



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

WEBLAB SCORM DE CLIENTES SIP

Alumno: Fernando Puentes Marchal

Tutor: Prof. D. Ildefonso Ruano Ruano
Depto.: Ingeniería de Telecomunicación

Junio, 2016



UNIVERSIDAD DE JAÉN

ESCUELA POLITÉCNICA SUPERIOR DE LINARES

Departamento de Ingeniería de Telecomunicación

CURSO ACADÉMICO

2015-2016

TRABAJO FIN DE GRADO

TÍTULO DEL TRABAJO

WEBLAB SCORM DE CLIENTES SIP

Grado en Ingeniería Telemática

AUTOR: Fernando Puentes Marchal

TUTOR: D. Ildefonso Ruano Ruano

Linares, septiembre de 2016

Fernando

**RUANO RUANO
ILDEFONSO -
02889873S**

Firmado digitalmente por RUANO
RUANO ILDEFONSO - 02889873S
Nombre de reconocimiento (DN):
c=ES, serialNumber=02889873S,
sn=RUANO RUANO,
givenName=ILDEFONSO, cn=RUANO
RUANO ILDEFONSO - 02889873S
Fecha: 2016.09.01 14:01:46 +02'00'

Agradecimientos

A mi tutor y profesor, D. Ildefonso Ruano Ruano, por ayudarme y motivarme durante todo el periodo de elaboración.

A mi novia, mis suegros y en especial a mis padres, por empujarme a seguir adelante.

Índice

1. RESUMEN	3
2. INTRODUCCIÓN	6
2.1. El protocolo SIP	8
2.1.1. Mensajes SIP	9
2.2. El protocolo SDP	13
2.2.1. Introducción.....	13
2.2.2. Parámetros de SDP	13
2.3. LMS	14
2.4. SCORM	14
2.4.1. Modelo de agregación de contenidos	15
2.4.2. Run-Time Environment	15
2.4.3. Secuenciación y navegación	16
3. OBJETIVOS.....	17
4. MATERIALES Y MÉTODOS	18
4.1. JSSIP	19
4.2. JQuery	19
4.3. RTCninja.....	19
4.4. Scorm RTE	20
4.5. El softphone SIP	21
4.5.1. Interfaz de usuario.....	22
4.6. El servidor SIP	29
4.7. Seguridad en SIP	30
4.7.1. Ataques.....	30
4.7.2. Servicios de seguridad	31
4.7.3. Mecanismos de seguridad.....	31
4.7.4. Acciones desarrolladas	31
5. RESULTADOS Y DISCUSIÓN.....	33

5.1. Como realizar comunicaciones	33
5.1.1. Llamada	36
5.1.2. Chat	37
5.2. Laboratorio remoto (WebLab)	38
5.2.1. Página 1. Introducción	38
5.2.2. Página 2. Teoría	40
5.2.3. Página 3. Servidor	45
5.2.4. Página 4. Cliente	46
5.2.5. Página 5. Evaluación final	54
6. CONCLUSIONES	61
7. ANEXOS	64
7.1. Instalación y configuración del servidor SIP	64
7.1.1. Instalación del servidor	64
7.1.2. Configuración del servidor	66
7.2. Pasos previos y requisitos para la ejecución del laboratorio remoto	73
7.2.1. Comprobación de micrófono y cámara	73
7.2.2. Navegador de ejecución	74
7.3. Manual de mantenimiento	75
7.4. Acrónimos	77
7.5. Índice de figuras	78
8. REFERENCIAS BIBLIOGRÁFICAS	81

1. RESUMEN

El presente Trabajo Fin de Grado (TFG) está dirigido a la creación de un Laboratorio Web (WebLab) para el estudio práctico del Protocolo de Inicio de Sesiones (SIP) (RFC3261, 2002). SIP es el protocolo más extendido para implementar comunicaciones de voz en redes de datos, más conocido como Voz sobre IP o VoIP (del inglés *Voice over IP*).

En el laboratorio creado se ha incrustado un cliente SIP en una página web accesible desde el Sistema de Gestión de Aprendizaje (*Learning Management System* en inglés o LMS) que se utiliza para establecer comunicaciones con otros clientes y servidores SIP. El cliente ha sido desarrollado como parte de este trabajo a partir de una versión libre existente en Internet e incluido en un módulo o paquete SCORM (*Sharable Content Object Reference Model*), que es el estándar de facto para contenidos de *e-learning*, para facilitar su reutilización, posibilidad de exportación y uso en diferentes LMS.

El cliente SIP elegido está realizado en lenguaje de programación JavaScript (*Julie C. Meloni, 2012*), para poder ser integrado en una página web y utiliza el protocolo WebSocket (WS) (<https://tools.ietf.org/html/rfc7118>) para enviar los mensajes SIP. A dicho cliente se le ha cambiado el aspecto y se le han añadido algunas funciones, como la posibilidad de mostrar los mensajes en la misma aplicación, mediante dos vistas (*vista completa y vista resumen*) o cambiar el idioma de la interfaz entre inglés o español.

Como ya se ha comentado, el WebLab se ha creado utilizando como soporte principal el conjunto de estándares más utilizado y extendido en docencia virtual que es SCORM. Se ha creado un paquete SCORM que contiene los contenidos teóricos y prácticos de SIP, se ha realizado tomando como referencia un modelo de 5 páginas que, una vez adaptado, permite proporcionar un contexto de trabajo integral en el que el alumno puede encontrar todos los recursos necesarios para completar con éxito las prácticas. Este paquete SCORM además va a evaluar a los alumnos sobre el protocolo SIP y almacenar datos de interés como las respuestas del alumno, tiempo que ha tardado en responder las preguntas y comentarios que el alumno quiera almacenar, entre otros.

La secuenciación de dichas páginas se muestra en *Imagen 1* y cada una de ellas se explica detalladamente en el apartado 5.2. *Laboratorio remoto (WebLab)*.

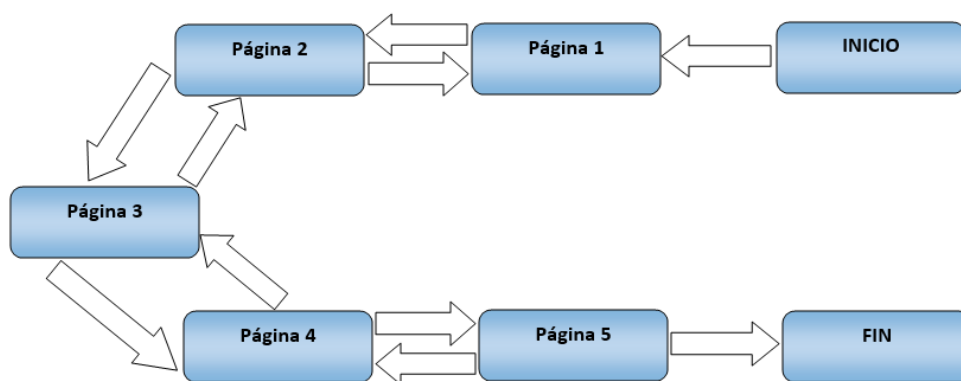


Imagen 1: Secuencia del SCORM

Fuente: (Ruano Ruano)

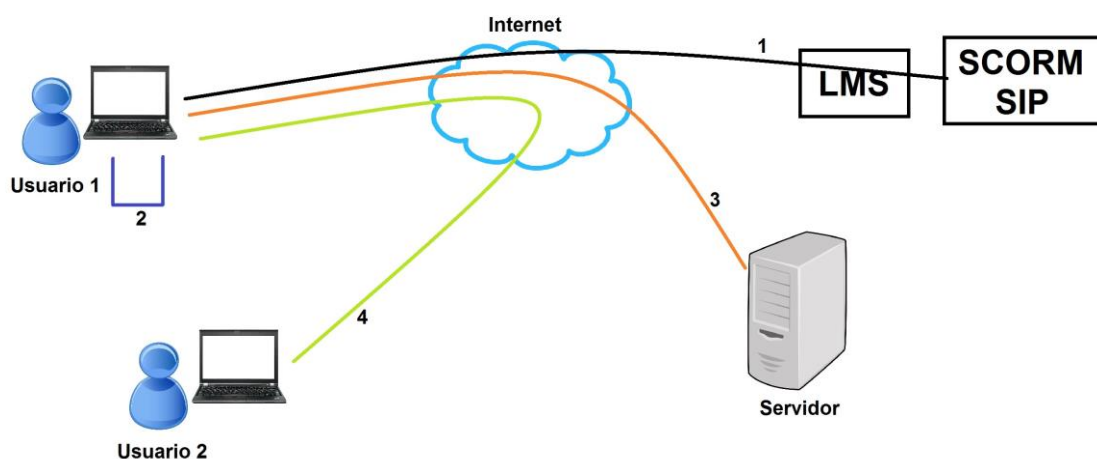


Imagen 2: Comunicaciones de las prácticas

Fuente: Propia

Imagen 2 muestra un esquema de las comunicaciones más importantes que tienen lugar en la práctica implementada en la página 4:

1. Los alumnos tras identificarse en el LMS acceden y descargan el paquete SCORM SIP que ejecutan en su navegador.
2. Tras instalar el servidor de forma local los alumnos deben registrarse en él utilizando el cliente SIP que se proporciona.
3. Esta comunicación se produce entre el cliente del alumno y el servidor instalado de forma remota.
4. Esta comunicación se produce entre dos alumnos que están registrados en el mismo servidor remoto.

En la realización del SCORM se han implementado funciones que permiten que se adapte al alumno y dar facilidades al alumno. Estas son las siguientes:

- Algunas preguntas son únicas para cada alumno, ya que la respuesta se basa en su id.
- El orden de las opciones de las preguntas de opción múltiple se muestra de forma aleatoria.
- El alumno conocerá en todo momento su nota en la página actual gracias a un cuadro que tendrá en su esquina superior derecha.
- El alumno sabrá si ha respondido bien a la pregunta en función del color que tome el botón de responder. Se explica detalladamente en el apartado 5.2. *Laboratorio remoto (WebLab)*.
- En cada pregunta se proporciona un apartado que muestra al alumno una ayuda para resolver la pregunta. Este apartado será accesible una vez haya respondido a la pregunta.
- En cada pregunta el alumno puede guardar un comentario propio.

El presente WebLab tiene una serie de características que lo hacen adecuado para la enseñanza del protocolo SIP. Algunas de estas características son:

- Se puede presentar en un Sistema de Gestión de Aprendizaje (LMS o *Learning Management System*) compatible con SCORM, la mayoría del mercado lo son, por ejemplo Moodle, el más utilizado en el mundo, o ILIAS, el utilizado en la Universidad de Jaén.
- Está basado en una estructura de contenidos SCORM.
- Incluye software en lenguaje JavaScript.
- Los alumnos que utilicen este WebLab pueden, entre otras cosas, realizar la instalación y configuración de un servidor SIP local, trabajar con el cliente SIP e intercambiar mensajes y llamadas en tiempo real con los compañeros.
- El WebLab interacciona con el LMS, lo que permite personalizar el contenido del mismo basándose en la identidad de los alumnos y seguir el trabajo realizado por ellos mediante una calificación que se almacena en el LMS.
- El cliente proporciona varias opciones visuales para poder ver los mensajes intercambiados, lo que facilitará que el alumno entienda mejor los mensajes del protocolo.

2. INTRODUCCIÓN

Actualmente, cada vez más recursos tecnológicos son utilizados en la enseñanza. A día de hoy se utilizan diversas tecnologías como ordenadores, software, Internet..., para mejorar las técnicas de educación y las capacidades de aprendizaje de los estudiantes.

Mediante un laboratorio remoto se pueden presentar imágenes, textos o aplicaciones en un navegador web con actividades para facilitar la enseñanza de la materia a la que está dirigida el WebLab.

Para poder poner a disposición de los estudiantes un laboratorio remoto es necesario que el mismo esté montado sobre un software adecuado y, a poder ser, conocido por los alumnos, como suele ser el LMS institucional de la organización educativa.

Al integrar el laboratorio remoto en el LMS de la Universidad de Jaén se permite el acceso al mismo a los usuarios autorizados (los alumnos que tienen acceso al WebLab en asignaturas que incluyan el protocolo SIP). Además facilita al profesor el seguimiento, control y evaluación del trabajo de los alumnos. Los alumnos tienen la posibilidad de utilizar un programa analizador de protocolos, como *wireshark* (<https://www.wireshark.org/>), para ver los mensajes, pero el hecho de tener que ver todos los mensajes de la red junto con estos hace que sea una tarea lenta y costosa.

Por ello en este WebLab se incorpora una aplicación que muestra solo los mensajes SIP, facilitando al alumno la búsqueda y estudio de los mismos.

Lo que se ha conseguido con este TFG es la obtención de un WebLab a través del que los alumnos aprenden a manejar un servidor proxy sencillo y un cliente SIP, intercambiando mensajes que podrán verse en tiempo real desde el propio cliente (solo mensajes SIP).

Para poder hacer unas prácticas completas de *VoIP SIP* y conocer si el cliente SIP del presente TFG está bien implementado es necesario utilizar un servidor SIP. Se han probado dos servidores SIP:

- *OfficeSIP* (<http://www.officesip.com/>): Este servidor tiene una fácil instalación y una difícil configuración. Además, tras un tiempo de estar instalado (en mi caso 2 días) este se bloqueaba y dejaba de funcionar, teniendo que desinstalarlo y volverlo a instalar. También se debe añadir a

esto que la mayoría de los clientes SIP estudiados no conectaban con éxito a él o se producía una pérdida de conexión entre cliente y servidor a los pocos segundos. Por estas razones se procedió a la búsqueda de otro servidor SIP.

- *PortSIP* (<http://www.portsip.com/>): Este servidor tiene una fácil instalación y cuenta con una interfaz web que permite configurar ciertos parámetros y añadir usuarios de forma sencilla. Al ver la compatibilidad entre este servidor y el cliente SIP escogido y la facilidad de configuración para el usuario se ha elegido este servidor para utilizar el cliente. Además, puede ser instalado tanto en un equipo local como en uno remoto.

Se han realizado pruebas con diferentes clientes SIP *open source* en busca de uno que sea funcional. Se han probado los siguientes:

- *SIPML5* (<https://www.doubango.org/sipml5/>): Este cliente no conectaba con ningún servidor y, por lo tanto, fue descartado.
- *JSCommunicator* (<https://github.com/opentelecoms-org/jscommunicator>): Este cliente no tuvo compatibilidad con ninguno de los servidores, por lo que fue descartado.
- *sip-js* (<https://github.com/theintencity/sip-js>): Este cliente es relativamente reciente (2011-2012). Se realizaron pruebas con él y estas fueron exitosas con el servidor *OfficeSIP*, pero la compatibilidad no era total (se desconectaba solo a los pocos segundos) y la interfaz tenía demasiados parámetros y se consideró buscar otra con mayor compatibilidad y más simple para el usuario.
- *JSSIP tryit* (<https://tryit.jssip.net/>): Este cliente tuvo compatibilidad total con el servidor *PortSIP*. Tiene pocos parámetros de configuración y una interfaz mucho mejor a los clientes mencionados anteriormente. Por estas razones se decidió trabajar sobre este cliente.

Anteriormente se realizó un proyecto relacionado con este llamado *WEBLAB SIP-JAVA BASADO EN SCORM*, realizado por *Ismael Malpica Herrera*, cuyo objetivo era la implementación de un cliente y un servidor SIP en Java e integrarlo en un paquete SCORM. Sin embargo, en la actualidad, muchas de las empresas responsables de los navegadores web están eliminando el soporte de plugins basados en Applets, por lo que están quedando cada vez más obsoletas. [107]

Por esta razón, en este caso se ha propuesto la realización de un cliente SIP en JavaScript que se pueda integrar en una web, ya que JavaScript es una tecnología que actualmente está dando grandes resultados en el desarrollo web.

2.1. El protocolo SIP

SIP es un protocolo de señalización del nivel de aplicación definido por la *IETF* (*Internet Engineering Task Force*) en la *RFC 3261* (2002) para la creación, modificación y terminación de sesiones con uno o más participantes. El término sesión podría ser descrito como un intercambio de datos entre varios participantes. Dichos datos son transportados mediante otros protocolos como *RTP/RTCP*, como se puede ver en *Imagen 3*.

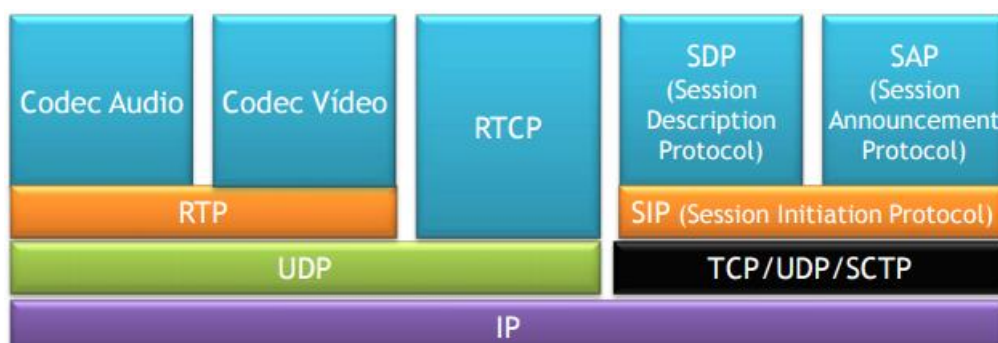


Imagen 3: Torre de Protocolos

Fuente: (Ruano Ruano)

El protocolo SIP se basa en un modelo cliente-servidor. El cliente envía solicitudes (*request*) al servidor y el servidor da respuestas (*responses*) a dichas solicitudes.

En SIP, los usuarios son identificados por un *Identificador de Recursos Uniforme* (URI). Un posible ejemplo de URI SIP sería <sip:anonimo@red.ujaen.es:5060>, donde *anónimo* se correspondería con el nombre de usuario, *red.ujaen.es* se correspondería con el dominio donde se encuentra el servidor y *5060* se correspondería con el puerto.

En SIP se pueden encontrar dos tipos de equipos:

- **Terminales de usuario:** Equipos finales que usan los usuarios. Pueden ser clientes o servidores. Tiene dos componentes fundamentales para funcionar como Agente de Usuario (UA):
 - o **Agente de usuario Cliente (UAC):** Aplicación que permite al terminal enviar solicitudes SIP y aceptar respuestas SIP provenientes de UAS.

- **Agente de usuario Servidor (UAS):** Aplicación que permite al terminal recibir solicitudes SIP y devolver respuestas SIP en nombre del usuario (Aceptación, Rechazo o Redirección).
- **Servidores SIP:** Equipos intermedios que añaden funcionalidad a las redes SIP. Existen tres tipos:
 - **Servidor de Registro (Registrar Server):** Acepta peticiones de registro de clientes (UAC) y mantiene registro de direcciones de cada usuario.
 - **Servidor Proxy (Proxy Server):** Recibe la solicitud del cliente (UAC), la analiza y decide el servidor (UAS) al que se la envía.
 - **Servidor de Redireccionamiento (Redirect Server):** Recibe la solicitud del cliente (UAC) y responde con la dirección/es donde la debe enviar.

2.1.1. Mensajes SIP

Hay dos tipos: Solicitudes y Respuestas. En *Imagen 4* se muestra el formato.

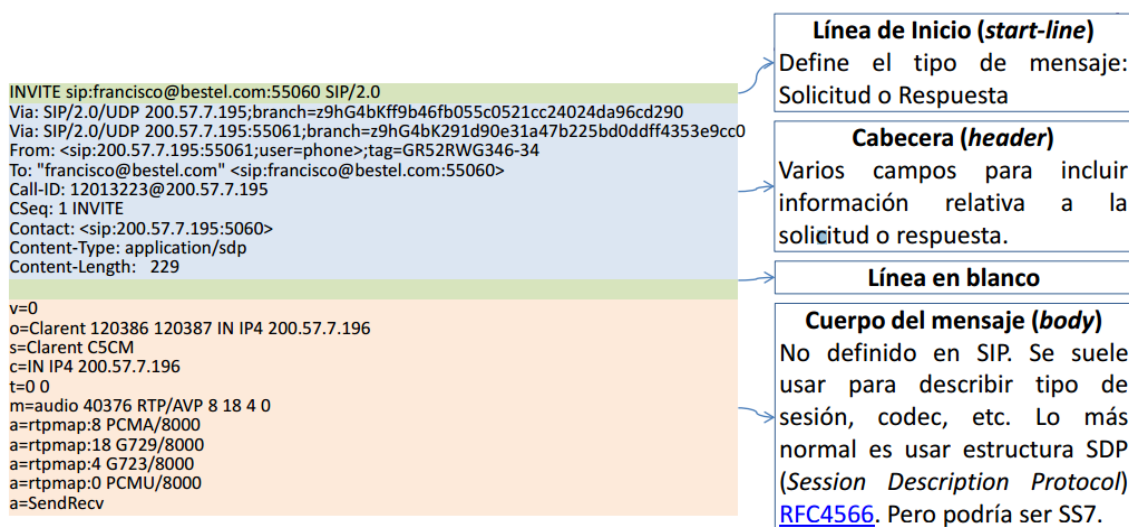


Imagen 4: Formato de un mensaje SIP

Fuente: (Ruano Ruano)

A continuación se explican los diferentes bloques del mensaje SIP:

- Línea de inicio (*start-line*): Puede ser de dos tipos:
 - Línea de solicitud (*Request-Line*): El formato se muestra en *Imagen 5*.

