**UNIVERSIDAD DE SONSONATE**

**FACULTAD DE INGENIERÍA Y CIENCIAS NATURALES**

**TEMA:**

DESARROLLO DE UN SERVIDOR MULTIMEDIA UTILIZANDO UNA RASPBERRY PI COMO DISPOSITIVO PRINCIPAL

**ANTEPROYECTO DE TRABAJO DE GRADUACIÓN**

**PRESENTADO POR:**

RICARDO CESAR CARRANZA GUTIERREZ

JAIME ENRIQUE HENRIQUEZ CRUZ

DANIEL SIGFREDO MATE HERNÁNDEZ

**PARA OPTAR AL GRADO DE**

INGENIERO EN SISTEMAS COMPUTACIONALES

**ABRIL 2019**

**SONSONATE, EL SALVADOR, CENTROAMÉRICA**

# AUTORIDADES DE LA UNIVERSIDAD

ANA MARIA DE LOS ANGELES RODRIGUEZ SALAZAR

RECTORA

LICENCIADO MAURICIO ALFREDO BERNAL HENRIQUEZ

FISCAL

DOCTOR ARONETTE DIAZ PINEDA

SECRETARIA GENERAL

INGENIERO RUBEN ALDREDO MONDOZA JÚAREZ

DECANO FACULTAD DE INGENIERIA Y CIENCIAS NATURALEZ

### JURADO EVALUADOR

PRESIDENTE

PRIMER VOCAL

SEGUNDO VOCAL

ASESOR

Ing. David Gildardo Rajo Castellanos

### INDICE GENERAL

[AUTORIDADES DE LA UNIVERSIDAD ii](#_Toc7081832)

[JURADO EVALUADOR iii](#_Toc7081833)

[INDICE GENERAL iv](#_Toc7081834)

[INTRODUCCION 1](#_Toc7081835)

[Capítulo I. Planteamiento del Problema 2](#_Toc7081836)

[1.1 El Problema de Investigación 3](#_Toc7081837)

[1.2 Objetivos 4](#_Toc7081838)

[1.3 Justificación 4](#_Toc7081839)

[1.4 Delimitación del Proyecto 5](#_Toc7081840)

[Capítulo II. Marco de la Investigación 6](#_Toc7081841)

[2 Marco de la Investigación 7](#_Toc7081842)

[2.1 WebSocket 7](#_Toc7081843)

[2.2 Progressive Web App 8](#_Toc7081844)

[2.3 Frameworks: 9](#_Toc7081845)

[2.4 Raspberry Pi 10](#_Toc7081846)

[2.5 Linux - Raspian OS 10](#_Toc7081847)

[2.6 Python. 11](#_Toc7081848)

[2.7 Django: 13](#_Toc7081849)

[2.8 Servidores Multimedia 14](#_Toc7081850)

[2.9 Acceso Multimedia universal 15](#_Toc7081851)

[2.10 UPNP: 15](#_Toc7081852)

[Capítulo III. Metodología de la Investigación 17](#_Toc7081853)

[3. Metodología de la Investigación 18](#_Toc7081854)

[3.1 Tipo de investigación 18](#_Toc7081855)

[3.2 Unidades de Análisis 19](#_Toc7081856)

[3.3 Variables y su Medición 19](#_Toc7081857)

[3.4 Procesamiento y Análisis de la Información 20](#_Toc7081858)

[Bibliografía 24](#_Toc7081859)

[Anexos 26](#_Toc7081860)

# INTRODUCCION

El presente trabajo que se describe a continuación tiene como principal objetivo el Desarrollo de un servidor multimedia para uso en aulas de recursos audiovisuales, utilizando una Raspberry Pi como dispositivo principal; potenciando de este modo la dinámica operativa de estas aulas, ya que facilitará el trabajo a los que hagan uso de estos recursos, evitará de igual forma el método tradicional de compartición de archivos multimedia, ya que se elimina por completo el uso de cables o dispositivos de almacenamiento extraíble, ofreciendo una solución basada completamente en tecnologías inalámbricas y en tiempo real.

De este modo, el planteamiento y desarrollo de este proyecto se fundamenta en el hecho de ofrecer una solución de vanguardia, fácil de usar, amigable y sobre todo funcional a los usuarios que hacen uso de las aulas multimedia, basados en el hecho de que, en su mayoría, estos usuarios tienen poco conocimiento sobre la manipulación avanzada de los equipos informáticos y de los medios que se utilizan para la proyección de contenidos multimedia.

