

Sistema de telemedicina OneHealth

Grado en Ingeniería Informática



Trabajo Fin de Grado

Autor:

Beatriz Asensi Prieto

Tutor/es:

Sergio Orts Escolano

Septiembre 2018



**MOTIVACIÓN**

Este proyecto se puso en marcha gracias a la motivación de hacer algún sistema que tuviese como punto fuerte la seguridad. Un sistema de telecomunicación médico-paciente requiere mucha seguridad ya que los datos que se manejan son muy sensibles, tales como historiales médicos, datos personales tanto de paciente como médico, etc. Además de esto, con esta idea damos una gran facilidad a la hora de comunicación en el ámbito de la medicina, ya que muchas personas no tienen la facilidad, ya sea económica, de movilidad o familiar para poder ir presencialmente a un centro de salud.

Personalmente, este proyecto pienso que nos ha servido para aprender muchos conceptos nuevos de informática, descubrir tecnologías y enfrentarnos a lo desconocido.

**AGRADECIMIENTOS**

**DEDICATORIA**

**ÍNDICE DE CONTENIDO**

**ÍNDICE DE FIGURAS**

**ÍNDICE DE TABLAS**

1. **INTRODUCCIÓN**

La videoconferencia en sus inicios, era una herramienta o sistema que solo unos pocos podían disfrutar, ya que esta tenía un coste muy elevado, empresas empoderadas con grandes presupuestos y conexión a Internet media podían permitirse ese lujo.

En la actualidad, la videoconferencia de alta calidad está al alcance de casi todo el mundo. Las mejoras en el acceso a Internet, las diferentes tecnologías, software y hardware, que nos permiten una mejor experiencia y facilidad para el acceso a Internet y a los diversos sistemas.

La videoconferencia ayuda a diversos sectores de la sociedad como a la educación, sanidad, informática, soporte y servicio, reuniones de negocios, etc… Esta herramienta hace que las empresas tengan menos costes a costa de un servicio barato como es esta tecnología.

En nuestro caso, vamos a aplicar esta tecnología a la medicina. La telemedicina o teleconsulta está creciendo cada día más, según la consultora Mordor Intelligence prevé que se facture 66.000 millones de dólares en 2021, con un desarrollo más elevado en el ámbito de la conectividad. La telemedicina tiene grandes beneficios tanto para las empresas como para los pacientes, ya que se puede dar atención sanitaria especializada a personas que viven en zonas lejanas a un centro de salud, ahorrar en el gasto de desplazamientos de los vehículos sanitarios, urgencias y personal médico. Gracias a estos datos, muchos especialistas y empresas del sector de la medicina confíen en la telemedicina.

1. **OBJETIVOS**
   1. **OBJETIVOS ESPECIFICOS**

El objetivo principal de este proyecto de fin de grado es la realización de un sistema de telecomunicación médico-paciente dotado de seguridad para entornos web y aplicaciones móviles. Además de esto tendremos un apartado de visualización de anteriores videoconferencias, las cuales estarán almacenadas y cifradas para una mayor seguridad. Y finalmente, integraremos este sistema en una plataforma eHealth, la cual gestionará la parte de citas, historiales médicos, consultas, etc.

* 1. **RELACION CON ASIGNATURAS**

En este apartado enumeraremos las asignaturas impartidas durante el Grado de Ingeniería Informática, las cuales han tenido una aportación directa o indirecta al desarrollo de este sistema.

**Fundamentos de las bases de datos y diseño de base de datos**

Estas dos asignaturas nos dieron los conocimientos básicos para la creación de una base de datos desde cero, viendo relaciones sencillas y no tan sencillas, cardinalidades y buenas prácticas para tener una base de datos robusta.

**Análisis y especificación de sistemas software**

Esta asignatura nos proporcionó material y herramientas para analizar una aplicación o sistema de cero y de esta manera sacar los requisitos principales y con ellos hacer diversos diagramas para visualizar mejor los objetivos y funcionalidades del sistema.

**Seguridad en el diseño de software**

En esta asignatura nos enseñaron las pautas básicas para proteger una aplicación o sistema, diferentes cifrados, como resolver vulnerabilidades y las diferentes catástrofes que han pasado a lo largo de la historia por no tener un sistema seguro.

**Aplicaciones distribuidas en internet**

En esta asignatura aprendimos tecnologías como nodejs y react, que en la actualidad son de los lenguajes más usados, además de cómo hacer un servidor rest con nodejs.

1. **ESTADO DEL ARTE**
   1. **SISTEMAS SIMILARES**

[**Médicos y pacientes**](http://www.medicosypacientes.com/articulo/un-nuevo-portal-permitira-consultar-la-historia-clinica-y-contactar-con-el-medico)

Esto es una página web para la consulta clínica y comunicarse con el médico de forma telemática. Es una plataforma web, la cual tiene diversas características:

* Inicio de sesión con clave o certificado digital.
* Consulta de tarjeta sanitaria.
* Consulta de centro de salud (mapa, números, etc).
* Hospital de referencia.
* Médico y enfermero de cabecera.
* Citas electrónicas.
* Ver historial clínico.
* Datos de facturación farmacéutica.
* Consulta online al médico de determinadas dudas sobre la asistencia sanitaria.
* Ver información del paciente.