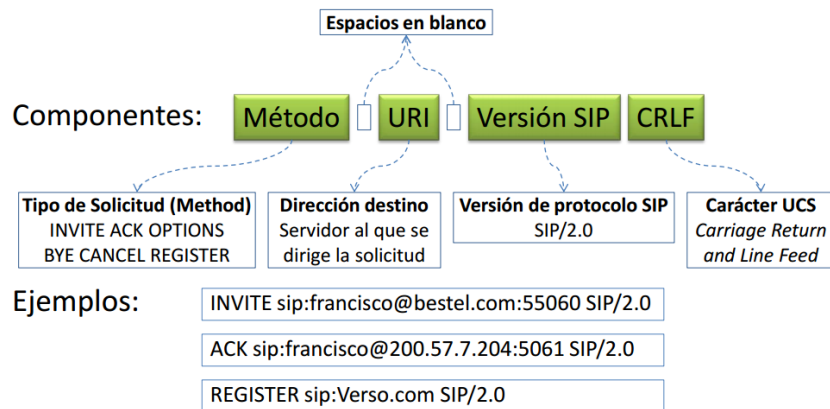


Imagen 5: Formato de la línea de inicio en mensajes de solicitud

Fuente: (Ruano Ruano)

Los métodos que pertenecen a la línea de solicitud son los siguientes:

- **INVITE:** Para invitar a participar en una sesión (llamada). En el cuerpo del mensaje se incluyen los parámetros necesarios.
- **ACK:** El llamante confirma al servidor que ha respondido a la llamada.
- **OPTIONS:** Usado para que un UA solicite información a un servidor sobre sus capacidades. Se suele utilizar antes de iniciar la llamada.
- **BYE:** Indica el fin de la sesión.
- **CANCEL:** Se utiliza para cancelar una solicitud en caso de no recibir respuesta de la misma.
- **REGISTER:** Se envía cuando un usuario quiere registrarse en un servidor SIP.

Recientemente se han introducido nuevos métodos, mostrados en Imagen 6.

Método	Descripción	Definido en
PRACK	Confirmación Provisional	RFC3262
SUBSCRIBE	Se suscribe para un evento de notificación del notificador.	RFC6665
NOTIFY	Notifica un nuevo evento a un subscritor	RFC6665
PUBLISH	Publica un evento al Servidor	RFC3903
INFO	Envía información durante una llamada (sesión) que no modifica su estado (p.e. transmitir coste de llamada en curso, tonos MF que se marquen, etc.)	RFC6086
REFER	Pide al destinatario que realice una petición SIP (p.e. Transferencia de llamada) y se le notifique el resultado de la misma.	RFC3515
MESSAGE	Transporta mensajes instantáneos mediante SIP.	RFC3428
UPDATE	Modifica el estado de una sesión (parámetros, p.e. codecs) sin cambiar el estado de la ventana de diálogo.	RFC3311

Imagen 6: Métodos SIP nuevos

Fuente: (Ruano Ruano)

- Línea de respuesta (*Reply-Line*): El formato se muestra en *Imagen 7*.

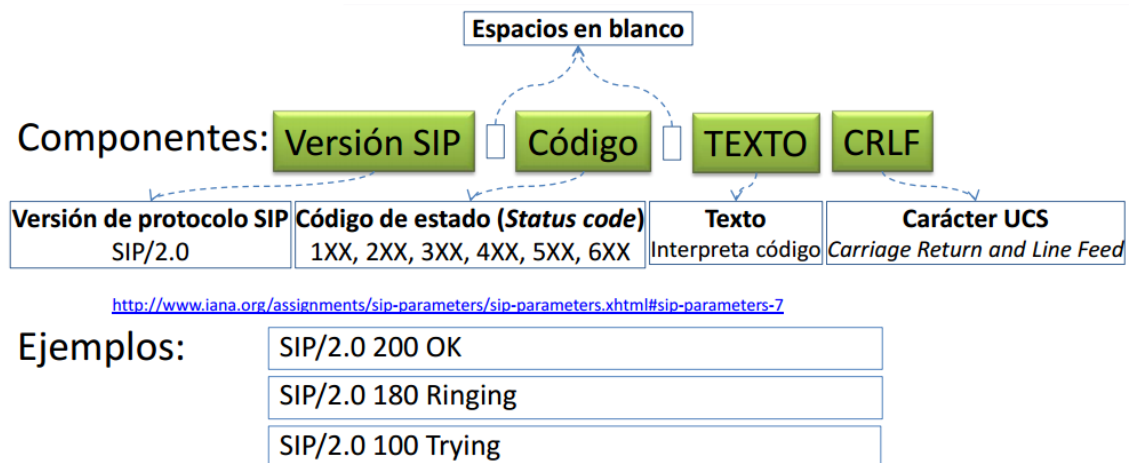


Imagen 7: Formato de la línea de inicio en mensajes de respuesta

Fuente: (Ruano Ruano)

Los diferentes códigos de estado son los mostrados en *Imagen 8*.

1xx = de información: Indican estado temporal de la comunicación 100 En progreso (intentando) 180 Timbre sonando 181 Llamada en desvío 182 Encolada 183 Sesión en progreso 199 Diálogo Temprano Terminado	408 Tiempo agotado para solicitud 410 Ido: El usuario existió, se ha ido y no está 412 Error de Petición Condicional 413 Solicitud demasiado larga 414 Solicitud URI demasiado larga 415 Tipo de medio no soportado 416 Esquema URI no soportado 417 Prioridad recurso desconocida 420 Extensión incorrecta: Servidor no soporta extensión SIP 421 Se requiere Extensión 422 Intervalo de sesión demasiado pequeño 423 Intervalo demasiado corto 424 Error información de Localizac. 428 Usar cabecera Identity 429 Proporcionar identidad de remitente 430 Error de Flujo 433 Anonimato no permitido 436 Error en cabecera Identity-Info 437 Certificado no soportado 438 Cabecera Identity No válido 439 Primer salto carece de soporte de salida 440 Superada Anchura Máxima 469 Paquete con error de Info 470 Necesario Consentimiento 480 Temporalmente no disponible 481 Llamada/Transacción no existe	482 Bucle detectado 483 Demasiados saltos 484 Dirección incompleta 485 Ambiguo 486 Ocupado 487 Solicitud terminada 488 No aceptable aquí 489 Evento Erróneo 491 Solicitud pendiente 493 Indescifrable: No pudo descifrar cuerpo S/MIME 494 Necesario Acuerdo de Seguridad 5xx = Errores de servidor 500 Error interno del servidor 501 No Implementado: La solicitud/método SIP no está implementado 502 Gateway incorrecto 503 Servicio no disponible 504 Tiempo agotado en servidor 505 Versión no soportada: Servidor no soporta versión SIP 513 Mensaje demasiado largo 580 Error de Precondición 6xx = Fallo general 600 Ocupado en todas partes 603 Rechazado 604 No existe en ninguna parte 606 No Aceptable
--	---	---

Imagen 8: Códigos de estado en mensajes de respuesta

Fuente: (Ruano Ruano)

- Cabecera (*Header*):

Formada por campos, de forma similar a HTTP.

Los campos incluyen información sobre el mensaje.

Tipos de campos:

- *Salto a salto (hop-by-hop)*: Pueden ser modificados o añadidos por los servidores por los que vaya pasando. Deben aparecer los primeros.
- *Extremo a extremo (end-to-end)*: No son interpretados por los servidores, solo por los clientes. Deben aparecer los últimos.

Los campos que más se utilizan son los siguientes:

- *Via*: Indica el transporte utilizado para el envío e identifica la ruta de la petición.
- *From*: Indica la dirección del origen de la petición.
- *To*: Indica la dirección del destino de la petición.
- *Call-ID*: Es el identificador único de cada llamada. Debe ser igual en todos los mensajes de una misma transacción.
- *Cseq*: Se inicia con un número e identifica de forma secuencial cada petición. La respuesta a una solicitud debe tener el mismo Cseq.
- *Contact*: Contiene una (o más) dirección que se puede utilizar para contactar con el usuario.
- *UserAgent*: Contiene el agente de usuario que realiza la comunicación.
- *Content-Length*: Longitud en octetos del cuerpo del mensaje.
- *Content-Type*: Indica el tipo de contenido del cuerpo del mensaje.

- **Cuerpo del mensaje (*Content*):**

Parte no obligatoria del mensaje SIP.

Puede contener:

- Información dirigida a una aplicación de usuario.
- Información de señalización.

Los equipos intermedios no lo analizan. Solo es interpretado por los equipos origen y destino.

Se suele utilizar el formato SDP en mensajes de señalización.

2.2. El protocolo SDP

2.2.1. Introducción

Definido en la RFC 4566.

No fue diseñado para SIP, se puede usar en otros entornos.

SDP es un protocolo diseñado para describir las características de una sesión multimedia.

2.2.2. Parámetros de SDP

El orden de los campos es siempre el mismo y es el mostrado en *Imagen 9*:

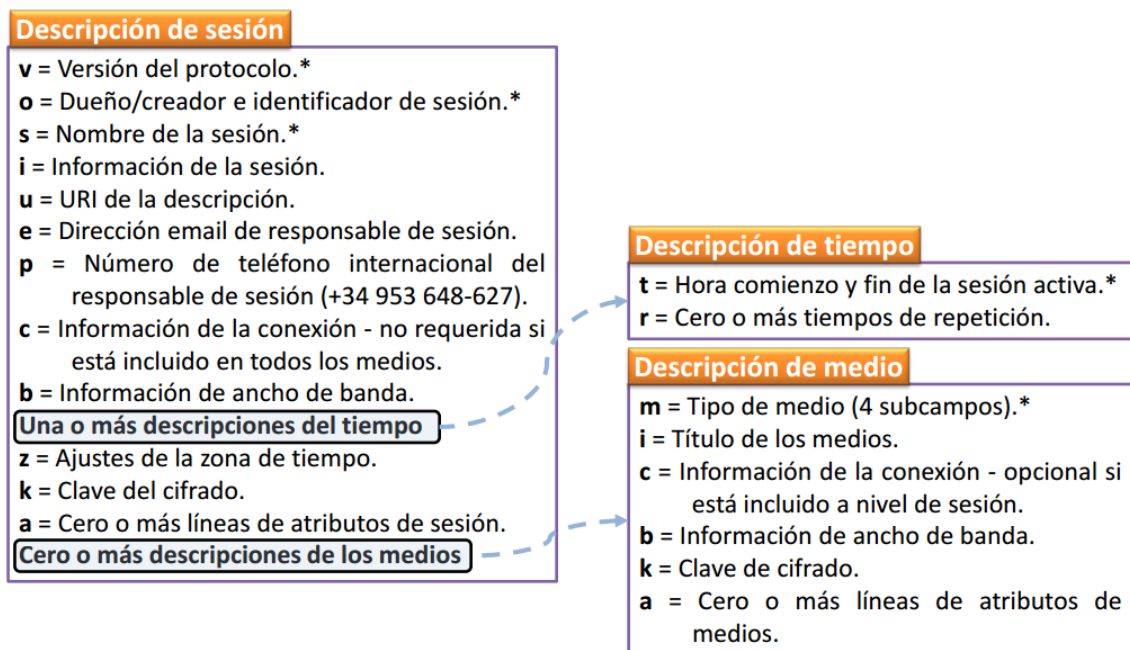


Imagen 9: Parámetros SDP

Fuente: (Ruano Ruano)

Hay campos obligatorios (v, o, s, t, m) y opcionales (todos los demás).

2.3. LMS

Un *LMS*, o sistema de gestión de aprendizaje, es un software instalado en un servidor web que se utiliza para gestionar, distribuir y controlar las actividades de formación no presencial de una institución u organización.

Las funciones del LMS son:

- Gestión de usuarios, recursos y actividades de formación.
- Administración del acceso.
- Controlar y supervisar el proceso de aprendizaje.
- Realizar evaluaciones.
- Generar informes.
- Gestionar servicios de comunicación como foros de discusión y videoconferencias, entre otros.

Actualmente la Universidad de Jaén cuenta con dos versiones de LMS activas, una para formación reglada y otra de pruebas con una versión más moderna (*versión 2004*). En el presente TFG se ha utilizado la *versión 2004* debido a que es más moderna y ofrece más funciones que la otra versión. El SCORM se ha integrado en la versión de evaluación de ILIAS, que tiene esta versión de LMS.

2.4. SCORM

SCORM (*Sharable Content Object Reference Model*) es un conjunto de estándares y especificaciones que ofrece la posibilidad de crear objetos pedagógicos estructurados para facilitar la enseñanza de alguna materia, cumpliendo los siguientes requisitos:

- Adaptabilidad: Personalizar el contenido en función de las necesidades de las personas.
- Interoperabilidad: Capacidad de utilizarse en otros emplazamientos y utilizando otras herramientas u otra plataforma.
- Accesibilidad: Se debe poder acceder a los componentes de enseñanza mediante un navegador web.
- Durabilidad: Los datos deben resistir la evolución de la tecnología sin necesitar ser renovados.
- Reusabilidad: Se debe poder reutilizar la información dentro de múltiples contextos y aplicaciones.

Existen varias versiones de SCORM, para la realización de este Trabajo Fin de Grado se ha utilizado la última publicada, la versión SCORM 2004 que está estructurada en 3 especificaciones o “libros técnicos”:

- Modelo de agregación de contenidos (CAM, *Content Aggregation Model*) (ref [103]).
- Entorno de Tiempo de Ejecución (RTE, *Run-Time Environment*) (ref [104]).
- Secuenciación y navegación (SN, *Sequencing and Navigation*) (ref [105]).

2.4.1. Modelo de agregación de contenidos

El modelo de agregación de contenidos, o *Content Aggregation Model* (CAM), incluye la forma de empaquetar el contenido para poder ser importado en un LMS. Para esto se crean archivos en formato XML que un LMS puede leer y extraer la información contenida en él.

Se compone de tres secciones: *el Modelo de Contenido, los Metadatos y el Empaquetado de Contenido*.

La sección de *Modelo de Contenido* especifica el contenido que se va a entregar. Si el contenido contiene más de un módulo, este modelo describe la relación entre ellos en forma de agregación, que pueden ser anidadas en forma de árbol.

La sección de *Metadatos* ofrece un mecanismo para describir el contenido utilizando un vocabulario común y predefinido. Cualquier sección de un curso SCORM puede ser descrita con metadatos.

La sección de *Empaquetado de Contenido* define como empaquetar un conjunto de objetos de aprendizaje y como estos deben ser leídos. En resumen, consiste en crear un archivo *zip* que contiene todos los ficheros necesarios, entre los que está el fichero denominado “*manifest.xml*” que define los contenidos de cada fichero, así como la relación que existe entre ellos.

2.4.2. Run-Time Environment

Incluye cómo debe ejecutarse el contenido mostrado en un navegador web y la forma en la que se comunica con el LMS. Está basado en JavaScript. Para ello se define un modelo de datos en los que están restringidas todas las comunicaciones posibles.

2.4.3. Secuenciación y navegación

Describe el orden de la presentación de los contenidos dependiendo de la navegación realizada por el usuario. Con este objetivo se elaboran los “*árboles de actividades*” que definen las posibles ordenaciones en función de las acciones que realice el usuario.

3. OBJETIVOS

Los objetivos propuestos para el presente TFG son los siguientes:

- Realizar o adaptar un cliente SIP Web con JavaScript para el envío, recepción y presentación en tiempo real de mensajes SIP.
- Uso de SCORM como soporte de software online en LMS.
- Estudio de establecimiento de sesiones SIP.
- Desarrollo de un paquete SCORM como plataforma orientada a docencia para que los alumnos realicen prácticas con el protocolo SIP.

El laboratorio remoto deberá realizarse de forma que sus usuarios cumplan los siguientes objetivos:

- Instalar y configurar un servidor proxy.
- Utilizar un cliente SIP para conectarse a un servidor.
- Visualizar los mensajes que intervienen en la comunicación.
- Entender cómo se realiza una comunicación mediante el protocolo SIP.
- Reconocer los diferentes tipos de mensajes SIP.
- Reconocer los diferentes campos de un mensaje SIP e interpretar su significado.
- Reconocer los diferentes campos de SDP e interpretar su significado.
- Capacidad de trabajar en grupo.

4. MATERIALES Y MÉTODOS

Para la implementación del laboratorio remoto se ha utilizado un SCORM ya desarrollado y se ha modificado tanto el código *HTML* como el *JavaScript* para que se muestre el texto deseado en las prácticas y se realicen todas las funcionalidades e interacciones con el LMS que debe tener el paquete SCORM necesario.

Como lenguaje de programación para elaborar las aplicaciones se ha elegido *JavaScript* ya que es un lenguaje que se imparte en varias asignaturas del Grado de Ingeniería Telemática y proporciona algunas ventajas con respecto a otros lenguajes, como por ejemplo *Java*. Tanto las applets como una aplicación en *JavaScript* se pueden integrar en un navegador web, pero actualmente es más fácil integrar una aplicación en *JavaScript* que un applet, ya que para visualizar el applet debemos configurar el navegador y asignar una serie de permisos, pero con una aplicación *JavaScript* esto no es necesario.

Algunas características de *JavaScript* son:

- El código se ejecuta en el cliente.
- Los scripts tienen capacidades limitadas, por motivos de seguridad.
- Es un lenguaje simple, lo que facilita la programación de aplicaciones.
- Es dinámico.
- Responde a eventos en tiempo real.

Para desarrollar el código *HTML* y *JavaScript* se ha utilizado un editor de texto llamado *Sublime Text*.

Sublime Text es un editor de texto muy visual que proporciona un fácil acceso a cada uno de los archivos y proporciona algunas facilidades como:

- Modificar los colores que aparecen para identificar los diferentes elementos del código.
- Seleccionar varias líneas a la vez y escribir en todas ellas a la vez.
- Autocompletar las etiquetas *HTML*.
- Función multiventana para ver varios documentos a la vez.

Como base para la realización del cliente SIP se ha utilizado el desarrollado por *José Luis Millán*. Partiendo de esta base se han modificado algunos archivos *HTML*, *CSS* y *JavaScript* y se han creado otras nuevas para desarrollar toda la interfaz de usuario.

4.1. JSSIP

JSSIP es una librería *open source* desarrollada por José Luis Millán, Iñaki Baz Castillo y Saúl Ibarra Corretgé basada en JavaScript (ref). Esta librería contiene todos los métodos empleados para realizar las comunicaciones SIP.

Se pueden destacar las siguientes características sobre la librería:

- Se ejecuta en navegadores.
- Utiliza el protocolo SIP sobre WebSocket (WS).
- Soporte para establecer llamadas de audio y video (WebRTC).
- Soporte para establecer envío de mensajes de texto.
- Esta desarrollada totalmente en JavaScript.
- Fácil de usar.
- Actualmente continúa su desarrollo con objeto de mejorar dicha librería.

4.2. JQuery

JQuery (<https://jquery.com/>) es una librería basada en JavaScript, creada inicialmente por John Resig, que simplifica la forma de interactuar con los elementos del código HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX (ref) a las páginas web.

Algunas de sus características son:

- Manipulación de la hoja de estilos.
- Posibilidad de realizar efectos, animaciones y eventos.
- Soporta extensiones.
- Compatible con la mayoría de los navegadores.

4.3. RTCninja

RTCninja (<https://www.npmjs.com/package/rtcninja>) es una librería basada en JavaScript que sirve para lidiar con los diferentes navegadores de forma transparente.

Esta librería requiere que se acceda a ella mediante una petición *HTTP*, por lo que la aplicación que la utilice debe estar en un servidor web.

4.4. Scorm RTE

Scorm Run-Time Environment se ocupa de los requisitos para el lanzamiento de objetos de contenido, el establecimiento de comunicación entre los sistemas de gestión de aprendizaje (LMS) y los objetos de contenido compatible (SCOs), y la gestión de la información de seguimiento que puede ser comunicada entre SCOs y LMSs.

Un SCO es un recurso destinado al aprendizaje y que es desarrollado para ejecutarse en una ventana web o en un marco dentro de una página programada mediante HTML.

Para la elaboración de las aplicaciones de este TFG se ha tenido que hacer referencia a la librería *RTE.js* y se han hecho uso de una serie de métodos y atributos que dicha librería contiene:

- *initialize()*: Inicializa las comunicaciones con el LMS.
- *terminate()*: Finaliza las comunicaciones con el LMS.
- *setExit()*: Establece la forma de salida de la página.
- *setSessionTime()*: Establece el tiempo que ha durado la sesión.
- *setNavRequest()*: Establece el destino al que se irá al salir de la página.
- *setCompletionStatus()*: Establece si la página está completa o incompleta.
- *getCompletionStatus()*: Obtiene si la página está completa o incompleta.
- *setSuccessStatus()*: Establece el estado de progreso de la página.
- *getSuccessStatus()*: Obtiene el estado de progreso de la página.
- *getLearnerId()*: Obtiene el id del alumno que inicia la sesión.
- *getCommentsLearnerCount()*: Obtiene el número de comentarios almacenados.
- *setScoreRaw()*: Establece el valor de la nota que tiene el alumno en la página.
- *getScoreRaw()*: Obtiene el valor de la nota que tiene el alumno en la página.
- *setScoreMin()*: Establece el valor mínimo de la nota de la página.
- *getScoreMin()*: Obtiene el valor mínimo de la nota de la página.
- *setScoreMax()*: Establece el valor máximo de la nota de la página.
- *getScoreMax()*: Obtiene el valor máximo de la nota de la página.
- *setCommentsLearnerAll()*: Almacena o modifica un comentario especificado.
- *getCommentsLearnerAll()*: Obtiene un comentario especificado.
- *setInteractionsDescription()*: Almacena o modifica la descripción de la interacción especificada.

- *setInteractionsAddNewI()*: Almacena o modifica una interacción especificada.
- *setInteractionsResponseAll()*: Almacena o modifica la respuesta del alumno a una interacción especificada.
- *setInteractionsInitTime()*: Establece el tiempo de inicio de la interacción al tiempo actual.

Además se han creado unas funciones que se ayudan de estos métodos para facilitar la utilización de algunos de los métodos anteriores:

- *init_test()*: Establece las notas mínima y máxima de la página de teoría y carga los comentarios o los inicializa junto con las interacciones.
- *init_Practicas()*: Establece las notas mínima y máxima de la página de prácticas y carga los comentarios o los inicializa junto con las interacciones.
- *init_final()*: Establece las notas mínima y máxima de la página del test final y carga los comentarios o los inicializa junto con las interacciones.
- *init_interactions()*: Inicializa las interacciones.
- *init_comments()*: Inicializa los comentarios.
- *cargarComentario()*: Obtiene el comentario deseado que está almacenado en el LMS.
- *guardarComentario()*: Guarda el comentario deseado en el LMS.
- *set_completionStatus_SuccessStatus()*: Identifica los valores que corresponden a las variables *completionStatus* y *successStatus* dependiendo de lo que haya hecho la persona al realizar la página.
- *delete_notas()*: Borra las notas de todas las preguntas de la página actual.

4.5. El softphone SIP

Un *softphone* es un software que utiliza *VoIP* dándole al usuario la posibilidad de hacer llamadas directamente desde su PC. En nuestro caso, el cliente SIP se puede configurar para conectarse al servidor proxy. Una vez hecho, el usuario podrá realizar un chat o una llamada de voz o videollamada con otro usuario registrado en el servidor proxy. Todo esto se realiza con el objetivo de visualizar los mensajes SIP que se intercambian para adquirir conocimientos sobre el protocolo SIP.

4.5.1. Interfaz de usuario

En esta sección se va a explicar la interfaz de usuario y las funciones que puede realizar.

Como se dijo anteriormente, el cliente se ha desarrollado a partir de un programa cliente utilizado como base. El cliente original era el mostrado en *Imagen 10* e *Imagen 11*:



Imagen 10: Interfaz inicial (Formulario)

Fuente: <https://tryit.jssip.net/>

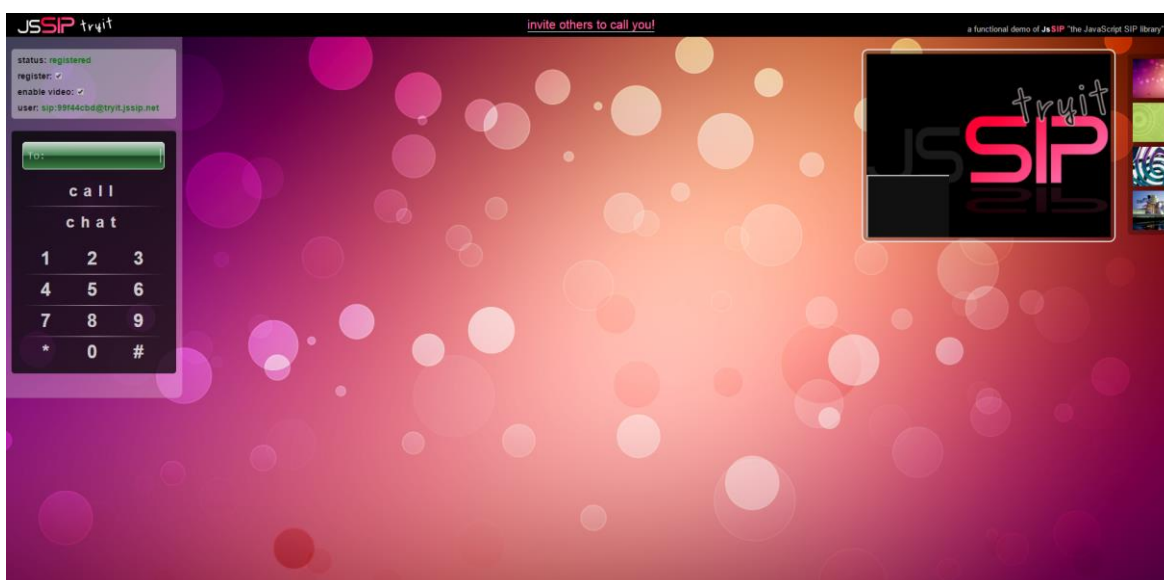


Imagen 11: Interfaz inicial (Cliente)

Fuente: <https://tryit.jssip.net/>

El resultado final de todas las modificaciones puede verse en *Imagen 12* e *Imagen 13*.

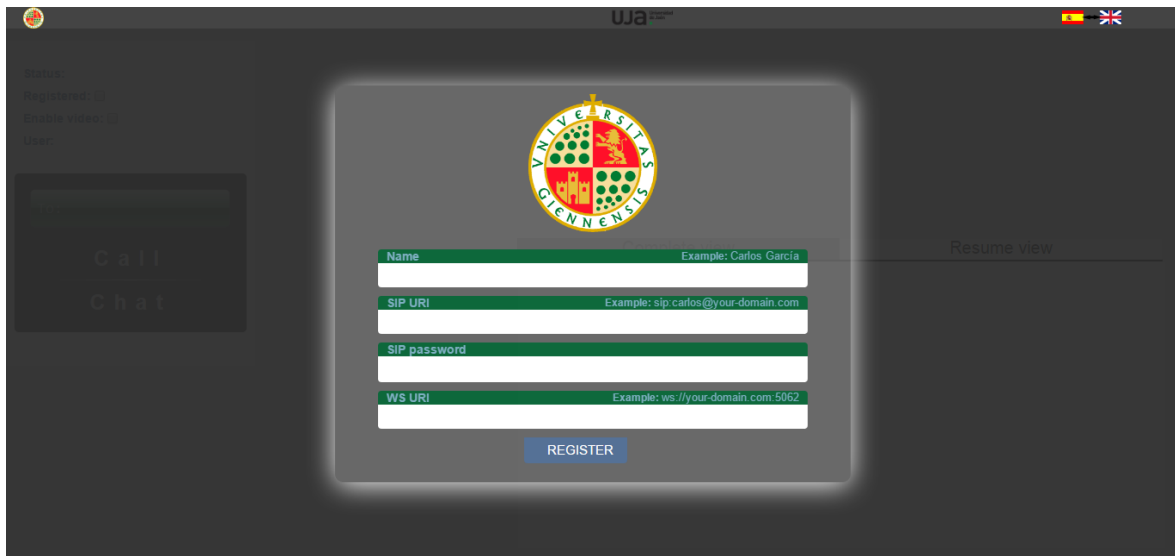


Imagen 12: Interfaz final (Formulario)

Fuente: Propia

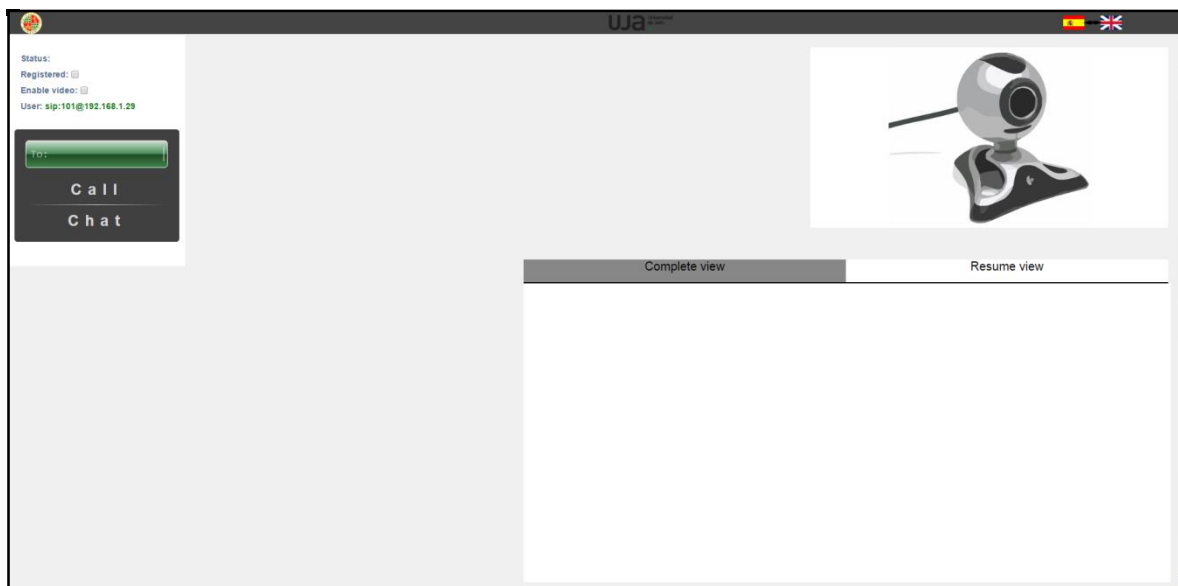


Imagen 13: Interfaz final (Cliente)

Fuente: Propia

Una vez se accede al cliente, el usuario puede encontrar un formulario de registro como el que se muestra en *Imagen 12*.

En primer lugar se va a explicar el funcionamiento de la barra de tareas, ya que ofrece una función que solo podrá utilizarse inicialmente, antes de registrarse en un servidor SIP.

Dicha función es la de cambiar el idioma de la interfaz entre inglés y español (los valores devueltos por la aplicación seguirán siendo en inglés). Esto se realiza al pulsar el botón que puede verse en *Imagen 14*.



Imagen 14: Botón idioma

Fuente: Propia

Una vez se elige el idioma deseado, el siguiente paso es que el usuario se registre mediante el formulario de registro (*Imagen 12*).

Como se puede apreciar tiene 4 campos y un botón para registrar:

- El campo *Name* (o *Nombre* en español) es para que el usuario introduzca su nombre de usuario.
- El campo *SIP URI* (o *URI SIP* en español) es para que el usuario introduzca su dirección URI SIP, que tiene el siguiente formato:

sip:[nombre de usuario]@[dominio del servidor]:[puerto]

- El campo *SIP password* (o *contraseña SIP* en español) es para que el usuario introduzca su contraseña de registro.
- El campo *WS URI* (o *URI WS* en español) es para que el usuario introduzca su dirección URI WS, que tiene el siguiente formato:

ws://[dominio del servidor]:[puerto]

- El botón *REGISTER* (o *REGISTRAR* en español) sirve para que el usuario envíe los datos del formulario al servidor y se realice el registro (una vez introducidos los valores en todos los campos). También se puede realizar pulsando la tecla "enter".

Cada campo tiene un ejemplo de posible valor a introducir en él para que el usuario pueda ver el formato de dicho campo.

Una vez realizado el registro se accede al cliente SIP. *Imagen 13* muestra una vista general del cliente SIP implementado para el presente TFG. Como se puede apreciar tiene 3 bloques bien diferenciados, que son:

- Información de usuario y teléfono (parte superior izquierda del cliente).
- Cámara origen y destino (parte superior derecha del cliente).
- Consola (parte inferior derecha del cliente).

4.5.1.1. Información de usuario y teléfono

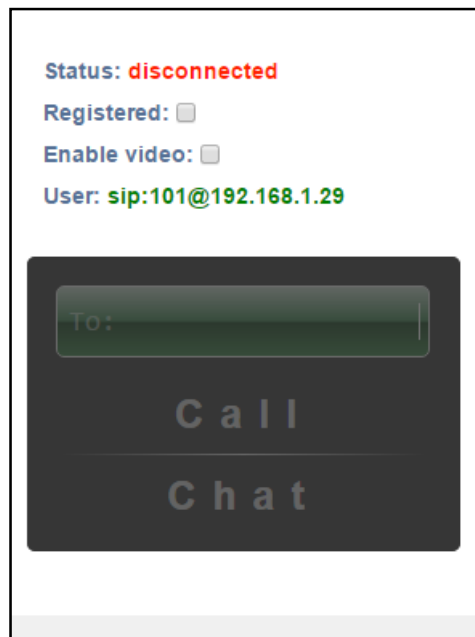


Imagen 15: Teléfono

Fuente: Propia

Imagen 15 muestra el bloque correspondiente a *Información de usuario y teléfono*. Como se puede ver se subdivide en dos:

- **Bloque de información:** Tiene 4 campos:
 - *Status* (o *Estado* en español): Indica el estado del registro.
 - *Registered* (o *Registrado* en español): Indica si está registrado o no. Si se pulsa vuelve a realizar un registro.
 - *Enable video* (o *Video habilitado* en español): Indica si la llamada será de audio o videollamada.
 - *User* (o *Usuario* en español): Indica la URI SIP del usuario.
- **Bloque de teléfono:** Tiene un campo en el que introducir el nombre del usuario al que queremos llamar o enviar un mensaje (solo el nombre de usuario, ya que la aplicación construye la URI SIP) y dos botones:
 - *Call* (o *Llamar* en español): Sirve para realizar una llamada al usuario indicado en el campo.
 - *Chat* (o *Chatear* en español): Sirve para abrir la ventana donde se envían mensajes al usuario indicado en el campo.

Si se introduce un nombre de usuario en el campo (en este caso 102) y se selecciona el botón *Chat* se abre una ventana mostrada en Imagen 16.

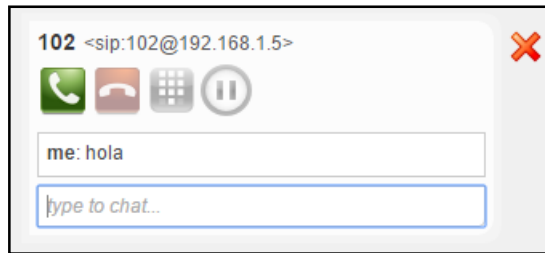


Imagen 16: Ventana de chat

Fuente: Propia

Al escribir una cadena en el campo situado en la zona inferior (en este caso "hola") y pulsar la tecla "enter" se le envía ese mensaje al usuario destino. Mientras haya algo escrito en dicho campo, al usuario destino se le mostrará una imagen en formato *gif* que le indicará que el otro extremo está escribiendo.

Como se puede observar en *Imagen 16*, sobre los mensajes hay 4 botones:

- El botón verde sirve para realizar una llamada al usuario destino. Esta será de audio, si no está seleccionada la opción "Enable video" del bloque de información y teléfono, o será videollamada, en caso contrario.
- El botón rojo sirve para colgar la llamada en ejecución.
- El botón con puntos sirve para mostrar un teclado con números. No tiene utilidad.
- El botón de pause sirve para pausar la llamada. En este momento ningún usuario puede escuchar al otro. Para volver a reanudar la llamada hay que volver a pulsar el botón.

En la esquina superior derecha de cada *chat* aparece una 'x' que al pulsarla cierra dicha ventana de chat.

4.5.1.2. Cámara origen y destino



Imagen 17: Ventana de cámaras

Fuente: Propia

Imagen 17 muestra el bloque correspondiente a *Cámara origen y destino*. Se subdivide en dos, aunque no se puede apreciar en la imagen porque ambas tienen color blanco de fondo por defecto:

- *Cámara origen*: Esta aparece en la esquina inferior izquierda dentro de un rectángulo más pequeño que la cámara destino.
- *Cámara destino*: Esta aparece en el resto del bloque y se ve mucho más grande que la cámara origen.

Este bloque solo se activa si la opción "Enable video" del bloque de *Información y teléfono* está habilitada. Se muestran en tiempo real ambas cámaras, con algo de retraso si la red está saturada.

4.5.1.3. Consola

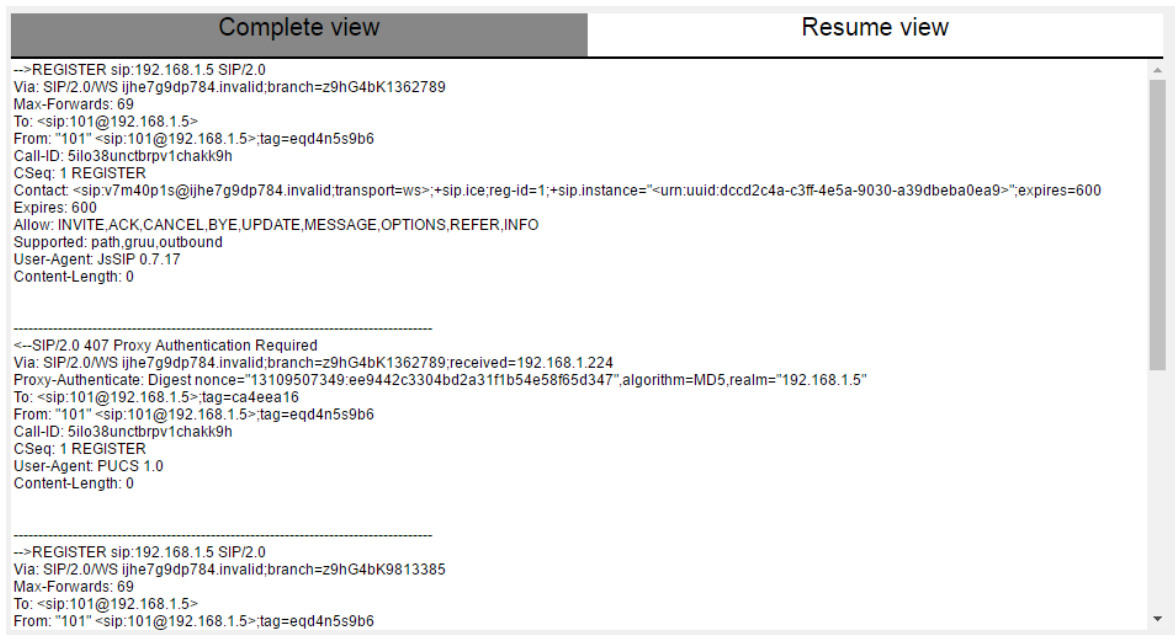


Imagen 18: Consola (Vista completa)

Fuente: Propia

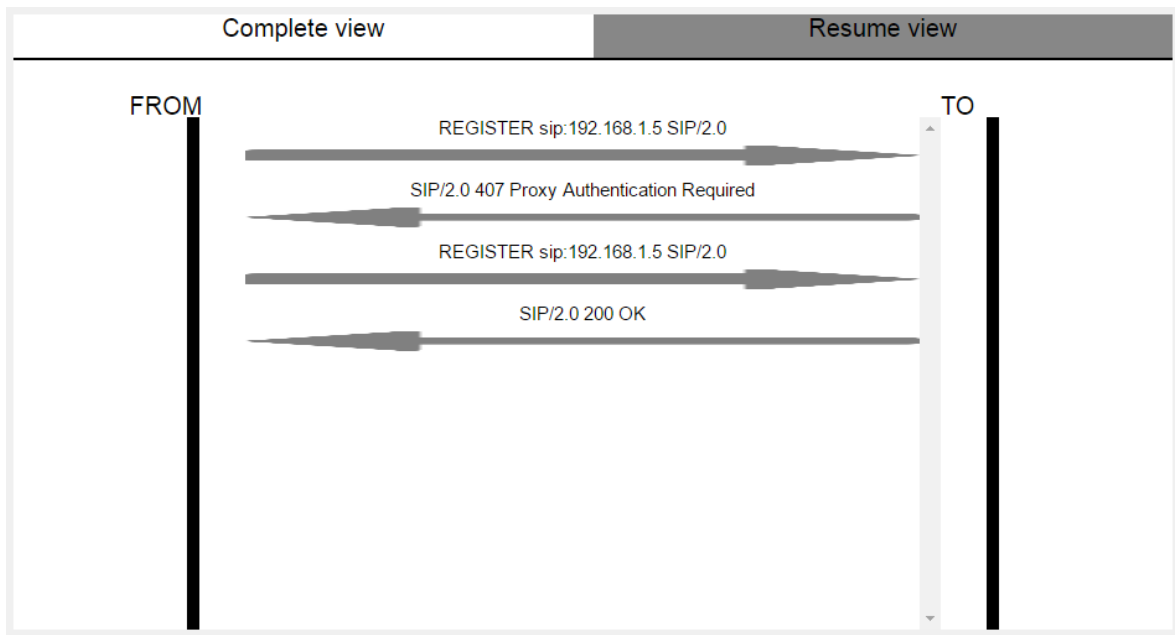


Imagen 19: Consola (Vista resumen)

Fuente: Propia

En este bloque es donde se visualizan los mensajes SIP en tiempo real. El bloque de consola se subdivide en 2 vistas:

- *Complete view* (o *Vista completa* en español): Como se puede ver en *Imagen 18*, esta vista muestra todos los mensajes en formato texto, cada uno separado del siguiente por una línea de "-----".
- *Resume view* (o *Vista resumen* en español): Como se puede ver en *Imagen 19*, esta vista muestra los mensajes como flechas con la línea de inicio del mensaje SIP encima de ella. Si la flecha va hacia la derecha es un mensaje enviado y si va hacia la izquierda es un mensaje recibido. Al pasar el ratón por encima de una flecha aparece una ventana en la que se muestra el mensaje SIP completo y, si se selecciona una flecha, esta ventana se queda bloqueada para poder copiar algún campo del mensaje. Para poder ver otros mensajes es necesario volver a pulsar dicha flecha.

