

**MECANISMO PARA EL ENRIQUECIMIENTO DE SERVICIOS EN TIEMPO REAL
SOBRE UN ENTORNO DE IPTV USANDO LA TECNOLOGÍA WEBRTC**



**CARLOS JULIAN DELGADO MUÑOZ
LEINER JOHAN MOSQUERA ARANDA**

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Grupo de Ingeniería Telemática
Línea de Investigación en Servicios Avanzados de Telecomunicaciones
Popayán, 2017

**MECANISMO PARA EL ENRIQUECIMIENTO DE SERVICIOS EN TIEMPO REAL
SOBRE UN ENTORNO DE IPTV USANDO LA TECNOLOGÍA WEBRTC.**



**Trabajo de Grado presentado como requisito para obtener el título de
Ingeniero en Electrónica y Telecomunicaciones**

**CARLOS JULIAN DELGADO MUÑOZ
LEINER JOHAN MOSQUERA ARANDA**

Directora: Mg(C) Mary Cristina Carrascal Reyes

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Grupo de Ingeniería Telemática
Línea de Investigación en Servicios Avanzados de Telecomunicaciones
Popayán, 2017**

TABLA DE CONTENIDO

	PAG.
ANEXO A.....	5
DOMINIOS Y ARQUITECTURA FUNCIONAL IPTV	5
1. Dominios de IPTV	5
2. Arquitectura funcional de IPTV según la ITU	6
Funciones de aplicación	7
2.1. Funciones de aplicación IPTV	7
2.2. Funciones de preparación de contenido	7
2.3. Funciones de protección de servicio y contenido (SCP)	7
Funciones de Control del Servicio	7
2.4. Bloque funcional de control de servicio IPTV	7
2.5. Bloque funcional de perfil de usuario del servicio	7
Funciones de entrega de contenido	7
2.6. Funciones de control de ubicación y distribución del contenido	8
2.7. Funciones de almacenamiento y entrega de contenido	8
ANEXO B.....	8
DIAGRAMAS DE SECUENCIA	8
ANEXO C.....	15
MANUAL DE USUARIO DEL SISTEMA SOLUCIÓN.	15
ANEXO D.....	19
GUIA DE CONSTRUCCIÓN DEL SISTEMA SOLUCIÓN.....	19
ANEXO E.....	22

Diagrama para la arquitectura	22
Modularización del mecanismo propuesto	22
Modularización del entorno IPTV y su servicio VoD	23
Modularización del servicio de videollamada usado como caso de estudio	24
Descripción de entorno de pruebas y escenario.....	26
Descripción del entorno.....	26
Descripción del escenario	28
Pruebas de funcionalidad	28
Organización de la información	28
Uso de la técnica de caja negra	30
Resultados Obtenidos	31
Pruebas de desempeño	31
Módulo Conmutador.....	33
Módulo Funciones WebRTC	34
Módulo Controlador.....	35

ANEXO A

DOMINIOS Y ARQUITECTURA FUNCIONAL IPTV

1. Dominios de IPTV

La arquitectura de IPTV soporta la existencia de múltiples entidades con diferentes regiones de control administrativo e intereses particulares. IPTV extiende básicamente sobre cinco dominios principales [].

1. El **dominio del consumidor** que presenta los servicios al usuario final para su consumo.
2. El dominio del proveedor de red encargado de la conexión entre los clientes, el dominio del proveedor del servicio y de plataforma.
3. El dominio del proveedor de plataforma, es el responsable de proporcionar servicios comunes tales como la autenticación de usuario y demás al proveedor de del servicio, además también es posible el aprovisionamiento de otros servicios aparte de los servicios de IPTV, como servicios de comunicación personalizados.
4. El dominio del proveedor del servicio, como su nombre lo indica, es el responsable de suministrar a los clientes los servicios propietarios.
5. El dominio del proveedor de contenido el cual que puede ser propio o licenciado esta encargado de vender contenidos a los proveedores de servicios.

Los dominios mencionados anteriormente para el despliegue de los servicios de IPTV son consignados en la Figura 7 “Dominios de IPTV”, como se puede observar es combinado el dominio de proveedor de plataforma y el dominio de proveedor de servicio en un solo dominio según el criterio de la ITU, estos dominios no definen un modelo comercial ya que un proveedor puede cubrir varios dominios.

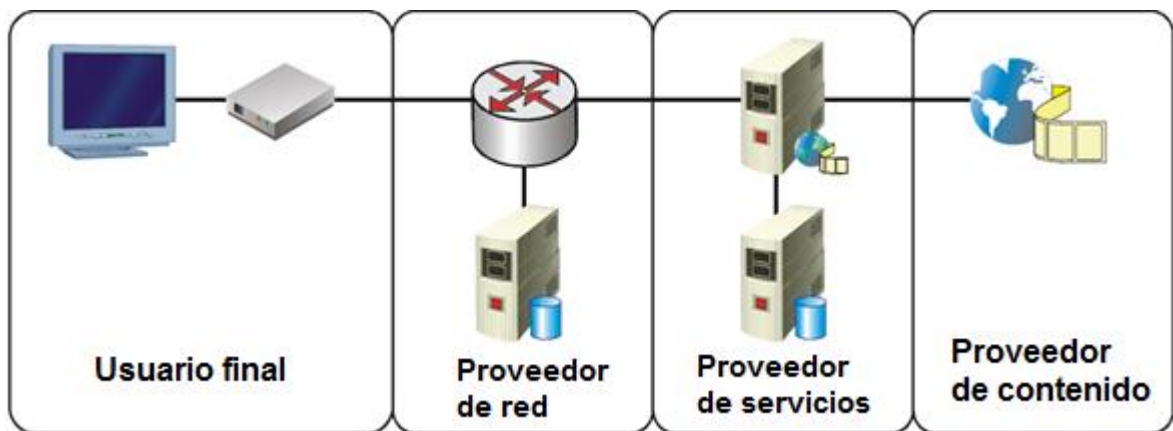


Figura 1: Dominios de IPTV

El presente trabajo de grado está enmarcado en el dominio del proveedor de servicio entendido como la combinación del dominio de proveedor de plataforma y el dominio de proveedor de servicio definidos por el OIPF anteriormente.

2. Arquitectura funcional de IPTV según la ITU

La arquitectura funcional para los servicios de IPTV ofrece una descripción detallada de los dominios nombrados anteriormente, la arquitectura funcional une los dominios de Proveedor de red y Proveedor de servicio ya que una separación de dominios por factores comerciales u operativos no es práctica para la descripción funcional de la arquitectura.

En la Figura 8 “Arquitectura funcional” se evidencia una vista general de la arquitectura funcional de IPTV, resaltando con un cuadro rojo punteado las entidades funcionales sobre las cuales es llevado a cabo el desarrollo de este proyecto.

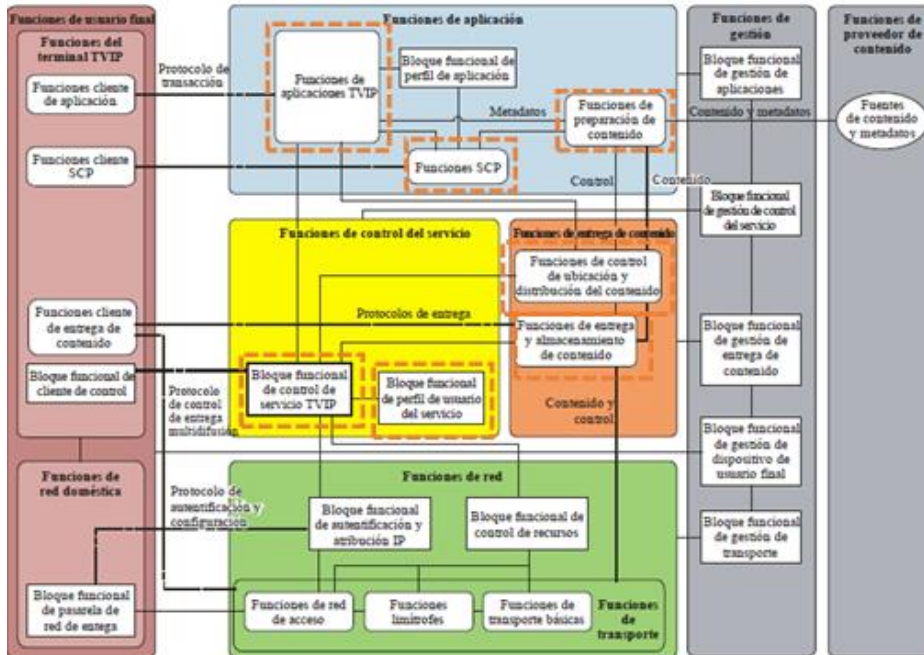


Figura 2: Arquitectura funcional IPTV

Es importante notar que, según las convenciones definidas en el documento oficial, los bloques rectangulares representan los bloques funcionales en la arquitectura de IPTV, los rectángulos redondeados representación una agrupación particular de funciones, las líneas continuas representan una relación directa entre funciones o bloques funcionales y las discontinuas representan relaciones lógicas entre funciones de usuario final.

Esta arquitectura general es usada para implementaciones de IPTV no basadas en NGN y con algunas adiciones de funciones es usada también en implementaciones de IPTV basado en NGN e IPTV basado en NGN e IMS, para el presente trabajo de grado es seleccionada la arquitectura IPTV no basado en NGN ya que cuenta con funciones más generales y es coherente con la selección de perfil abierto realizada anteriormente.

A continuación, se describen las tareas que deben ser ejecutadas en cada bloque funcional.

Funciones de aplicación

2.1. Funciones de aplicación IPTV

Las funciones de la aplicación IPTV le permiten al terminal IPTV seleccionar contenido.

Cuando es recibida una solicitud del terminal IPTV, las funciones de aplicación IPTV proceden a la autorización de la aplicación y la ejecución de la lógica del servicio, considerando el perfil de usuario, los metadatos del contenido y otra información obtenida de las entidades pertinentes. Las funciones de aplicación IPTV también pueden comunicarse con las funciones de entrega de contenido para preparar la entrega de contenido al terminal IPTV a través de las funciones de entrega de contenido.

2.2. Funciones de preparación de contenido

Las funciones de preparación de contenido controlan la preparación y agregación de contenidos tales como la programación de VoD, la difusión de canales de TV, los metadatos y los datos de la CPG¹, tal y como se reciben de las funciones de proveedor de contenido.

2.3. Funciones de protección de servicio y contenido (SCP)

La protección del contenido comprende el control del acceso al mismo y su protección mediante métodos tales como la ciframiento. La protección del servicio incluye la autenticación y autorización de acceso a los servicios.