Todo eso fundamentado en el marco teórico de la investigación, donde se describen los conceptos y tecnologías que se utilizarán para el desarrollo del proyecto, entre las cuales se pueden mencionar: Python (Lenguaje de programación), Django (Framework basado en Python para el desarrollo web), Angular, Ionic para el desarrollo de Web App y App móviles.

# Capítulo I. Planteamiento del Problema

### 1.1 El Problema de Investigación

Es muy común el uso de las salas multimedia o del uso de material multimedia en los salones de clases, aumentan la comprensión de los contenidos a través de ayudas audiovisuales, con el fin de fortalecer los conocimientos transmitidos a los estudiantes. De igual forma cada vez más las instituciones educativas como escuelas, colegios y universidades cuentan con el equipo necesario para el uso de estas salas multimedia, a saber, proyector de cañón, computadoras, internet, etc. lo que facilita el acceso de estos recursos, principalmente, a los docentes y por consiguiente a los estudiantes.

Tomando en cuenta que estos recursos antes mencionados son, mayormente, utilizados por docentes o catedráticos con poco o mucho conocimiento técnico para la manipulación de los mismos, ocasionando confusión o dificultad al momento de manipular dicha tecnología, gastando tiempo valioso de las clases, problemas tales como: los archivos a utilizar hay que transferirlos de un dispositivo USB a la computadora conectada al proyector de cañón, los archivos multimedia se almacenan en un Smartphone y no se cuenta con un medio de transferencia ágil para dicho dispositivo, como un cable, conexión bluetooth, etc.; entre otros problemas técnicos que surgen.

Es por eso que se propone el desarrollo de un servidor multimedia, el cual consta de una Raspberry Pi, dispositivo del tamaño de una tarjeta de crédito, conectada al proyector de cañón u otro dispositivo de salida como monitor o TV a través de un cable HDMI, siendo este el receptor de los archivos multimedia que se deseen compartir, ya sea, videos, imágenes, audios, incluso presentaciones, a través de una aplicación móvil o una aplicación web desde una computadora. Todo esto facilitará la manipulación de los archivos multimedia, ya que se evitará el uso de cables, se hará más rápido el proceso de compartir los contenidos y también se cubrirá un mayor número de usuarios.

### 1.2 Objetivos

#### 1.2.1 Objetivo General:

Desarrollar un servidor multimedia, para utilizarlo en un aula de recursos audiovisuales, utilizando una Raspberry Pi como dispositivo principal.

#### 1.2.2 Objetivos Específicos:

Estudiar la situación actual de los recursos multimedia en salas o aulas de centros educativos, empresas e instituciones gubernamentales.

Desarrollar el prototipo funcional del Servidor Multimedia, la aplicación móvil y la Web App para uso desde una PC como cliente para compartir los archivos deseados.

Validar el correcto funcionamiento del servidor multimedia en un ambiente real, instalando la solución de software en un dispositivo Raspberry Pi.

### 1.3 Justificación

La presente investigación y desarrollo de software se enfoca en la construcción de un servidor multimedia que facilite la dinámica operativa de las aulas o salas multimedia. Construido bajo diversas y recientes tecnologías de desarrollo, tales como: Python (Lenguaje de programación), Django (Framework basado en Python para el desarrollo web), Angular, Ionic para el desarrollo de Web App y App móviles.

Esta iniciativa surge bajo el hecho de que se vuelve tediosa y a veces complicada, la tarea de compartir contenido multimedia en un aula o sala de conferencias, ya que no se cuentan con los conocimientos técnicos necesarios o el proceso es complicado. Este servidor multimedia viene a solucionar esta brecha y brindar un sistema más amigable y dinámico para dicha tarea, proporcionando una aplicación móvil como herramienta para controlar y compartir el contenido multimedia que se desea, además de una aplicación web si se desea compartir contenido desde una computadora, todo esto con la mayor simplicidad posible para entregar una experiencia de usuario más elegante y fácil de utilizar eliminando el uso de cables ya que todo se hará bajo tecnologías inalámbricas y en plataformas web.

### 1.4 Delimitación del Proyecto

Para este proyecto se desarrollará un prototipo funcional de un servidor multimedia, para compartir contenido audiovisual en salas multimedia de la Universidad de Sonsonate y en un ambiente local o en una red local, sin tener comunicación con la Internet.

