
- [32] Ecured, «Ecured,» [En línea]. Available: <https://www.ecured.cu/PHP>. [Último acceso: Mayo 2018].
- [33] Python Software Foundation, «General Python FAQ,» 26 Noviembre 2018. [En línea]. Available: <https://docs.python.org/3/faq/general.html>.
- [34] Selenium, «Selenium,» [En línea]. Available: <https://www.seleniumhq.org/>.
- [35] Selenium, «Selenium,» [En línea]. Available: <https://www.seleniumhq.org/projects/webdriver/>.
- [36] AngularJS, «What Is AngularJS?,» [En línea]. Available: <https://docs.angularjs.org/guide/introduction>.
- [37] Materialize, «MaterializaCSS,» [En línea]. Available: <https://materializecss.com/>.
- [38] INIDE, «INIDE,» 2017. [En línea]. Available: [http://www.inide.gob.ni/ECH/Publicacion%20ECH%20I%20trimestre%202017\\_Ago2017.pdf](http://www.inide.gob.ni/ECH/Publicacion%20ECH%20I%20trimestre%202017_Ago2017.pdf).
- [39] SeoCoaching, «Seocoaching,» 14 Marzo 2016. [En línea]. Available: <http://seocoaching.co/que-es-un-crawler-o-aranas-de-la-web-y-que-hacen/>.
- [40] W3schools, «W3schools,» 21 diciembre 2017. [En línea]. Available: <https://www.w3schools.com/css/>.
-



- 
- [41] conceptodefinicion, «conceptodefinicion,» 26 Enero 2015. [En línea]. Available: <http://conceptodefinicion.de/html/>.
- [42] Mozilla, «Mozilla,» 20 Diciembre 2017. [En línea]. Available: [https://developer.mozilla.org/es/docs/Learn/Getting\\_started\\_with\\_the\\_web/JavaScript\\_basics](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics).
- [43] S. Manzur, «Mexired,» 2015. [En línea]. Available: <http://www.mexired.com/blog/que-es-jquery>.
- [44] Mozilla Foundation, «MDN web docs,» [En línea]. Available: <https://developer.mozilla.org/es/docs/Glossary/jQuery>. [Último acceso: Mayo 2018].
- [45] Mozilla Foundation, «MDN web docs,» [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/Reference/API>. [Último acceso: Mayo 2018].



## 8 Anexos

### 8.1 Costos del proyecto

El desarrollo del prototipo tiene un costo de \$1540, de los que derivan \$1500 por el desarrollo de la interfaz web, control remoto y web Crawler, más \$40 del costo de la Raspberry pi 3 modelo B. Sin embargo, al producir 1000 ejemplares en masa los costos por dispositivos disminuirían notablemente ya que el desarrollo de la aplicación divide sus costos entre la cantidad de ejemplares, dando un total de \$1.5 por dispositivo, sumando la compra de la Raspberry por \$40 y con un margen de ganancia del 68.67%, el dispositivo tendría un valor en el mercado de \$69.99.

### 8.2 Ejemplo de Web Crawler

```

1  from selenium import webdriver
2  from selenium.webdriver.firefox.options import Options
3  from selenium.common.exceptions import NoSuchElementException
4  from time import sleep
5
6  options = Options()
7  options.add_argument('--headless')
8  url = "https://animeflv.net/browse"
9  file = open("animes.js", "w", encoding='utf-8')
10
11 def anime(url, num_anime):
12     try:
13         driver = webdriver.Firefox(options=options)
14         driver.get(url)
15         pre = "https://animeflv.net"
16         ul_animes = driver.find_element_by_class_name('ListAnimes')
17         animes = ul_animes.find_elements_by_class_name('Anime')
18         for i in range(0, 10):
19             div_description = animes[i].find_element_by_class_name('Description')
20             name_anime = animes[i].find_element_by_class_name('Title').text
21             url_anime = div_description.find_element_by_class_name('Button').get_attribute('href')
22             img_anime = animes[i].find_element_by_tag_name('img').get_attribute('src')
23             file.write("\n\t\t\t\tID_A:" + "" + str(num_anime) + "" + ",")
24             file.write("\n\t\t\t\tTitulo:" + "" + name_anime + "" + ",")
25             file.write("\n\t\t\t\tImagen:" + "" + pre+img_anime + "" + ",")
26             num_anime = num_anime + 1
27         try:
28             driver2 = webdriver.Firefox(options=options)
29             driver2.get(url_anime)
30             generos = driver2.find_element_by_class_name('Nvgnrs').find_elements_by_tag_name('a')
31             genero = ""
32             for r in range(0, Len(generos)): ...
33             description = driver2.find_element_by_class_name('Description').find_element_by_tag_name('p').text
34             file.write("\n\t\t\t\tDescripcion:" + "" + description + "" + ",")
35             file.write("\n\t\t\t\tGenero:" + "" + genero + "" + ",")
36             file.write("\n\t\t\t\tEpisodios: [")
37
38             episodios = driver2.find_element_by_class_name('ListCaps').find_elements_by_tag_name('li')
39             for j in reversed(range(0, Len(episodios))): ...
40
41         finally:
42             driver2.quit()
43     finally:
44         driver.quit()
45     anime(url, num_anime)