Funciones de Control del Servicio

2.4. Bloque funcional de control de servicio IPTV

El bloque funcional de control de servicio IPTV ofrece las funciones necesarias para tramitar las peticiones de inicio, modificación y terminación del servicio, realizar el control de acceso al servicio, establecer y mantener los recursos de red y de sistema necesarios para dar soporte a los servicios IPTV solicitados por el cliente IPTV.

2.5. Bloque funcional de perfil de usuario del servicio

El bloque funcional de perfil de usuario del servicio:

- Almacena el perfil de servicio de usuario final (es decir, los servicios IPTV a los que está abonado).
- Almacena el estado de presencia del usuario final (por ejemplo, en línea/fuera de línea).

Funciones de entrega de contenido

Las funciones de entrega de contenido (CDF) ejecutan funcionalidades de memoria intermedia (caché) y almacenamiento y entregan el contenido con arreglo a lo solicitado por las funciones de usuario final.

Las funciones de entrega de contenido interactúan con las funciones del usuario final (por ejemplo, la funcionalidad de control de reproducción).

Las funciones de entrega de contenido soportan mecanismos de unidifusión, multidifusión o ambos.

¹ CPG: Guías de contenido programado.

Las funciones de entrega de contenido constan de las siguientes funciones:

- Funciones de control de distribución y ubicación del contenido.
- Funciones de entrega y almacenamiento de contenido.

2.6. Funciones de control de ubicación y distribución del contenido

Las funciones de control de ubicación y distribución del contenido están encargadas de lo siguiente:

- Controlar la distribución de contenido.
- Seleccionar las funciones de almacenamiento y entrega de contenido adecuadas a las funciones de usuario final en función de determinados criterios, por ejemplo, la información recabada y la capacidad del terminal.

2.7. Funciones de almacenamiento y entrega de contenido

Las funciones de almacenamiento y entrega de contenido están encargadas de entregar el contenido al cliente utilizando las funciones de red (por ejemplo, mecanismos de unidifusión y/o multidifusión).

Las funciones de almacenamiento y entrega de contenido tienen a cargo lo siguiente:

- la entrega de contenido al usuario final.
- el almacenamiento del contenido y de la información conexas.
- la notificación del estado (por ejemplo, estado de carga y disponibilidad) a las funciones de control de ubicación y distribución del contenido.

Las tareas anteriormente descritas son consideradas para la construcción de funciones del sistema diseñado, aunque no son implementadas todas ya que algunas se escapan del alcance del proyecto.

ANEXO B

DIAGRAMAS DE SECUENCIA

El caso de uso “Registrar” es el primer caso de uso llevado a cabo en el sistema, su objetivo es almacenar nuevos perfiles de usuario que permitan el ingreso de nuevos actores al sistema, como consecuencia de este caso se desencadena el caso de uso “AccederGuíaContenido”, como se puede observar en la siguiente imagen del paso 5 al 8.

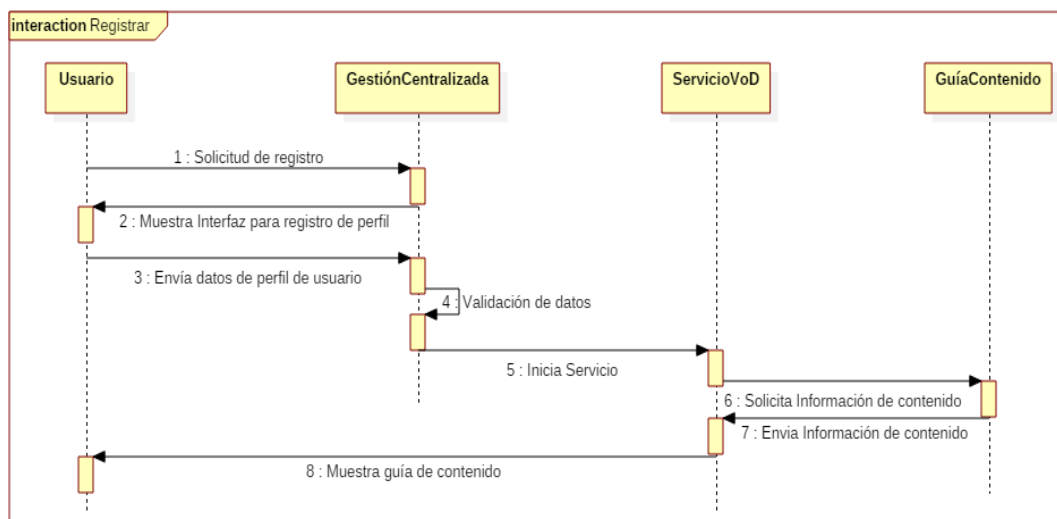


Figura 3 Diagrama de secuencia: caso de uso Registrar

Los pasos que son llevados a cabo para la ejecución del caso de uso “Registrar” son:

1. Solicitud de registro: El usuario desde la interfaz de inicio del sistema selecciona la opción de registrarse, indicándole al sistema su intención.
2. Muestra Interfaz para registro de perfil: Después de detectar la intención del usuario, el sistema despliega la interfaz construida para obtener los datos acordados ara el perfil de usuario.
3. Envía datos de perfil de usuario: Sobre la interfaz de captura de datos para el perfil de usuario, son llenados los campos dispuesto en el formulario y posteriormente son enviados estos datos para su validación.
4. Validación de datos: Durante su validación, el sistema verifica que la información llenada por el usuario sea adecuada, después procede a hacer el registro de esta en la base de datos.

El resto de pasos expuesto serán tratados en diagramas posteriores.

El caso de uso “AsociarContenidoMultimedia” exclusivo del administrador y descrito en la siguiente figura, permite la agregación de nuevo contenido multimedia para su despliegue en el servicio de VoD construido, este además incluye el caso de uso “AsociarContenidoInformativo” con el cual se agrega información referente al nuevo contenido agregado.

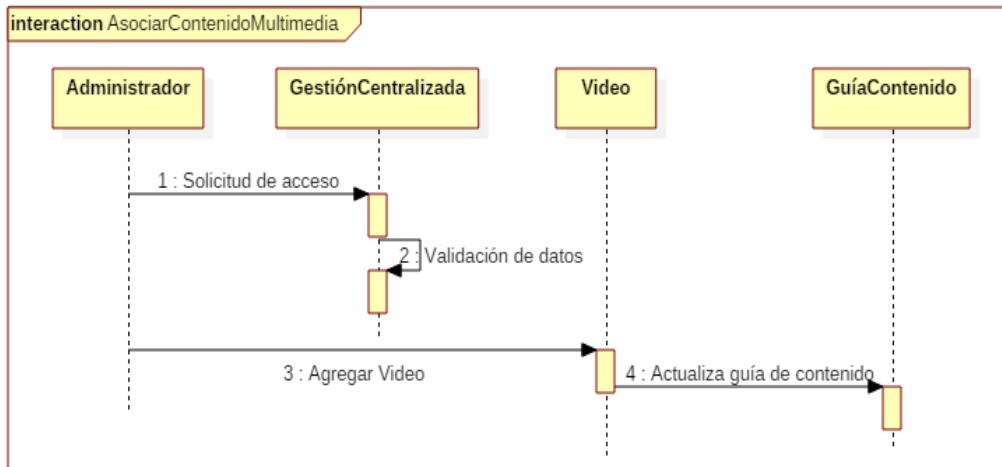


Figura 4 Diagrama de secuencia: caso de uso AsociarContenidoMultimedia

Los pasos involucrados en este caso de uso son:

1. Solicitud de acceso: Todos los actores del sistema deben realizar un proceso de validación antes de usar cualquier servicio del sistema, en este caso el administrador envía una solicitud de acceso enviando su alias y clave.
2. Validación de datos: Cuando el sistema recibe el alias y clave del usuario, este compara los datos enviados con los almacenados durante el registro del perfil de usuario, si los datos coinciden, valida si el actor es un usuario o un administrador y procede a desplegar los servicios propios de cada quien.
3. Agregación de Video: Después de acceder al sistema el usuario podrá agregar información del video que desea adicionar, como su ubicación relativa en el servidor web, nombre del video, descripción del contenido, tamaño, entre otros.

El caso de uso “AccederGuíaContenido” es el caso de uso central del sistema, es a partir de este que se extienden la gran mayoría de casos de uso, su objetivo es mostrar al usuario la guía de contenido presente para el servicio de VoD, al igual que para el caso anterior, es iniciado el proceso con una identificación del actor, aunque a diferencia del anterior este caso de uso es exclusivo para el usuario.

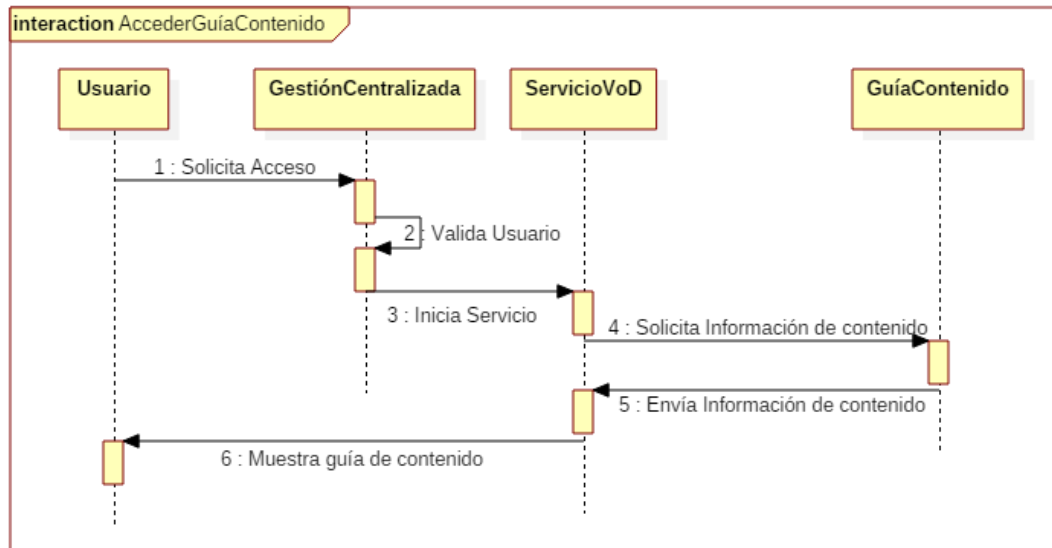


Figura 5 Diagrama de secuencia: caso de uso AccederGuíaContenido

A continuación son presentados los pasos ejecutados por el sistema para dar cumplimiento al actual caso de uso, ya que los pasos 1 y 2 son idénticos al caso de uso anterior se omite su explicación en este caso:

3. Inicia Servicio: Después de la identificación del usuario, el sistema solicita el despliegue del servicio de VoD sobre la interfaz principal, la cual contiene la guía de contenido.
4. Solicita Información de contenido: Durante el inicio del servicio de VoD, este solicita la información del contenido multimedia disponible para el usuario.
5. Envía Información de contenido: Como respuesta a la solicitud es enviada la información asociada a cada video disponible.
6. Muestra guía de contenido: La información entregada por la “GuíaContenido” es desplegada sobre la interfaz principal a manera de tabla, cada fila contiene un enlace que le permite al usuario consumir el contenido seleccionado, tal como es explicado en el siguiente caso de uso.