El dispositivo a utilizar será una Raspberry Pi, que funcionará como servidor y una interfaz web como cliente que se podrá cargar desde una PC para el envío del contenido multimedia, además el desarrollo de una aplicación para dispositivos móviles multiplataforma, instalable en iOS y Android, que también servirá como cliente.

### Capítulo II. Marco de la Investigación

### 2 Marco de la Investigación

### 2.1 WebSocket

Internet se ha creado en gran parte a partir del llamado paradigma solicitud/respuesta de HTTP. Un cliente carga una página web y no ocurre nada hasta que el usuario hace clic en la página siguiente. Allá por 2005, AJAX empezó a hacer que Internet pareciera más dinámico. Aun así, todas las comunicaciones HTTP eran dirigidas por el cliente, lo que requería la interacción del usuario o hacía necesario preguntarle periódicamente cada vez que se cargaban nuevos datos del servidor.

La especificación WebSocket define un API que establece conexiones "socket" entre un navegador web y un servidor. Dicho con otras palabras: existe una conexión persistente entre el cliente y el servidor, y ambas partes pueden empezar a enviar datos en cualquier momento.

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor. La API de WebSocket está siendo normalizada por el W3C, mientras que el protocolo WebSocket ya fue normalizado por la IETF como el RFC 6455.

Debido a que las conexiones TCP comunes sobre puertos diferentes al 80 son habitualmente bloqueadas por los administradores de redes, el uso de esta tecnología proporciona una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios WebSocket sobre un único puerto TCP (a costa de una pequeña sobrecarga del protocolo).

En el lado del cliente, WebSocket está ya implementado en Mozilla Firefox 8, Google Chrome 4 y Safari 5, así como la versión móvil de Safari en el iOS 4.2.1​ y en Internet Explorer 10.

### 2.2 Progressive Web App

Las aplicaciones web progresivas usan las APIs web modernas junto a una estrategia de mejoras progresivas para crear aplicaciones multi-plataforma. Estas apps funcionan en todos los dispositivos y ofrecen varias características que brindan al usuario las mismas ventajas y mejoras que las aplicaciones nativas. Una Progressive Web App es:

* **Progresiva:** funciona para todos los usuarios, sin importar la elección de navegador, porque está construida con mejora progresiva como principio central.
* **Adaptable:** se adapta a cualquier factor de formulario, sea escritorio, móvil, tablet o lo que venga en el futuro.
* **Independiente de la conectividad:** mejorada con service workers para trabajar sin conexión o con redes de mala calidad.
* **Estilo app:** al usuario le parece una app con interacciones y navegación estilo app, porque está construida con modelo de shell de app.
* **Fresca:** siempre actualizada gracias al proceso de actualización de service worker.  
  Segura: emitida vía HTTPS para evitar intromisiones y para garantizar que el contenido no se haya manipulado.
* **Descubrible:** se puede identificar como "app" gracias al manifiesto W3C y al alcance de registro de service worker, lo que permite que los motores de búsqueda la encuentren.  
  **Posibilidad de volver a interactuar:** facilita la posibilidad de volver a interactuar a través de funciones como notificaciones push.
* **Instalable:** les permite a los usuarios "conservar" las apps que les resultan más útiles en su pantalla principal sin la molestia de una tienda de app.
* **Vinculable:** se puede compartir fácilmente vía URL, no requiere instalación compleja.

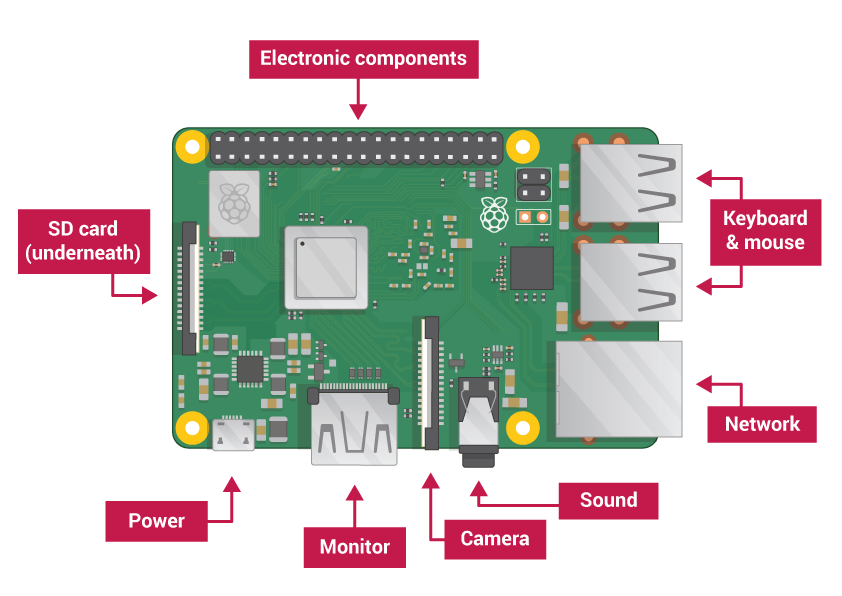
### 2.3 Frameworks:

Un framework o marco de trabajo de software proporciona una forma estándar de crear e implementar aplicaciones. Es un entorno de software universal y reutilizable que proporciona una funcionalidad particular como parte de una plataforma de software más grande para facilitar el desarrollo de aplicaciones, productos y soluciones de software. Los frameworks de software pueden incluir programas de soporte, compiladores, bibliotecas de código, conjuntos de herramientas e interfaces de programación de aplicaciones (API) que reúnen a todos los diferentes componentes para permitir el desarrollo de un proyecto o sistema:

Frameworks de desarrollo para PWA:

* **Angular:** Angular es una plataforma que facilita la creación de aplicaciones con la web. Angular combina plantillas declarativas, inyección de dependencia, herramientas de extremo a extremo y mejores prácticas integradas para resolver los desafíos de desarrollo. Angular permite a los desarrolladores crear aplicaciones que viven en la web, el móvil o el escritorio.
* **Ionic:** Ionic Framework es un kit de herramientas de UI (Interfaz de Usuario o User Interface por sus siglas en inglés) de código abierto para crear aplicaciones móviles y de escritorio de alta calidad y rendimiento utilizando tecnologías web (HTML, CSS y JavaScript), basado en Angular.

### 2.4 Raspberry Pi



Es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo coste desarrollado en el Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de informática.

El Raspberry Pi usa principalmente sistemas operativos GNU/Linux. [Raspbian](https://es.wikipedia.org/wiki/Raspbian" \o "Raspbian), una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la fundación para iniciarse.

### 2.5 Linux - Raspian OS

Raspbian es un sistema operativo gratuito basado en Debian optimizado para el hardware Raspberry Pi. Un sistema operativo es el conjunto de programas y utilidades básicos que hacen que su Raspberry Pi se ejecute. Sin embargo, Raspbian ofrece más que un sistema operativo puro: viene con más de 35,000 paquetes, software precompilado en un formato agradable para una fácil instalación en tu Raspberry Pi.

La compilación inicial de más de 35,000 paquetes de Raspbian, optimizada para el mejor rendimiento en la Raspberry Pi, se completó en junio de 2012. Sin embargo, Raspbian todavía está en desarrollo activo con un énfasis en mejorar la estabilidad y el rendimiento de la mayor cantidad posible de paquetes de Debian.

Raspbian es una de las distribuciones principales más antiguas y populares de la Raspberry Pi. Aunque se desarrolló de forma independiente, la Raspberry Pi Foundation también mantiene su propia versión recomendada de Raspbian que puede instalar usando el instalador [NOOBS](https://www.raspberrypi.org/downloads/noobs/) de [Foundation](https://www.raspberrypi.org/downloads/noobs/) .

La última versión se basa en el puerto ARM de la distribución de escritorio Debian Stretch, que mejora el rendimiento bajo el capó y ofrece nuevas aplicaciones en comparación con la versión anterior basada en Jessie.

Para la instalación, necesitará una tarjeta microSD con un tamaño mínimo de 8 GB. En el primer arranque, accederás al entorno de escritorio Pixel ligero y elegante. También puede elegir entre miles de programas adicionales, como el paquete LibreOffice o IceDove, todos los cuales pueden instalarse utilizando el administrador de paquetes integrado de Raspbian.

### 2.6 Python.

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

Es un [lenguaje interpretado](https://es.wikipedia.org/wiki/Lenguaje_interpretado), no compilado, usa [tipado dinámico](https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/caracteristicas.html#python-tipado-dinamico), [fuertemente tipado](https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/caracteristicas.html#python-fuertemente-tipado). Es [multiplataforma](https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/caracteristicas.html#python-multiplataforma), lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistema operativos.

Es un lenguaje de programación [multiparadigma](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_multiparadigma), el cual soporta varios paradigma de programación como [orientación a objetos](https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion9/poo.html#python-poo), [estructurada](https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion5/programacion_estructurada.html#python-programacion-estructurada), [programación imperativa](https://es.wikipedia.org/wiki/Programaci%C3%B3n_imperativa) y, en menor medida, [programación funcional](https://es.wikipedia.org/wiki/Programaci%C3%B3n_funcional).