```

Ilustración 15: Ejemplo de Crawler

```

1 var Animes =
2 [
3 {
4   ID_A:'0',
5   Titulo:'Tsurune: Kazemai Koukou Kyuudoubu',
6   Imagen:'https://animeflv.nethttps://animeflv.net/uploads/animes/covers/3058.jpg',
7   Descripción:'Cuando Minato Narumiya comienza como estudiante en la Preparatoria Prefectural Kazemai no tarda en recibir una invitación
8   Genero:'Deportes,Drama,Escolares',
9   Episodios: [
10    {
11      URL:'https://fex.net/play/387597123478/894158039'
12    },
13    {
14      URL:'https://fex.net/play/387597123478/917937585'
15    },
16    {
17      URL:'https://fex.net/play/387597123478/938092754'
18    },
19    {
20      URL:'https://fex.net/play/387597123478/954789827'
21    },
22    {
23      URL:'https://fex.net/play/387597123478/970464711'
24    },
25    {
26      URL:'https://fex.net/play/387597123478/987730760'
27    }
28  ]
29 },
30 {
31 },
32 },
33 },
34 },
35 },
36 },
37 },
38 },
39 },
40 },
41 },
42 },
43 },
44 },
45 },
46 },
47 },
48 },
49 },
50 },
51 },
52 },
53 },
54 },
55 },
56 },
57 },
58 },
59 },
60 },
61 },
62 },
63 },
64 },
65 },
66 },
67 },
68 },
69 },
70 },
71 },
72 },
73 },
74 },
75 },
76 },
77 },
78 },
79 },
80 },
81 },
82 },
83 },
84 },
85 },
86 },
87 },
88 },
89 },
90 },
91 },
92 },
93 },
94 },
95 },
96 },
97 },
98 },
99 },
100 },
101 },
102 },
103 },
104 },
105 },
106 },
107 },
108 },
109 },
110 },
111 },
112 },
113 },
114 },
115 },
116 },
117 },
118 },
119 },
120 },
121 },
122 },
123 },
124 },
125 },
126 },
127 },
128 },
129 },
130 },
131 },
132 },
133 },
134 },
135 },
136 },
137 },
138 },
139 },
140 },
141 },
142 },
143 },
144 },
145 },
146 },
147 },
148 },
149 },
150 },
151 },
152 },
153 },
154 },
155 },
156 },
157 },
158 },
159 },
160 },
161 },
162 },
163 },
164 },
165 },
166 },
167 },
168 },
169 },
170 },
171 },
172 },
173 },
174 },
175 },
176 },
177 },
178 },
179 },
180 },
181 },
182 },
183 },
184 },
185 },
186 },
187 },
188 },
189 },
190 },
191 },
192 },
193 },
194 },
195 },
196 },
197 },
198 },
199 },
200 },
201 },
202 },
203 },
204 },
205 },
206 },
207 },
208 },
209 },
210 },
211 },
212 },
213 },
214 },
215 },
216 },
217 },
218 },
219 },
220 },
221 },
222 },
223 },
224 },
225 },
226 },
227 },
228 },
229 },
230 },
231 },
232 },
233 },
234 },
235 },
236 },
237 },
238 },
239 },
240 },
241 },
242 },
243 },
244 },
245 },
246 },
247 },
248 },
249 },
250 },
251 },
252 },
253 },
254 },
255 },
256 },
257 },
258 },
259 },
260 },
261 },
262 },
263 },
264 },
265 },
266 },
267 },
268 },
269 },
270 },
271 },
272 },
273 },
274 },
275 },
276 },
277 },
278 },
279 },
280 },
281 },
282 },
283 },
284 },
285 },
286 },
287 },
288 },
289 },
290 },
291 },
292 },
293 },
294 },
295 },
296 }

```

**Ilustración 16: Ejemplo de archivo JSON generado automáticamente**



# Manual de Usuario



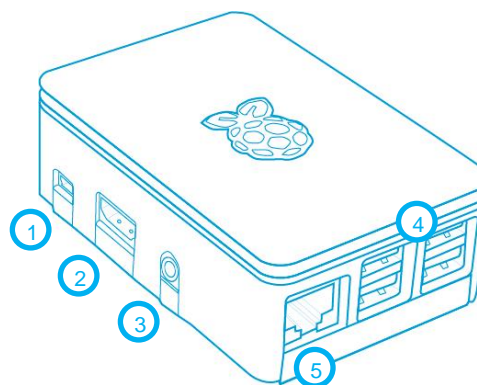
## 8.3 Manual de usuario

### Requisitos

- Televisor con entrada HDMI.
- Raspberry Pi.
- Dispositivo Móvil.
- Aplicación lectora de código QR.
- Conexión a internet de 2 Mbps mínimo.