El caso de uso “ConsumirContenido” extiende del anterior caso, su objetivo es permitir el consumo del contenido multimedia disponible en el sistema, es activado a través de un enlace adicionado en la guía de contenido, con el cual el usuario puede seleccionar el video que desee ver y expresarle al sistema su intención.

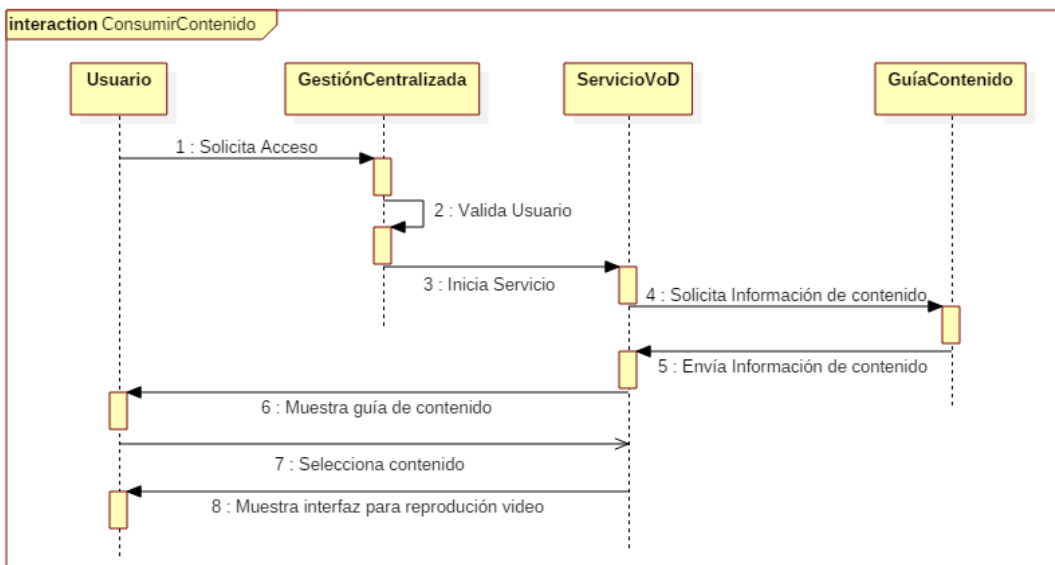


Figura 6 Diagrama de secuencia: caso de uso ConsumirContenido

Los pasos del 1 al 6 son similares a los descritos en el caso de uso anterior por lo cual serán omitidos en el actual caso de uso, a continuación son descritos los pasos 7 y 8 para el cumplimiento del caso de uso “ConsumirContenido”:

7. Selecciona contenido: El usuario selecciona un video de los disponibles en la guía de contenido que le ha sido mostrada.
8. Muestra interfaz para reproducción video: Cuando el sistema recibe la selección realizada por el usuario, despliega una nueva interfaz dedicada a la reproducción del video seleccionado.

El primer caso de uso dedicado exclusivamente al servicio de Videollamada es “VerConectados”, extiende del caso de uso “AccederGuíaContenido”, su objetivo es mostrar cuales usuarios se encuentran conectados al servicio de Videollamada y permitir el establecimiento de una comunicación con estos.

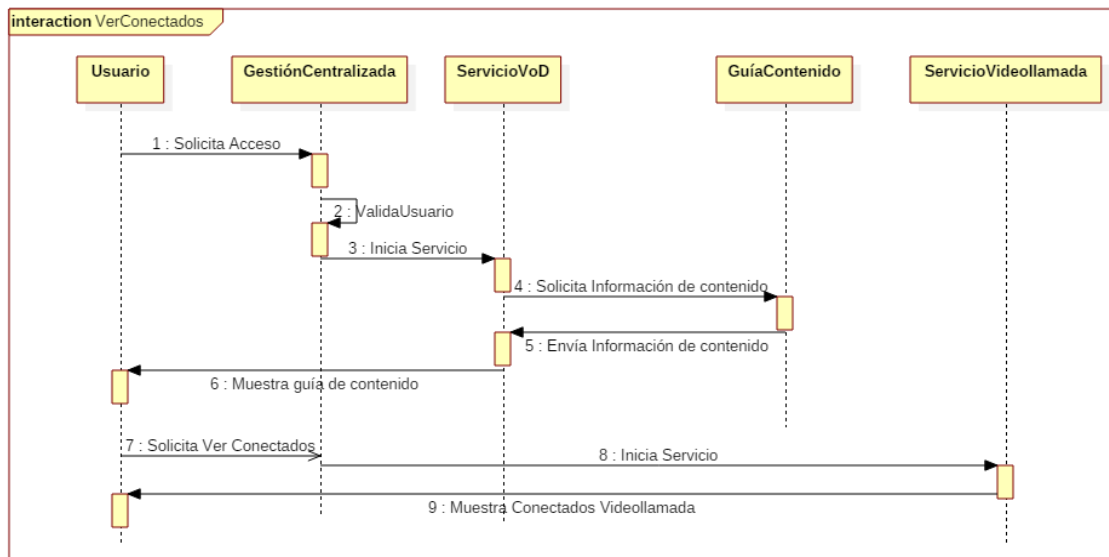


Figura 7 Diagrama de secuencia: caso de uso VerConectados

Los pasos del 1 al 6 expuestos en la figura anterior fueron descritos en el caso de uso “AccederGuíaContenido” por lo cual no serán descritos en este caso, por el contrario, los pasos 7,8 y 9 necesarios para el cumplimiento del caso de uso “VerConectados” serán descritos a continuación:

7. Solicita Ver Conectados: Estando en la interfaz principal del sistema el usuario solicita ver que otros usuarios se encuentran conectados al servicio de Videollamada
8. Inicia Servicio: La solicitud realizada por el usuario conlleva al inicio del servicio de Videollamada, el cual es ejecutado en este paso.
9. Mostrar Conectados WebRTC: Durante su inicio, el servicio de Videollamada despliega una nueva interfaz donde listará los usuarios que se encuentran actualmente conectados al servicio.

El caso de uso más importante del servicio de Videollamada es “LlamarUsuarioRemoto”, como su nombre lo indica su objetivo es establecer una llamada entre usuarios conectados al servicio, extiende del caso de uso “VerConectados” pues es a partir de la lista usuarios conectados que es posible dar inicio a una comunicación, incluye los casos de uso “EnviarNotificaciones” y “SolicitarPeriféricos” mostrados en los pasos 5, 7 y 8,11 respectivamente de la siguiente figura.

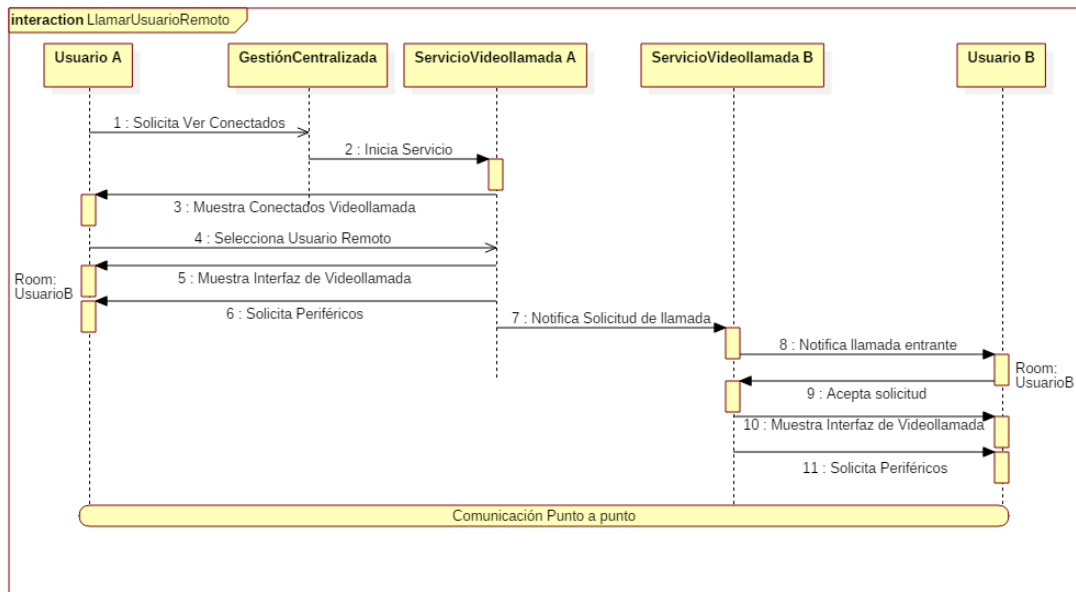


Figura 8 Diagrama de secuencia: caso de uso LlamarUsuarioRemoto

A continuación son descritos los pasos necesarios para la ejecución del actual caso de uso y los casos de uso incluidos. Los pasos 1,2 y 3 no serán descritos ya que fueron mencionados en el caso de uso anterior

4. Selecciona Usuario Remoto: De la lista de usuarios conectados, el usuario A puede seleccionar a quien desea llamar.
5. Muestra Interfaz de Videollamada: El servicio de Videollamada A despliega la interfaz construida para la comunicación entre usuarios, esta interfaz se convertirá una *Room* con el identificador del usuario B, al cual hemos llamado.
6. Solicitud de Periféricos: Durante el proceso de carga de la interfaz de Videollamada es ejecutado el caso de uso incluido “SolicitarPeriféricos”, el cual tiene como objetivo pedirle al usuario la autorización para que el servicio pueda disponer de sus periféricos, en este caso la cámara y el micrófono.
7. Notifica Solicitud de llamada: En paralelo al paso 5 y 6, el sistema envía una notificación al servicio Videollamada B, alojado en equipo del usuario B.
8. Notifica llamada entrante: El servicio de Videollamada B despliega sobre la interfaz en la cual se encuentra el usuario B una ventana emergente, indicándole que está siendo llamado.
9. Acepta solicitud: Dentro de la ventana emergente desplegada en el paso anterior, el usuario B puede elegir si acepta o no la solicitud, si acepta la solicitud son ejecutados los pasos 10 y 11 del mismo modo que los pasos 5 y 6, si por el contrario se rechaza la solicitud, se cierra la ventana emergente y puede continuar usando el sistema.

Finalmente es establecido el canal de comunicación punto a punto por donde se intercambiarán datos correspondientes a la Videollamada.