En Python, el formato del código (p. ej., la indentación) es estructural.

* **Propósito general:** Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.
* **Fuertemente tipado:** El [fuertemente tipado](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_fuertemente_tipado) significa que el tipo de valor no cambia repentinamente. Un [string](https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion3/tipo_cadenas.html#python-str) que contiene solo dígitos no se convierte mágicamente en un número. Cada cambio de tipo requiere una conversión explícita.
* **Tipado dinámico:** El [tipado dinámico](https://es.wikipedia.org/wiki/Tipado_din%C3%A1mico) significa que los objetos en tiempo de ejecución (valores) tienen un tipo, a diferencia del tipado estático donde las variables tienen un tipo.
* **Multiplataforma:** Python es [multiplataforma](https://es.wikipedia.org/wiki/Multiplataforma), lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos. Quiere decir que no se debe compilar el código antes de su ejecución. En realidad, sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.
* **Interactivo:** Python dispone de un intérprete por la línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.
* **Orientado a Objetos:** La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
* **Funciones y librerías:** Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip

### 2.7 Django:

Django es un framework web diseñado para realizar aplicaciones de cualquier complejidad en unos tiempos muy razonables. Está escrito en Python y tiene una comunidad muy amplia, que está en continuo crecimiento.

Por qué usar Django

Los motivos principales para usar Django son:

* **Es muy rápido:** Si tenéis un startup, tenéis prisa por terminar vuestro proyecto o, simplemente, queréis reducir costes, con Django podéis construir una aplicación muy buena en poco tiempo.
* **Viene bien cargado:** Cualquier cosa que necesitéis realizar, ya estará implementada, sólo hay que adaptarla a vuestras necesidades. Ya sea porque hay módulos de la comunidad, por cualquier paquete Python que encontréis o las propias aplicaciones que Django trae, que son muy útiles.
* **Es bastante seguro:** Podemos estar tranquilos con Django, ya que implementa por defecto algunas medidas de seguridad, las más clásicas, para que no haya SQL Injection, no haya Cross site request forgery (CSRF) o no haya Clickjacking por JavaScript. Django se encarga de manejar todo esto de una manera realmente sencilla.
* **Es muy escalable:** Podemos pasar desde muy poco a una aplicación enorme perfectamente, una aplicación que sea modular, que funcione rápido y sea estable.
* **Es increíblemente versátil:** Es cierto que en un principio Django comienza siendo un Framework para almacenar noticias por sitios de prensa, blogs y este estilo de webs, pero con el tiempo ha ganado tanta popularidad que se puede usar para el propósito que queráis.

### 2.8 Servidores Multimedia

Un servidor de medios se refiere a un dispositivo de computadora dedicado o a un software de aplicación especializado, que va desde una máquina de clase empresarial que proporciona video a pedido hasta, más comúnmente, una pequeña computadora personal o NAS (Network Attached Storage) para el hogar, dedicado a almacenar diversos medios digitales (es decir, videos / películas digitales, audio / música y archivos de imágenes).

Por definición, un servidor de medios es un dispositivo que simplemente almacena y comparte medios. Esta definición es imprecisa y puede permitir que varios dispositivos diferentes se llamen Media Servers. Puede ser un simple almacenamiento conectado a la red , una PC con sistema de cine en casa (HTPC) con Windows XP Media Center Edition , MediaPortal o MythTV, o un servidor web comercial que aloja medios para un gran sitio web. En una configuración doméstica, un servidor de medios actúa como un agregador de información: video, audio, fotos, libros, etc. Estos diferentes tipos de medios (ya sean originados en DVD, CD, cámara digital). o en forma física) se almacenan en el disco duro del servidor de medios. El acceso a estos está disponible desde una ubicación central.

También se puede usar para ejecutar aplicaciones especiales que permiten a los usuarios acceder a los medios desde una ubicación remota a través de Internet.

El único requisito para un servidor de medios es un método para almacenar medios y una conexión de red con suficiente ancho de banda para permitir el acceso a esos medios. Dependiendo de los usos y aplicaciones que ejecute, un servidor de medios puede requerir grandes cantidades de RAM o una CPU potente y de múltiples núcleos. Se puede usar un RAID para crear una gran cantidad de almacenamiento, aunque generalmente no es necesario en un servidor de medio doméstico usar un RAID que genere un aumento del rendimiento porque las velocidades actuales de transferencia de la red doméstica son más lentas que las de la mayoría de los discos duros actuales. Sin embargo, un RAID. La configuración también se puede usar para evitar la pérdida de los archivos de medios debido a una falla del disco. Muchos servidores de medios también tienen la capacidad de capturar los medios.