[**Policlínica IUMET**](http://www.iumet.es/servicios-telemedicina.php)

El instituto Universitario de Medicina Telemática IUMET surge para dar a los Centros de Reconocimiento de Conductores (CRC). Este sistema permite hacer consultas instantáneas de exploración psicofísica a los pacientes, dar apoyo formativo para ampliar las habilidades de los profesionales del Centro de Reconocimiento, asesoraminento y promueve la investigación en el ámbito de la seguridad vial. Además tiene un apartado de teleoftalmología.

[**Vida**](https://play.google.com/store/apps/details?id=es.vida.android&referrer=adjust_reftag%3DceZwEgCrXcwSj%26utm_source%3DVIDA.es%26utm_campaign%3Dfirstpage-app-promo)

Este es una aplicación para plataformas móvil o Tablet y está disponible para los sistemas operativos Android e IOS. Esta aplicación tiene diversas características:

* Video consultas para:
  + Preocupaciones generales de salud
  + Alergias
  + Asma
  + Infecciones sinusales
  + Herpes labial
  + Resfriados
  + Infección del tracto urinario
* Reserva de citas para la video consulta
* El médico podrá recetar al paciente

Este sistema es de pago, el cual tiene un precio de 20€ la consulta.

[**Sanitas**](https://www.sanitas.es/landings/seguros/cambiate-blua/noclientes/ownmedia/)

Sanitas es un seguro médico muy conocido, el cual tiene un servicio de videoconsulta llamado blua. Este sistema web también tiene lo básico de una aplicación eHealth.

* Video consulta médica con especialistas
* Programas de asesoramiento por videoconsulta
* Información de centros
* Reserva de citas
* Consulta de pruebas

Esta aplicación tiene unas cuotas mensuales o trimestrales, las cuales todas contienen el servicio de videoconsulta.

[**HealthTap**](https://www.healthtap.com/)

Este es un sistema para plataformas web y móvil con las siguientes características:

* Información inmediata de conocimientos médicos
* Respuestas inmediatas de los especialistas
* Consejos
* Noticias relevantes
* Valoraciones de medicamentos
* Consultas por chat de texto y video
* Asesoramiento de especialistas
* Tratamientos y recetas
* Consultas de laboratorio y resultados.
* Listados a medida
* Recordatorios médicos

Es una aplicación gratuita.

* 1. **WEBRTC**

WebRTC, también conocido como Web Real-Time Communications, es un proyecto de código abierto, promovido por Google, Mozilla y otros, que permite comunicaciones en tiempo real sin plug-ins a través de una API JavaScript. Facilita las aplicaciones de llamada de voz, chat de video y compartimiento de archivos entre navegadores. El códec soportado actualmente para WebRTC es VP8. WebRTC utiliza un servidor denominado Servidor de Conferencias Web que en conjunto con un Servidor STUN es requerido para proveer la página inicial y sincronizar las conexiones entre los dos nodos WebRTC.

La seguridad y el cifrado en WebRTC no son opcionales, ya que esta tecnología tiene características integradas nativas que se encargan de los problemas de seguridad. Además WebRTC ofrece cifrado punto a punto entre extremos sobre cualquier servidor garantizando las comunicaciones en tiempo real, privadas y seguras.

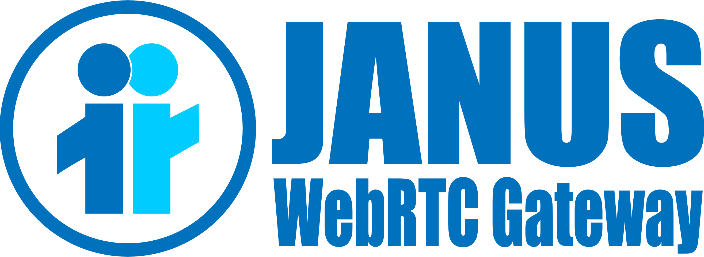
WebRTC requiere del usuario para permitir explícitamente el acceso a su cámara y micrófono. Esto garantiza que el usuario sea consciente de que su cámara y micrófono se encenderán. Cuando el usuario permite el acceso, un punto rojo aparecerá en esa solapa, proporcionando una clara indicación, de que la solapa tiene acceso a los medios.

Para que WebRTC transfiera datos en tiempo real, los datos primero se cifran utilizando el método DTLS (Datagram Transport Layer Security). Este es un protocolo integrado en todos los navegadores soportados por WebRTC desde el principio (Chrome, Firefox y Opera).  En una conexión DTLS cifrada, el espionaje y la manipulación de información no pueden tener lugar

Aparte de DTLS, WebRTC también encripta los datos de vídeo y audio a través del método SRTP (Secure Real-Time Protocol) garantizando que las comunicaciones IP – su voz y el tráfico de vídeo – no puedan ser escuchados o vistos por terceros no autorizados.

* 1. **FRAMEWORKS**

**Janus**

****Janus es un software creado con el propósito de ser una puerta de enlace. Este framework permite una comunicación WebRTC entre navegadores, pudiendo intercambiar mensajes JSON y transmitir RTP/RTCP. Toda la lógica de este sistema está implementada en la parte del servidor. Janus usarse para implementar aplicaciones como priebas de eco, conferencias, grabadoras, pasarelas SIP y similares.