ANEXO C MANUAL DE USUARIO DEL SISTEMA SOLUCIÓN.


En este anexo es presentado el manual de usuario del sistema implementado en donde fue puesto a prueba el mecanismo propuesto y los servicios que sirvieron como caso de estudio, en este manual se describen los pasos que un usuario común debe realizar para usar el sistema de manera correcta.

1. Como primer paso, inicialmente el usuario accede a través de una URL del sitio donde está alojado el sistema, es importante notar que es necesario que el usuario este registrado en el sistema para poder acceder a todos los servicios allí prestados, para lo cual existen dos interfaces iniciales que pueden ser accedidas a través de un menú lateral desplegado. La primera (ver Figura N° 9) corresponde a la interfaz para registrarse en el sistema; cómo es posible observar, en ella aparecen todos los campos requeridos para completar y almacenar la información del consumidor en la base de datos del proveedor, seguido a esto ya es posible que el usuario ingrese al sistema por medio de la interfaz de inicio de sesión en donde es necesario ingresar los campos de nombre de usuario y su contraseña mostrados en la Figura N° 10.



The image shows a registration form titled "SIGN IN". It contains four input fields: "Name", "UserName", "Password 1 *", and "Password 2 *". A dark button labeled "Registrarse" is positioned at the bottom left of the form area.

Figura 9 Página de registro de usuarios



The image shows a login form titled "LOGIN". It contains two input fields: "UserName" with the text "cjdelgado" entered, and "Password" which is masked with seven dots. A dark button labeled "Entrar" is positioned at the bottom left of the form area.

Figura 10 Página de inicio de sesión.

2. Como segundo paso, está la página principal del sistema, que se muestra en la imagen N° 11, esta interfaz muestra la guía de contenido del servicio de VoD de IPTV

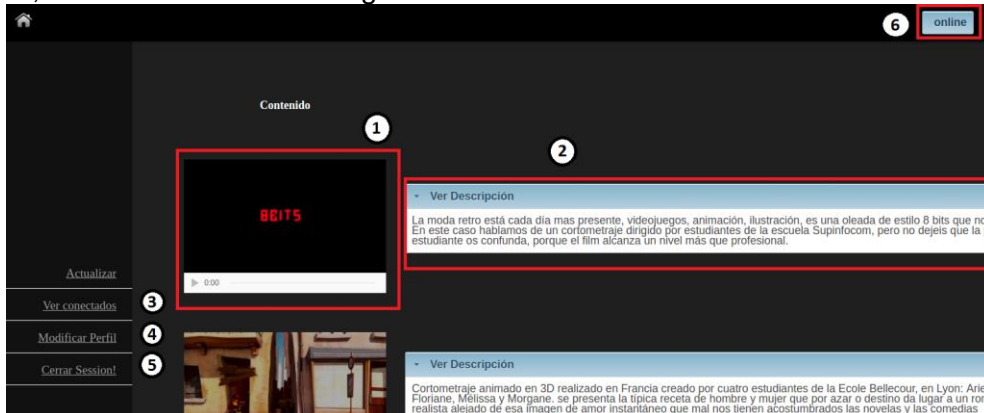


Figura 11 Catálogo del contenido multimedia

En “1” es posible ver la miniatura de un video que el consumidor puede elegir para ver en pantalla completa, en “2” se muestra la descripción de dicho video. Por otro lado, la interfaz posee un menú lateral en donde se encuentran otras opciones, en “3” está el enlace correspondiente para ver la lista de otras personas conectadas y que pueden ser llamadas cuando se desee, cabe resaltar que cada usuario elige en qué estado permanecer en el sistema, es decir, conectado o desconectado, para ello existe un botón ubicado en “6” que permite cambiar de estado cuando se quiera; en “4” el usuario tiene la opción de modificar su perfil para actualizar la información que este contenga. Y por último se encuentra el enlace para cerrar sesión cuando se quiera salir del sistema.

3. Como se mencionó antes si un usuario desea ver un video en especial, solo tiene que dar clic encima de él en la guía de contenidos, después de ello se muestra el contenido en modo pantalla completa como se muestra en la imagen N° 12.



Figura 12 Contenido multimedia en pantalla completa

4. Por otro lado, en usuario tiene la capacidad de realizar una video llamada con otro usuario, para cual debe listar a aquellos que se encuentren activos, es decir, conectados, en la imagen N° 13 se muestra dicha lista y en ella solo tiene que escoger a quien llamar.

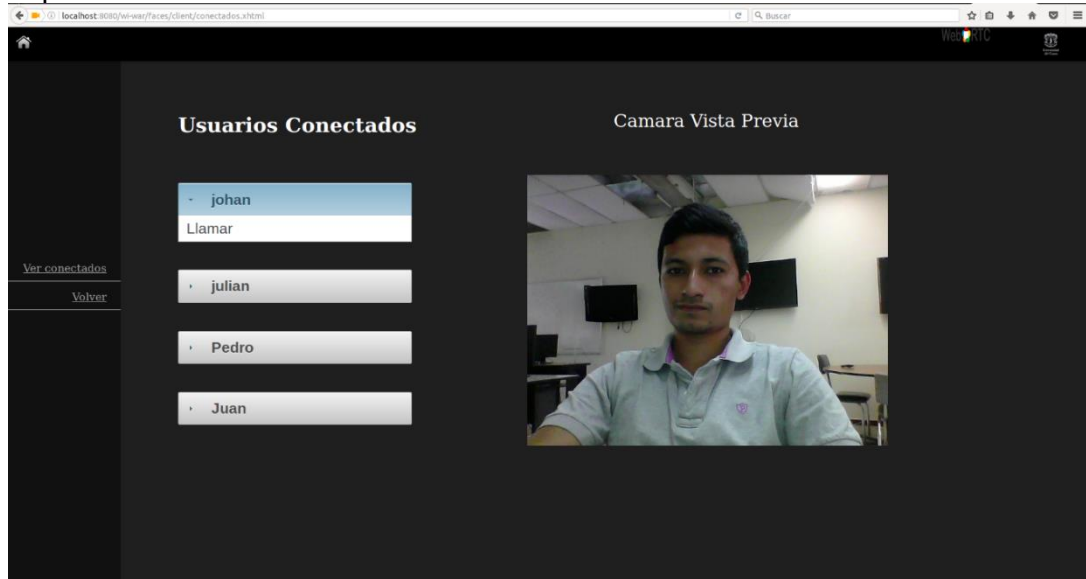


Figura 13 Usuarios conectados y vista previa de cámara usuario local.

5. Una vez inicie una llamada el usuario tendrá que aceptar el uso de sus dispositivos periféricos como la cámara y el micrófono.



Figura 14 Solicitud de uso de periféricos.

6. Cuando el usuario remoto responda la llamada hecha, entonces en la pantalla se dispondrán de dos recuadros para visualizar la video llamada, cada usuario en ella, además tendrá la opción cuando se desee en el menú de la izquierda, el cual lo llevará hasta la última página visitada.



Figura 15 Videollamada con WebRTC.

7. Si el usuario está activo para ser llamado, la notificación de llamada entrante se visualizará en la interfaz en donde dicho usuario se encuentre navegando y una vez finalizada el servicio se retorna a la página donde se encontraba.

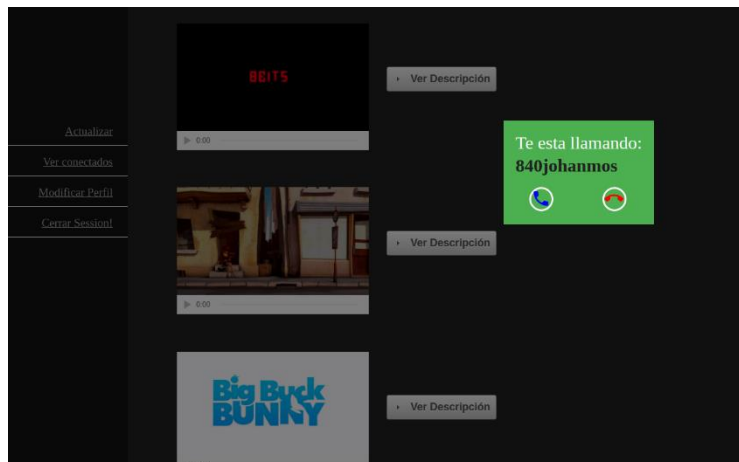


Figura 16 Notificación de llamada entrante.

ANEXO D

GUIA DE CONSTRUCCIÓN DEL SISTEMA SOLUCIÓN

La construcción del sistema solución esta basa en el patrón modelo-vista-controlador, para la construcción del modelo se hace uso de las herramientas prestadas por Netbeans para la creación automática de la capa de persistencia, en ella se abstraen las tablas de la base de datos en clases convirtiendo cada columna en un atributo de la clase, un ejemplo de estas clases se evidencia en la *Figura 17* donde en la línea 44 y 45 se observa la instanciación de un objeto de la clase “User” mapeado desde la base de datos; además de proveernos el mapeo a objetos, el modelo contiene las funciones necesarias para el manejo de la base de datos, es decir, alberga las funciones de consulta, edición, creación y eliminación de las tablas en la base de datos, específicamente para el actual proyecto de investigación el modelo se concentra en las clases EJB.

En la imagen a continuación (Ver Figura 17) se aprecia el uso de las clases del modelo junto con algunas otras variables necesarias para la lógica de la aplicación.

```
42 @EJB
43 private UserEJB userEJB;
44 private User user;
45 private User userRemote;
46 private List<User> listaConectados;
47 private String ipAddress = "192.168.0.14";
48 private Video videoSelected;
49 private String timeUserVideo;
50 Long time_start, time_end;
```

Figura 17 Notificación de llamada entrante.