### 2.9 Acceso Multimedia universal

El Acceso Multimedia Universal (Universal Multimedia Access - UMA) es la capacidad de un sistema o aplicación de acceder a contenido multimedia desde cualquier terminal a través de cualquier red. La tecnología UMA pretende poner a disposición de los usuarios diferentes representaciones de la misma información de forma transparente, adaptándola a diferentes terminales, redes de acceso y preferencias de usuario. La información sólo se crea una vez y el sistema UMA se encarga de personalizar el contenido deseado de la forma más rápida posible.

El objetivo es satisfacer una petición de contenido de forma eficaz y con garantía de obtener la versión más adecuada en función de las condiciones de acceso, independientemente del dispositivo utilizado por el usuario: teléfono móvil, PDA (Personal Digital Assistant), PC doméstico, web TV, estaciones de trabajo, etc. La red de acceso: Ethernet, xDSL, LAN, GSM, GPRS, UMTS, cable, etc.

El formato original del contenido. Las características del entorno de uso (variables en cada petición). Las preferencias del usuario: mensaje de texto o de voz, video en color o blanco y negro, calidad requerida, etc.

### 2.10 UPNP:

Universal Plug and Play es un protocolo diseñado para simplificar la configuración de redes.

Sus usos principales son 3:

* Anunciar la presencia de servicios UPnP (lo que permite encontrar un equipo de red cuya presencia se ha anunciado previamente, como una cámara)
* Atravesar el router (nombre, IGD, Internet Gateway Device o NAT, Network Traversal)
* Compartir audio y vídeo a través de la red local, lo que se denomina UPnP AV, el cual está soportado por el controlador edomus a través de los Actuadores UPnP

UPNP es una función que permite que los dispositivos en su red doméstica se descubran entre sí y accedan a ciertos servicios. A menudo, esto se usa para transmitir medios entre dispositivos en una red. Toneladas de diferentes dispositivos son compatibles con la transmisión UPnP en la actualidad (o su prima, DLNA), lo cual es genial, porque significa que puede copiar o descargar medios a la computadora de su hogar y transmitirlos directamente a su televisor, sin necesidad de transferir archivos.

El protocolo UPnP que le permite transmitir medios a través de su red no es el mismo que el de UPnP en su enrutador, que es notoriamente inseguro. El uso de UPnP para transmitir medios entre las computadoras de su propia casa no requiere que UPnP esté habilitado en su enrutador, ni representa un riesgo para la seguridad.

Hay muchas formas diferentes de configurar UPnP, y variará dependiendo de cada dispositivo que use. Explicaré cómo funciona y cómo configurarlo usando los métodos más fáciles que he usado.

### Capítulo III. Metodología de la Investigación

### 3. Metodología de la Investigación

### 3.1 Tipo de investigación

Principalmente el tipo de Investigación desarrollado en este proyecto es: La Investigación Descriptiva, la cual estudia cada una de las situaciones necesidades y eventos, es decir, como es y cómo se manifiesta determinado fenómeno; además, busca especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis.

En este caso al desarrollar esta investigación se determinarán las propiedades, características y perfiles que existan en el proceso o construcción del servidor multimedia en aulas donde se utilicen los recursos audiovisuales, esto con el fin de llegar a una definición clara y precisa de las variables y bases teóricas a tomar en cuenta al momento de analizar los insumos de la investigación.

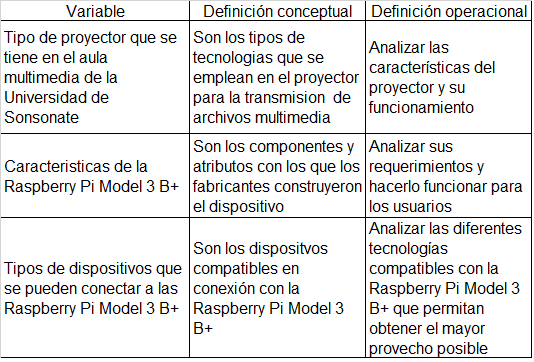
### 3.2 Unidades de Análisis

* Salas y aulas multimedia de la Universidad de Sonsonate
* Servidor Multimedia
* Desarrollo de aplicaciones móviles multiplataforma
* Raspberry Pi Model 3 B+