### Raspberry Pi 3 modelo B

1. Puerto de Carga.
2. Entrada HDMI.
3. Salida de Audio
4. Puertos USB.
5. Conector de Red RJ45.



### Instrucciones

Conecte la Raspberry al televisor utilizando el cable HDMI, proceda a conectar el cable de red al router y por último conecte la Raspberry al tomacorriente.



## Control Remoto



Una vez encendida la Raspberry aparecerá en nuestra pantalla un mensaje de bienvenida con un código QR (1) con el fin de hacer uso del control remoto desde nuestros dispositivos móviles (2), Abrimos nuestra aplicación para escanear el código y esperamos que cargue el control remoto en nuestro dispositivo.

## Estructura del control remoto

### Funciones Básicas

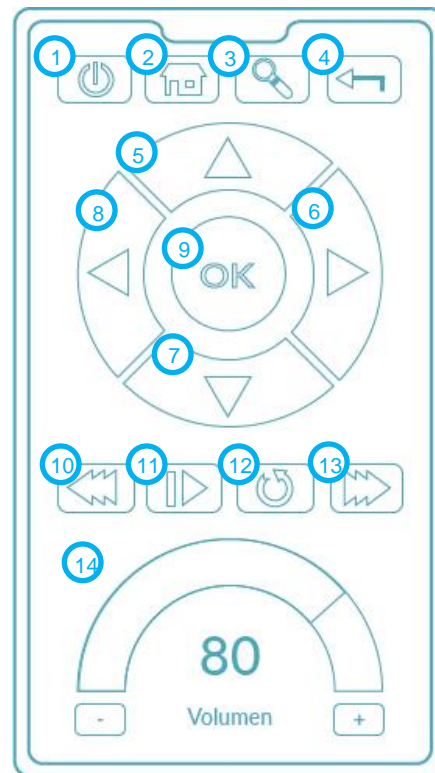
1. Apagar.
2. Recargar página.
3. Buscar.
4. Regresar.

### Desplazamiento

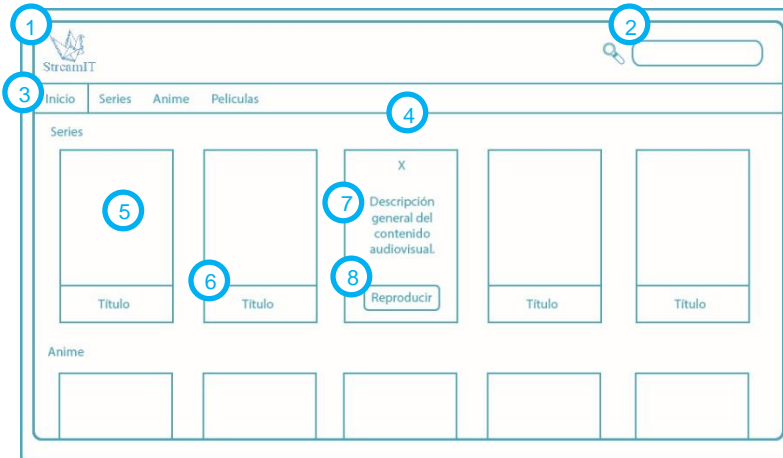
5. Arriba.
6. Derecha.
7. Abajo.
8. Izquierda.
9. Realizar Acción.

### Manipulación de Video

10. Retroceder.
11. Reproducir / Detener.
12. Reiniciar. Avanzar.
13. Volumen.



## Sitio Principal



1. Logo.
2. Campo de Búsqueda.
3. Menú de Categorías.
4. Contenido.
5. Imagen de Portada.
6. Título.
7. Descripción.
8. Reproducir.

## Interfaz de reproducción de contenido audiovisual

1. Cerrar la ventana.
2. Ocultar Detalles.
3. Mostrar Detalles.
4. Contenido Audiovisual.
5. Título.
6. Categoría.
7. Descripción
8. Lista de capítulos  
(Disponible para anime y series).