En la siguiente imagen se visualiza la función encargada de consultar en la base de datos de usuarios aquellos que se encuentran conectados al sistema y disponibles para ser llamados, la lista encontrada es desplegada en la interfaz correspondiente al cliente.

```
66 public List<User> obtenerConectados(){
67     User user = new User();
68     user.setState(Boolean.TRUE);
69     List<User> listaConectados=new ArrayList<>();
70
71     Query query=em.createNamedQuery("User.findByState");
72     query.setParameter("state", user.getState());
73     try {
74         listaConectados=query.getResultList();
75     } catch (Exception e) {
76         e.printStackTrace();
77         return null;
78     }
79
80     return listaConectados;
81 }
```

Figura 18: Consulta de Conectados

En la siguiente función se visualiza el proceso en el cual se recibe como parámetro el identificador del usuario llamante, con este dato es consultado a la base de datos los datos requeridos del usuario en la base de datos.

```

520     public String showWebrtc(String roomRemote) {
521         time_start = System.currentTimeMillis();
522
523         userRemote = new User();
524         userRemote = userEJB.getUserbyRoom(roomRemote);
525
526         time_end = System.currentTimeMillis();
527         System.out.println("Tiempo de Ejecución " + (time_end - time_start) + " ms");
528
529         return "webrtc/webrtc.xhtml?faces-redirect=true";
530     }

```

Figura 29: Consulta de información de usuario

La siguiente función es la utilizada para el establecimiento de la comunicación entre dos puntos, esta función se ejecuta en el módulo controlador en el servidor cuando un usuario responde a una llamada entrante mientras disfruta del servicio de VoD, con ella es controlado el servicio de video bajo demanda para que el tiempo de reproducción e información del uso del servicio en esa sesión no se pierda, cabe resaltar que dentro de esta función se imprime el tiempo de ejecución de la función para verificar su desempeño en el despliegue del servicio, estos datos fueron expresados en el capítulo de pruebas de la monografía.

```

532     public String responseCall() {
533         time_start = System.currentTimeMillis();
534
535         userRemote = new User();
536         userRemote = userEJB.getUserbyRoom(userCalling);
537         userLocal = new User();
538         userRemote = user;
539
540         if (userRemote != null) {
541             if ("home".equals(lastPage)) {
542                 time_end = System.currentTimeMillis();
543                 System.out.println("Tiempo de Ejecución " + (time_end - time_start) + " ms");
544                 return "webrtc/webrtc.xhtml?faces-redirect=true";
545             } else {
546                 ///////////////////////////////////////////////////Se actualiza la tabla de current Time//////////////////////////////////////
547                 UserVid uv = new UserVid();
548                 uv = uservidEJB.getTimeByUser(user, videoSelected);
549                 uv.setCurrentTime(currentTime);
550                 uservidEJB.updateCurrentTime(uv);
551                 uservid = new UserVid();
552                 uservid = uservidEJB.getTimeByUser(user, videoSelected);
553                 time_end = System.currentTimeMillis();
554                 System.out.println("Tiempo de Ejecución " + (time_end - time_start) + " ms");
555                 return "webrtc/videoFullScreen.xhtml?faces-redirect=true";
556             }
557         } else {
558             time_end = System.currentTimeMillis();
559             System.out.println("Tiempo de Ejecución " + (time_end - time_start) + " ms");
560             return "webrtc/conectados.xhtml?faces-redirect=true";
561         }
562     }

```

Figura 20: Función responder llamada

La siguiente función es ejecutada en el navegador del cliente, fue realizada usando lenguaje JavaScript, y se encarga del inicio de la transmisión punto a punto de contenido multimedia, con ella es posible imprimir el audio y video en las interfaces correspondientes tanto del usuario local como del remoto.

```

93 <script type="text/javascript">
94
95     var endButton = document.getElementById("endButton");
96     endButton.onclick = end;
97
98     var aula;
99
100
101     function start() {
102         alerta();
103         aula = "#{userMB.userRemote.room}";
104         var webrtc = new SimpleWebRTC({
105             localVideoEl: 'localVideo',
106             remoteVideosEl: 'remoteVideo',
107             autoRequestMedia: true
108         });
109
110         webrtc.on('readyToCall', function () {
111             webrtc.joinRoom('' + aula);
112         });
113     }
114
115
116     function end() {
117         location.href = "http://#{userMB.ipAddress}:8080/WI-war/faces/client/index.xhtml";
118     }
119
120
121 </script>

```

Figura 21: Función inicio de llamada

En la siguiente fragmento de código se visualiza el modo en el cual se envían mensajes de alerta y notificación de llámate entrante y finalización a través de un Socket.

```

54 <script type="text/javascript">
55     var socket = io.connect('http://#{userMB.ipAddress}:3000');
56     function alerta() {
57
58
59         var mensaje = {
60             de: "#{userMB.user.room}",
61             para: "#{userMB.userRemote2.room}",
62             cierre: "activa"
63         }
64
65         socket.emit('chat message', mensaje);
66         return false;
67     }
68
69
70     function colgar() {
71         var mensaje = {
72             de: "#{userMB.user.room}",
73             para: "#{userMB.userRemote2.room}",
74             cierre: "inactiva"
75         }
76         socket.emit('chat message', mensaje);
77         return false;
78     }
79
80
81
82     socket.on("#{userMB.user.room}", function (data) {
83         var cierre = data['cierre'];
84         if (cierre == "inactiva") {
85             alert("La llamada ha sido finalizada");
86             document.getElementById('form_clg:clg').click();
87         }
88     });
89
90 </script>

```

Figura 22: Mensajes de control

ANEXO E

Diagrama para la arquitectura

Modularización del mecanismo propuesto

En la siguiente figura (ver figura 1) es explicado el modo de funcionamiento y la interacción de los módulos definidos para el mecanismo propuesto.

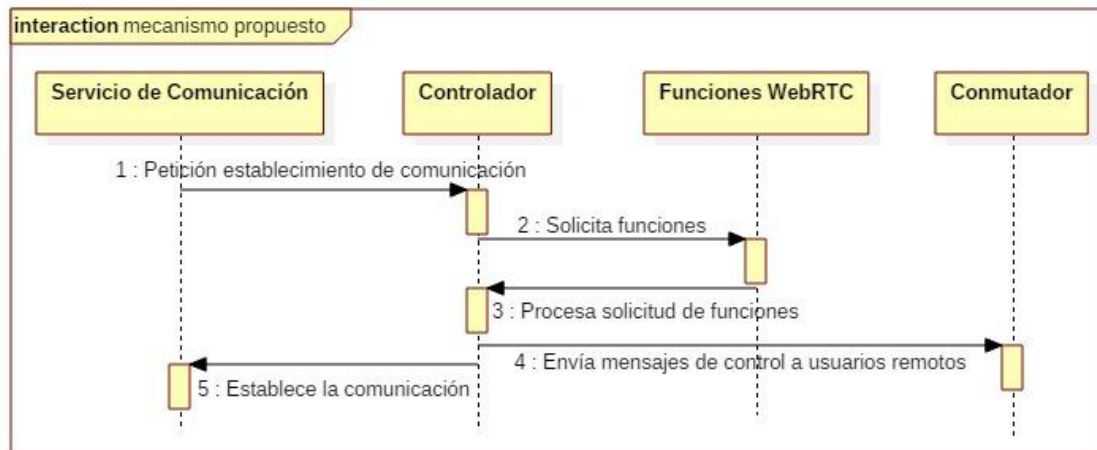


Figura 1: Diagrama de secuencia para el mecanismo propuesto

Módulos:

a) Funciones WebRTC

Este módulo es el encargado de hacer posible el inicio de una comunicación, en él es realizada la captura multimedia y la creación del canal de datos por el cual es enviado el flujo de información durante la ejecución del servicio

b) Conmutador

En este módulo fue realizada la implementación de un servidor Node JS, encargado de la recepción, control y envío de los mensajes de notificación a cada usuario que inicia una comunicación a través de una Videollamada. Estos mensajes permiten al proveedor de servicio controlar e identificar el estado de las comunicaciones establecidas, posibilitando así, una mejor gestión de los usuarios y un enriquecimiento en los servicios en tiempo real de WebRTC.

c) Controlador

En este módulo se realiza la gestión centralizada del software, es decir, es el corazón de la solución que realiza la toma de decisiones dependiendo de las acciones realizadas por un usuario conectado al sistema.

d) Servicio de Comunicación

Este bloque representa el servicio de Videollamada implementado como caso de estudio en el sistema para poner en marcha todos los módulos del mecanismo.

Secuencias:

1. Solicitud de establecimiento de comunicación

El bloque de los servicios de comunicación indica cuando un usuario desea iniciar un servicio y realiza una petición al módulo controlador.

2. Procesa solicitud de funciones

El modulo controlador realiza la petición al bloque de funciones WebRTC y le pasa la información del usuario llamante.

3. Contenido multimedia y mensajes de control

El módulo de funciones es el encargado de reenviar información de la configuración de la comunicación hecha para el inicio de una Videollamada, como por el ejemplo el objeto Peerconnection.

4. Envía mensajes de control a usuario remoto

Una vez el controlador recibe la información de conexión y del usuario llamante se activa el modulo encargado de enviar la notificación de llamada al usuario remoto llamado conmutador

5. Recibe mensajes de control de usuario remoto

El controlador recibe la respuesta del usuario remoto con el fin de determinar si ha aceptado la llamada, de ser positiva la respuesta envía el contenido multimedia del usuario B hacia el usuario llamante y se da inicio a la comunicación en tiempo real a través del canal de datos.

Modularización del entorno IPTV y su servicio VoD

Con el fin de visualizar la prestación e interacción de las tecnologías de WebRTC e IPTV, fueron definidos dos servicios entre los cuales se encuentra el servicio de VoD como fue explicado en la monografía, es por ello que a continuación es mostrado el diagrama de secuencia (ver figura 2) donde es evidenciado el modo de operación y ejecución del servicio en cuestión:

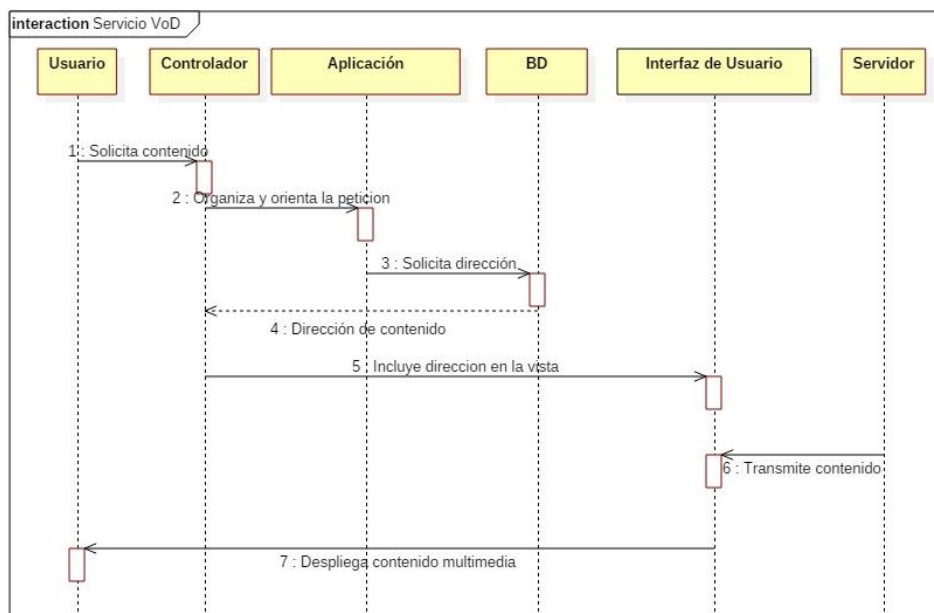


Figura 2: Diagrama de secuencia para el servicio de video bajo demanda

La base de datos almacena la información de cada contenido, entre las cuales se encuentra la dirección de alojamiento en el servidor de contenidos.