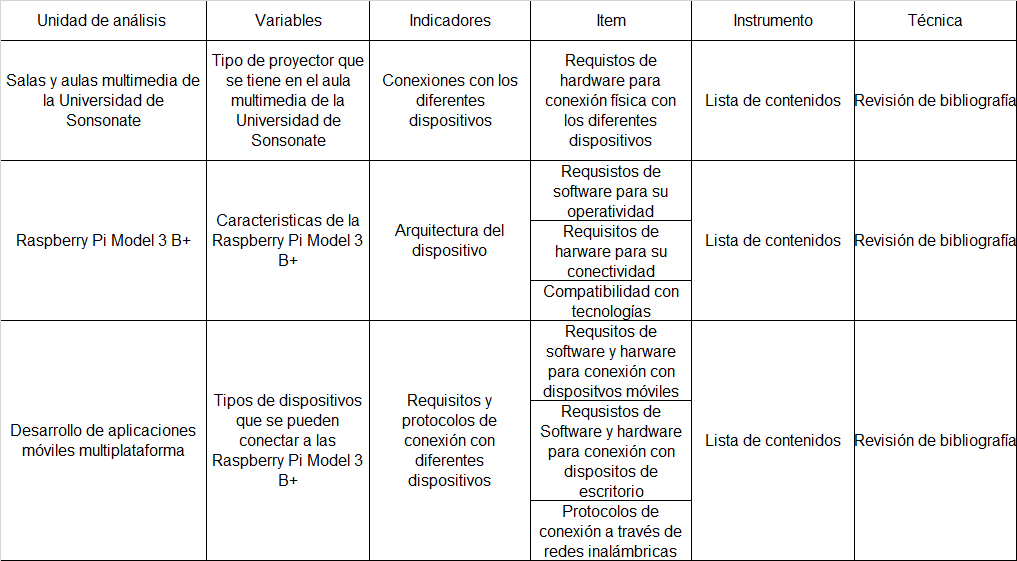
### 3.3 Variables y su Medición

#### 3.3.1 Definición de Variable

A continuación se presentan las definiciones de variables asociadas a las unidades de análisis para determinar sus indicadores de medición

.

#### 3.3.2 Medición de Variables



### 3.4 Procesamiento y Análisis de la Información

La consulta de bibliografía, permitirá obtener información sobre las diferentes tecnologías a utilizar en el desarrollo del tema de investigación, para luego hacer un análisis descriptivo de los diferentes acontecimientos del tema en estudio.

#### 3.4.1 Técnicas de investigación

Se utilizarán diferentes técnicas de recopilación de información, para garantizar una investigación que incluya todos los aspectos importantes que intervienen en la propuesta de solución del tema en estudio:

* Revisión de bibliografía
* Observación

Revisión Bibliográfica: El objetivo principal de esta modalidad es realizar una investigación documental, es decir, recopilar información ya existente sobre un tema o problema. Puedes obtener esta información de diversas fuentes como, por ejemplo, revistas, artículos científicos, libros, material archivado y otros trabajos académicos. Esta investigación documental proporciona una visión sobre el estado del tema o problema elegido en la actualidad.

La observación: es la técnica de recogida de la información que consiste básicamente, en observar, acumular e interpretar las actuaciones, comportamientos y hechos de las personas u objetos, tal y como las realizan habitualmente, todo ello con el fin de obtener determinada información necesaria para una investigación. En este proceso se busca contemplar en forma cuidadosa y sistemática cómo se desarrolla dichas características en un contexto determinado, sin intervenir sobre ellas o manipularlas. Es la técnica de información básica, sobre la que se sustenta todas las demás, ya que se establece la relación básica entre el sujeto que observa y el objeto observado, que es el inicio de toda comprensión de la realidad. En este caso se empleará la Lista de Control, CheckLists u hojas de Verificación, instrumento correspondiente a la observación, para tomar nota del contenido del módulo de Gestión de Flujo de Efectivo y Gestión de Comandas en cocina.

#### 3.4.2 Instrumento de Medición

Para esta investigación se utilizarán como instrumentos de medición lo siguiente:

* Listado de contenido
* CheckLists

Listado de contenido: Una tabla de contenido es un reflejo de la estructura de un documento y contiene los títulos de los temas y subtemas que forman el documento.