La licencia de este sistema es gratuita y está distribuido bajo los términos de la Licencia Pública General GNU versión 3. Este sistema está creado por la empresa [Meetecho](http://www.meetecho.com/en/).

Demo del programa: <https://janus.conf.meetecho.com/videocalltest.html>

**Licode**

Licode es una plataforma de comunicaciones WebRTC de código abierto creada por [Lynckia](http://lynckia.com/).

****Esta plataforma está basada en las tecnologías de WebRTC, es 100% compatible con las últimas versiones estables de Google Chrome. Al estar basado en WebRTC cualquier usuario puede utilizar esta herramienta en sus navegadores, sin la necesidad de instalar ningún plugin ni extensión. Permite incluir sala de videoconferencia, la transmisión, la grabación de cualquier cosa multimedia.

Para poder trabajar con este framework una de las restricciones es que tenemos que tener Ubuntu 14.04 LTS o Mac OS X.

La licencia de este software está bajo los términos de [CC BY 3.0](https://creativecommons.org/licenses/by/3.0/).

**Medooze**

****Medooze es un servidor multimedia que te permite recibir y enviar videos mediante la tecnología de WebRTC. Este servidor esta soportado por los sistemas operativos Linux, Mac OS X y Raspberry Pi.

Este sistema está creado por [Medooze](http://www.medooze.com/).

Licencia [MIT](https://opensource.org/licenses/MIT).

**MediaSoup**

****MediaSoup tiene como objetivo ser un SFU WebRTC, el cual recibe trasmisiones de audio y video de los participantes de una sala de conferencias y retransmite a todos los demás. Ser un módulo nodejs en el servidor, ser un SDK en la parte de cliente, ser minimalista, que todos los navegadores y ORTC existentes sean compatibles y que no tenga un protocolo específico.

Este software está hecho por Iñaki Baz Castillo y José Luis Millán.

Demo: <https://demo.mediasoup.org/?roomId=xoqaerz3>

Licencia [ISC](https://opensource.org/licenses/ISC).

**Kurento**

1. **MATERIALES Y MÉTODOS**
   1. **HERRAMIENTAS SOFTWARE**

Para la realización de este proyecto hemos utilizado diferentes herramientas software para crear la documentación y almacenar los diferentes archivos.

* + 1. **DOCUMENTACIÓN**

Para la realización de la documentación hemos utilizado:

* Microsoft Word 2013

Microsoft Word es un programa informático orientado al procesamiento de textos. Este fue creado por la empresa Micorsoft y está incluido en la suite informática Microsoft Office.

* + 1. **DISEÑO**
* Draw.io

Draw.io es un sistema de código abierto para crear diagrama de todo tipo. Este está creado por Gaudenz Alder y David Benson. En nuestro caso hemos lo hemos utilizado para la realización de diagramas de casos de uso, da clase, E-R y mockups.

* + 1. **DESARROLLO**
* Sublime

Sublime Text es un editor de texto y editor de código fuente, el cual está escrito en C++ y Python para los plugins que tiene integrados. Este programa no es de código abierto y se debe de adquirir una licencia para su uso continuado, aunque existe una versión de evaluación que no tiene fecha de caducidad que es la que nosotros estamos utilizando.

* Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para distintos sistemas operativos. El programa es de código abierto y gratuito, aunque la descarga oficial está bajo software propietario.

* GitHub

GitHub es un sistema de control de versiones, se utiliza normalmente para la creación y almacenamiento de código fuente de programas. El software está escrito por Ruby on Rails, pero actualmente es propiedad de Microsoft.

* Ubuntu 16.06

Ubuntu es un sistema operativo de código abierto, distribuido por Linux y basado en una arquitectura Debian. Fue creado por Mark Shuttleworth, director de la compañía Canonical. Ubuntu es distribuido de manera gratuita y es financiado mediante diversos servicios vinculados al sistema y mediante servicio técnico.

* Oracle VM VirtualBox

Oracle VM VirtualBox es un software de virtualización para arquitecturas x86/amd64. Es un producto multiplataforma, el cual permite ser ejecutado en varios sistemas operativos (Mac OS, Windows, Linux u Oracle Solaris). Pertenece a Oracle y tiene la versión privativa y la versión Open Source.

* Node.js

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado por Tyan Lienhart Dahl. Está bajo una licencia MIT.

* React

React js es una líbreria de Javascript creada por Facebook, el cual permite crear interfaces de usuario, aplicaciones SPA (single page application) más eficientes. Está librería funciona tanto en el lado de cliente como en el de servidor. Utiliza el patrón MVC (Modelo-Vista-Controlador). Fue creado por Jordan Walke y está bajo una licencia MIT.

* Google Keep

Google Keep es una aplicación para organizar de manera sencilla información a través de tarjetas. Está disponible para Android y en Google Drive como aplicación web.Ha sido desarrollada por Google Inc.

* Google Drive

Google Drive es una aplicación de almacenamiento de archivos. Está disponible a través de la web y aplicaciones Android e IOS. Tiene 15GB gratuitos para almacenar diferentes archivos ampliables mediante diferentes planes de pago. Fue creado Google Inc bajo la licencia EULA.