La *vista completa* ofrece una visión completa de todos los mensajes, mientras que la *vista resumen* muestra una visión general de los mensajes, con posibilidad de ver todo el contenido del mensaje deseado.

La *vista completa* es útil cuando hay pocos mensajes, ya que se pueden localizar ciertos mensajes fácilmente. Pero cuando tenemos muchos mensajes es recomendable utilizar la *vista resumen*, ya que solo con ver la línea de inicio de los mensajes SIP y el sentido de la flecha es más fácil localizar mensajes específicos.

4.6. El servidor SIP

El servidor utilizado para este proyecto es el denominado *Port SIP*, debido a los resultados producidos y a la compatibilidad con el cliente SIP implementado.

Port SIP es un proveedor de *VoIP* que ofrece comunicaciones visuales *end-to-end*, que permiten establecer un balance en la eficiencia del servidor.

Otra ventaja por la que se ha elegido este servidor es su sencilla instalación y configuración, la cual se explica detalladamente en el *anexo 7.1. Instalación y configuración del servidor SIP*.

En resumen, algunas de sus características más importantes son:

- Fácil instalación.
- Fácil configuración y mantenimiento.
- Es muy compatible, escalable y estable.
- No se necesitan demasiados conocimientos de telecomunicaciones, ya que presenta una interfaz visual de navegador que simplifica la configuración.
- Está basado en WebRTC.

4.7. Seguridad en SIP

4.7.1. Ataques

Los ataques y amenazas que se pueden realizar contra una comunicación SIP son los siguientes:

- *Modificación y finalización de sesiones*: Un atacante podría crear e introducir mensajes en una comunicación, pudiendo incluso finalizar la sesión con un mensaje BYE o cambiar los parámetros de la sesión con otro mensaje INVITE, entre otros ataques.
- *Secuestro de registro*: Este ataque consiste en que el atacante se registra suplantando al usuario, de forma que todo el tráfico destinado a dicho usuario también será enviado al atacante.
- *Suplantación de servidor*: Este ataque consiste en que el atacante genere mensajes de respuesta ante peticiones del cliente, pudiendo redirigir al usuario a recursos inapropiados o inseguros.
- *Manipulación del cuerpo del mensaje*: Este ataque podría realizarlo cualquier atacante que este en medio de un intercambio de mensajes SIP y consistiría en que el atacante coge el mensaje, modifica su contenido y genera un mensaje idéntico con el nuevo contenido.
- *Denegación de servicio*: Un atacante podría enviar tantas peticiones como para ocupar un servidor por completo y que el resto de usuarios no puedan acceder debido a que el servidor les responde que está lleno.

4.7.2. Servicios de seguridad

Para evitar los ataques mencionados en el punto anterior, los principales servicios de seguridad requeridos por el protocolo SIP son:

- *Autenticación*: Es el proceso que asegura que el usuario con el que me estoy comunicando es quien dice ser.
- *Integridad*: Con este servicio se asegura que los mensajes SIP no fueron modificados ni manipulados por un atacante.
- *Confidencialidad*: Con este servicio se previene que un atacante pueda obtener algún tipo de información incluida en el mensaje SIP.

4.7.3. Mecanismos de seguridad

Existen mecanismos que se pueden utilizar para proporcionar al protocolo SIP los servicios indicados en el punto anterior:

- *IPsec*: Es un conjunto de protocolos de nivel de red que soporta el intercambio seguro de paquetes en la capa IP.
- *Transport Layer Security (TLS)*: Otorga seguridad a la capa de transporte sobre protocolos orientados a conexión, como TCP.
- *SIPS URI*: Este tipo de URI es como la URI SIP pero con la mejora de que exige que el recurso debe ser alcanzado de manera segura con TLS. Para utilizar este tipo de URI se usa el mismo formato pero con el esquema *sips*.
- *S/MIME*: Los mensajes SIP pueden utilizar el estándar MIME en su cuerpo. Este estándar contiene un mecanismo, llamado S/MIME, para proporcionar seguridad a sus contenidos.

4.7.4. Acciones desarrolladas

En cada uno de los mensajes SIP generados por el cliente SIP, existen etiquetas llamadas *tags* en los campos *From* y *To* de la cabecera. Estos *tag* sirven para identificar, de manera única, la transacción a la que pertenecen.

De esta forma, todos los mensajes de una misma transacción tendrán el mismo *tag*. Este cambiará al pasar de transacción.

Además, el servidor tiene un sistema de autenticación, que hace que al recibir un mensaje SIP de un usuario, este necesite una autenticación para que el servidor lo retransmita al destino.

5. RESULTADOS Y DISCUSIÓN

El resultado del presente TFG es la creación de un *WebLab* basado en una estructura SCORM que puede ser utilizado en cualquier LMS compatible con SCORM y ha sido probado y depurado en ILIAS, el LMS institucional de la Universidad de Jaén.

El *WebLab* implementado permite a los usuarios:

- Adquirir conocimientos sobre el protocolo SIP y realizar prácticas personalizadas.
- Ser evaluados mediante test que aparecerán en algunas páginas.
- Manejar un servidor Proxy SIP y un cliente SIP basado en JavaScript.

Mediante el uso de la *especificación RTE*, se ha conseguido ejecutar el contenido en un navegador web y que este se comunique con el LMS. Esta comunicación permite el envío de comentarios, puntuaciones, timestamp y respuestas del usuario, entre otros. El lenguaje utilizado ha sido JavaScript.

Además se han creado 5 SCOs y se ha establecido una secuencia entre ellos tal y como se muestra en el flujo grama del apartado 5.2.5. *Página 5. Evaluación final*.

En dicho SCORM se ha incrustado un cliente SIP basado en JavaScript destinado al envío y presentación de los mensajes SIP en tiempo real, con el objetivo de que el usuario entienda que está enviando y recibiendo la aplicación.

5.1. Como realizar comunicaciones

En los dos siguientes subapartados se van a explicar los pasos necesarios para realizar las comunicaciones, desde la finalización del registro, que se explicará en este apartado.

Para el registro tenemos el esquema mostrado en *Imagen 20*:

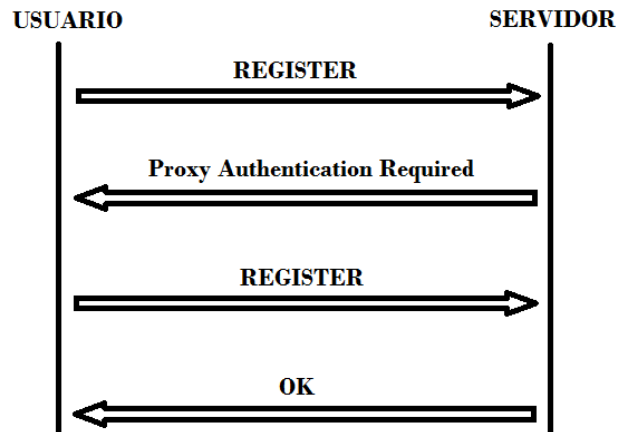


Imagen 20: Intercambio de mensajes SIP (Registro)

Fuente: Propia

Paso 1

El cliente envía el mensaje SIP de tipo *REGISTER*. Imagen 21 muestra un ejemplo.

```

-->REGISTER sip:weblab.ujaen.es SIP/2.0
Via: SIP/2.0/WS mhm2ofpd6nau.invalid;branch=z9hG4bK3873454
Max-Forwards: 69
To: <sip:102@weblab.ujaen.es>
From: "102" <sip:102@weblab.ujaen.es>;tag=ktsp9n187p
Call-ID: 9icajn9lagaq11ascf7o1d
CSeq: 1 REGISTER
Contact: <sip:a3r10eto@mhm2ofpd6nau.invalid;transport=ws>;+sip.ice;reg-id=1;+sip.instance="urn:uuid:b5332333-5b2f-4759-baa4-aab8855a0771">;expires=600
Expires: 600
Allow: INVITE,ACK,CANCEL,BYE,UPDATE,MESSAGE,OPTIONS,REFER,INFO
Supported: path,gruu,outbound
User-Agent: JsSIP 0.7.17
Content-Length: 0
  
```

Imagen 21: Ejemplo REGISTER 1

Fuente: Propia

Paso 2

El servidor no acepta el registro ya que requiere que el mensaje *REGISTER* incluya la autorización necesaria (clave), por lo que le devuelve un mensaje en el que se indica que se requiere la autenticación. Imagen 22 muestra un ejemplo.

```

<--SIP/2.0 407 Proxy Authentication Required
Via: SIP/2.0/WS mhm2ofpd6nau.invalid;branch=z9hG4bK3873454;received=217.217.199.19
Proxy-Authenticate: Digest nonce="13116092645:2c3b8108c8c4b175e8738f07a26cdc3f",algorithm=MD5,realm="weblab.ujaen.es"
To: <sip:102@weblab.ujaen.es>;tag=5330dc10
From: "102" <sip:102@weblab.ujaen.es>;tag=ktsp9n187p
Call-ID: 9icajn9lagaq11ascf7o1d
CSeq: 1 REGISTER
User-Agent: PUCS 1.0
Content-Length: 0
  
```

Imagen 22: Ejemplo REGISTER 2

Fuente: Propia

Paso 3

El cliente SIP vuelve a enviar el mensaje de tipo *REGISTER* con la autorización.

Imagen 23 muestra un ejemplo.

```
-->REGISTER sip:weblab.ujaen.es SIP/2.0
Via: SIP/2.0/WS mhm2ofpd6nau.invalid;branch=z9hG4bK5199377
Max-Forwards: 69
To: <sip:102@weblab.ujaen.es>
From: "102" <sip:102@weblab.ujaen.es>;tag=ktsp9n187p
Call-ID: 9icajn9lagaq11ascf7o1d
CSeq: 2 REGISTER
Proxy-Authorization: Digest algorithm=MD5, username="102", realm="weblab.ujaen.es", nonce="13116092645:2c3b8108c8c4b175e8738f07a26cdc3f",
uri="sip:weblab.ujaen.es", response="a4a5210296843e3f7049902a27bcdc29"
Contact: <sip:a3r10eto@mhm2ofpd6nau.invalid;transport=ws>;+sip.ice;reg-id=1;+sip.instance="urn:uuid:b5332333-5b2f-4759-baa4-
aab8855a0771">;expires=600
Expires: 600
Allow: INVITE,ACK,CANCEL,BYE,UPDATE,MESSAGE,OPTIONS,REFER,INFO
Supported: path,gruu,outbound
User-Agent: JsSIP 0.7.17
Content-Length: 0
```

Imagen 23: Ejemplo REGISTER 3

Fuente: Propia

Paso 4

El servidor le devuelve un mensaje *OK* si todo va bien. *Imagen 24* muestra un ejemplo.

```
<--SIP/2.0 200 OK
Via: SIP/2.0/WS mhm2ofpd6nau.invalid;branch=z9hG4bK5199377;received=217.217.199.19
Require: outbound
Contact: <sip:a3r10eto@mhm2ofpd6nau.invalid;transport=ws>;reg-id=1;+sip.instance="urn:uuid:b5332333-5b2f-4759-baa4-
aab8855a0771">;expires=600;+sip.ice
To: <sip:102@weblab.ujaen.es>;tag=624d8c2b
From: "102" <sip:102@weblab.ujaen.es>;tag=ktsp9n187p
Call-ID: 9icajn9lagaq11ascf7o1d
CSeq: 2 REGISTER
User-Agent: PUCS 1.0
Content-Length: 0
```

Imagen 24: Ejemplo REGISTER 4

Fuente: Propia

Una vez visto el registro, los siguientes apartados se explicarán la llamada y el chat entre dos usuarios.

5.1.1. Llamada

En el esquema mostrado en *Imagen 25* se puede observar el envío de mensajes SIP típico para establecer una llamada entre dos usuarios.

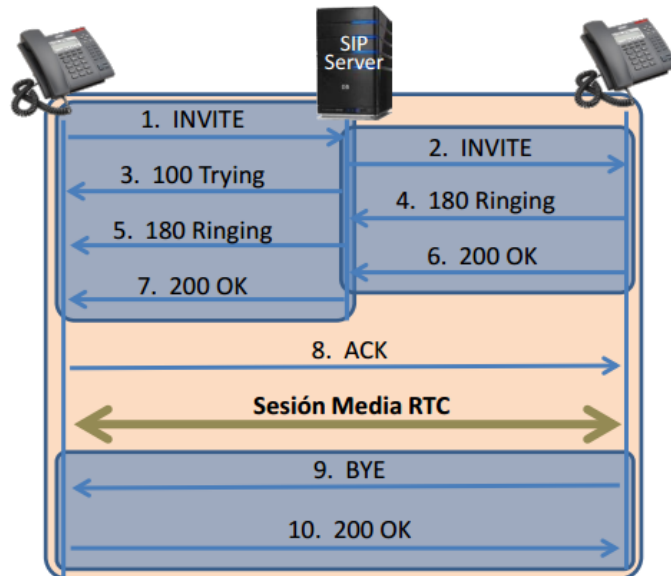


Imagen 25: Intercambio de mensajes SIP (Llamada)

Fuente: (Ruano Ruano)

Paso 1

Se envía un mensaje *INVITE* al servidor y este devuelve un mensaje *Trying*.

Paso 2

Si el mensaje *INVITE* llega al destino, este devolverá un mensaje *Ringing* al servidor y luego este lo enviará al origen de la llamada para que reconozca que el destino está sonando.

Paso 3

Cuando el destino descuelga la llamada, envía un mensaje *OK* al servidor y luego este lo envía al otro extremo. Una vez que el que establece la llamada recibe el mensaje *OK*, este envía un mensaje *ACK* y comienza la comunicación de datos extremo a extremo.

Paso 4

Una vez finalizada la llamada, se envía un mensaje *BYE* y el otro extremo responde con un mensaje *OK*.

5.1.2. Chat

En el esquema mostrado en *Imagen 26* se puede observar el intercambio de mensajes SIP típico para el envío de un mensaje entre dos usuarios.

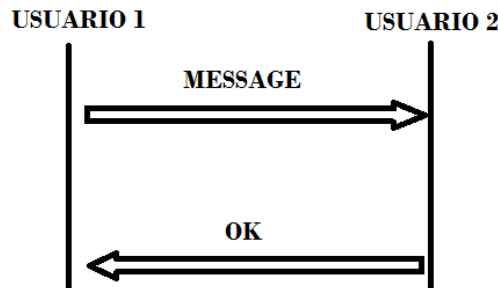


Imagen 26: Intercambio de mensajes SIP (Chat)

Fuente: Propia

Paso 1

USUARIO1 envía un mensaje *MESSAGE* en el que se incluye el mensaje “hola”.

Imagen 27 muestra un ejemplo.

```
-->MESSAGE sip:101@weblab.ujaen.es SIP/2.0
Via: SIP/2.0/WS mhm2ofpd6nau.invalid;branch=z9hG4bK8606670
Max-Forwards: 69
To: <sip:101@weblab.ujaen.es>
From: "102" <sip:102@weblab.ujaen.es>;tag=qpcpgp6a6i
Call-ID: 4i0i9kbbbs5s0kqadc5
CSeq: 3715 MESSAGE
Proxy-Authorization: Digest algorithm=MD5, username="102", realm="weblab.ujaen.es", nonce="13116094610:49e6f27840675fadb6df91331180ad0a",
uri="sip:101@weblab.ujaen.es", response="9c5f62485d9a59d065578b39f26c19a3"
Content-Type: text/plain
Allow: INVITE,ACK,CANCEL,BYE,UPDATE,MESSAGE,OPTIONS,REFER,INFO
Supported: outbound
User-Agent: JsSIP 0.7.17
Content-Length: 4

hola
```

Imagen 27: Ejemplo MESSAGE 1

Fuente: Propia

Paso 2

USUARIO2 devuelve un mensaje *OK*. *Imagen 28* muestra un ejemplo.

```
<--SIP/2.0 200 OK
Via: SIP/2.0/WS mhm2ofpd6nau.invalid;branch=z9hG4bK8606670;received=217.217.199.19
To: <sip:101@weblab.ujaen.es>;tag=a843c912
From: "102" <sip:102@weblab.ujaen.es>;tag=qpcpgp6a6i
Call-ID: 4i0i9kbbbs5s0kqadc5
CSeq: 3715 MESSAGE
User-Agent: PUCS 1.0
Content-Length: 0
```

Imagen 28: Ejemplo MESSAGE 2

Fuente: Propia

5.2. Laboratorio remoto (WebLab)

Una vez realizado el cliente, se ha procedido a la implementación de un paquete SCORM (versión 2004), con el objetivo de crear un WebLab en el que los alumnos sean capaces de utilizar la aplicación desarrollada y adquirir conocimientos sobre el protocolo SIP. Los textos e imágenes del WebLab se han programado mediante HTML y JavaScript.

El laboratorio remoto se divide en 5 bloques o páginas SCO.

5.2.1. Página 1. Introducción

Esta página se divide en 2 bloques.

En el primer bloque, el usuario podrá visualizar los datos de la asignatura a la que pertenece el laboratorio y los objetivos y competencias que debería adquirir con la realización de las prácticas. La estructura de este bloque es la siguiente:

- *Universidad*: Indica el nombre de la universidad en la que se desarrolla el laboratorio remoto.
- *Titulación*: Indica el nombre de la titulación para la que está orientado el WebLab.
- *Asignatura*: Indica el nombre de la asignatura para la que está orientado el WebLab.
- *Planificación*: Indica el curso y el cuatrimestre en el que se imparte la asignatura a la que se ha asignado el WebLab.
- *Contexto del ejercicio*: Indica el temario en el que está basado el laboratorio remoto.
- *Objetivos*: Muestra los diferentes objetivos y competencias que el usuario deberá adquirir con la realización del laboratorio remoto.
- *Documentación necesaria*: Muestra los enlaces donde puede obtenerse la teoría necesaria para realizar el WebLab.

En el segundo bloque, se muestra la descripción de cada una de las páginas del laboratorio remoto, con el objetivo de que el usuario entienda lo que se va a ver y exigir en cada página, de forma resumida.

Por último, se encuentran dos botones: uno para pasar a la siguiente página y otro para finalizar el WebLab. Estos botones estarán en todo momento activos, pudiendo ser utilizados por el usuario cuando este desee.

En *Imagen 29* se puede ver el contenido de la página 1.



INTRODUCCIÓN AL WEBLAB

Universidad:	Universidad de Jaén
Titulación:	Grado en Ingeniería Telemática
Asignatura:	Sistemas de telefonía
Planificación:	4º Curso // 1º Cuatrimestre
Contexto del ejercicio:	Tema del protocolo SIP
Objetivos:	• Instalar y configurar un servidor proxy. • Utilizar un cliente SIP para conectarse a un servidor. • Visualizar los mensajes que intervienen en la comunicación. • Entender cómo se realiza una comunicación mediante el protocolo SIP. • Reconocer los diferentes tipos de mensajes SIP. • Reconocer los diferentes campos de un mensaje SIP e interpretar su significado. • Reconocer los diferentes campos de SDP e interpretar su significado. • Capacidad de trabajar en grupo.
Documentación necesaria:	Tema de SIP

Estructura y desarrollo de cada parte del WebLab

El contenido consta de 5 páginas:

- Página 1. Introducción (esta página):** En ella se explica de forma general la estructura del WebLab y los objetivos que se conseguirán al superarlo.
- Página 2. Teoría:** En esta página se verán todos los conceptos teóricos específicos de las prácticas a realizar que proporcionan una gran ayuda para superarlas. Además, después de la teoría hay un test para comprobar que se han asimilado estos conocimientos y se está en disposición de realizar las prácticas incluidas en las siguientes páginas de este laboratorio.
- Página 3. Servidor:** Para poder acceder a esta página se debe superar el test de la página 2. En ella se realizará la instalación de un servidor proxy para poder realizar una comunicación con el cliente. En esta página no hay test.
- Página 4. Cliente:** En esta página se encuentran las prácticas donde se tiene un cliente SIP realizado con JavaScript y se debe utilizar según se indica en cada una de las prácticas. La evaluación se realizará mediante preguntas, tanto de tipo test como de escribir algún valor.
- Página 5. Evaluación final:** Para poder acceder a esta página se deben superar las tareas planteadas en la página 4. En esta página encontrarás 10 preguntas tipo test en las que se comprobará si se han entendido los trabajos realizados en las prácticas y que se han superado los objetivos especificados en esta página.

Para completar el WebLab se debe superar **cada una de las páginas** de este.

Todas las evaluaciones se superan con **más de un 5**.

Puedes repetir cada página tantas veces como quieras. Para ello debes volver a la página anterior y luego darle a siguiente para volver a la página que quieres repetir. Una vez que aceptes repetir una página, **la nota anterior de esa página se borrará**.

En cada pregunta podrás ver que hay tres botones ("Pregunta X", "Explicación" y "Comentario"). Solo puedes ver la explicación una vez hayas respondido a la pregunta y pulsado "Comprobar". Además, en cada pregunta podrás dejar un comentario para recordar alguna idea.

Al responder una pregunta, si el color del botón "Comprobar" se pone **rojo** significa que has cometido un error y si se pone **verde** es que has respondido correctamente.

Como se podrá observar en las siguientes páginas, en la esquina superior derecha habrá una ventana pequeña con la **nota** de dicha página. A continuación se muestran sus utilidades:

- En las páginas en las que haya alguna evaluación aparecerá esta ventana y te indicará en todo momento tu nota actual en esa evaluación.
- El color del texto estará en **rojo** para notas inferiores a 5 y **verde** para notas superiores.




Imagen 29: Página 1. INTRODUCCIÓN

Fuente: Propia

5.2.2. Página 2. Teoría

En esta página se muestran al usuario los contenidos teóricos más importantes de la teoría que necesitará para la realización y superación de las prácticas. La información que se muestra en esta página está estructurada de la siguiente forma:

- *Introducción*: Se muestra al usuario información previa que debe conocer sobre el protocolo SIP.
- *Elementos SIP*: Se muestra al usuario los diferentes elementos que intervienen en una comunicación SIP.
- *Terminales de usuario*: Se muestran y explican al usuario los diferentes Agentes de Usuario (UA) que se pueden dar en el protocolo SIP.
- *Servidores SIP*: Se muestran y describen los diferentes tipos de servidores SIP que existen.
- *Conceptos básicos*: Se muestra información que el usuario necesita conocer, como el formato de una URI SIP y el concepto de dominio.
- *Mensajes SIP*: Se muestran y describen al usuario todos los mensajes del protocolo SIP que se verán en las prácticas.
- *SDP*: Se muestran los contenidos teóricos necesarios sobre el protocolo SDP.
- *Escenario*: Se muestra al usuario un ejemplo de llamada entre dos usuarios y se explica el proceso que se produce paso a paso.

Imagen 30 muestra una parte del apartado de teoría.

Introducción

El objetivo de esta página es que el alumno entienda el funcionamiento del protocolo SIP y los mensajes intercambiados por este protocolo en la comunicación entre dos clientes y un servidor proxy. La teoría que se muestra a continuación es solo un recordatorio de algunos conceptos necesarios para la realización de las prácticas. Si necesitas ver algo más puedes acceder al tema de teoría haciendo clic [aquí](#).

SIP: Estándar desarrollado por IETF en 1999 (RFC 2543 y RFC 3261 en 2002) para comunicaciones multimedia entre usuarios.

Es un protocolo de control (señalización) de nivel de aplicación para la creación, modificación y terminación de sesiones con uno o más participantes.



Figura 1. Torre de protocolos

Arquitectura basada en modelo Cliente-Servidor; Cliente hace solicitudes (request) a servidor, que responde (response) para aceptarlas o rechazarlas.

1. Elementos SIP

En SIP tenemos dos tipos de elementos:

- Terminales de usuario: Equipos finales que usan los usuarios. Pueden ser clientes o servidores.
- Servidores SIP: Equipos intermedios que añaden funcionalidad a redes SIP.

2. Terminales de usuario

Los Agentes de Usuario (UA) manejan la señalización SIP. Los Terminales tienen dos componentes fundamentales para operar como UA:

- UAC. Agente de usuario Cliente (User Agent Client): Aplicación que permite que el terminal pueda realizar o iniciar una llamada. Envía

Imagen 30: Página 2. TEORÍA

Fuente: Propia

Tras la teoría, el usuario deberá superar un test de 10 preguntas, tanto tipo test como de rellenar huecos, puntuado sobre 10 puntos. No se le permitirá responder 2 veces a la misma pregunta y el usuario podrá pasar a la siguiente página si este ha obtenido más de un 5. También hay un botón para volver a la página 1 y otro para finalizar el WebLab.

Se puede destacar también que el orden de las opciones de las preguntas de opción múltiple se muestra de forma aleatoria y que, de forma visual, el alumno sabrá si ha respondido bien o mal a la pregunta, ya que el botón de esa pregunta cambiará de color y podrá ver la nota actual en el recuadro superior derecho. El color verde significa que ha respondido bien a la pregunta y el color rojo significa que ha respondido mal a la pregunta.

Además, una vez respondida la pregunta (bien o mal), se habilita la opción de explicación, que muestra una ayuda para que el usuario recuerde la parte de teoría que se corresponde con la pregunta.

El usuario puede repetir el test tantas veces como quiera. Solo debe volver a la página 2 para que se le muestre la opción de repetir o conservar la nota anterior.

El usuario puede guardar un comentario para cada pregunta en cualquier momento y, si lo desea, puede modificar el comentario.

A continuación se muestran todas las preguntas del test en *Imagen 31*, *Imagen 32*, *Imagen 33*, *Imagen 34* e *Imagen 35*.

The image shows a screenshot of a web-based test interface. It contains two question cards, each with a title bar, a question text, a text input field, and a 'Comprobar' button.

Pregunta 1 Explicación Comentario

URI SIP (Bien: 1 punto / Mal: -0.25 puntos)

¿Cuál sería tu URI SIP completa si tu usuario es "asdf", el servidor está en el dominio "ujaen.es" y el puerto es "5062"? (Escribe en minúscula y sin espacios)

Comprobar

Pregunta 2 Explicación Comentario

Línea de solicitud (Bien: 1 punto / Mal: -0.25 puntos)

¿Cuál de los siguientes métodos no es de la línea de solicitud de un mensaje SIP?:

- ☐ CANCEL.
- ☐ OK.
- ☐ INVITE.
- ☐ REGISTER.

Comprobar

Imagen 31: Página 2. TEST 1

Fuente: Propia

Pregunta 3	Explicación	Comentario
<u>Parámetros SDP</u> (Bien: 1 punto / Mal: -0.25 puntos)		
¿Cuál de los siguientes parámetros SDP es obligatorio?:		
☐ k.		
☐ o.		
☐ a.		
☐ b.		
Comprobar		

Pregunta 4	Explicación	Comentario
<u>SDP</u> (Bien: 1 punto / Mal: -0.25 puntos)		
¿Es posible alterar el orden de los parámetros SDP?:		
☐ Si.		
☐ No.		
Comprobar		

Imagen 32: Página 2. TEST 2

Fuente: Propia

Pregunta 5	Explicación	Comentario
<u>Mensajes finalización de llamada</u> (Bien: 1 punto / Mal: -0.25 puntos)		
¿Qué tipo de mensaje SIP se envía cuando quieres finalizar la llamada?:		
☐ BYE.		
☐ INVITE.		
☐ REGISTER.		
☐ CANCEL.		
Comprobar		

Pregunta 6	Explicación	Comentario
<u>Campos de la cabecera</u> (Bien: 1 punto / Mal: -0.25 puntos)		
¿Qué campo del mensaje SIP debes mirar si quieres conocer el identificador único de una llamada?:		
☐ Via.		
☐ To.		
☐ Cseq.		
☐ Call-Id.		
Comprobar		

Imagen 33: Página 2. TEST 3

Fuente: Propia

Pregunta 7	Explicación	Comentario
<u>Respuesta de éxito</u> (Bien: 1 punto / Mal: -0.25 puntos)		
¿Qué código de la línea de respuesta indica que hay éxito en la petición?:		
☐ 4xx.		
☐ 1xx.		
☐ 2xx.		
☐ 3xx.		
Comprobar		

Pregunta 8	Explicación	Comentario
<u>Servidores SIP</u> (Bien: 1 punto / Mal: -0.25 puntos)		
¿Cuál de los siguientes no es un servidor SIP?:		
☐ Servidor de registro.		
☐ Servidor de redireccionamiento.		
☐ Servidor de gestión.		
☐ Servidor proxy.		
Comprobar		

Imagen 34: Página 2. TEST 4

Fuente: Propia

Pregunta 9	Explicación	Comentario
<u>Mensajes SIP</u> (Bien: 1 punto / Mal: -0.25 puntos)		
¿Qué parte del mensaje SIP no es obligatoria?:		
☐ Línea en blanco.		
☐ Cabecera.		
☐ Línea de inicio.		
☐ Cuerpo.		
Comprobar		

Pregunta 10	Explicación	Comentario
<u>Mensajes y Servidores SIP</u> (Bien: 1 punto / Mal: -0.25 puntos)		
¿Qué servidor analiza el cuerpo del mensaje SIP?:		
☐ Ninguno.		
☐ Servidor de registro.		
☐ Servidor de redireccionamiento.		
☐ Servidor proxy.		
Comprobar		

Imagen 35: Página 2. TEST 5

Fuente: Propia

5.2.3. Página 3. Servidor

En esta página es donde se descarga el instalador del servidor utilizado para este WebLab, que es *PortSIP*.

Además se indica como instalarlo y configurarlo paso a paso para que funcione correctamente en la práctica a realizar.

Esta página no tiene ningún tipo de evaluación, ya que será evaluada en la página siguiente.

A continuación se muestran 2 imágenes, una de la parte de instalación (*Imagen 36*) y otra de la parte de configuración (*Imagen 37*).

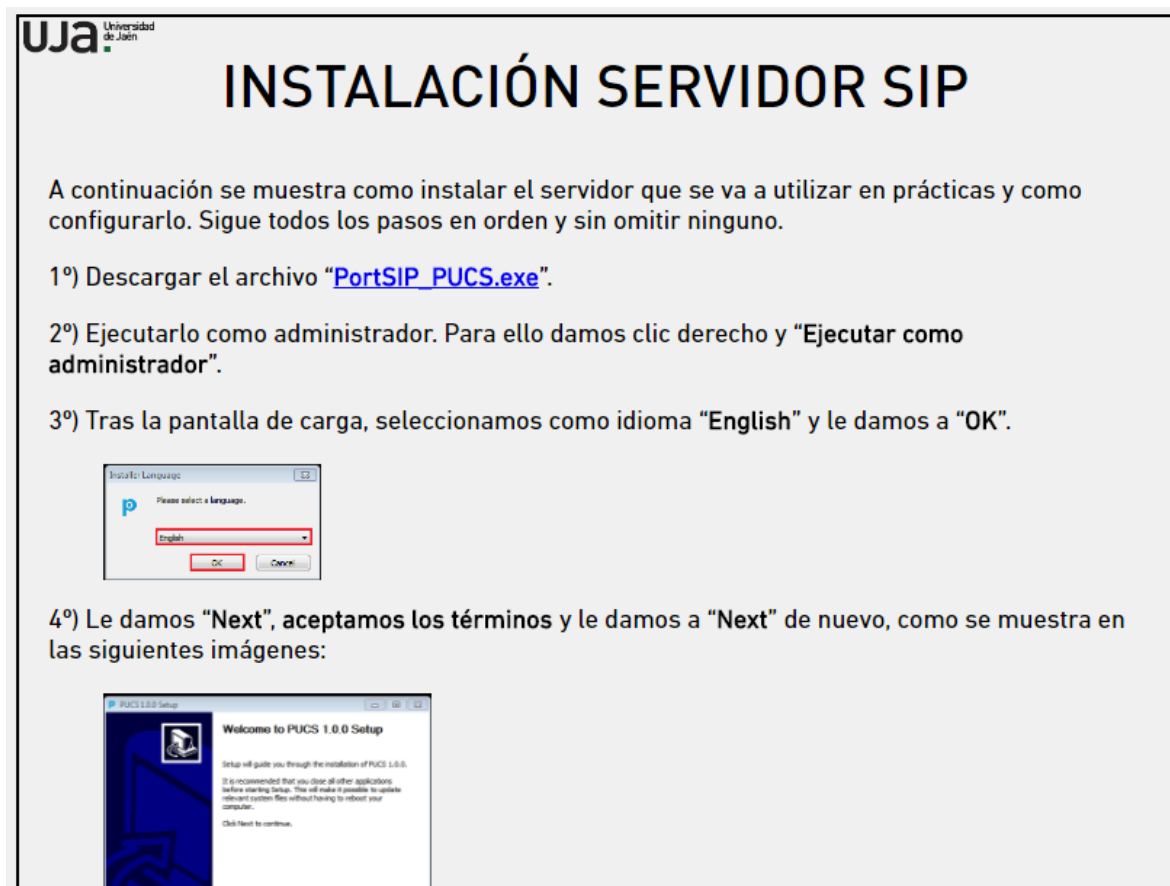


Imagen 36: Página 3. SERVIDOR (Instalación)

Fuente: Propia

CONFIGURACIÓN SERVIDOR SIP

Una vez instalado el servidor solo queda configurarlo. A continuación te mostraré los pasos para configurar correctamente el servidor. Síguelos en orden y no te saltes ninguno a menos que se especifique en algún paso.

1º) Para acceder a la configuración del servidor tienes 3 opciones:

- Si marcaste la opción "Launch Console" se te tiene que haber abierto la página de login.
- Si has instalado correctamente el servidor te tiene que haber aparecido en el escritorio un acceso directo llamado "PUCS Management Console", por lo que al hacer doble clic en él te envía a la página de login.
- Puedes abrir el navegador y poner "127.0.0.1:8888" en la barra de direcciones.

Los tres pasos te llevarán a la siguiente página de login en el navegador



2º) Ahora debes acceder a la configuración. Para ello introduce los siguientes datos y luego dale a "Login":

- Username: admin

Imagen 37: Página 3. SERVIDOR (Configuración)

Fuente: Propia

5.2.4. Página 4. Cliente

En esta página se muestra el cliente SIP que debe utilizar el usuario para realizar las prácticas, un manual del cliente para que el usuario aprenda todas sus funciones y una serie de tareas que han sido divididas en tres bloques. Cada bloque tendrá unas tareas que serán evaluadas mediante preguntas tipo test de opción múltiple y respuesta única o preguntas de rellenar huecos.

La página está puntuada sobre 10 y en total tiene 20 preguntas. El criterio de puntuación será el siguiente:

- Pregunta correcta: +0.5 puntos.
- Pregunta incorrecta: -0.125 puntos.

En cada pregunta se podrá incluir un comentario y este se podrá modificar tantas veces como sea necesario.

Además, las preguntas tipo test pueden ocultarse o mostrarse por bloques, facilitando al usuario el acceso al cliente SIP y a las diferentes preguntas.

También habrá una opción llamada "solución" que muestra la respuesta correcta a la pregunta, de forma que el alumno pueda pensar sobre su error. Esta opción solo será accesible una vez respondida la pregunta, de forma correcta o incorrecta.

El orden de las opciones de cada pregunta se mostrará de forma aleatoria.

El usuario sabrá de forma visual como ha respondido a la pregunta, ya que el botón de "Comprobar" se pondrá de otro color. Si el color es verde significa que el usuario ha respondido bien. Si el color es rojo significa que el usuario ha respondido mal.

Para superar esta página se requiere que el usuario obtenga más de un 5 en la página. El usuario puede ver la nota actual en el recuadro superior derecho y podrá repetir tantas veces como desee las prácticas de cliente, incluso habiendo obtenido más de un 5. Solo debe volver a la página 4.

Podrá volver en cualquier momento a la página anterior con el botón correspondiente.

En *Imagen 38* se muestra el formato de la página final.

Imagen 38: Página 4. CLIENTE (Manual de usuario)

Fuente: Propia

A continuación se muestran las 20 preguntas del test en *Imagen 39*, *Imagen 40*, *Imagen 41*, *Imagen 42*, *Imagen 43*, *Imagen 44*, *Imagen 45*, *Imagen 46*, *Imagen 47*, *Imagen 48*, *Imagen 49* e *Imagen 50*.

Pregunta 1	Solución	Comentario
<u>Campo Content-Length: (Bien: 0.5 puntos / Mal: -0.125 puntos)</u>		
Escribe el valor del campo "Content-Length" del mensaje de tipo REGISTER:		
Comprobar		

Pregunta 2	Solución	Comentario
<u>Campo FROM y TO: (Bien: 0.5 puntos / Mal: -0.25 puntos)</u>		
¿Coincide la URI SIP del campo "FROM" del mensaje de tipo REGISTER con la URI SIP del campo "TO" del mensaje de tipo OK?:		
☐ Si.		
☐ No.		
Comprobar		

Imagen 39: Página 4. TEST 1

Fuente: Propia

Pregunta 3	Solución	Comentario
<u>Campo CSeq: (Bien: 0.5 puntos / Mal: -0.25 puntos)</u>		
En las comunicaciones entre cliente y servidor, a cada solicitud que envía el cliente, el servidor le devuelve una respuesta. ¿El campo "CSeq" es el mismo en el mensaje de solicitud y en el de su mensaje de respuesta asociado?:		
☐ No.		
☐ Si.		
Comprobar		

Pregunta 4	Solución	Comentario
<u>Campo Call-Id: (Bien: 0.5 puntos / Mal: -0.25 puntos)</u>		
¿El campo "Call-ID" es el mismo en todos los mensajes del proceso de registro que puedes observar?:		
☐ Si.		
☐ No.		
Comprobar		

Imagen 40: Página 4. TEST 2

Fuente: Propia

Pregunta 5 Solución Comentario

Campo Allow: (Bien: 0.5 puntos / Mal: -0.25 puntos)

El campo "Allow", ¿qué métodos de línea de solicitud incluye? (Elegir la más correcta):

- ☐ Todos los métodos que incluye el protocolo SIP.
- ☐ Métodos que se permiten en esta comunicación.
- ☐ Métodos que no se permiten en esta comunicación.

Comprobar

Pregunta 6 Solución Comentario

Campo User-Agent: (Bien: 0.5 puntos / Mal: -0.125 puntos)

Introduce el valor del campo "User-Agent" del mensaje de tipo REGISTER y el del mensaje de tipo OK: (Escribelos tal cual aparece en el mensaje)

REGISTER

OK

Comprobar

Imagen 41: Página 4. TEST 3

Fuente: Propia

Pregunta 7 Solución Comentario

Mensajes SIP: (Bien: 0.5 puntos / Regular: 0.25 puntos / Mal: -0.25 puntos)

¿Por qué hay 2 mensajes de solicitud de tipo REGISTER y sus respuestas asociadas?: (Elegir la más correcta)

- ☐ Porque se requiere autenticación en el servidor.
- ☐ Porque hay que hacer un registro doble (otro después del primero).
- ☐ Porque el primero ha fallado y se ha enviado otro.

Comprobar

Imagen 42: Página 4. TEST 4

Fuente: Propia

Pregunta 8 Solución Comentario

Autorización de MESSAGE: (Bien: 0.5 puntos / Mal: -0.25 puntos)

En que mensaje de tipo MESSAGE se envía la autenticación del cliente:

- ☐ Segundo MESSAGE.
- ☐ Primer MESSAGE.

Comprobar

Pregunta 9 Solución Comentario

Campo Content-Length: (Bien: 0.5 puntos / Mal: -0.125 puntos)

Escribe el valor del campo "Content-Length" del mensaje de tipo MESSAGE:

Comprobar

Imagen 43: Página 4. TEST 5

Fuente: Propia

Pregunta 10 Solución Comentario

Campo Content-Type: (Bien: 0.5 puntos / Mal: -0.125 puntos)

Escribe el valor del campo "Content-Type" del mensaje de tipo MESSAGE que contiene el mensaje que has enviado:

Comprobar

Imagen 44: Página 4. TEST 6

Fuente: Propia

Pregunta 11 Solución Comentario

Campos TO y FROM de MESSAGE: (Bien: 0.5 puntos / Mal: -0.25 puntos)

¿Coincide la URI SIP del campo "TO" del mensaje de tipo MESSAGE con tu usuario y la URI SIP del campo "FROM" con el usuario de tu compañero?:

- ☐ Si.
- ☐ No.

Comprobar

Pregunta 12 Solución Comentario

Campo Call-Id: (Bien: 0.5 puntos / Mal: -0.25 puntos)

¿Coincide el campo "Call-ID" en todos los mensajes de la misma transacción?:

- ☐ Si.
- ☐ No.

Comprobar

Imagen 45: Página 4. TEST 7

Fuente: Propia

Pregunta 13 Solución Comentario

Contenido del mensaje: (Bien: 0.5 puntos / Mal: -0.25 puntos)

¿En qué parte del mensaje de tipo MESSAGE puedes ver el mensaje del compañero?:

- ☐ Cuerpo.
- ☐ Línea de inicio.
- ☐ Cabecera.

Comprobar

Imagen 46: Página 4. TEST 8

Fuente: Propia

Pregunta 14 Solución Comentario

Protocolo para describir sesiones: (Bien: 0.5 puntos / Mal: -0.125 puntos)

Como se denomina el protocolo que se utiliza en el cuerpo del mensaje de solicitud INVITE, el cual nos proporciona información sobre la sesión: (siglas en mayúsculas)

Comprobar

Pregunta 15 Solución Comentario

Campo Content-Type: (Bien: 0.5 puntos / Mal: -0.125 puntos)

Escribe el valor del campo "Content-Type" del mensaje de tipo INVITE que has enviado: (Escribelo tal cual aparece en el mensaje)

Comprobar

Imagen 47: Página 4. TEST 9

Fuente: Propia

Pregunta 16 Solución Comentario

Parámetro SDP: (Bien: 0.5 puntos / Mal: -0.125 puntos)

¿En que parámetro del SDP que aparece en el contenido del INVITE podemos ver el tipo de medio y puerto, entre otros campos?: (Escribelo en minúscula)

Comprobar

Imagen 48: Página 4. TEST 10

Fuente: Propia

Pregunta 17

Solución Comentario

Mensaje Trying: (Bien: 0.5 puntos / Mal: -0.125 puntos)

Escribe el número que hay en el campo "Content-Length" del mensaje de tipo Trying:

Comprobar

Ahora que tu compañero cuelgue la llamada y contesta a las siguientes preguntas:

Pregunta 18

Solución Comentario

Fin de llamada: (Bien: 0.5 puntos / Mal: -0.125 puntos)

Escribe el nombre del tipo de mensaje que finaliza la llamada, y el que te avisa de que el otro extremo ha recibido ese mensaje: (Escribe en mayúsculas)

Comprobar

Imagen 49: Página 4. TEST 11

Fuente: Propia

Pregunta 19

Solución Comentario

Mensaje BYE: (Bien: 0.5 puntos / Mal: -0.125 puntos)

Escribe el valor del campo "Content-Length" del mensaje de tipo BYE:

Comprobar

Pregunta 20

Solución Comentario

Transporte: (Bien: 0.5 puntos / Mal: -0.125 puntos)

¿Qué tipo de transporte se está utilizando para enviar los datos en la comunicación? (Ayuda: mirar parámetros de "SDP"): (siglas en mayúscula)

Comprobar

Imagen 50: Página 4. TEST 12

Fuente: Propia

5.2.5. Página 5. Evaluación final

En esta última página se muestra un test de evaluación final con 10 preguntas tipo test, cada una con 5 opciones, 4 opciones más 1 de "No sabe/No contesta". La puntuación máxima será de 10 puntos y el criterio de puntuación será el siguiente:

- Pregunta correcta: +1 punto.
- Pregunta incorrecta: -0.25 puntos.
- No sabe/No contesta: +0 puntos.

En cada pregunta se podrá incluir un comentario y este se podrá modificar tantas veces como sea necesario.

También habrá una opción llamada "explicación" que muestra una ayuda para encontrar la solución. Esta opción solo será accesible una vez respondida la pregunta, de forma correcta, incorrecta o No sabe/No contesta.

Además, de cada pregunta se guardará el timestamp y el tiempo que el usuario tarde en contestar, entre otros datos de la interacción. El orden de las opciones de cada pregunta se mostrará de forma aleatoria, excepto la opción "No sabe/No contesta", que se mostrará siempre la última.

En esta página se muestra solo una pregunta a la vez. Por ello hay un menú con 10 botones numerados que indican el número de la pregunta a la que hacen referencia.

El usuario sabrá de forma visual como ha respondido a la pregunta, ya que tanto el botón de "Comprobar" como su número en el menú de preguntas se pondrán de otro color. Si el color es verde significa que el usuario ha respondido bien. Si el color es rojo significa que el usuario ha respondido mal. Si el color es gris significa que el usuario ha respondido No sabe/No contesta.

Para superar el test final se requiere que el usuario obtenga más de un 5 en el test. El usuario puede ver la nota actual en el recuadro superior derecho y podrá repetir tantas veces como desee el test final, incluso habiendo obtenido más de un 5. Solo debe volver a la página 5.

Podrá volver en cualquier momento a la página 4 con el botón correspondiente.

En *Imagen 51* se muestra el formato de la página final.

UJA

Universidad

de Jaén

EVALUACIÓN FINAL

0/10

Consejos:

1. En esta página hay 10 preguntas tipo test de las cuales solo podrás ver una a la vez hasta llegar a la última. Para pasar a la siguiente pregunta debes seleccionar el número de la pregunta a la que quieres ir en el menú que está encima de la pregunta.
2. El test se supera con más de un 5. Si has sacado menos debes repetirlo hasta obtener más de un 5. Puedes repetirlo tantas veces como quieras incluso si has sacado más de un 5.
3. Piensa bien antes de responder ya que solo podrás responder una vez. Después de responder tendrás acceso a la explicación de la pregunta para que puedas ver la solución. En caso de alguna queja en la corrección de una pregunta avisa al profesor.
4. Puedes escribir un comentario en cada pregunta que te ayude a recordar algo.
5. Al contestar tendrás acceso a la explicación de la parte de teoría que pertenece a esa parte. No se dice la solución.
6. Al responder una pregunta, si el color del botón "Comprobar" se pone rojo significa que has cometido un error y si se pone verde es que has respondido correctamente.
7. Las preguntas contestadas aparecerán en el menú de color verde si se han respondido correctamente, de color rojo si se han respondido mal o de color gris si se ha respondido "No sabe/No contesta".

1

2

3

4

5

6

7

8

9

10

Pregunta 1

Explicación

Comentario

Mensajes SIP: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

¿Qué mensaje se envía para llamar a otro usuario/compañero?:

- ☐ BYE.
- ☐ ACK.
- ☐ REGISTER.
- ☐ INVITE.
- ☐ No sabe/No contesta.

Comprobar

Imagen 51: Página 5. EVALUACIÓN FINAL

Fuente: Propia

En *Imagen 62* puede verse el flujo grama del Laboratorio Web.

A continuación se muestran las 10 preguntas del test en *Imagen 52*, *Imagen 53*, *Imagen 54*, *Imagen 55*, *Imagen 56*, *Imagen 57*, *Imagen 58*, *Imagen 59*, *Imagen 60* e *Imagen 61*.

Pregunta 1

Explicación

Comentario

Mensajes SIP: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

¿Qué mensaje se envía para llamar a otro usuario/compañero?:

- ☐ BYE.
- ☐ ACK.
- ☐ REGISTER.
- ☐ INVITE.
- ☐ No sabe/No contesta.

Comprobar

Imagen 52: Página 5. TEST 1

Fuente: Propia

Pregunta 2 Explicación Comentario

0/10

Datos conexión cliente SIP: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

Dado el siguiente mensaje, identifica los campos necesarios y elige la configuración correcta que se ha utilizado para conectarse al servidor:

```
REGISTER sip:weblab.ujaen.es SIP/2.0
Via: SIP/2.0/WS 7c5bfq7j8rke.invalid;branch=z9hG4bK4825582
Max-Forwards: 69
To: sip:101@weblab.ujaen.es
From: "101" sip:101@weblab.ujaen.es;tag=r03if8mt13
Call-ID: 13nfep5cftp7uo50rc0ks
CSeq: 1 REGISTER
Contact: sip:101@weblab.ujaen.es;transport=ws;+sip.ice;reg-id=1;+sip.instance="urn:uuid:067b311d-7ec1-4d4d-a421-66dd94701617"
Expires: 600
Allow: INVITE,ACK,CANCEL,BYE,UPDATE,MESSAGE,OPTIONS,REFER,INFO
Supported: path,gruu,outbound
User-Agent: JsSIP 0.7.17
Content-Length: 0
```

☐ User SIP URI: "sip:101@weblab.ujaen.es"
 Server URI: "udp://weblab.ujaen.es:5060"

☐ User SIP URI: "sip:101@weblab.ujaen.es"
 Server URI: "udp://weblab.ujaen.es:5062"

☐ User SIP URI: "sip:101@weblab.ujaen.es"
 Server URI: "ws://weblab.ujaen.es:5062"

☐ User SIP URI: "sip:101@weblab.ujaen.es"
 Server URI: "tcp://weblab.ujaen.es:5062"

☐ No sabe/No contesta.

Comprobar

Imagen 53: Página 5. TEST 2

Fuente: Propia

Pregunta 3 Explicación Comentario

Campo mensaje SIP: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

Si te quieres comunicar con un usuario/compañero que está registrado en tu mismo servidor. ¿Qué campo necesitas saber para comunicarte con él?:

☐ Puerto del usuario.
☐ Puerto del servidor.
☐ URI SIP del usuario.
☐ URI SIP del servidor.
☐ No sabe/No contesta.

Comprobar

Imagen 54: Página 5. TEST 3

Fuente: Propia

Pregunta 4 Explicación Comentario

Comunicación SIP: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

Quando realizas una llamada a otro usuario se generan una serie de mensajes que por lo general tienen una secuencia establecida (vista en teoría). De las siguientes opciones, elige la correcta: (cuidado con el orden y sentido de los mensajes SIP)

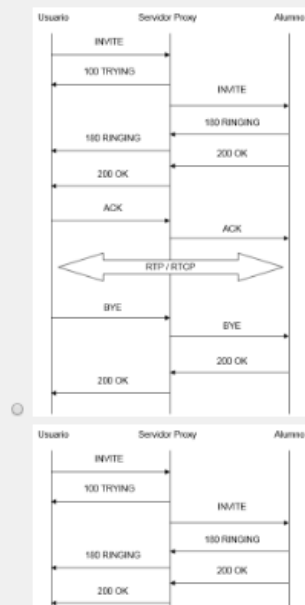


Imagen 55: Página 5. TEST 4

Fuente: Propia

Pregunta 5 Explicación Comentario

Campo mensaje SIP 2: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

Si tienes varias llamadas a la vez y tú solo quieres ver los mensajes de una de ellas. ¿Qué campo te indica los mensajes que pertenecen a esa llamada?:

- ☐ Contact.
- ☐ User-Agent.
- ☐ Cseq.
- ☐ Call-Id.
- ☐ No sabe/No contesta.

Comprobar

Imagen 56: Página 5. TEST 5

Fuente: Propia

Pregunta 6 Explicación Comentario

Campo SDP: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

En el siguiente ejemplo de parámetro SDP, ¿cuál de las siguientes afirmaciones es correcta?:

m=video RTP/AVP 15 3 0

- ☐ Video en puerto sobre RTP y puede usar el codec 15, 3 o 0.
- ☐ Video en puertos 15, 3 y 0 sobre RTP y codecs .
- ☐ Video en puerto sobre RTP y usa los codecs 15, 3 y 0.
- ☐ Video en puertos 15, 3 o 0 sobre RTP y codecs .
- ☐ No sabe/No contesta.

Imagen 57: Página 5. TEST 6

Fuente: Propia

Pregunta 7 Explicación Comentario

Servidor: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

¿En qué apartado de los siguientes se añaden los usuarios al servidor instalado en la página 3?:

- ☐ Extensions.
- ☐ Media Server.
- ☐ Domain and transports.
- ☐ Settings.
- ☐ No sabe/No contesta.

Imagen 58: Página 5. TEST 7

Fuente: Propia

Pregunta 8 Explicación Comentario

Identificar usuario origen: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

Dado el siguiente mensaje, identifica la URI SIP del usuario que te ha enviado el mensaje:

```
MESSAGE sip:03v21444@4h00tqe7pl00.invalid;transport=ws SIP/2.0
Via: SIP/2.0/WS 192.168.133.14:5062;branch=z9hG4bK-524287-1----425128297750134c;rport
Max-Forwards: 19
To: sip:101@weblab.ujaen.es
From: "102" sip:102@weblab.ujaen.es;tag=dd17c043
Call-ID: Dyej3GnklmvphHEM-Zfljg..
CSeq: 3 MESSAGE
Allow: REGISTER,INVITE,ACK,CANCEL,OPTIONS,BYE,REFER,NOTIFY,SUBSCRIBE,UPDATE,INFO,MESSAGE,PRACK
Content-Type: text/plain
User-Agent: PUCS 1.0
Content-Length: 4

Hola
```

- ☐ sip:101@weblab.ujaen.es
- ☐ sip:102@ujaen.es
- ☐ sip:102@weblab.ujaen.es
- ☐ sip:101@ujaen.es
- ☐ No sabe/No contesta.

Imagen 59: Página 5. TEST 8

Fuente: Propia

Pregunta 9 Explicación Comentario

Tipo de mensaje SIP: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

¿A qué tipo de solicitud SIP está asociado el siguiente mensaje?:

```
SIP/2.0 100 Trying
Via: SIP/2.0/WS 192.168.133.14:5062;branch=z9hG4bK-524287-1---783dda19bd50707c;rport
To: sip:101@weblab.ujaen.es
From: "102" sip:102@weblab.ujaen.es;tag=404d582b
Call-ID: QRLm-t9WZbbN5ehUX-bKJQ..
CSeq: 1 INVITE
Supported: timer,ice,replaces,outbound
Content-Length: 0
```

- ☐ ACK.
- ☐ Trying.
- ☐ INVITE.
- ☐ OK.
- ☐ No sabe/No contesta.

Comprobar

Imagen 60: Página 5. TEST 9

Fuente: Propia

Pregunta 10 Explicación Comentario

Campo mensaje SIP 3: (Bien: 1 punto / Mal: -0.25 puntos / No sabe/No contesta: 0 puntos)

Según el mensaje mostrado en la siguiente imagen, ¿qué puerto se utiliza para enviar el mensaje al servidor?:

```
INVITE sip:03v21444@4h00tqe7pl00.invalid;transport=ws SIP/2.0
Via: SIP/2.0/WS 192.168.133.14:5062;branch=z9hG4bK-1---783dda19bd50707c;rport
Max-Forwards: 19
Contact: sip:102@150.214.100.80:80;ob;transport=ws
To: sip:101@weblab.ujaen.es
From: "102" sip:102@weblab.ujaen.es;tag=404d582b
Call-ID: QRLm-t9WZbbN5ehUX-bKJQ..
CSeq: 1 INVITE
Session-Expires: 1800
Min-SE: 90
Accept-Language: en
Allow: REGISTER,INVITE,ACK,CANCEL,OPTIONS,BYE,REFER,NOTIFY,SUBSCRIBE,UPDATE,INFO,MESSAGE,PRACK
Content-Type: application/sdp
Supported: timer,replaces,norefersub,dialog,100rel
User-Agent: PUCS 1.0
Content-Length: 2641

v=0
o=mozilla... THIS_IS_SDPARTA-46.0.1 466437660179815839 0 IN IP4 0.0.0.0
.
.
.
```

- ☐ 5062.
- ☐ 5060.
- ☐ .
- ☐ 8080.
- ☐ No sabe/No contesta.

Comprobar

Imagen 61: Página 5. TEST 10

Fuente: Propia

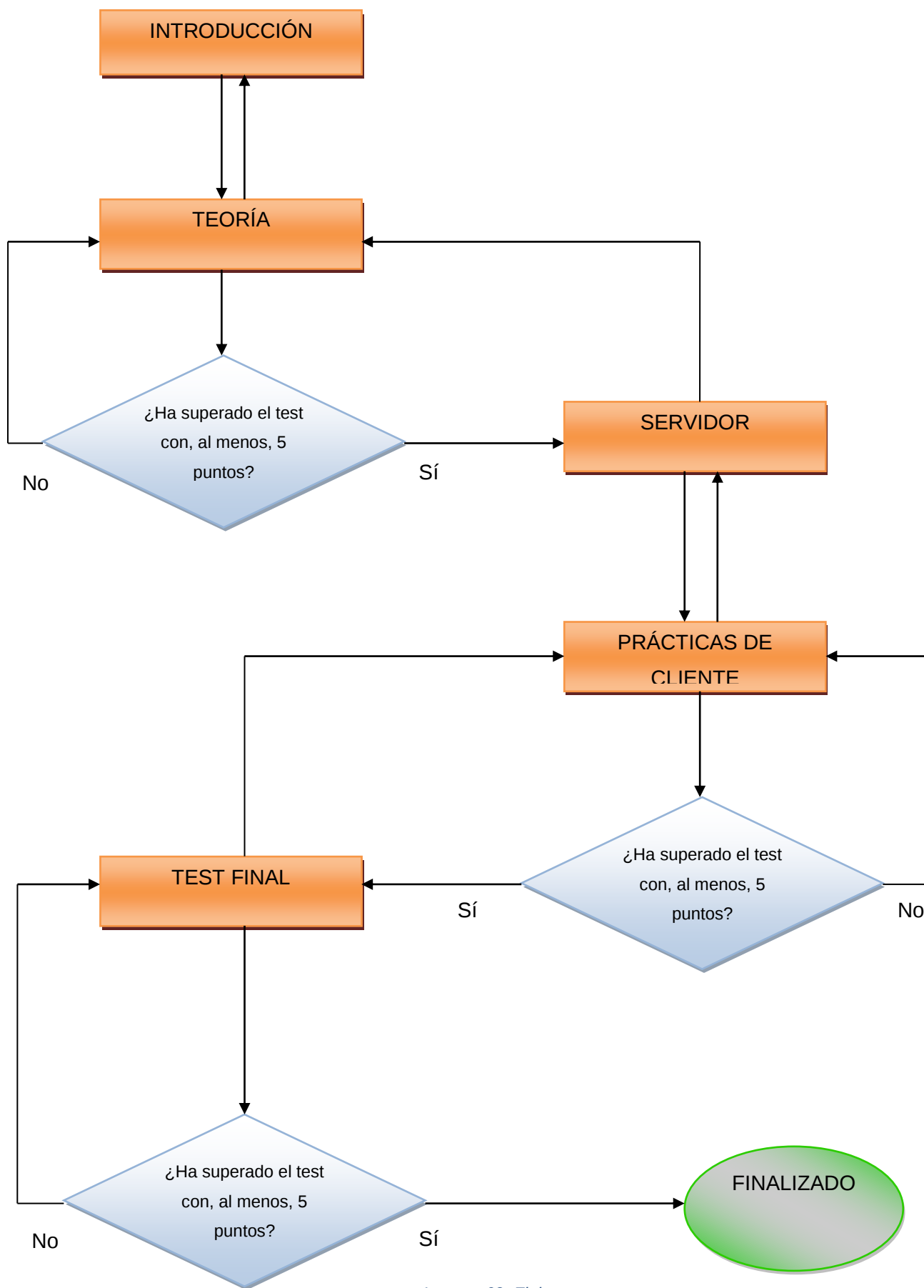


Imagen 62: Flujo grama

Fuente: Propia

6. CONCLUSIONES

El resultado del presente TFG ha sido la creación de un WebLab basado en SCORM e integrado en el LMS de la Universidad de Jaén, con el objetivo de que sea utilizado por los alumnos relacionados con asignaturas en las que se estudie el protocolo SIP.

Para conseguir dicho resultado se han realizado las siguientes acciones:

- Búsqueda de un servidor proxy para realizar el registro de los usuarios.
- Desarrollo de un softphone basado en JavaScript que actúa como cliente SIP para el envío, recepción y presentación de mensajes SIP en tiempo real. Además se permite establecer comunicaciones con otros usuarios, tanto de voz como de video.
- Desarrollo de una interfaz sencilla que permite visualizar los mensajes SIP enviados y recibidos a través de 2 posibles vistas.
- Presentación del cliente SIP y servidor.
- Uso de SCORM para integrar el cliente JavaScript en el laboratorio remoto.
- Desarrollo de un paquete SCORM, con el objetivo de que los alumnos conozcan el protocolo SIP y lo practiquen, mediante las prácticas planteadas.
- Personalización del WebLab basada en la identidad del usuario.
- Consecución de la integración WebLab-LMS.

El ámbito de funcionamiento del cliente puede ser tanto en red local como red pública, ya que se han realizado pruebas en un servidor público de la Universidad de Jaén y estas han sido exitosas, permitiendo que los alumnos puedan realizar el laboratorio remoto desde el lugar que quieran.

Todos los objetivos indicados al inicio del TFG han sido cumplidos y, además, las aplicaciones tienen algunas funcionalidades no contempladas inicialmente, como sonidos de recepción de mensajes y tono de llamada, entre otros.

Se han realizado pruebas con muchos usuarios, 10 equipos, y se han obtenido resultados exitosos, tanto en el envío de mensajes como en el establecimiento de llamadas. Un punto a destacar, aunque la multiconferencia no es un objetivo del presente TFG, sería que si varios usuarios llaman a un mismo destinatario, este solo podrá escuchar al último usuario al que descuelgue la llamada.

Gracias a la elaboración del presente TFG, el autor ha desarrollado una serie de habilidades y características:

- Mejora de los conocimientos de la programación *JavaScript*.
- Mejora de los conocimientos de la programación *HTML*.
- Aprendizaje del funcionamiento del entorno *Sublime Text*.
- Aprendizaje del funcionamiento y desarrollo de un laboratorio remoto basado en *SCORM*.
- Mejora de la capacidad de generar acciones a partir de indicaciones.

Se ha desarrollado un cliente SIP que se incluye en la página 4 del WebLab que presenta las siguientes características:

- Se puede utilizar para colaborar en la enseñanza y aprendizaje del protocolo SIP y la VoIP.
- Es fácil de usar.
- Se puede utilizar para comunicarse con otros usuarios, lo cual fomenta el trabajo en equipo.
- Ofrece 2 posibles vistas para visualizar los mensajes, lo que facilita la localización y posicionamiento de los mismos.
- Posibilidad de realizar llamadas de voz o video llamadas y envío de mensajes de texto.

Algunas observaciones que se deben indicar sobre el cliente son las siguientes:

- Si se introducen mal los datos de registro y este falla, se debe recargar el cliente para su correcto funcionamiento.
- Si se llama a varios usuarios, solo se escuchará al último que descuelgue la llamada.
- La página donde se incruste el cliente SIP debe ser considerada como segura por el navegador, ya que si no se considera como segura, el navegador *Google Chrome* no mostrará los permisos del micrófono y la cámara y, por lo tanto, no se podrá establecer la llamada. Se recomienda utilizar *mozilla firefox*.
- Debe existir en el equipo un micrófono y cámara (como mínimo el micrófono) bien configurados para que el equipo los detecte. Si no existen, el cliente SIP solo podrá enviar y recibir mensajes de texto.
- Algunas veces al poner la llamada en pause, esta se detiene y finaliza la llamada.

- Los usuarios que se quieran comunicar deben estar registrados en el mismo servidor proxy, ya que en la implementación del cliente, al realizar la llamada se toma el servidor de la URI SIP del usuario origen de la comunicación.

Como líneas de futuro se podrían incluir las siguientes:

- El correcto funcionamiento de la multiconferencia.
- Corrección de fallos mencionados anteriormente.

7. ANEXOS

7.1. Instalación y configuración del servidor SIP

7.1.1. Instalación del servidor

1º) Descargar el archivo “**PortSIP_PUCS.exe**”.

2º) Ejecutarlo como administrador. Para ello damos clic derecho y “**Ejecutar como administrador**”.

3º) Tras la pantalla de carga, seleccionamos como idioma “**English**” y le damos a “**OK**”, como se muestra en *Imagen 63*.

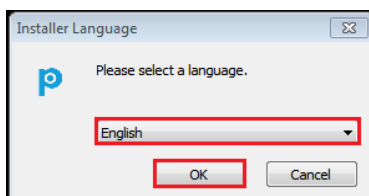


Imagen 63: Anexo I. Idioma

Fuente: Propia

4º) Le damos “**Next**”, **aceptamos los términos** y le damos a “**Next**” de nuevo, como se muestra en *Imagen 64*.

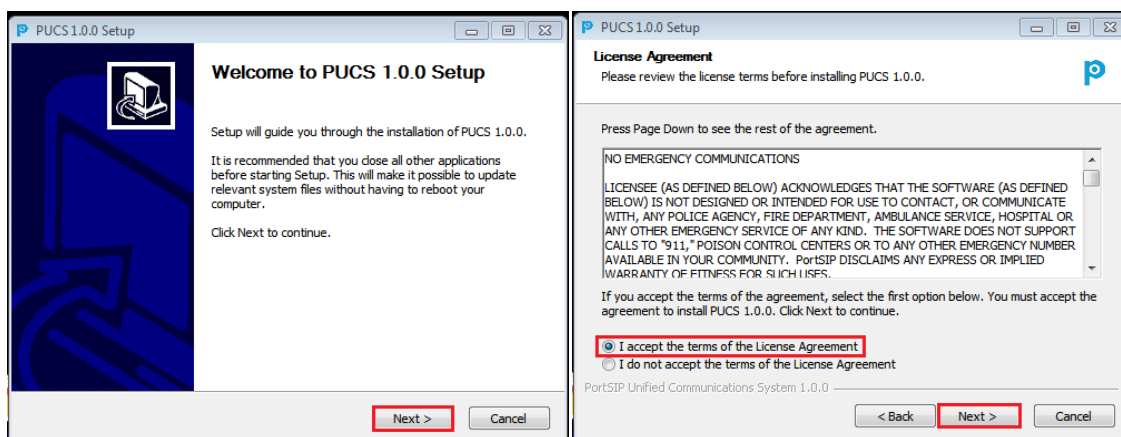


Imagen 64: Anexo I. Licencia

Fuente: Propia

5º) Seleccionamos la **ruta de instalación** (en nuestro caso lo dejaremos con el valor por defecto) y le damos a **“Next”**. Ahora nos pedirá el **puerto** en el que escuchará el servidor (en nuestro caso dejaremos su valor por defecto) y le daremos a **“Install”** para que comience la instalación, como se muestra en *Imagen 65* e *Imagen 66*.

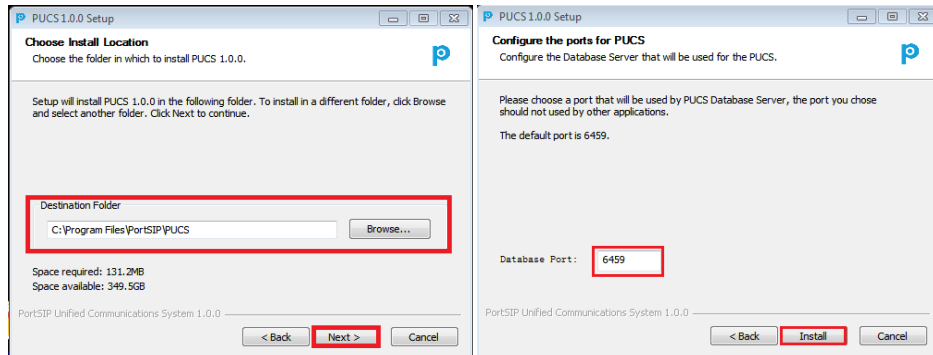


Imagen 65: Anexo I. Ubicación

Fuente: Propia

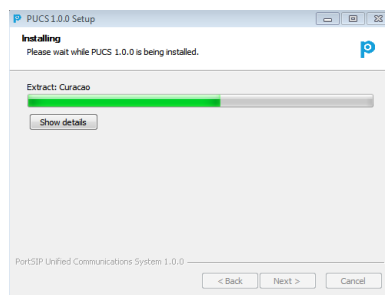


Imagen 66: Anexo I. Proceso de instalación

Fuente: Propia

6º) Una vez finalizada la instalación le damos a **“Finish”**, como se muestra en *Imagen 67*.

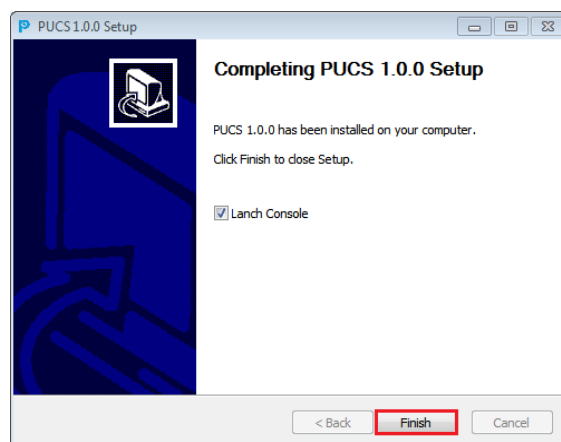


Imagen 67: Anexo I. Proceso de finalización de instalación

Fuente: Propia

Nota: Puedes dejar la opción **“LanchConsole”** seleccionada para que te lleve a la página de configuración directamente o puedes desmarcarlo para que no te lleve (recomiendo dejarlo marcado para continuar con la configuración del servidor).

7.1.2. Configuración del servidor

Una vez instalado el servidor solo queda configurarlo. A continuación te mostraré los pasos para configurar correctamente el servidor. Síguelos en orden y no te saltes ninguno a menos que se especifique en algún paso.

1º) Para acceder a la configuración del servidor tienes 3 opciones:

-Si marcaste la opción “LanchConsole” se te tiene que haber abierto la página de login.

-Si has instalado correctamente el servidor te tiene que haber aparecido en el escritorio un acceso directo llamado “PUCS Management Console”, por lo que al hacer doble clic en él te envía a la página de login.

-Puedes abrir el navegador y poner la siguiente dirección en la barra de direcciones:

“127.0.0.1:8888”

Los tres pasos te llevarán a la página de login en el navegador mostrada en *Imagen 68*.

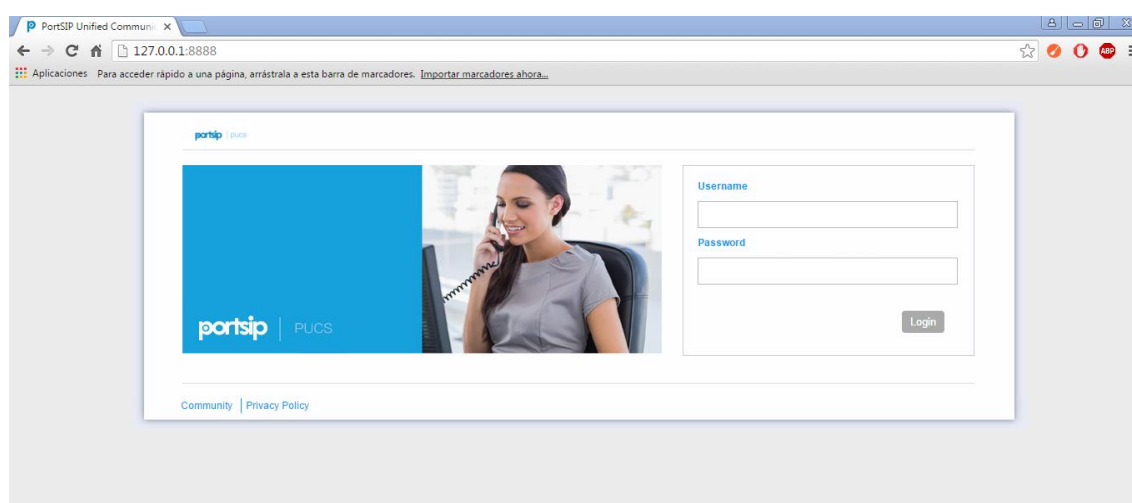


Imagen 68: Anexo I. Interfaz de configuración

Fuente: Propia

2º) Ahora debes acceder a la configuración. Para ello **introduce** los siguientes datos y luego dale a **“Login”**:

-Username: **admin**

-Password: **admin**

Nota: Si te has equivocado al introducir los datos te recomiendo que **recargues la página** ya que esta puede que no te coja bien los datos.

3º) Una vez hecho el login debes abrir la **consola de windows**, ya que ahora hay que especificar la dirección IP en la que va a estar el servidor y esa dirección será la de tu PC, como se muestra en *Imagen 69*. Para ver tu dirección IP introduce **“ipconfig”** en la consola si estás usando Windows o **“ifconfig”** si estás usando Linux.

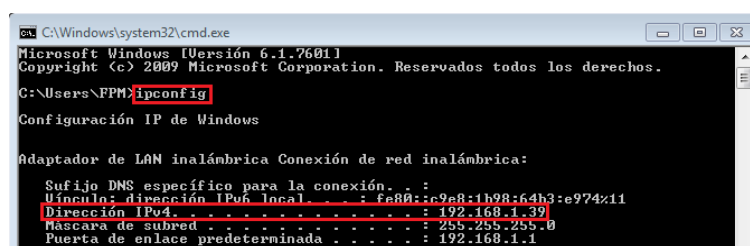


Imagen 69: Anexo I. Identificar dirección IP

Fuente: Propia

4º) Una vez que haces login y sabes la dirección IP de tu PC, ya estás listo para comenzar a configurar el servidor. Tras el login te aparecerá la siguiente pantalla:

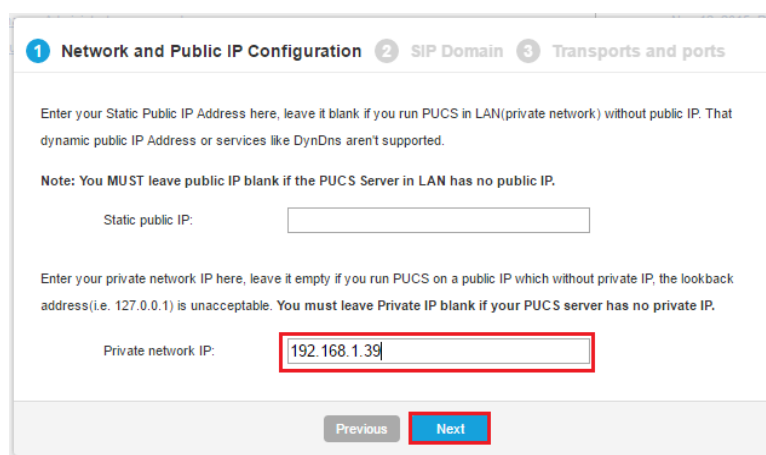


Imagen 70: Anexo I. Dirección o dominio del servidor 1

Fuente: Propia

Como ves en *Imagen 70* debes **introducir tu dirección IP** en el apartado **“Private Network IP”** ya que la red del laboratorio es privado y luego darle a **“Next”**. Ahora te sale la segunda pantalla, en la que tendrás que **introducir tu dirección IP** otra vez y darle a **“Next”**, como se muestra en *Imagen 71*.

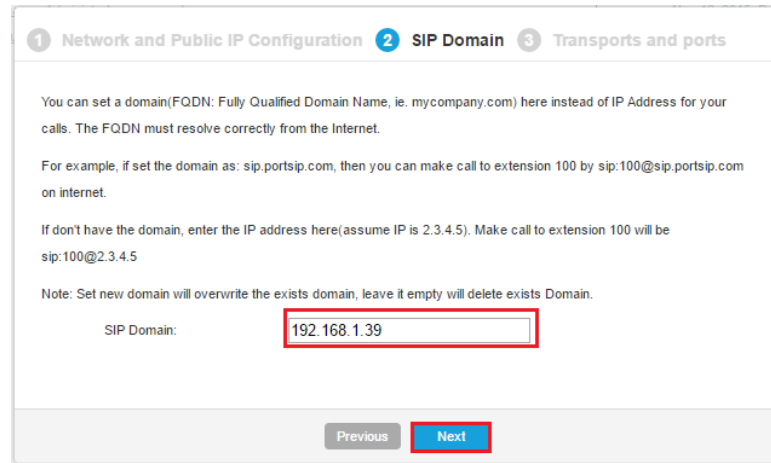


Imagen 71: Anexo I. Dirección o dominio del servidor 2

Fuente: Propia

Por último, en la tercera pantalla dejamos el valor por defecto, como podéis ver en *Imagen 72* y le damos a **“save”**:

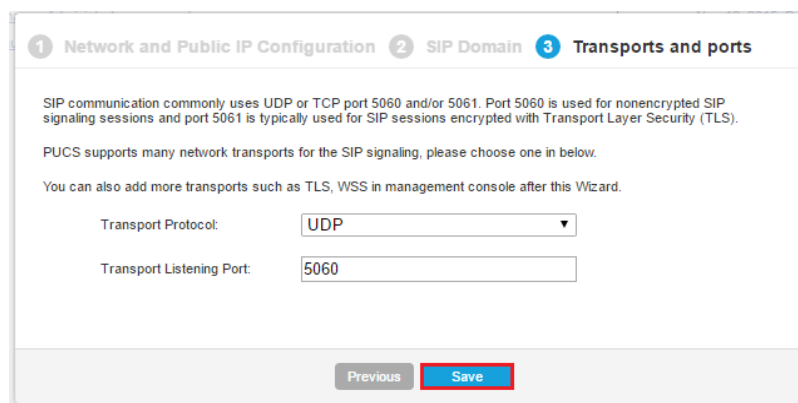


Imagen 72: Anexo I. Transporte inicial del servidor

Fuente: Propia

Finalmente, terminarás en la página que se muestra en *Imagen 73*.

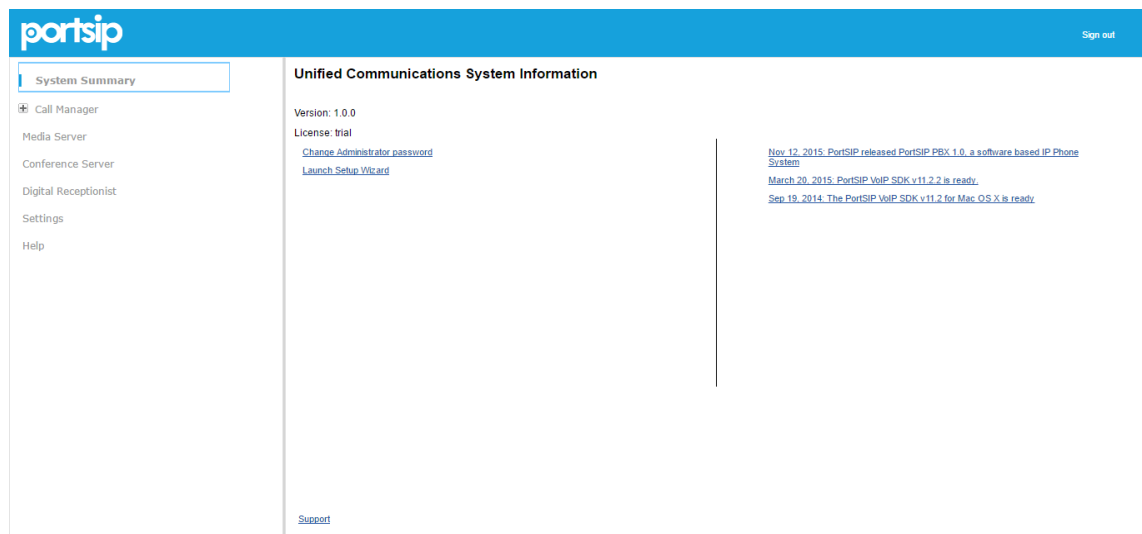


Imagen 73: Anexo I. Interfaz del servidor

Fuente: Propia

5º) Como transporte vamos a utilizar WebSocket (WS), por lo que ahora vamos a añadir este transporte. Para ello ir a **“Call Manager”->“Domain and transports”**, como se muestra en *Imagen 74*.

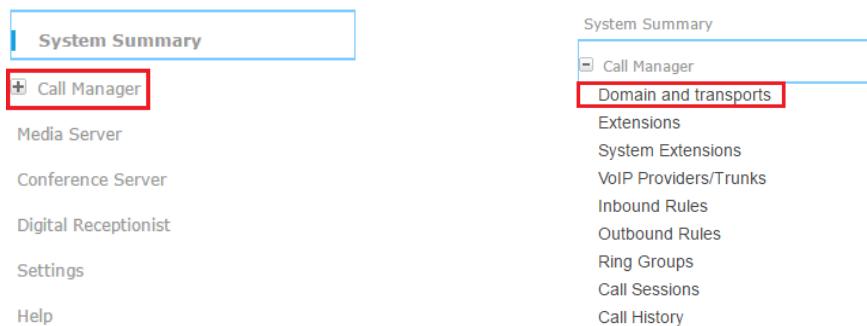


Imagen 74: Anexo I. Ruta del apartado transporte

Fuente: Propia

Y esto te llevará a la página mostrada en *Imagen 75*.

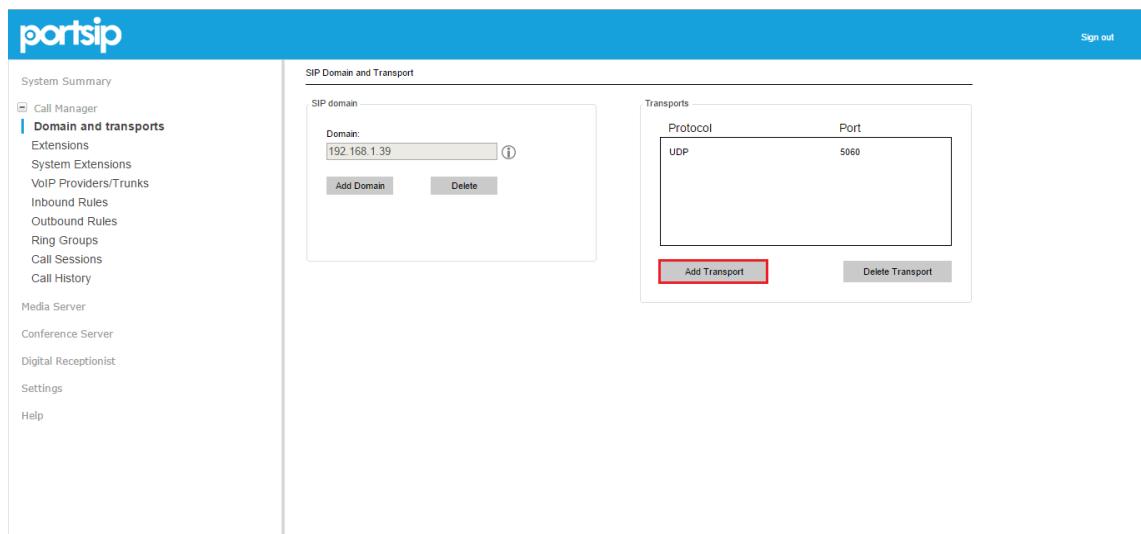


Imagen 75: Anexo I. Apartado transporte

Fuente: Propia

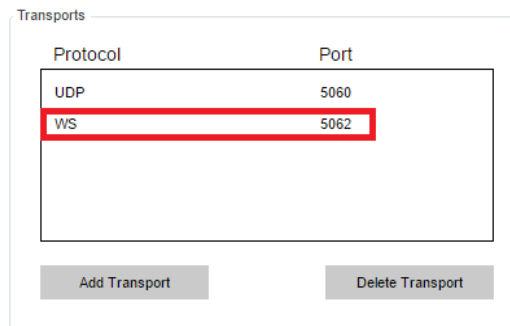
Hacemos clic en **“AddTransport”** y dejamos los datos como se ve en *Imagen 76* y le damos a **“Apply”**:

The 'Add Transport' dialog box is shown. It has two main sections. The 'Transport:' section includes 'Transport protocol:' with a dropdown menu showing 'WS' (highlighted with a red rectangle) and 'Transport Port:' with a text input showing '5062'. The 'TLS/WSS Settings:' section includes 'Certificate file:', 'Private key file:', 'Root certificate file:', 'Certificate Private Key Password:', and 'Client Verification:' (set to 'None'). Each field has a redacted input area and an information icon. At the bottom right are 'Cancel' and 'Apply' buttons, with the 'Apply' button highlighted by a red rectangle.

Imagen 76: Anexo I. Agregar transporte WS

Fuente: Propia

Para comprobar que se ha añadido correctamente te tiene que haber aparecido en “Transports” lo que se muestra en *Imagen 77*.



Protocol	Port
UDP	5060
WS	5062

Add Transport Delete Transport

Imagen 77: Anexo I. Comprobación de transportes

Fuente: Propia

6º) Ahora vamos a añadir a los usuarios. Para ello ir al apartado “**Extensions**”, como se muestra en *Imagen 78*.

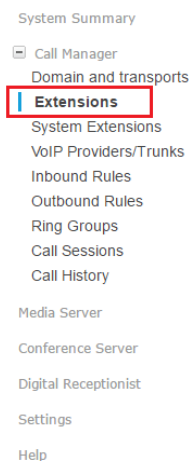
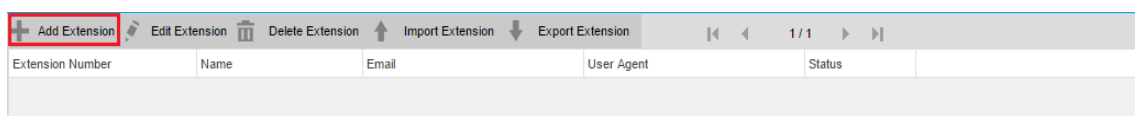


Imagen 78: Anexo I. Ruta al apartado de extensiones

Fuente: Propia

Una vez allí te aparecerá sin ningún usuario. Para añadir uno dale a “**Add Extensión**”, como se muestra en *Imagen 79*.



+ Add Extension Edit Extension Delete Extension Import Extension Export Extension 1 / 1

Extension Number	Name	Email	User Agent	Status	
------------------	------	-------	------------	--------	--

Imagen 79: Anexo I. Tabla de usuarios

Fuente: Propia

Y te saldrá la ventana de configuración mostrada en *Imagen 80*.

Edit Extension settings and click OK or Apply to save changes.

General Voice Mail Forwarding Rules

User Information

Configure user information below

Extension Number		*	i
Password		*	i
First Name			i
Last Name			i
Email			i

☐ Disable this extension [i](#)

☐ Disallow External Calls [i](#)

Cancel Apply

Imagen 80: Anexo I. Agregar usuario

Fuente: Propia

Solo debes añadir “**ExtensionNumber**” y “**Password**” y darle a “**Apply**”, como se muestra en *Imagen 80*. Por ejemplo:

-ExtensionNumber: 101

-Password: 101

Como se muestra en *Imagen 81*, ahora ese usuario aparece en la pantalla “Extensions”.

portsip Sign out

System Summary

Call Manager

Domain and transports

Extensions

System Extensions

VoIP Providers/Trunks

Inbound Rules

Outbound Rules

Ring Groups

Call Sessions

Call History

Media Server

Conference Server

Digital Receptionist

Settings

Help

+ Add Extension Edit Extension Delete Extension Import Extension Export Extension

Extension Number	Name	Email	User Agent	Status
101				

Imagen 81: Anexo I. Comprobar usuario

Fuente: Propia

Nota: Añade tantos usuarios como necesites.

7º) Tras añadir a los usuarios ya tendrás el servidor configurado.

7.2. Pasos previos y requisitos para la ejecución del laboratorio remoto

El Laboratorio Web está incluido en un formato SCORM en forma de una aplicación JavaScript. Para poder utilizar la aplicación se debe tener un micrófono (la WebCam es opcional) conectado al PC y que funcione correctamente, ya que si no se detecta ningún micrófono, cuando se quiera establecer una llamada no se podrá. Además, el navegador debe reconocer la página donde está el cliente como segura (ver apartado 7.2.2. *Navegador de ejecución*).

No se necesita añadir ningún permiso extra a los que ya vienen configurados por defecto. Solo será necesario dar permiso de micrófono y cámara al hacer la llamada.

7.2.1. Comprobación de micrófono y cámara

En Windows, para comprobar si el micrófono está bien configurado hay que ir a "Panel de control" y acceder a "Sonido". En él buscamos el micrófono y comprobamos que al hablar sube el color verde de la barra de la derecha, como puede verse en *Imagen 82*.

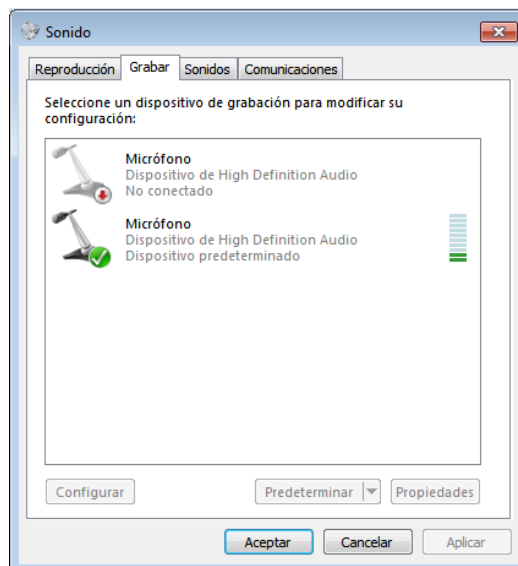


Imagen 82: Anexo II. Comprobar micrófono

Fuente: Propia

En Windows, para comprobar si la cámara está bien configurada podemos utilizar cualquier aplicación que utilice la webcam y ver si funciona.

7.2.2. Navegador de ejecución

La aplicación requiere que el navegador acceda al micrófono y cámara para poder realizar llamadas a otro usuario. *Google Chrome* no reconoce el dominio "ev.ujaen.es" como seguro, por lo que el Laboratorio Web se podrá ejecutar y todo funcionará perfectamente, excepto al llamar, porque *Google Chrome* no nos mostrará la opción de aceptar los permisos del micrófono y, por lo tanto, no se podrá establecer la llamada. Para poder ejecutar llamadas en el Laboratorio Web utilizando *Google Chrome* hay que ejecutar el siguiente comando en consola:

START chrome.exe --unsafely-treat-insecure-origin-as-secure="http://ev.ujaen.es" --user-data-dir=/test/only/profile/dir http://ev.ujaen.es

Con este comando se le está diciendo al sistema que abra el navegador *Google Chrome* y con los dos parámetros siguientes se le está indicando que se abra en la web "http://ev.ujaen.es" y que la tome como segura. En esta ventana que se abre si se podrán realizar llamadas.

Para facilitar la ejecución del comando al usuario se puede utilizar un archivo con extensión ".bat" para que solo tenga que hacer doble clic sobre él.

Para crear un archivo con extensión ".bat" se debe copiar el comando indicado anteriormente y pegarlo en un archivo de texto. Una vez hecho esto se guardan los cambios y en lugar de poner extensión ".txt" ponemos ".bat".

En *mozilla firefox* no es necesario realizar nada ya que actualmente si muestra la opción de aceptar los permisos del micrófono.

7.3. Manual de mantenimiento

Este manual se presenta con el objetivo de facilitar la posible modificación futura del código implementado. Se deberán realizar los siguientes pasos:

- Instalación de un editor de texto. Se recomienda Sublime Text.
- Una vez instalado, a continuación se indica donde se localizan los archivos importantes:
 - index.html: En la ruta raíz del proyecto (./), donde además se encuentran las diferentes páginas que componen el WebLab: "page1.html", "page2.html", "page3.html", "page4.html" y "page5.html". También el archivo "imsmanifest.xml".
 - Imágenes: En la ruta ./images.
 - Archivos js: En la ruta ./js.
 - Archivos css: En la ruta ./css.
 - Archivos de fuente de texto: En la ruta ./fonts.
 - Archivos de audio: En la ruta ./sounds.
 - Otros archivos: En la ruta ./documents.
 - RTE.js: En la ruta ./_ejs_library/scripts.
- Si se quiere añadir alguna función al cliente:
 - Crear un archivo con extensión ".js" en la carpeta js y otro con ".css" en la carpeta css.
 - Si se incluyen archivos de imagen o de audio se deben incluir en la carpeta correspondiente, para seguir la estructura del proyecto.
 - Añadir un elemento en el código del archivo "index.html" e incluir los dos archivos anteriores, con el siguiente código:


```
<script src="[tu ruta sin corchetes]" type="text/javascript"></script>
```



```
<link rel="stylesheet" href="[tu ruta sin corchetes]" type="text/css"></link>
```
 - Añadir la funcionalidad y el estilo que desees en los archivos creados. Para comunicaciones con el LMS se deben utilizar las funciones del archivo "RTE.js".
- Una vez añadida la nueva funcionalidad, se debe comprimir todo el proyecto en un zip. Para ello vamos a la carpeta raíz, seleccionamos todos los archivos y carpetas y hacemos clic derecho y seleccionamos "enviar a -> carpeta comprimida (en zip)". Esto creará un archivo que será el que tenemos que subir a la página del WebLab para crear un SCORM nuevo.

- Si se quiere añadir una nueva página al WebLab o eliminarla, se debe modificar el archivo *"imsmanifest.xml"*.

- Para añadir una nueva página hay que copiar el siguiente código, modificando lo que está entre corchetes ([]):

```
<item identifier="[nombre]" identifierref="[id]" isvisible= "true">
<title>[titulo]</title>
</item>
```

```
<resource identifier="[id]" href="[ruta del
html]" adlcp:scormType="sco" type="webcontent">
<file href="[ruta del html]"/>
<dependency identifierref="API-DEP"/>
<dependency identifierref="API-DEP1"/>
</resource>
```

- Para eliminar una página hay que eliminar el código anterior del archivo *"imsmanifest.xml"*.

7.4. Acrónimos

TFG: Trabajo Fin de Grado.

WebLab: Laboratorio Web.

SIP: Session Initiation Protocol.

SDP: Session Description Protocol.

RTP: Real-time Transport Protocol.

RTCP: Real Time Control Protocol.

IETF: Internet Engineering Task Force.

VoIP: Voice over IP.

WS: WebSocket.

HTML: HyperText Markup Language.

HTTP: Hypertext Transfer Protocol.

JS: JavaScript.

PC: Personal Computer.

SCORM: Shareable Content Object Reference Model.

SCOs: Shareable Content Objects.

LMS: Learning Management System.

RTE: Run-Time Environment.

URI: Uniform Resource Identifier.

UA: User Agent.

UAC: User Agent Client.

UAS: User Agent Server.

CAM: Content Agregation Model.

7.5. Índice de figuras

Imagen 1: Secuencia del SCORM	4
Imagen 2: Comunicaciones de las prácticas	4
Imagen 3: Torre de Protocolos.....	8
Imagen 4: Formato de un mensaje SIP	9
Imagen 5: Formato de la línea de inicio en mensajes de solicitud.....	10
Imagen 6: Métodos SIP nuevos	10
Imagen 7: Formato de la línea de inicio en mensajes de respuesta	11
Imagen 8: Códigos de estado en mensajes de respuesta	11
Imagen 9: Parámetros SDP	13
Imagen 10: Interfaz inicial (Formulario)	22
Imagen 11: Interfaz inicial (Cliente)	22
Imagen 12: Interfaz final (Formulario)	23
Imagen 13: Interfaz final (Cliente)	23
Imagen 14: Botón idioma	24
Imagen 15: Teléfono	25
Imagen 16: Ventana de chat	26
Imagen 17: Ventana de cámaras	27
Imagen 18: Consola (Vista completa)	28
Imagen 19: Consola (Vista resumen)	28
Imagen 20: Intercambio de mensajes SIP (Registro)	34
Imagen 21: Ejemplo REGISTER 1	34
Imagen 22: Ejemplo REGISTER 2	34
Imagen 23: Ejemplo REGISTER 3	35
Imagen 24: Ejemplo REGISTER 4	35
Imagen 25: Intercambio de mensajes SIP (Llamada)	36
Imagen 26: Intercambio de mensajes SIP (Chat)	37
Imagen 27: Ejemplo MESSAGE 1	37
Imagen 28: Ejemplo MESSAGE 2	37
Imagen 29: Página 1. INTRODUCCIÓN	39
Imagen 30: Página 2. TEORÍA	41
Imagen 31: Página 2. TEST 1	42
Imagen 32: Página 2. TEST 2	43
Imagen 33: Página 2. TEST 3	43
Imagen 34: Página 2. TEST 4	44
Imagen 35: Página 2. TEST 5	44

Imagen 36: Página 3. SERVIDOR (Instalación)	45
Imagen 37: Página 3. SERVIDOR (Configuración)	46
Imagen 38: Página 4. CLIENTE (Manual de usuario).....	47
Imagen 39: Página 4. TEST 1.....	48
Imagen 40: Página 4. TEST 2.....	48
Imagen 41: Página 4. TEST 3.....	49
Imagen 42: Página 4. TEST 4.....	49
Imagen 43: Página 4. TEST 5.....	50
Imagen 44: Página 4. TEST 6.....	50
Imagen 45: Página 4. TEST 7.....	51
Imagen 46: Página 4. TEST 8.....	51
Imagen 47: Página 4. TEST 9.....	52
Imagen 48: Página 4. TEST 10.....	52
Imagen 49: Página 4. TEST 11.....	53
Imagen 50: Página 4. TEST 12.....	53
Imagen 51: Página 5. EVALUACIÓN FINAL	55
Imagen 52: Página 5. TEST 1.....	55
Imagen 53: Página 5. TEST 2.....	56
Imagen 54: Página 5. TEST 3.....	56
Imagen 55: Página 5. TEST 4.....	57
Imagen 56: Página 5. TEST 5.....	57
Imagen 57: Página 5. TEST 6.....	58
Imagen 58: Página 5. TEST 7.....	58
Imagen 59: Página 5. TEST 8.....	58
Imagen 60: Página 5. TEST 9.....	59
Imagen 61: Página 5. TEST 10.....	59
Imagen 62: Flujo grama.....	60
Imagen 63: Anexo I. Idioma	64
Imagen 64: Anexo I. Licencia.....	64
Imagen 65: Anexo I. Ubicación	65
Imagen 66: Anexo I. Proceso de instalación	65
Imagen 67: Anexo I. Proceso de finalización de instalación	65
Imagen 68: Anexo I. Interfaz de configuración	66
Imagen 69: Anexo I. Identificar dirección IP	67
Imagen 70: Anexo I. Dirección o dominio del servidor 1.....	67
Imagen 71: Anexo I. Dirección o dominio del servidor 2.....	68
Imagen 72: Anexo I. Transporte inicial del servidor.....	68

Imagen 73: Anexo I. Interfaz del servidor.....	69
Imagen 74: Anexo I. Ruta del apartado transporte.....	69
Imagen 75: Anexo I. Apartado transporte.....	70
Imagen 76: Anexo I. Agregar transporte WS.....	70
Imagen 77: Anexo I. Comprobación de transportes	71
Imagen 78: Anexo I. Ruta al apartado de extensiones	71
Imagen 79: Anexo I. Tabla de usuarios.....	71
Imagen 80: Anexo I. Agregar usuario.....	72
Imagen 81: Anexo I. Comprobar usuario.....	72
Imagen 82: Anexo II. Comprobar micrófono.....	73

8. REFERENCIAS BIBLIOGRÁFICAS

RFC3261. (2002). Obtenido de <https://tools.ietf.org/html/rfc3261>

RFC4566. (2006). Obtenido de <https://tools.ietf.org/html/rfc4566>

Ruano Ruano. Apuntes de la asignatura de Sistemas de Telefonía. Tema 5: Arquitectura de VoIP SIP. Recuperado el 15 de Noviembre de 2015

Julie C. Meloni. (2012). Programación HTML5, CSS3 Y JavaScript, ANAYA.

Ostyn, C. (2005). In the Eye of the SCORM. An introduction to SCORM 2004 for Content Developers. Kirkland: Ostyn Consulting. Recuperado el 20 de Noviembre de 2015, de http://www.ostyn.com/standards/docs/Eye_Of_The_SCORM_draft.pdf

Advanced Distributed Learning Initiative. SCORM 2004 4th Edition. (2009). Recuperado el 20 de Noviembre de 2015, de <http://www.adlnet.gov/scorm/>

[103] Advanced Distributed Learning (ADL), "Content Aggregation Model (CAM) SCORM 2004 4th Ed. v1.1." p. 234, 2009.

[104] Advanced Distributed Learning (ADL), "Run-Time Environment (RTE) SCORM 2004 4th Ed. v1.1." p. 244, 2009.

[105] Advanced Distributed Learning (ADL), "Sequencing and Navigation (SN) SCORM 2004 4th Ed. v1.1." p. 234, 2009.

"The Rise and Fall of the Java Applet: Creative Coding's Awkward Little Square" by Michael Byrne, February 2, 2016, <http://motherboard.vice.com/read/a-brief-history-of-the-java-applet>

[107] Dalibor, "Moving to a Plugin-Free Web," Topic-Oracle, 2016. [Online]. Available: https://blogs.oracle.com/java-platform-group/entry/moving_to_a_plugin_free. [Accessed: 20-Jul-2016].