1. Solicita contenido multimedia

El terminal del usuario realiza la petición del contenido multimedia al controlador, dependiendo de que video el usuario desee mirar.

2. Organiza y orienta petición

El controlador recibe peticiones del usuario desde el navegador web, este la procesa y solicita la respuesta a la función correspondiente.

3. Solicita dirección de contenido multimedia

Dentro de la lógica de la aplicación, están los métodos encargados de hacer la consulta a la base de datos de cada contenido, la dirección de estos contenidos es almacenada en una lista para que puedan ser accedidos cuando sea requerido.

4. Responde con dirección contenido multimedia

Una vez obtenida la información de la base de datos, son armadas las direcciones completamente, adicionando la IP del servidor de contenidos a la ruta del archivo que el usuario ha solicitado.

5. Incluye dirección de contenido multimedia en la vista

El controlador una vez conseguida la dirección correspondiente llama al archivo XHTML y le pasa lo obtenido para que sea desplegado.

6. Trasmite contenido multimedia

El servidor de contenidos envía el video seleccionado a la página XHTML con la cual es respondida la petición del usuario.

7. Despliegue contenido multimedia

El navegador cliente interpreta el archivo XHTML con la respuesta del sistema permitiendo que el usuario puede disfrutar del contenido solicitado.

Modularización del servicio de videollamada usado como caso de estudio

Por otro lado, está el servicio de videollamada implementado usando la tecnología WebRTC, en la siguiente figura (ver figura 3) es plasmada la secuencia en la cual se ejecutan las funciones para el establecimiento de una comunicación entre dos puntos.

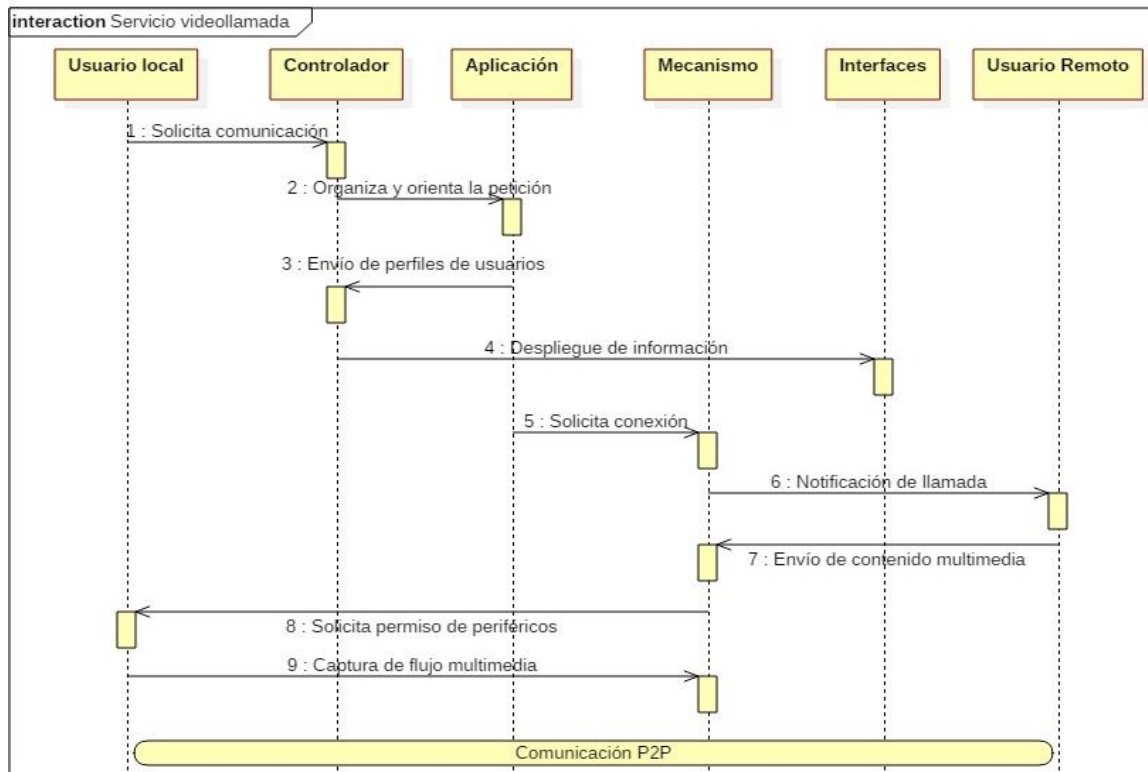


Figura 3: Diagrama de secuencia para el servicio de videollamada

El modo de ejecución es el siguiente:

1. Solicita establecimiento de comunicación

El usuario a través de la aplicación web desarrollada puede visualizar a otros usuarios conectados, además puede elegir llamar a quien desee.

2. Organiza y orienta petición.

El controlador recibe peticiones del usuario desde el navegador web, este las procesa y solicita la respuesta a las funciones correspondientes.

3. Envía información usuario local y remoto

El sistema consulta en la base de datos los perfiles de usuario y los tiene a disposición cuando estos se requieran, en este caso, cuando se solicita el establecimiento de una videollamada es cargada la información tanto del usuario local, quien llama, como del usuario remoto, quien es llamado.

4. Incluye información de los usuarios dentro de contenido XHTML

Despliega información del usuario local y remoto en las interfaces como sus nombres e incluye datos dentro del código JavaScript para ejecutar efectivamente el proceso de comunicación.

5. Solicita establecimiento de comunicación

Dentro del sistema solución es ejecutada la lógica encargada de invocar funciones del mecanismo que, permiten establecer la comunicación entre usuarios. Como primer paso está el envío de notificación de llamada entrante al usuario remoto para que este decida si

contestar o rechazarla, mientras esto sucede, la API WebRTC se prepara para iniciar la comunicación.

6. Usuario Remoto

- **6.1: Notifica solicitud de comunicación**

Es enviada una notificación al usuario remoto indicándole que el usuario local desea establecer una comunicación, por lo que tiene dos opciones: contestar o rechazar, si el consumidor rechaza la llamada entonces puede seguir disfrutando de la aplicación.

- **6.2: Envía contenido multimedia usuario remoto**

Si el usuario remoto acepta la llamada entrante y permite el uso de sus dispositivos periféricos, se envía el contenido multimedia para que sea desplegado en la interfaz correctamente y se visualice usando contenedores HTML.

7. Usuario Local

- **7.1: Solicita permiso de periféricos**

Cuando un usuario local inicia una llamada, es invocada la función encargada de hacer la gestión de los dispositivos periféricos, en ella se le pregunta al usuario si da acceso a estos. Si el usuario no permite el uso de dispositivos periféricos entonces la llamada se cancela o se despliega solamente con audio o video dependiendo si comparte su cámara web o su micrófono.

- **7.2: Captura flujo multimedia de periféricos**

Si un usuario acepta dar permiso a sus dispositivos periféricos, inicia a la captura de video y/o audio.

8. Envía flujo multimedia de comunicación

Una vez el usuario remoto ha aceptado la comunicación y ambos usuarios otorguen permisos al sistema sobre sus dispositivos, el contenido multimedia como audio y video es transmitido directamente entre ellos y se agrega en la interfaz web desarrollada para el servicio de videollamada.

9. Despliega contenido multimedia.

Tanto el usuario local como el usuario remoto visualizan el contenido multimedia de audio y video, transmitido en el paso anterior. En este punto es posible decir que la llamada ha sido establecida satisfactoriamente.

Descripción de entorno de pruebas y escenario.

En esta sección se consiga información complementaria del entorno de pruebas y escenarios sobre los cuales se dispuso el sistema solución, y finalmente los resultados obtenidos durante el proceso de evaluación.

Descripción del entorno

El mecanismo implementado en el marco de esta investigación es puesto a prueba para determinar su comportamiento en situaciones adversas, es por ello que los módulos (Conmutador, Funciones WebRTC y Controlador) fueron evaluados en una prueba de estrés. Las pruebas fueron realizadas mediante funciones JavaScript para los módulos Conmutador y Funciones WebRTC, en el Anexo D se muestra las funciones básicas utilizadas para el establecimiento de una comunicación y se especifica el modo de

obtención de los datos de medición de cada una de ellas. Para la evaluación del módulo Controlador se realizó un proceso similar con utilidades incluidas en el lenguaje JAVA.

Las pruebas fueron llevadas a cabo en el “Laboratorio de Televisión Digital” de La Universidad del Cauca utilizando un escenario de pruebas compuesto por tres equipos como se puede apreciar en la siguiente figura (ver figura 4).

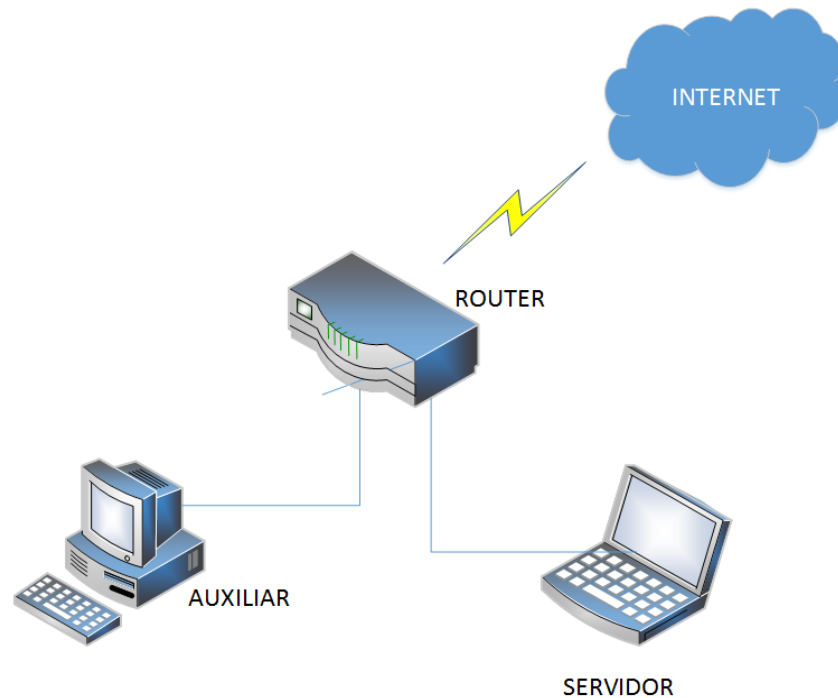


Figura 4: Arquitectura Pruebas de desempeño

El dispositivo “ROUTER” fue configurado para conectar los dispositivos en una red aislada del resto del laboratorio, buscando evitar posibles interferencias externas durante las pruebas, sin embargo, fue necesario conectar el “ROUTER” a internet ya que el módulo Funciones WebRTC requiere de servidores de señalización y librerías externas; las características de este son:

- Router Tenda W316R
- Wireless-N 150 Mbps
- Cable UTP Cat 5e 100Mbps

El equipo “Servidor” fue destinado para el almacenamiento y ejecución del sistema desarrollado, albergando también los tres módulos del mecanismo puesto a prueba, las características del equipo portátil son:

- Sistema operativo Ubuntu 14.04 LTS x64
- Procesador Intel Core i5 2.6 GHz

- Memoria RAM 6GB
- Cámara Web

Por último, el equipo “Auxiliar” utilizado como “tester²” del mecanismo, encargado de ejecutar las peticiones y almacenar los datos resultantes de las pruebas cuenta con las siguientes características:

- Sistema operativo Ubuntu 14.04 LTS x64
- Procesador Inter Core i7 3.4 GHz
- Memoria RAM 4GB

Descripción del escenario

El escenario utilizado para la ejecución de las pruebas de funcionalidad y desempeño fueron los siguientes:

El equipo ‘Servidor’ contaba con un Sistema operativo *Ubuntu 14.04 LTS x64*, en él fue necesario la instalación del servidor de aplicaciones Glassfish 4.1 para la ejecución del código fuente y la construcción de las funcionalidades del sistema solución implementadas en lenguaje Java; al ser una aplicación web, con un navegador web es posible consumir los servicios, en este escenario de pruebas fue usado Google Chrome como navegador principal, aunque cabe notar que otros navegadores como Firefox Mozilla pudieron ser usado sin ningún inconveniente.

Para las pruebas de funcionalidad específicamente fueron puestos a prueba cada requerimiento definido por separado en donde se aplicó la técnica de la caja negra para evaluar los resultados obtenidos por el sistema construido y compararlos con los resultados esperados

Pruebas de funcionalidad

El primer escenario de pruebas verifica el cumplimiento de los requisitos funcionales planteados para el sistema solución, estos requisitos se pueden encontrar en las secciones 2.5.1, 2.6.1 y 2.6.2; para tal fin fue adoptada la técnica de evaluación funcional denominada “caja negra” [81].

Organización de la información

Antes de usar plenamente la técnica de caja negra, los parámetros de entrada usados en la prueba deben ser organizados, en la tabla 1 se aprecia el listado de requisitos obtenidos del capítulo 2, a cada requisito se le hace corresponder el caso de uso que involucra, es decir, se identifica que acción dentro del sistema cumple con el requisito funcional, finalmente estos casos de uso son agrupados según el componente que impactan del mecanismo propuesto, esto con el fin de garantizar que solo se pongan a prueba los

² *Tester*: Dispositivo externo al sistema, utilizado para ejecutar el plan de pruebas sin invadir los equipos sobre los cuales se despliega la solución desarrollada.

requisitos funcionales que afectan los componentes directamente involucrados con el mecanismo propuesto.

Tabla 1 Vinculación Casos de Uso y Requerimientos

Componente	Caso de Uso	Requisitos	N° Prueba
Características del mecanismo propuesto	Solicitar Periféricos	El mecanismo ejecuta la solicitud de los periféricos disponibles en el dispositivo del usuario final	01
	Llamar Usuario Remoto	El mecanismo establece canales de comunicación punto a punto (conforme a lo establecido por la API WebRTC) con otros usuarios conectados y su posterior cierre al terminar el proceso de comunicación	02
		Las funciones provistas por el mecanismo deben ser ejecutadas dentro de páginas HTML que sirvan como interfaz para el cliente, con el fin de permitir una interacción entre servicios a nivel de presentación	03
	Enviar Notificaciones	Establece mensajes propietarios para identificar el estado de la comunicación, los cuales posibilitan el control de la misma	04
	Enviar Notificaciones	El mecanismo conmuta mensajes de control entre los usuarios conectados al servicio	05
	Registrar	Adiciona al perfil de usuario la clave "room" que representa el identificador único para los servicios de comunicación WebRTC	06
	Registrar	Adiciona al perfil de usuario la clave "state" la cual es el estado de conexión o desconexión al servicio.	07
	Requisitos para la implementación de un servicio de Videollamada en un entorno IPTV	Registrar	El acceso al servicio de Videollamada será concedido únicamente a usuarios registrados, estos deberán autenticarse antes de poder acceder al servicio
Acceder Guía Contenido		Durante el consumo de servicios de IPTV como VoD, el sistema le permite al usuario cambiar su estado	09

		entre “activo” e “inactivo” en el servicio de Videollamada	
	Llamar Usuario Remoto	La aplicación permite al usuario iniciar, finalizar y responder Videollamadas	10 equivalente a prueba 2
	Ver Conectados	Para el inicio de una Videollamada el usuario puede visualizar que usuarios están disponibles. Solo es posible establecer una comunicación si ambos usuarios están activos	11
	Ver Conectados	En una página previa al establecimiento de la llamada, el usuario del servicio puede visualizar la captura de video de la cámara local	12

Posterior a la organización de los requisitos y casos de uso, se procede a describir el paso a paso que debe ejecutarse en el sistema para poner en marcha el caso de uso, además se debe describir el resultado que se espera obtener con este paso a paso; la tabla 2 presenta un formato para organizar la información anterior, dándole un identificador único a cada requisito funcional, además de un espacio para la descripción paso a paso sobre el sistema y finalmente un lugar para el resultado esperado.

Tabla 2 Formato para informativo pruebas de desempeño

Prueba N° XX	
Título	Requisito que será puesto a prueba.
Pasos	Se expone el paso a paso que debe ejecutarse sobre el sistema para cumplir con el requisito a evaluar.
Resultado esperado	Espacio para narrar el resultado que se espera obtener con la ejecución del paso a paso anteriormente descrito

En el documento principal de la monografía podemos encontrar los 12 requisitos funcionales descritos según el formato de la tabla anterior (ver sección 4.2.2 de la monografía).

Uso de la técnica de caja negra

La técnica de caja negra compara el resultado que se espera obtener al realizar una acción sobre el sistema y el que realmente se obtiene, si ambos resultados son idénticos se deduce que el requerimiento funcional se cumple.

En la siguiente figura (ver figura 5) se aprecia de manera gráfica el proceso lógico que se lleva a cabo con la técnica de caja negra, las acciones que se realizan sobre el sistema se

consideradas como entradas de este, las salidas del sistema serán los resultados que arroje el sistema y puedan ser percibidos por el usuario.

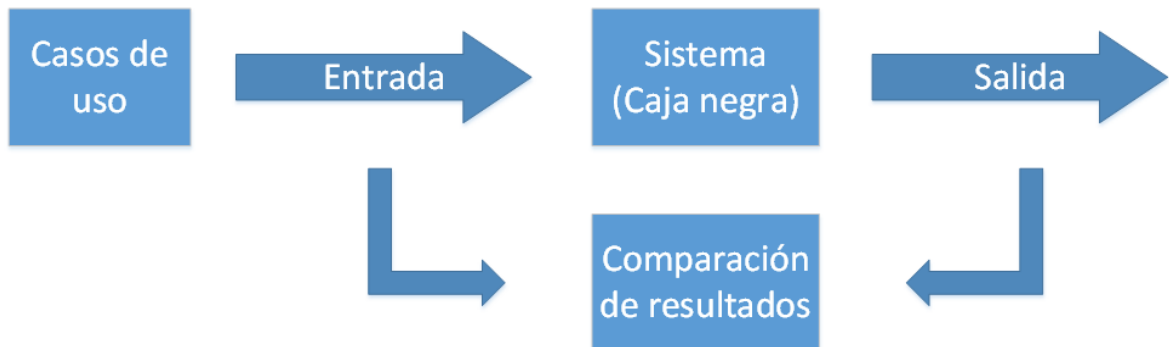


Figura 5: Diagrama lógico técnico de caja negra

Resultados Obtenidos

En el presente trabajo de grado se realizaron un total de 12 pruebas funcionales con la técnica de caja negra, el 100% de las pruebas arrojaron resultados positivos concluyendo de esta manera que el mecanismo propuesto cumple en su totalidad con los requisitos funcionales planteados en el capítulo 2 de la monografía. Para ver el proceso individual de cada prueba funcional el lector puede remitirse a la sección 4.2.3 del documento monografía.

Pruebas de desempeño

En el segundo escenario de pruebas el mecanismo propuesto es evaluado con el fin de determinar su comportamiento en situaciones adversas, por ello los módulos (Conmutador, Funciones WebRTC y Controlador) fueron evaluados con una prueba de desempeño.

Una prueba de desempeño busca comprobar que el sistema cumple con los requisitos de operación acordados, es decir, que tiene tiempos de respuesta apropiados para su correcto funcionamiento, en nuestro caso específico un tiempo total de respuesta menor a 400 milisegundos según lo definido en la sección 2.4.3.2 de la monografía.

Las pruebas de desempeño fueron realizadas mediante funciones JavaScript para los módulos Conmutador y Funciones WebRTC, en el **Anexo D** se muestra las funciones básicas utilizadas para el establecimiento de una comunicación y se especifica el modo de obtención de los datos de medición de cada una de ellas. Para la evaluación del módulo Controlador se realizó un proceso similar con utilidades incluidas en el lenguaje JAVA.

En las siguientes tablas (ver tablas 3, 4, 5 y 6) son consignadas algunas muestras de los resultados obtenidos durante las pruebas de desempeño, realizadas sobre el modulo Conmutador. Si el lector desea conocer la totalidad de los datos obtenidos debe dirigirse al documento "resultados.xlsx" incluido en formato digital junto a los anexos.

Tabla 3 Resultado de la evaluación Modulo Conmutador

Cantidad de peticiones	Tiempo de respuesta (ms)
1	75,42
10	65,635
20	371,43
30	82,29
40	78,195
50	77,74
60	90,49
70	89,695
80	99,28
90	71,89
100	102,255

Tabla 4 Resultado de la evaluación funciones WebRTC

Cantidad de peticiones	Tiempo de respuesta
1	17,615
10	2,77
20	3
30	2,655
40	2,8
50	3,74
60	2,82
70	2,78
80	2,87
90	2,63
100	2,77

Tabla 5 Resultado de la evaluación módulo Controlador (Inicio de llamada)

Cantidad de peticiones	Tiempo de respuesta
1	13
10	12
20	10

30	2
40	7
50	19
60	8
70	4
80	42
90	6
95	13

Tabla 6 Resultado de la evaluación módulo Controlador (Finalización de llamada)

Cantidad de peticiones	Tiempo de respuesta
1	3
10	9
20	3
30	6
40	2
50	5
60	45
70	4
80	5
90	8
95	6

A partir de los datos obtenidos anteriormente, se procede a realizar un mapeo de estos datos a gráficas, para tal fin fue utilizada la herramienta libre SCiDAVis [82].

Módulo Conmutador

El módulo conmutador, encargado de reenviar información entre usuarios, es puesto a prueba realizando hasta 100 peticiones secuenciales desde el cliente y cronometrando el tiempo que tarda en ir y volver una petición, la petición enviada desde el cliente hacia el conmutador lleva un mensaje comúnmente utilizado en el establecimiento de la comunicación y va dirigido al mismo emisor, es decir, el mensaje enviado sale del cliente A, pasa por el módulo conmutador y retorna al cliente A, de esta manera se cronometra el tiempo que tarda el mensaje en ir y volver al cliente A.

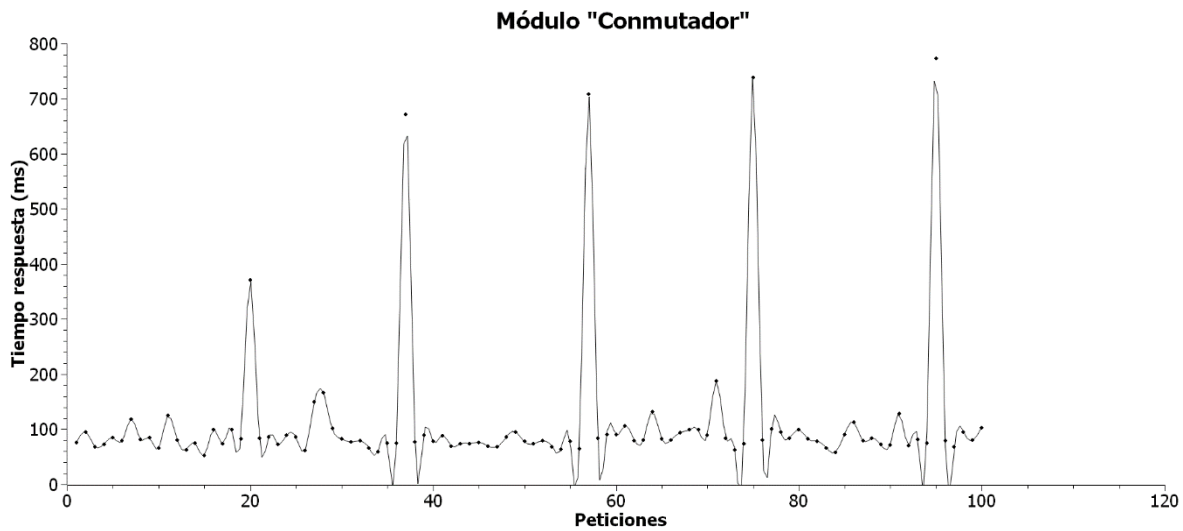


Figura 6: Peticiones Bloque "Conmutador"

Los resultados obtenidos durante la prueba son consignados en la *figura 6*, en donde se observa un tiempo de respuesta que oscila entre los 50 y 150 milisegundos. Durante una hipotética puesta en producción del mecanismo, será necesario sumar a este tiempo el retardo que genere la red al enviar la petición a un cliente remoto.

Módulo Funciones WebRTC

La agrupación de las Funciones WebRTC fue evaluada por medio de peticiones secuenciales de los métodos utilizados para el establecimiento de una comunicación, para realizar este ejercicio fue necesario construir una nueva página web dedicada a la ejecución de las funciones, también fue necesario establecer como llamante y llamado al mismo usuario, eliminando así la interacción humana durante la prueba.

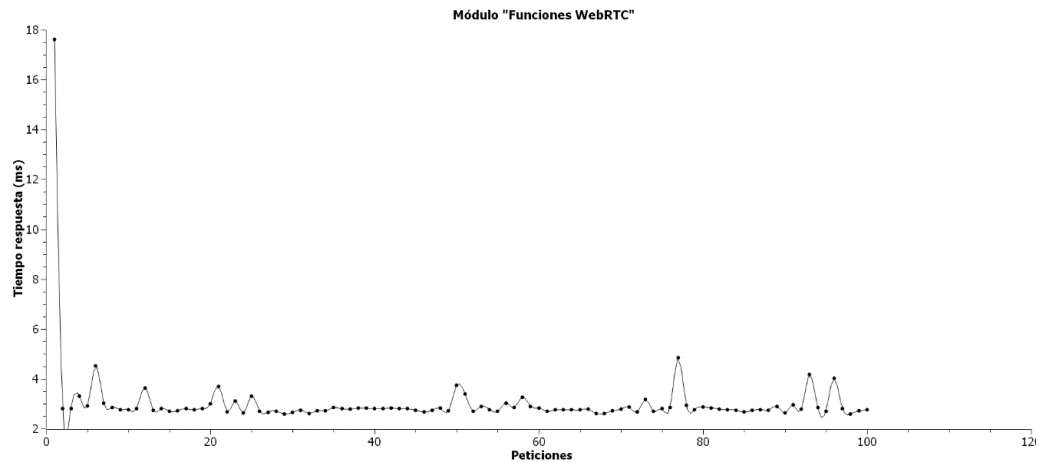


Figura 7: Peticiones Bloque "Funciones WebRTC"

La primera petición realizada tardó aproximadamente 18 milisegundos, ya que debe descargar la librería construida por "WebRTC Project", después de tener este archivo localmente los tiempos de respuesta de las funciones WebRTC oscilan entre los 2 y 4 milisegundos, como se puede apreciar en la figura 7.

El tiempo de procesamiento bajo obtenido con las funciones WebRTC permite inferir que este bloque no será un limitante para la prestación de los servicios de comunicación en tiempo real que dispongan del mecanismo propuesto.

Módulo Controlador

El módulo controlador agrupa las funciones lógicas para el inicio y terminación de los servicios implementados con base en el mecanismo desarrollado. Debido a que estos dos

procesos se realizan por separado, fue necesario probar cada uno independientemente, dando como resultado las figuras 8 y 9.

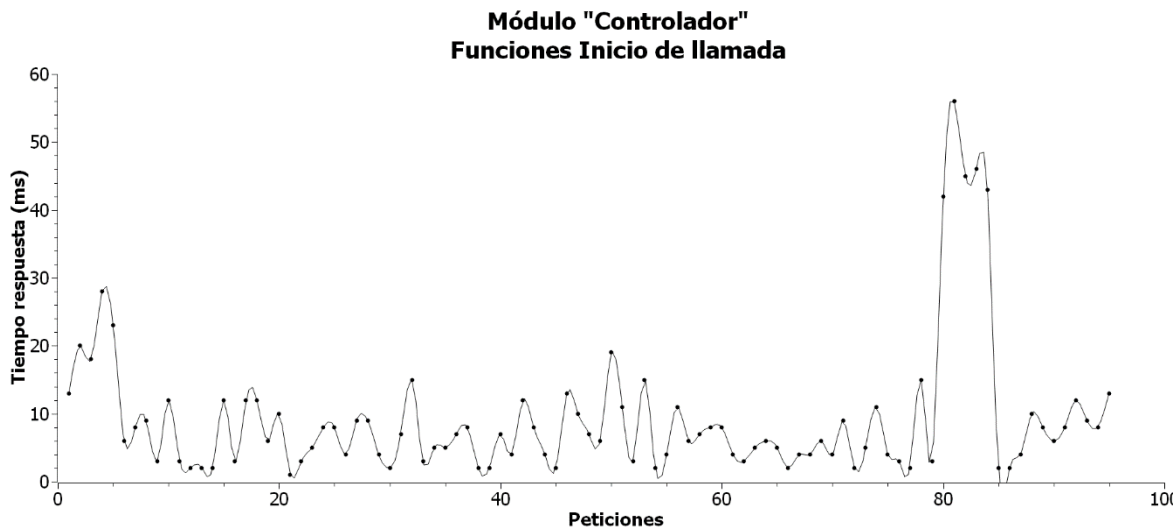


Figura 3: Peticiones Bloque "Controlador", Inicio de llamada

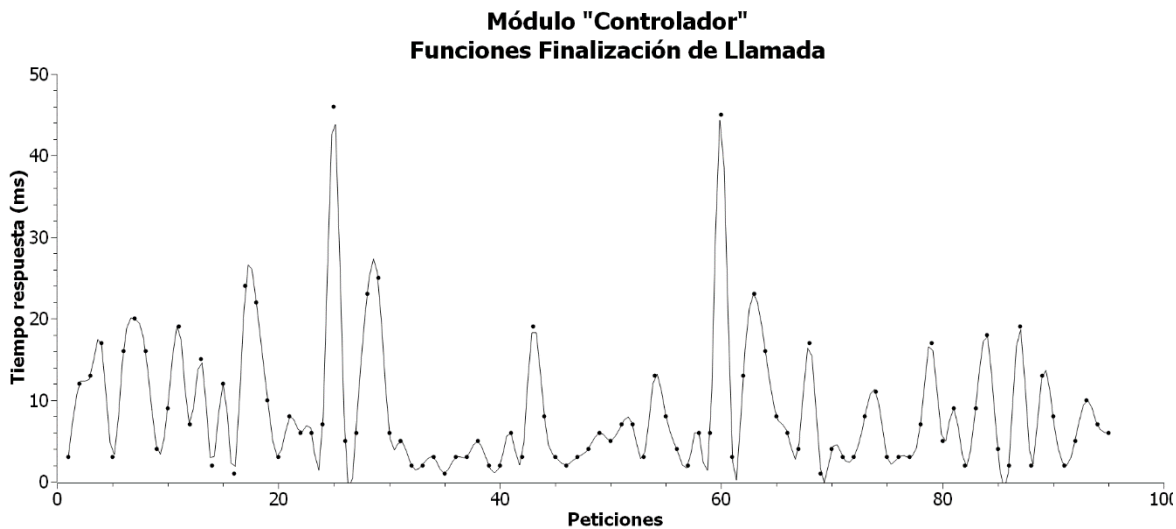


Figura 4: Peticiones Bloque "Controlador", Finalización de llamada

Los resultados obtenidos con la prueba del módulo controlador tienen una mayor variación que las pruebas anteriores, entre los 2 y los 60 milisegundos con una mayor concentración de datos alrededor de los 10 milisegundos.

Durante el funcionamiento normal en el sistema desarrollado, cada módulo es ejecutado secuencialmente, iniciando el Controlador y finalizando con el Conmutador, esto para el

inicio de una llamada; por lo que los tiempos anteriormente cronometrados deben sumarse dando un tiempo promedio de 91,27 milisegundos para la ejecución del sistema desarrollado, cabe resaltar que este tiempo está dentro de los parámetros considerados como aceptables para la comunicación en tiempo real establecidos por la ITU-T y definidos en la sección **¡Error! No se encuentra el origen de la referencia..**