* 1. **HERRAMIENTAS HARDWARE**

Para la realización de este Trabajo de Fin de Grado hemos tenido que utilizar dos dispositivos muy importantes:

* Cámara

Hemos usada la cámara integrada en el ordenador portátil, BisonCam, NB Pro.

* Micrófono

Hemos usado el micrófono integrado del ordenador portátil, Realtek High Definition Audio.

1. **ESPECIFICACIÓN DE REQUISITOS**
   1. **CASOS DE USO**

A continuación definiremos los diversos casos de uso de nuestra aplicación. Los casos de uso nos ayudan a expresar el comportamiento deseado del sistema, además podemos establecer las necesidades según la necesidad de cada usuario (actor).

* Ver pacientes disponibles para videollamada

|  |  |
| --- | --- |
| Caso de Uso: Ver pacientes disponibles para videollamada | |
| Actor: Médico | |
| Curso Normal | **Alternativo** |
| 1. El médico inicia sesión en el sistema. | * 1. Si el médico no está registrado, se le informará que debe de hacerlo. |
| 1. El médico va al apartado de videollamada. |  |
| 1. El médico buscará por el DNI del paciente y tendrá la lista de los conectados. |  |
| 1. El médico llamará al paciente con el que quiere hablar. |  |
| 1. El sistema avisará al paciente |  |
| 1. El paciente aceptará o denegará la llamada iniciada. |  |

* Ver videollamada anterior

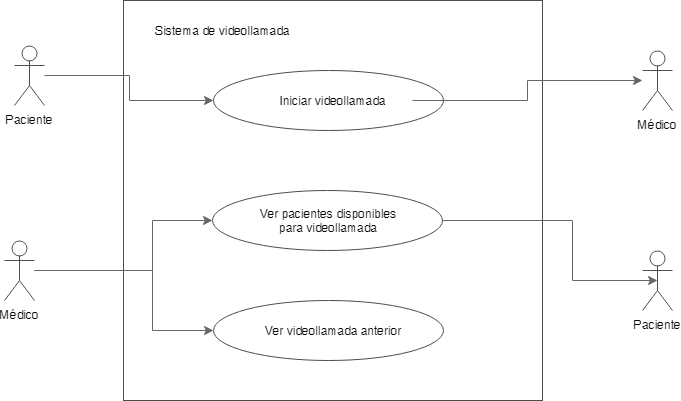
|  |  |
| --- | --- |
| Caso de Uso: Ver videollamada anterior | |
| Actor: Médico |  |
| Curso Normal | **Alternativo** |
| 1. El médico iniciará sesión en el sistema. | * 1. Si el médico no está registrado, se le informará que debe de hacerlo. |
| 1. El médico irá al apartado de historiales de paciente. |  |
| 1. El médico buscará la fecha cuando ocurrió la videollamada. |  |
| 1. El médico podrá visualizar la videollamada en el sistema. |  |

* Iniciar videollamada

|  |  |
| --- | --- |
| Caso de Uso: Iniciar videollamada | |
| Actor: Paciente | |
| Curso Normal | **Alternativo** |
| 1. El paciente inicia sesión en el sistema. | * 1. Si el paciente no está registrado, se le informará que debe de hacerlo. |
| 1. El paciente va al apartado de videollamada. |  |
| 1. El paciente introducirá el código de su cita. |  |
| 1. El paciente llamará al médico con el que tiene la cita |  |
| 1. El sistema avisará al médico. |  |
| 1. El médico aceptará o denegará la llamada iniciada. |  |

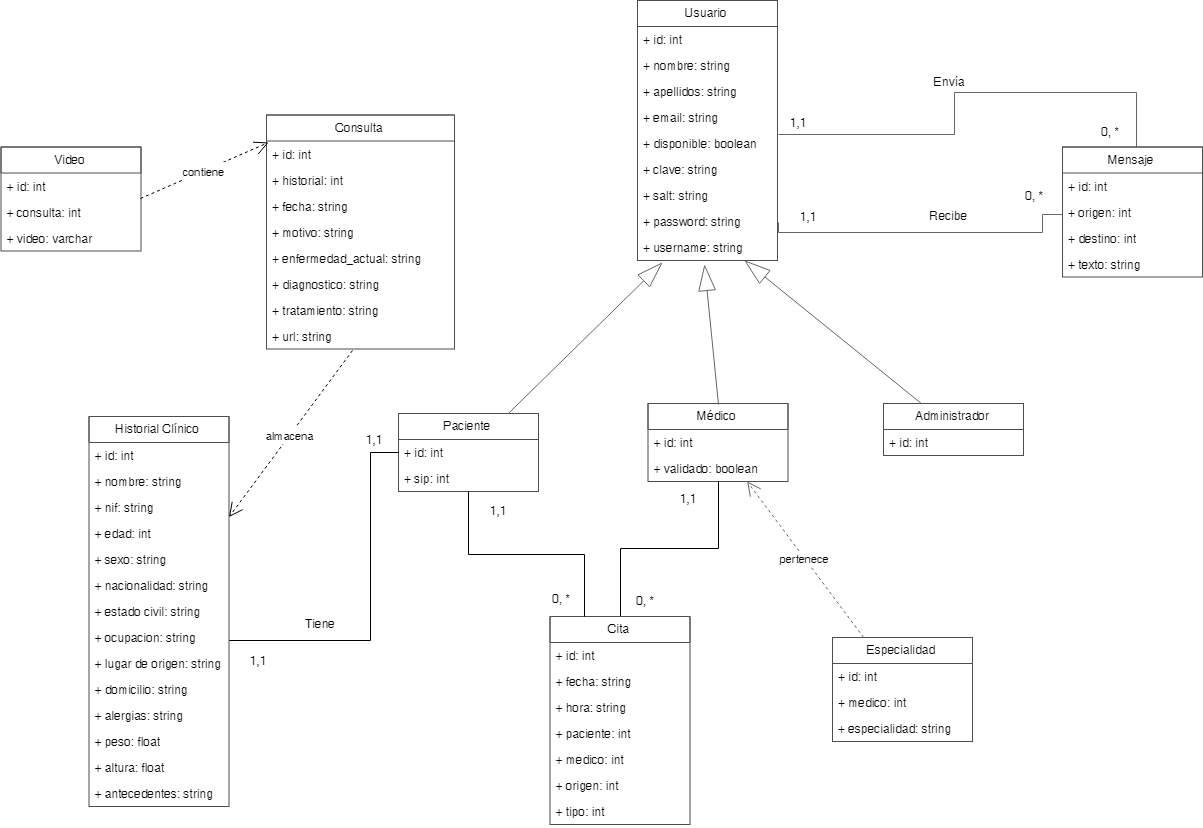
* 1. **DIAGRAMA DE CASOS DE USO**

Representación gráfica de los casos de uso.



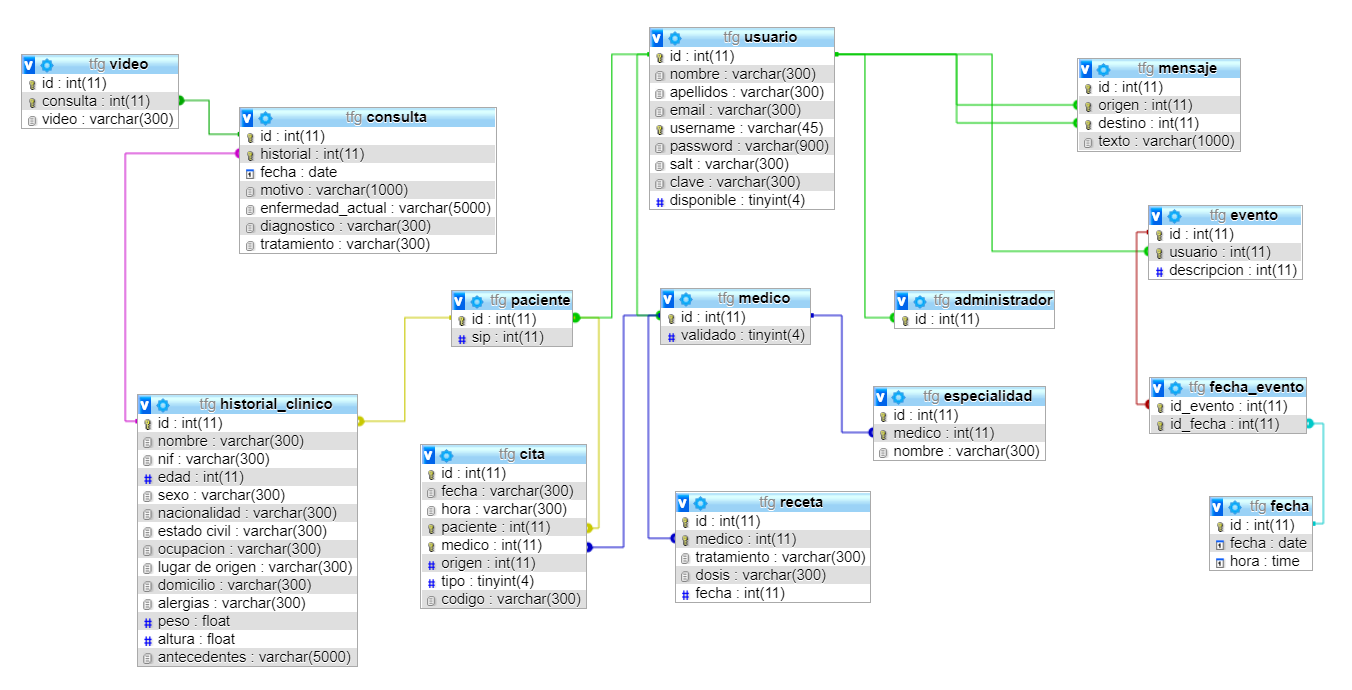
* 1. **DIAGRAMA DE CLASES**

En este apartado mostraremos nuestro diagrama de clases. Estos diagramas se caracterizan por ser estáticos, por tanto no describen acciones y muestran las entidades que existen y las relaciones entre ellas.



* 1. **ESQUEMA ENTIDAD RELACIÓN**

A continuación vamos a definir el esquema Entidad Relación de nuestra base de datos. Con este tipo de esquema podemos mostrar las diferentes entidades, relaciones y atributos que nuestra base de datos posee.

****

* 1. **REQUISITOS FUNCIONALES**

En este apartado enumeraremos los diferentes requisitos funcionales. Estos son declaraciones de los servicios que debe proporcionar el sistema, la forma en que deben reaccionar a las entradas y cómo se debe comportar en situaciones particulares. Definiremos a cada uno con un identificador, nombre, una breve descripción y una prioridad, la cual puede tomar los valores: baja, media, alta.

|  |  |
| --- | --- |
| Identificador: | RF1 |
| Nombre: | Llamar a un médico |
| Descripción: | Un paciente podrá llamar a un médico teniendo una cita previa concertada. |
| Prioridad | Alta |

Tabla 1. Requisito funcional 1

|  |  |
| --- | --- |
| Identificador: | RF2 |
| Nombre: | Llamar a un paciente |
| Descripción: | Un médico podrá llamar a un paciente sin necesidad de tener ninguna cita concertada con antelación. El médico podrá buscar al paciente por su Tarjeta Sanitaria Individual SIP. |
| Prioridad | Alta |

Tabla 2. Requisito funcional 2

|  |  |
| --- | --- |
| Identificador: | RF3 |
| Nombre: | Contestar a una llamada |
| Descripción: | Cualquier usuario ya sea médico o paciente podrá contestar a una llamada entrante. La notificación de la llamada se mostrará en una ventana superpuesta en la página actual. |
| Prioridad | Alta |

Tabla 3. Requisito funcional 3

|  |  |
| --- | --- |
| Identificador: | RF4 |
| Nombre: | Denegar una llamada |
| Descripción: | Cualquier usuario ya sea médico o paciente puede denegar cualquier llamada entrante. |
| Prioridad | Alta |

Tabla 4. Requisito funcional 4

|  |  |
| --- | --- |
| Identificador: | RF5 |
| Nombre: | Colgar una llamada |
| Descripción: | Cualquier usuario ya sea médico o paciente puede colgar una llamada en la que es participe. |
| Prioridad | Alta |

|  |  |
| --- | --- |
| Identificador: | RF6 |
| Nombre: | Listar videos de un paciente |
| Descripción: | Un médico podrá ver un listado de videos grabados en anteriores video consultas de un paciente. |
| Prioridad | Media |

|  |  |
| --- | --- |
| Identificador: | RF7 |
| Nombre: | Ver videos |
| Descripción: | Un médico podrá visualizar un video concreto de una video llamada hecha anteriormente con un paciente. |
| Prioridad | Alta |

* 1. **REQUISITOS NO FUNCIONALES**

Ahora definiremos los requisitos no funcionales, estos van a definir las restricciones o funciones ofrecidas por el sistema, aspectos técnicos. Estos requisitos siguen la misma estructura definida en el apartado anterior.

|  |  |
| --- | --- |
| Identificador: | RNF1 |
| Nombre: | Fiabilidad del sistema |
| Descripción: | El sistema en su totalidad debe de dar una sensación de fiabilidad al cliente, teniendo una recuperación de fallos rápida y una ocurrencia de fallos baja. |
| Prioridad | Alta |

|  |  |
| --- | --- |
| Identificador: | RNF2 |
| Nombre: | Seguridad |
| Descripción: | El sistema debe de contar con un nivel de seguridad, para que toda la información personal del paciente y médico esté protegida. |
| Prioridad | Alta |

|  |  |
| --- | --- |
| Identificador: | RNF3 |
| Nombre: | Tiempos |
| Descripción: | El sistema debe de poder notificar una llamada al usuario que se quiere llamar rápidamente, entre 30 segundos y 1 minuto. |
| Prioridad | Alta |

|  |  |
| --- | --- |
| Identificador: | RNF4 |
| Nombre: | Usabilidad |
| Descripción: | El sistema tiene que ser fácil de entender e intuitivo. |
| Prioridad | Alta |

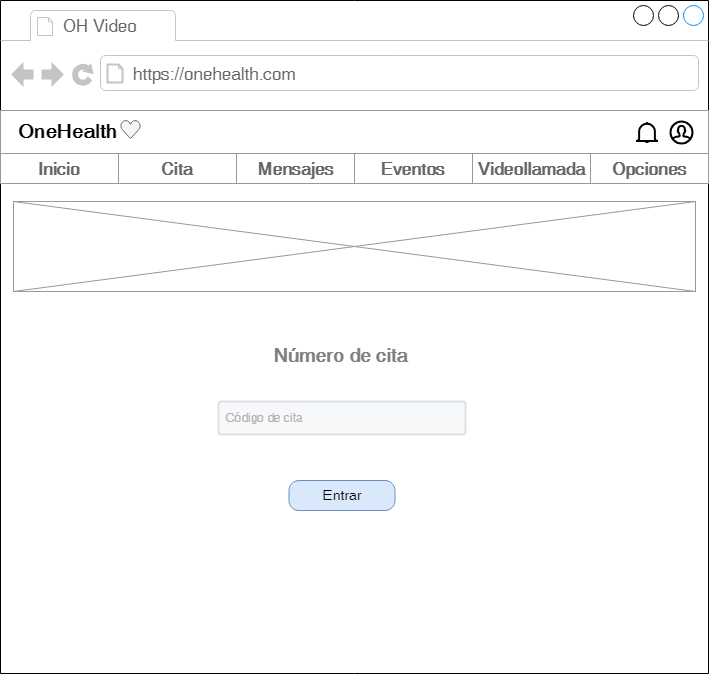
|  |  |
| --- | --- |
| Identificador: | RNF5 |
| Nombre: | Recursos |
| Descripción: | El sistema debe de poder acceder a una cámara y micro interno o externo del dispositivo con el que se está accediendo a la web. |
| Prioridad | Alta |

* 1. **REQUISITOS DE INTERFAZ**
     1. **MOCKUPS**

Este apartado está orientado a los mockups. Los mockups son diseños de alta fiabilidad, donde podemos representar toda la información que va a contener una vista, pudiendo establecer colores, tipos de letra y demostrar alguna funcionalidad básica. A continuación están los diferentes mockups diseñados, organizados por el usuario.

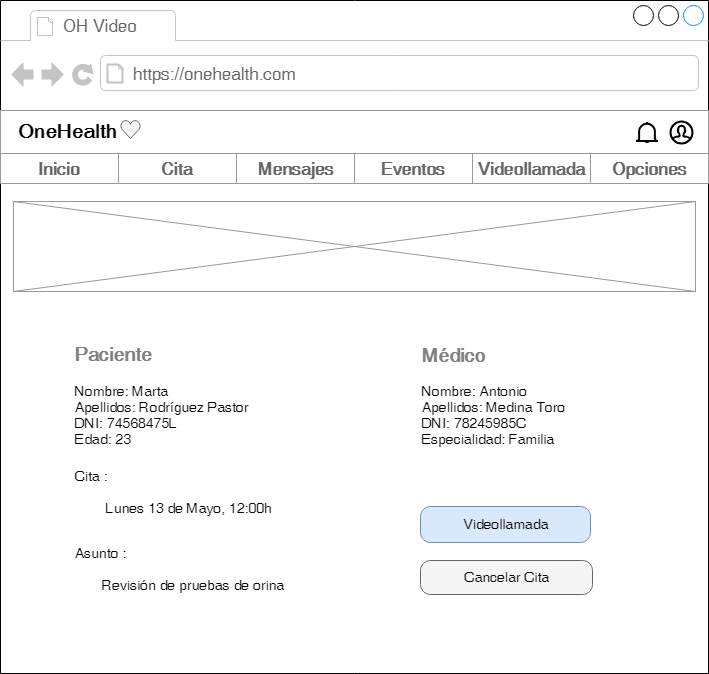
* + - 1. **Parte del paciente**
* Vista de comprobación de código de cita

En esta vista el usuario conectado podrá introducir su código de cita anteriormente generado en el apartado de crear citas. Si el código es correcto, el sistema procederá a enviarle otra vista. Para que el código sea válido tiene que estar ese código al paciente, ser el día de la cita y que sea la hora, con un intervalo de 10 minutos menos y 20 minutos más.



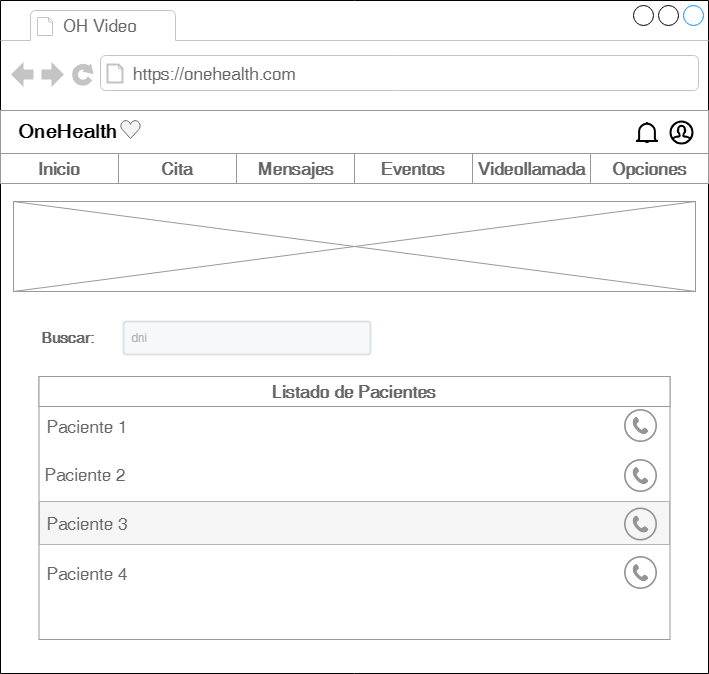
* Vista de información de cita de video consulta

En esta vista será la siguiente a la de comprobación de código y podemos encontrar en ella toda la información sobre la cita, como la hora, fecha, datos del paciente y médico. Además de esto, esta vista permitirá llamar al médico con el que tiene la cita o cancelar dicha cita.



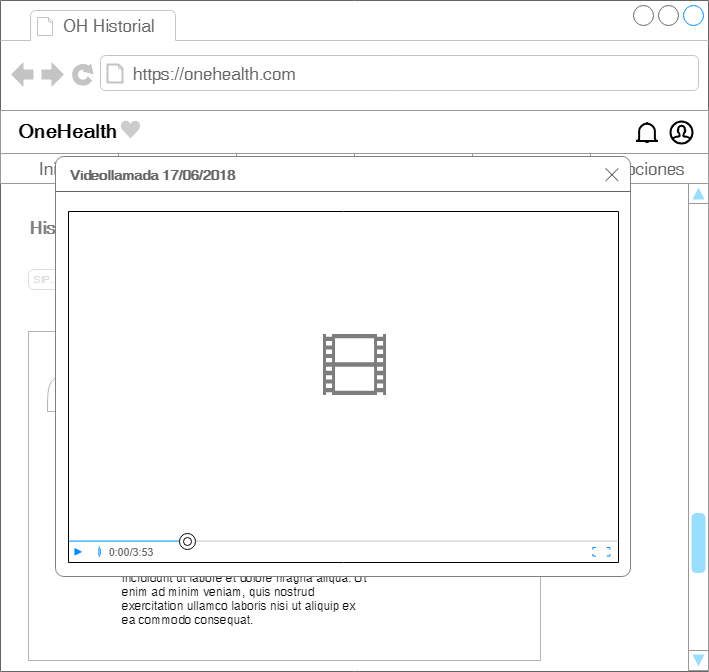
* + - 1. **Parte del médico**
* Vista de listado de pacientes conectados

Esta vista permitirá al médico conectado ver los pacientes que hay ahora mismo conectados a la plataforma, para hacer una videollamada con ellos. Este listado permite hacer una búsqueda de paciente a través de su número de tarjeta sanitaria individual SIP.

****

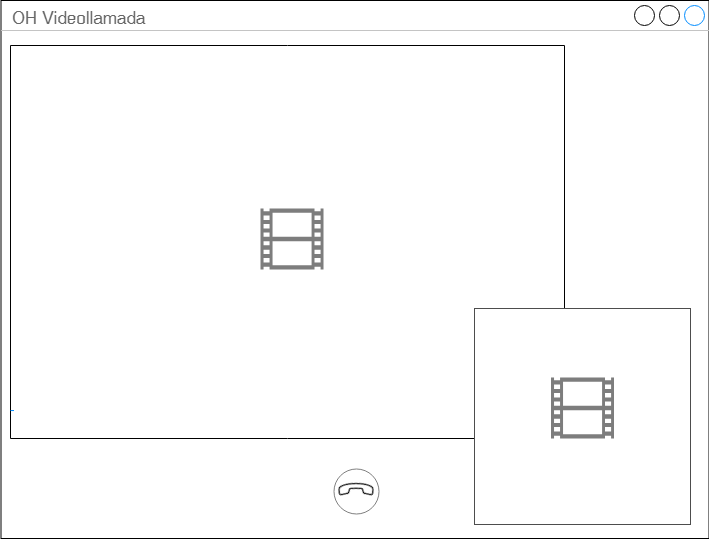
* Vista de reproducción de una video consulta

Esta vista permite al médico conectado ver un video grabado en una videollamada hecha anteriormente. Este apartado estará situado en la parte de consultas de un paciente.



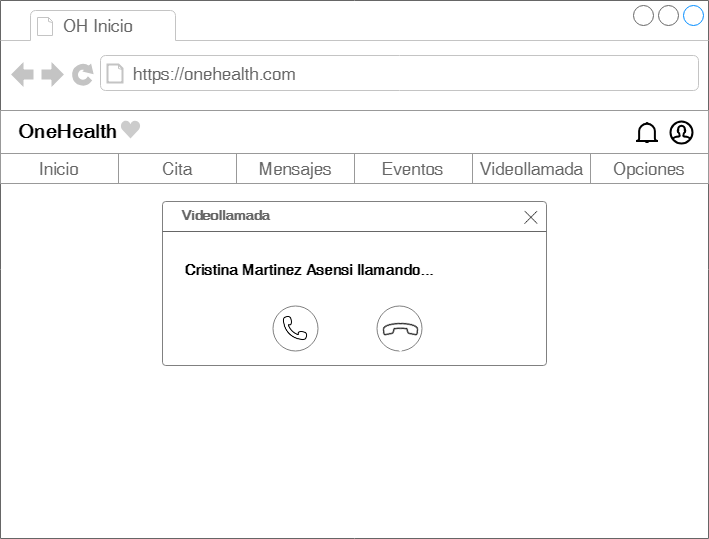
* + - 1. **Parte genérica**
* Vista de video llamada

Esta vista será genérica, esto quiere decir que cualquier usuario tanto paciente como médico verá lo mismo en una videollamada. Tendremos la imagen de usuario con el que estamos en grande y una imagen pequeña de nuestra imagen.



* Vista de una llamada recibida

Esta vista refleja cómo se le notifica a un usuario que tiene una llamada entrante. Este usuario tendrá a su disposición un botón para contestar y otro para denegar la llamada.



1. **DISEÑO Y ARQUITECTURA**
   1. **CLIENTE**
   2. **SERVIDOR**
2. **SEGURIDAD**
3. **DISEÑO E IMPLEMENTACIÓN**
   1. **ESTRUCTURA DEL PROYECTO**
   2. **SOCKETS**
   3. **KURENTO**
4. **CONCLUSIONES**
   1. **REVISION DE OBJETIVOS**
   2. **CONCLUSIÓN FINAL**
5. **BIBLIOGRAFÍA**
6. **ANEXOS**