Una tabla de contenidos puede contener o no el número de página y puede establecerse un enlace directo a los puntos contenidos en la tabla. La tabla de contenido suele figurar al principio de un libro y es lo primero que miramos cuando queremos saber de qué temas trata el documento.

CheckLists: Es un instrumento útil preparado para recopilar información de un objeto de estudio basado en la observación. Luego de recolectar la información a través de las técnicas de investigación se procederá a la clasificación, procesamiento y análisis de la información, recolectada para proporcionar una conclusión.

#### 3.4.3 Metodología de desarrollo del prototipo del servidor multimedia

La metodología de desarrollo del proyecto de software será: La Metodología ágil SCRUM, prevaleciendo los 4 principios del manifiesto ágil (Konwledge, 2013): 50

1. Los individuos y su interacción, por encima de los procesos y las herramientas.

2. El software que funciona, frente a la documentación exhaustiva.

3. La colaboración con el cliente, por encima de la negociación contractual.

4. La respuesta al cambio, por encima del seguimiento de un plan.

### 

### Bibliografía

**Referencias Bibliográficas**

Sampieri, R. H. (2014). *Metodologia de la Investigación.* Mexico, Distrito Federal: MC GRAW HILL.

**Referencias Web**

Álvarez, M. Á. (19 de 11 de 2003). *Lenguaje de programación de propósito general, orientado a objetos, que también puede utilizarse para el desarrollo web*. Recuperado el 19 de 04 de 2019, de Desarrollo Web: https://desarrolloweb.com/articulos/1325.php

Angular. (12 de Enero de 2019). *Angular*. Recuperado el 25 de Febrero de 2019, de angular: https://angular.io/docs

COVANTEC. (01 de Diciembre de 2018). *PYTHON*. Recuperado el 17 de Febrero de 2019, de ACERCA DE PYTHON: https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/caracteristicas.html

desarrolloweb. (01 de Octubre de 207). *desarrolloweb.com*. Recuperado el 13 de Febrero de 2019, de Python: https://desarrolloweb.com/articulos/1325.php

Docs. (18 de octubre de 2018). *Ionic Framework*. Recuperado el 21 de Febrero de 2019, de Ionic Framework: https://ionicframework.com/docs/intro

eedemon documentation. (13 de Septiembre de 2018). *documentation UPnP*. Recuperado el 23 de Febrero de 2019, de UPnP: https://doc.eedomus.com/es/index.php/UPnP

MDN Wed doc. (15 de Febrero de 2018). *WebSocket*. Recuperado el 15 de Febrero de 2019, de WebSocket: https://developer.mozilla.org/es/docs/Web/API/WebSocket

Openwebinars.net. (03 de Agosto de 2018). *Django*. Recuperado el 19 de Febrero de 2019, de Conoce Django: https://openwebinars.net/blog/que-es-django-y-por-que-usarlo/

Raspian. (02 de Diciembre de 2018). *Raspian host*. Recuperado el 09 de Febrero de 2019, de Raspian host: https://www.raspbian.org/

Web Fundamentals. (10 de Enero de 2018). *Progressive Web App*. Recuperado el 19 de Febrero de 2019, de Progressive Web App: https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/?hl=es

Wikipedia. (15 de Diciembre de 2017). *Media Server*. Recuperado el 18 de Febrero de 2019, de Servidor Multimedia: https://en.wikipedia.org/wiki/Media\_server

Wikipedia. (18 de Mayo de 2018). *Multimedia Universal*. Recuperado el 22 de Febrero de 2019, de Acceso Multimedia Universal: https://es.wikipedia.org/wiki/Acceso\_Multimedia\_Universal

Wikipedia. (13 de Noviembre de 2018). *Software framework*. Recuperado el 21 de Febrero de 2019, de Software framework: https://en.wikipedia.org/wiki/Software\_framework

Wikipedia. (05 de Agosto de 2018). *WebSocket*. Recuperado el 20 de Febrero de 2019, de WebSocket: https://es.wikipedia.org/wiki/WebSocket

### Anexos

#### Anexo 1 Cronograma de Actividades

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CRONOGRAMA DE ACTIVIDADES |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ACTIVIDADES | JULIO |  |  |  | AGOSTO |  |  |  | SEPTIEMBRE |  |  |  | OCTUBRE |  |  |  | NOVIEMBRE |  |  |  | DICIEMBRE |  |  |  |
|  | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Recolección de Información |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Análisis de los Resultados |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Desarrollo de la Propuesta |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Preparación del Informe Final |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Defensa Oral del Informe Final |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Graduación |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |