

RAE

1. **TIPO DE DOCUMENTO:** Trabajo de grado para optar por el título de INGENIERO DE SISTEMAS
2. **TITULO:** DESARROLLO DE UN PROTOTIPO DE UNA APLICACIÓN WEB PARA AGENDAR CITAS MÉDICAS DOMICILIARIAS EN CONTEXTO DE COVID-19 EN COLOMBIA
3. **AUTORES:** Manuel Eduardo Morales Perez y Juan Gabriel Peña Rodríguez
4. **LUGAR:** Bogotá, D.C
5. **PALABRAS CLAVE:** Aplicación Web, Citas Médicas, Bases de Datos no Relacionales, NodeJS, Angular, JavaScript, HTML, CSS, Servidor, Patrón MVC.
6. **DESCRIPCIÓN DEL TRABAJO:** Una aplicación que permita asignar a los pacientes citas médicas a domicilio en donde no tenga que salir de sus casas y estar expuestos a contagio frente al COVID-19 para los ciudadanos colombianos.
7. **LINEAS DE INVESTIGACIÓN:** Metodologías ágiles, Bases de datos no relacionales, Tecnologías actuales soportada por la mayoría de los navegadores en sus ultimas versiones, Tecnologías webs más usadas y potentes del mercado actual.
8. **METODOLOGÍA:** Metodología ágil basada en iteraciones. Metodología ideal para proyectos pequeños con pocos integrantes y bajos recursos.
9. **CONCLUSIONES:** Con la información recopilada sobre las nuevas tecnologías, se tomó la decisión de realizar la aplicación en AngularJS para ahorrar tiempo creando la arquitectura del proyecto y la organización que se desea tener. Por lo que una de las conclusiones a las que se llegó fue que el tipo de lenguaje que se usó es compatible con diferentes navegadores y fácil de modificar, haciendo que por cualquier motivo un desarrollador externo al proyecto pueda leer el código y continuar con la codificación.

DESARROLLO DEL PROTOTIPO DE UNA APLICACIÓN WEB PARA
AGENDAR CITAS MÉDICAS DOMICILIARIAS EN CONTEXTO DE
COVID-19 EN COLOMBIA

JUAN GABRIEL PEÑA RODRÍGUEZ
MANUEL EDUARDO MORALES PÉREZ

UNIVERSIDAD DE SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS

BOGOTÁ, D.C 2021

**DESARROLLO DEL PROTOTIPO DE UNA APLICACIÓN WEB PARA
AGENDAR CITAS MÉDICAS DOMICILIARIAS EN CONTEXTO
DE COVID-19 EN COLOMBIA**

**JUAN GABRIEL PEÑA RODRÍGUEZ
MANUEL EDUARDO MORALES PÉREZ**

TRABAJO DE GRADO

**Tutor: YAMIL BUENAÑO PALACIOS
Ingeniero de sistemas**



**UNIVERSIDAD DE
SAN BUENAVENTURA**

**UNIVERSIDAD DE SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
BOGOTÁ
2021**

TABLA DE CONTENIDO

Planteamiento Del Problema.....	10
Objetivos De La Investigación.....	11
Objetivo General	11
Objetivos Específicos	11
Pregunta De Investigación	12
Justificación	13
Alcance	14
Marco Teórico.....	15
Metodología	29
Términos Relevantes en OpenUp.....	29
Riesgos del Proyecto	31
Plan Del Proyecto.....	34
Análisis y Diseño del Sistema.....	35
Levantamiento De Información	35
Análisis De Información	42
Arquitectura.....	64
Diseño del sistema.....	75
Desarrollo Del Sistema	83
Casos de Prueba	83
Sistema Desarrollado.....	87
Glosario.....	94
Términos de Negocio Relevantes.....	94
Términos Técnicos Relevantes.....	95
Conclusiones	97
Referencias.....	99
Anexos	101
Anexo 1 – Cronograma del Proyecto	101

Anexo 2 – Encuesta.....	102
Anexo 3 - Resolución 1552.....	106
Anexo 4 - Diccionario de datos.....	109
Anexo 5 – Manual de Instalación.....	113
Anexo 6 - Manual de Usuario (Paciente).....	124
Anexo 7 – Manual de Usuario (Médico administrador)	128

ÍNDICE DE TABLAS

Tabla 1 – Principios de OpenUp.....	20
Tabla 2 – Tareas y Artefactos	29
Tabla 3 – Riesgos del Proyecto.....	31
Tabla 5– Resumen de Interesados el Sistema.....	38
Tabla 6 – Gerente de Servicios de Salud (Stakeholder)	41
Tabla 7 – Médico Jefe de Servicios (Stakeholder)	41
Tabla 8 – Médico (Stakeholder)	41
Tabla 9 – Paciente (Stakeholder)	42
Tabla 10 – Constructor del Sistema (Stakeholder)	42
Tabla 11 – Requerimientos Funcionales.....	45
Tabla 12 – Rangos del Sistema.....	47
Tabla 13 – Skateholders frente y sus Características.....	48
Tabla 14 – Importancia de Características Para los Skateholders	49
Tabla 15 – Registro (Descripción caso de Uso).....	53
Tabla 16 – Registrar Médico (Descripción caso de Uso)	54
Tabla 17 – Eliminar Médico (Descripción caso de Uso).....	55
Tabla 18 – Ingresar al Sistema (Descripción caso de Uso)	56
Tabla 19 – Ingresar al Sistema Mediante G+ (Descripción caso de Uso)	57
Tabla 20 – Cambio de Contraseña (Descripción caso de Uso).....	57
Tabla 21 – Crear hospital (Descripción caso de Uso).....	59
Tabla 22 – Modificar Hospital (Descripción caso de Uso).....	59
Tabla 23 – Eliminar Hospital (Descripción caso de Uso).....	60
Tabla 24 – Consultar Hospitales (Descripción caso de Uso).....	61
Tabla 25 – Agendar Cita (Descripción caso de Uso).....	62
Tabla 26 – Cancelar Cita (Descripción caso de Uso)	63
Tabla 27 – Consultar Cita (Descripción caso de Uso).....	64
Tabla 28 – Overview del Sistema (Arquitectura)	66
Tabla 29 – Actores y Roles.....	66
Tabla 30 – Aplicación Web (Componentes).....	71

Tabla 31 – Controladores (Componente).....	71
Tabla 32 – Modelos (Componente)	71
Tabla 33 – Registro de Usuario (Casos de Prueba)	83
Tabla 34 – Ingreso de Usuario (Casos de Prueba).....	84
Tabla 35 – Ingreso incorrecto de Usuario (Casos de Prueba).....	85
Tabla 36 – Solicitud de Cita Médica (Casos de Prueba)	85
Tabla 37 – Visualizar Citas médicas (Casos de Prueba).....	86
Tabla 38 – Creación de Hospital (Casos de Prueba).....	87

TABLA DE ILUSTRACIONES

Ilustración 1- Esfuerzos en OpenUp	19
Ilustración 3 – Fases de la Metodología (Open up)	19
Ilustración 5 – características y subcaracterísticas de Calidad	46
Ilustración 6 – Resumen del Sistema (Caso de Uso)	52
Ilustración 7 – Resumen del Sistema (Caso de Uso)	52
Ilustración 8 – Registro (Caso de Uso)	52
Ilustración 9 – Ingresar al sistema (Caso de Uso).....	55
Ilustración 10 – Crear hospital (Caso de Uso)	58
Ilustración 11 – Agendar cita (Caso de Uso)	61
Ilustración 12 – Agendar cita (Caso de Uso)	63
Ilustración 13 - Arquitectura MVC.....	69
Ilustración 14 – Componente (Diagrama).....	70
Ilustración 15 – Cita (Documento no Relacional)	72
Ilustración 16 - Hospital (Documento no Relacional)	72
Ilustración 17 - Medico (Documento no Relacional)	73
Ilustración 18 – Usuario (Documento no Relacional)	73
Ilustración 19 - Despliegue (Diagrama).....	74
Ilustración 20 – Secuencia (Diagrama).....	75
Ilustración 21 - Modelo de datos	76
Ilustración 22 – Mapa Mind de Navegación del Sistema	77
Ilustración 23 – Login (Wireframe).....	78
Ilustración 24– Registro paciente (Wireframe).....	78
Ilustración 25 – Solicitud de Cita (Wireframe).....	79
Ilustración 26 – Listado de citas (Wireframe)	79
Ilustración 27 – Lista de usuarios (Wireframe)	80
Ilustración 28 – Lista de Hospitales (Wireframe).....	81
Ilustración 29 – Lista de Hospitales (Wireframe).....	81
Ilustración 30– Lista de Médicos (Wireframe).....	81
Ilustración 31 – Registro (Wireframe).....	82

Ilustración 32 – Pantalla de Ingreso al Sistema	88
Ilustración 33 – Pantalla de Registro	89
Ilustración 34- Pantalla de Solicitud de Citas	89
Ilustración 35 – Pantalla de Listado de Citas.....	90
Ilustración 36 – Pantalla de Listado de Usuarios.....	90
Ilustración 37 – Pantalla de Listado de Hospitales	91
Ilustración 38 – Pantalla de Creación de Hospital	92
Ilustración 39 – Pantalla de Consulta de Médicos	92
Ilustración 40 – Pantalla de Registro de Médicos.....	93
Ilustración 41 – Pantalla de Visualización de Gráficas	93

Planteamiento Del Problema

Colombia se encuentra en una emergencia sanitaria desde el 6 de marzo de 2020 debido a la llegada del COVID- 19. Esto ha conllevado a que los ciudadanos tengan que estar en confinamiento, lo que impide que realicen actividades cotidianas como acudir a sus citas o controles médicos por miedo al contagio. A causa de esto, el incremento de diferentes dolencias o enfermedades médicas es inminente, siendo así necesaria la creación de nuevas tecnologías que promuevan la salud en el país.

Objetivos De La Investigación

Objetivo General

Desarrollar el prototipo de una aplicación web que permita agendar citas médicas domiciliarias en contexto de Covid-19 en Colombia.

Objetivos Específicos

1. Realizar el levantamiento de información asociada a las tecnologías más utilizadas en la industria.
2. Seleccionar una metodología ágil con la cual se desarrolle el prototipo de la aplicación web, y que se ajuste a los integrantes y recursos del proyecto.

Pregunta De Investigación

¿Cómo desarrollar el prototipo de una aplicación web para citas médicas a domicilio en contexto de Covid-19 en Colombia?

Justificación

En el último reporte oficial realizado por la Organización Panamericana de la Salud (2021), el 4 de mayo de 2021, Colombia reporta 2.919.805 casos de COVID-19, representando un incremento nacional de 0,5%. Debido a esta situación sanitaria por la que pasa el país, resulta necesaria la adaptación de nuevas tecnologías para suplir la demanda de servicios médicos, así como la reducción del riesgo de un posible contagio a través de la interacción con otras personas en un centro médico, o durante el desplazamiento hacia este.

Igualmente, y debido a que la situación actual ha sido inesperada para la población mundial y nacional, no se encuentra evidencia suficiente de aplicaciones web que se enfoquen en agendar citas médicas para evitar el contagio del virus, por lo que es necesario la adaptación de estas al contexto lo más pronto posible.

Por esta razón, se desarrolla un prototipo de una aplicación web que beneficie a la población colombiana al momento de requerir servicios y/o atención médica, ya que se ha evidenciado el aumento de enfermedades asociadas al sedentarismo, como la obesidad, enfermedades del corazón, colesterol alto, síndrome metabólico, entre otros (OMS, 2021). Es así, que la aplicación permitirá que el paciente agende citas desde cualquier dispositivo, y obtenga la atención médica en el lugar que indique.

Alcance

Para la elaboración de este proyecto, se debe realizar un levantamiento de información el cual ayuda para mejorar el entendimiento de la problemática e identificar cuáles son los requerimientos para la entrega como producto final. A su vez, hacer un análisis de las tecnologías más usadas, enfocándose en las más estables y que cuenten con mayor soporte y documentación con el fin de desarrollar un producto funcional que se adapte con los requerimientos identificados y entregar un software intuitivo de fácil acceso y rápido mantenimiento.

Con el estudio y análisis anterior, se entregará un prototipo funcional el cual permita al paciente pedir citas médicas domiciliarias a un especialista asociado a un centro médico, pudiendo elegir el paciente la fecha y hora que más se acople a su conveniencia eligiendo entre diferentes profesionales por especialidad, ofreciendo la posibilidad de cancelar la cita con anticipación en caso de contra tiempos externos que perjudiquen el cumplimiento a la cita por parte del especialista o paciente.

Marco Teórico

Para empezar, el COVID-19 es la enfermedad causada por el coronavirus (SARS-Cov-2) y que fue detectado por primera vez el 31 de diciembre de 2019, en un grupo de casos de neumonía vírica en Wuhan (República Popular de China). Los síntomas más habituales del COVID-19 son fiebre, tos seca y cansancio; los menos frecuentes son la pérdida del gusto u olfato, congestión nasal, conjuntivitis, dolor de garganta, dolor de cabeza, dolores musculares o articulares, náuseas o vómitos, diarrea, entre otros; y los más graves son disnea, pérdida de apetito, confusión, dolor en el pecho y temperatura alta (Organización Mundial de la Salud, 2019).

En el caso de Colombia, frente al COVID-19, hay 4.801.050 casos confirmados, 67.940 casos activos, 4.598.176 recuperados y 121,216 fallecidos, siendo así las zonas más afectadas Bogotá, Antioquia, Cundinamarca, Valle y Huila (INS, 2021), presentando el incremento del número de personas contagiadas que necesitan una cama de cuidado intensivo (UCI). Frente a esto, la ocupación UCI llegó a más del 85%, según el Ministerio de Salud, por lo que el país tiene una disponibilidad del 14.3% en unidades de cuidado intensivo (Semana, 2021).

Igualmente, desde el primer brote del virus en Colombia, se han tomado como medidas de prevención (1) la cuarentena, en la que cualquier persona que haya estado en contacto con alguien infectado permanece separada de los demás en su casa; y (2) el aislamiento, en el que las personas que presentan síntomas de COVID-19, se encuentran separados de las demás personas en centros médicos donde pueda recibir atención clínica (OMS, 2021).

Es importante resaltar que, durante este confinamiento por COVID-19 se ha incrementado la frecuencia de conductas sedentarias, debido a que la persona puede pasar entre 3 a 4 o más horas al día en posición sedante. Es así, que el sedentarismo ha sido descrito como un factor de riesgo importante de mortalidad y, de la misma manera, la inactividad física ha demostrado tener efectos negativos en la salud, incrementando el riesgo de padecer enfermedades crónicas como obesidad, diabetes, hipertensión arterial y cáncer (Duarte et.al., 2020).

Debido a los síntomas anteriormente mencionados, junto con otras dolencias físicas cotidianas, así como el manejo de citas o controles médicos rutinarios por parte de los pacientes, deben seguir su curso normal. Por esto, el desarrollo de una aplicación web se vuelve primordial, ya que esta se adapta a las necesidades de los usuarios en tiempos de pandemia, debido a que estas tecnologías digitales permiten el distanciamiento físico, que es primordial para frenar la propagación del virus (Ciudades y Gobiernos Locales Unidos, 2020).

Para un mejor entendimiento sobre el desarrollo del prototipo de una aplicación web, es importante aclarar ciertos conceptos relevantes en la realización de este proyecto, tales como ingeniería de Software, procesos de desarrollo, proceso unificado, OpenUp, ciclo de vida y Lenguaje de Modelado Unificado (UML).

La ingeniería de Software es una disciplina en el ámbito de la ingeniería que comprende todos los aspectos del desarrollo de software desde las etapas tempranas que comprenden las especificaciones del sistema hasta el despliegue y mantenimiento de este, una vez el desarrollo ha finalizado. La ingeniería comprende los procesos técnicos de desarrollo como la gestión de proyectos de software, desarrollo de herramientas, métodos y teorías en las cuales se apoyan los ingenieros para la construcción de software (Sommerville, 2005).

Para producir un software de calidad, los ingenieros que implementan esta disciplina lo hacen mediante un enfoque sistemático y organizado. No obstante, aunque la ingeniería consiste en elegir el método más apropiado para una determinada situación, un enfoque más informal de desarrollo se podría implementar en algunos casos. Este desarrollo informal es ideal en plataformas web, las cuales requieren de una mezcla entre el desarrollo de software y diseño gráfico (Sommerville, 2005).

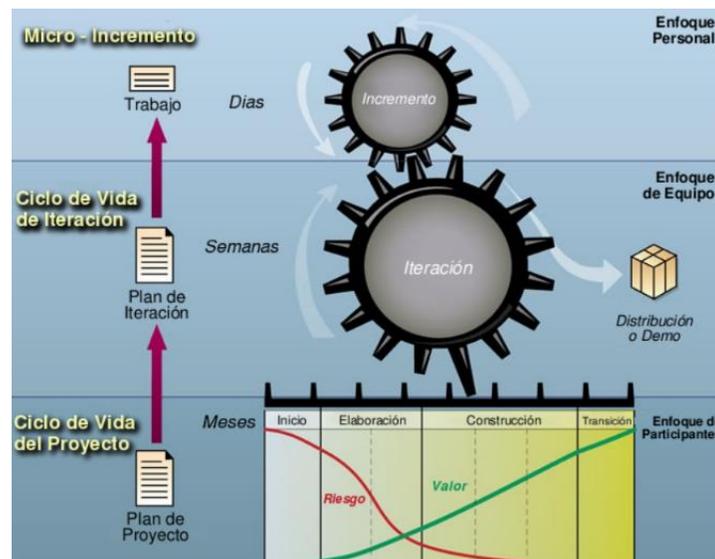
Por otro lado, cuando se habla de la construcción de software, entran en escena una serie de variables necesarias para el comienzo y finalización exitosa de un proyecto, las cuales son (a) recursos, (b) conocimientos, (c) tiempo y (d) esfuerzos. De esta manera, la ingeniería de software establece el camino a seguir. Un posible rumbo a tomar se define como lo Procesos de desarrollo de software, en donde se establecen las conexiones entre los distintos elementos que logren dar respuesta a las incógnitas *quién* debe hacer, *qué*, *cuándo* y *cómo* se debe realizar. La primera incógnita hace referencia a los recursos humanos en un proyecto, la segunda, a lo que se quiere desarrollar mediante actividades basadas en roles, que responden a su vez al cómo y al cuándo (Noriega, 2015).

Por otra parte, un proceso de software completo y con frecuencia se le conoce como *metodología*, que ayuda a los equipos de desarrollo a definir roles y actividades junto a las técnicas más recomendadas. Asimismo, se usa el término *método* para hacer referencia a notaciones las cuales se aplican a una o muchas actividades pertenecientes al proceso de desarrollo. De esta manera, el proceso unificado de desarrollo es un conjunto de actividades necesarias para transformar los requisitos del usuario en sistema un software (Jacobson et.al., 2000).

Más a detalle, es un marco para desarrollar software que se ejecuta de manera iterativa y orientada a objetos, y se está dirigido por casos de usos, en el que los ingenieros desarrollan modelos de diseño para más adelante implementarlos, teniendo presente la arquitectura la cual presenta aspectos dinámicos y estáticos más importante de un sistema. También, es fuertemente enfocada a los riesgos y está dividida entre las fases que pueden incrementarse (incremental), pueden ser recorridas o tener más de un ciclo de vida (iterativa). En otras palabras, es una metodología enfocada principalmente para guiar el proceso de desarrollo de software en sus aspectos más técnicos (Jacobson et.al., 2000).

Figura 1

Esfuerzos en OpenUp



Nota: tomado de Gimson (2012). Metodologías ágiles y desarrollo basado en conocimiento.

De otra manera, OpenUp es una evolución del proceso unificado, el cual contiene un mínimo de ayudas para los equipos, y que están diseñadas para hacer más efectivo el desarrollo de software. Asimismo, posee una filosofía ágil y práctica enfocándose en actividades colaborativas del desarrollo de software sin dejar a un lado las características

fundamentales del proceso unificado. Este se desarrolla en un ciclo iterativo detectando cualquier riesgo de forma temprana permitiendo demostrar resultados al cliente en cualquier etapa del curso del proyecto (Gimson, 2012).

De la misma manera, este trabajo se caracteriza por estar organizado por pequeños incrementos que representan pequeñas unidades de trabajo donde se genera un progreso, que es capaz de ser medible en horas o pocos días. Este es realizado por un equipo organizado y comprometido que trabaja de manera incremental. Es así, que se divide en iteraciones que son medidas por semanas en donde se enfoca a la entrega de resultados por parte del equipo de la manera más rápida posible y, al mismo tiempo, define qué se tiene que entregar y el cómo se realizará (Ríos et.al., 2013).

Es importante aclarar que el proyecto se divide en cuatro etapas: (1) inepción, (2) elaboración, (3) construcción y (4) transición. Estas fases otorgan a los miembros del equipo puntos de decisión a lo largo de todo el proyecto permitiendo vigilancia sobre este, en todo momento, por lo que las decisiones de seguir o no adelante con el proyecto se toman en los momentos oportunos. Y de la mano, el cronograma define el ciclo de vida de este lo cual genera como resultado final una aplicación correctamente desplegada (Ríos et.al., 2013).

Principios de OpenUp

OpenUp se divide en cuatro principios primordiales, creando las bases para la implementación de los roles y tareas y en la manera en que estos se realizan: (1) generar soluciones que maximicen los beneficios para todas las partes del proyecto, cumpliendo con todas las restricciones que posee este. También, equilibrar todas las prioridades relevantes con el objetivo de maximizar el valor para todas las partes interesadas en el proyecto (Ríos et.al., 2013; Gimson, 2012).

El (2) principio hace referencia a realizar trabajo en conjunto para alinear los intereses y compartir conocimientos, a la vez que se promueven prácticas que fomentan el buen ambiente entre todos los miembros del equipo, generando así una colaboración y comprensión compartida del proyecto. (3) Tener como prioridad la arquitectura en las fases tempranas permitiendo al equipo enfocarse, para disminuir los riesgos del proyecto y planificar el desarrollo entre todos los miembros del equipo (Ríos et.al., 2013; Gimson, 2012).

Y (4) promover prácticas que permiten a todos los miembros del equipo evolucionar para generar retroalimentación de forma temprana y constante de todas las partes interesadas o miembros del equipo, con el fin de mostrar el valor incremental para cada uno de ellos (Ríos et.al., 2013; Gimson, 2012).

OpenUp: Proceso Ágil

OpenUp utiliza una filosofía ágil que se enfoca en la colaboración en el desarrollo de software. Así lo refiere Balduino (2007): “La mayoría de las practicas ágiles reconocidas tienen como propósito que el equipo se comuniquen entre sí para así lograr una comprensión compartida del proyecto, eliminando así los resultados improductivos y la pérdida de tiempo”.

Los principios de OpenUp soportan, cada una, una declaración del manifiesto ágil, como se observa en la Tabla 1.

Tabla 1

Principios de OpenUp

Principios de OpenUp	Principios del manifiesto ágil
Trabajo colaborativo para alinear los intereses y socializar comprensión acerca del proyecto	Individuo e iteraciones sobre el procesos y herramientas

Equilibrar todas las prioridades involucradas para aumentar el valor de cada interesado del proyecto	La atención continua con el cliente sobre otros temas externos
Centrarse en la arquitectura en fases tempranas disminuyendo riesgos	Software funcional sobre elaboración de documentos
Fomentar la retroalimentación constante para todos los Stakeholders	Adaptación a cambios sobre el seguimiento de un plan

Nota: tomado de Gimson (2012); Sommerville (2005).

Por otro lado, OpenUp es contenida dentro de las disciplinas (1) entorno, (2) gerencia de proyecto, (3) requerimientos, (4) desarrollo, (5) pruebas y (6) despliegue (Ríos et.al., 2013).

Entorno

Para esta disciplina se explica cómo moldear los procesos y las herramientas de OpenUp con la finalidad de aplicar a un determinado proyecto u organización (Ríos et.al., 2013).

Gerencia de proyecto

En esta disciplina se declara la manera de capacitar y apoyar a cada miembro del equipo de desarrollo, teniendo presente los riesgos que posee la construcción de software y cómo hacerles frente. Los principales objetivos para esta disciplina son: (a) promover acuerdos entre los interesados priorizando la secuencia del trabajo; (b) fomentar la colaboración entre los miembros del equipo y desarrollar planes a corto y largo plazo para el proyecto, (c) contribuir en la creación de un entorno de trabajo ideal, maximizando la productividad de cada miembro de trabajo, (d) retroalimentación sobre el avance del proyecto a todas las partes interesadas, y (e) generar un marco para gestionar riesgos del proyecto con el fin de adaptación a posibles cambios (Ríos et.al., 2013).

Requerimientos

Para esta disciplina, se explica la forma de comprueban y gestionan todos los requisitos para el sistema que se pretende desarrollar. La finalidad de los requerimientos es: (a) comprender el problema a resolver, (b) comprender lo que el sistema debe hacer, definiendo los requisitos para efectuar la solución, (c) entender las necesidades de todos los interesados del proyecto acerca de qué necesitan o requieren los usuarios, (d) definir el alcance del sistema, (e) planificar las iteraciones, (f) un primer acercamiento a la estimación de costos y tiempos, y (g) la arquitectura, que se plantea partiendo de los requisitos según su importancia y con el propósito de desarrollarla robusta y estable par que soporte el sistema (Ríos et.al., 2013).

Desarrollo

Para esta disciplina se explica la manera de diseñar e implementar una solución técnica según la arquitectura desarrollada, que vaya de la mano con los requerimientos propuestos, teniendo los siguientes objetivos: (a) construir el sistema de forma iterativa e incremental, (b) moldear el diseño con el objetivo de coincidir con el entorno de ejecución, (c) validar que las unidades técnicas usadas en la construcción del sistema funcionen tal cual como se solicita, y (d) convertir los requisitos del sistema en un diseño funcional (Gimson, 2012).

Teniendo en cuenta cada iteración y las tareas relacionadas a ella, se desarrollan para generar una versión del sistema donde cada vez sea más estable. Al momento de trabajar en el sistema cada desarrollador se hará uso de la arquitectura estando sujetos a las limitaciones de esta (Gimson, 2012).

Pruebas

Para esta disciplina se definen el mínimo de tareas básicas para la planificación, implementación y ejecución del sistema con sus respectivas pruebas. La finalidad de esta disciplina es: (a) validar conjeturas propuestas pertenecientes a las especificaciones y requisitos mediante la demostración clara, (b) comprobar que cada uno de los requisitos aplique de una manera óptima al sistema, y (c) validar que el producto final funcione de acuerdo con los requerimientos (Gimson, 2012).

Despliegue

Para esta disciplina se explica cómo se planifica y se despliega una solución, teniendo en cuenta todo el conjunto de actividades que hacen que el producto final esté disponible para su uso (Ríos et.al., 2013).

Ciclo de vida

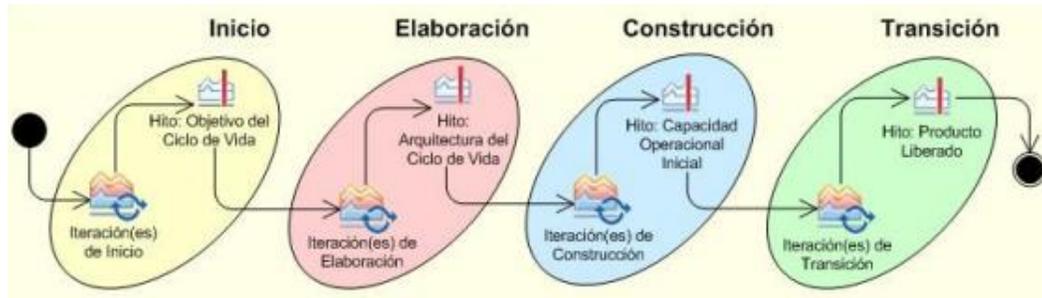
El ciclo de vida, o las fases de OpenUp, permiten a cada integrante del equipo de desarrollo aportar micro incrementos que son el resultado del trabajo unas cuantas horas, o máximo un par de días. Este progreso se mide verificando de manera diaria los micro incrementos presentados, permitiendo que el producto final evolucione en función de estos micro incrementos (Ríos et.al., 2013).

El objetivo de esta metodología es siempre ayudar al equipo de desarrollo, priorizando el producto final sobre la elaboración de documentación extensa. Las fases del proyecto proporcionan al cliente una visión general del proyecto, también genera transparencia y utilidades con el fin de controlar la financiación del riesgo y su retorno estimado (Ríos et.al., 2013).

Como se mencionó anteriormente, OpenUp es un proceso iterativo e incremental, que cuenta con cuatro fases que se dividen, a su vez, en iteraciones, así como se evidencia en la Figura 2.

Figura 2

Fases de OpenUp



Nota: tomado de Ríos, S., Hinojosa, C., & Delgado, R. (2013).

Incepción

Para esta fase se identifican las necesidades de cada uno de los participantes del proyecto, las cuales son plasmadas en el objetivo de este. También, identifican los ámbitos, límites y criterios de adaptación, casos de uso y un maquetado de bajo nivel de la planificación (Ríos et.al., 2013).

Elaboración

En esta fase se define la arquitectura del sistema. Se desarrolla un plan del proyecto donde se plasman los requisitos y una arquitectura estable. El proceso de desarrollo, herramientas, infraestructura y el entorno de desarrollo se identifican. Para el fin de fase, se debe tener una definición acertada de casos de uso y *stakeholders* (Ríos et.al., 2013).

Construcción

En esta fase los componentes y funcionalidades del sistema que hagan falta por implementar se realizan, se prueban y se integran. Los resultados que se obtienen son a través

de incrementos que tienen que ser desarrollados rápidamente, sin olvidar la calidad de lo que se está desarrollando (Ríos et.al., 2013).

Transición (Transition)

Esta fase hace referencia a la introducción del producto final en la comunidad de usuarios, siempre y cuando el producto tenga una madurez considerable. La fase de transición cuenta con las sub-fases de pruebas asociadas para versiones beta o alfa y capacitación para los usuarios del sistema, además de miembros del equipo encargados del mantenimiento de este. Dependiendo de la respuesta obtenida de los usuarios, se puede dar el caso de realizar algunas modificaciones en las entregas por si es necesario implementar alguna funcionalidad más (Ríos et.al., 2013).

Cada fase tiene las iteraciones que le hagan falta, teniendo en cuenta el dominio del negocio (que tienen los participantes del proyecto), la tecnología que se ha usado, la dificultad que presenta la arquitectura, el tamaño del proyecto y de otros factores que podrían traducirse en un retraso. Cada iteración puede o no tener una longitud dependiendo de las características del proyecto, aunque las iteraciones de un mes son las que se recomiendan (Gimson, 2012).

Partiendo que OpenUp es una metodología ágil, esta propone a los equipos una manera rápida para planear sus iteraciones: desglosar el trabajo según su iteración en un WBS (Work Breakdown Structure) y un WBS para todo el proceso del proyecto, o bien de extremo a extremo. En cada fase se desarrolla y se entregan versiones del producto estables y que funcionan. Cuando se finaliza una iteración se representa como un hito menor, que a su vez ayuda a la realización del hito mayor (Gimson, 2012).

Lenguaje de Modelado Unificado (UML)

El lenguaje de modelado unificado (UML) es un lenguaje que se basa en diagramas para la construcción y documentación de un sistema. Un modelo es una solución para entender mejor el sistema a desarrollar. En otras palabras, un modelo otorga los planos de un sistema software, donde se incluyen una visión global del sistema o el comportamiento de una de sus partes (Vidal et.al., 2012).

El UML es completamente independiente de las fases de diseño y de otros lenguajes de programación que se usan en la construcción de software, sin embargo, se basa en la programación orientada a objetos (Vidal et.al., 2012).

Es importante que el sistema a desarrollar sea estudiado desde distintas posiciones debido a que el usuario final tiene una visión diferente de la que posee un programador o un analista. UML incorpora distintos diagramas y notaciones graficas textuales destinadas a enseñar el sistema desde los diferentes puntos de vista que tienen los diferentes interesados en un sistema y que, a su vez, pueden ser usadas en las diferentes fases del ciclo de desarrollo de software (Vidal et.al., 2012).

Ahora bien, frente a investigaciones realizadas relacionadas con el tema de este proyecto, Cotino (2020) analizó la utilidad de las herramientas tecnológicas como Apps, pasaportes biológicos electrónicos y sistemas de geolocalización. Indica, que estos instrumentos permiten gestionar las relaciones, contactos y movilidad de los afectados por el virus o para controlar el cumplimiento de las normas generales de confinamiento. Por esta razón, las aplicaciones web, a demás de realizar un control sanitario, pueden ser esenciales para la previsión y asignación de servicios de salud.

De igual manera, el autor refiere que estos sistemas o aplicaciones informáticas, psoibilitan el autodiagnóstico por medio de la introducción de datos, implementar

evaluaciones para saber si se procede a hacer test, aconsejar permanencia en casa o acudir al centro médico, entre otras funciones (Cotino, 2020). Sin embargo, y como se evidencia, la aplicación no va más allá de los síntomas de la COVID-19, por lo que se reafirma la necesidad e importancia del desarrollo de una aplicación web que cubra con el requerimiento de otras enfermedades.

Del mismo modo, en la Fundación Universitaria Konrad Lorenz, Cantillo et.al. (2007) realizaron el diseño y la implementación de un sistema de información para la asignación de citas de consulta externa en diferentes áreas médicas. Esa aplicación se desarrolló bajo los lenguajes de programación Java Script, Visual Basic Script y ASP, utilizando la metodología RUP. Igualmente, utilizaron, como herramienta para el diseño de la página, Dreamweaver. Finalmente, analizaron los resultados y concluyeron que es un programa apropiado para la aplicación de lo requerido (control de citas médicas).

Por su parte, Franco (2013) desarrolló una aplicación web para la administración online de citas médicas en El Centro Médico de Orientación y Planificación Familiar, e igualmente utilizó la metodología RUP según su estructura dinámica (inicio, elaboración, construcción y transición). Como conclusiones, se encontró que la aplicación desarrollada cuenta con un gran número de pacientes, ya que les brinda una administración de citas médicas de forma rápida, transparente y segura.

Igualmente, Tolentino (2018) desarrolló una aplicación web basada en el patrón MVC para mejorar la gestión de citas médicas en una clínica de Chimbote, Perú. Se utilizaron indicadores (satisfacción del personal de admisión, duración de promedio de registro de cita médica, búsqueda de historia clínica y reporte) para dar cuenta del funcionamiento de la

aplicación. Se concluyó que los datos obtenidos refieren de un buen funcionamiento de la aplicación en relación con los indicadores anteriormente mencionados.

Finalmente, se retoma el estudio de Nolasco (2019) en el que se pretendía desarrollar una aplicación web para el control de citas médicas del centro de salud de San Jerónimo, en Purímac, Perú, con el objetivo de mejorar la atención médica, así como detectar deficiencias y proponer estrategias que favorezcan la optimización de los servicios. Se utilizaron las fases de la Programación Externa y las herramientas PMBOK (Project Management Body of Knowledge). La autora llegó a la conclusión de que los usuarios que utilizaban la aplicación se sentían mucho más conformes con la obtención de citas desde la aplicación.

Teniendo en cuenta los estudios anteriores, es importante recalcar que no se encuentra suficiente información a cerca del desarrollo de aplicaciones web para agendar citas médicas en un contexto de pandemia como el que se vive actualmente, y es por esto que se encuentra oportuno el desarrollo del presente proyecto.

Metodología

Para el desarrollo del sistema, se han evaluado distintas metodologías como lo son: Scrum, XP, Prince2, entre otras, siendo OpenUp la que mejor se ajusta al equipo, al tiempo y recursos destinados para este proyecto.

OpenUp establece el ciclo de vida para el desarrollo de software compatible con los principios sobre los que se fundamenta el Proceso Unificado donde se tiene en cuenta que está diseñado para que los equipos pequeños hagan uso de ella. OpenUp no dicta contenido para las fases de despliegue debido a que se centra en equipos únicos donde son tratados a nivel organizacional.

OpenUp es la metodología ideal para proyectos pequeños que no cuenten con muchos recursos dado a que la documentación extensa y su elaboración no son requeridos, además su implementación de micro incrementos permite mantener el control sobre el proyecto, lo cual fundamental para evitar los errores y maximizar las posibilidades de éxito.

Para resumir, se acordó entre los participantes del proyecto hacer uso de esta metodología ya que OpenUp es suficiente a un nivel básico y funcional: solo incluye contenido fundamental y no dicta guías o documentos en temas relacionados a los cuales los proyectos se pueden enfrentar. A su vez, este proceso puede ser extendido y personalizado según las diferentes necesidades que resultan a lo largo de cada una de las fases.

Términos Relevantes en OpenUp

OpenUp se divide en 2 dimensiones distintas totalmente relacionada entre ellas: “*contenido de método y contenido de proceso*”. El contenido de método hace referencia a elementos: productos de trabajo, artefactos, tareas y roles. En esta dimensión es donde todas las

cosas mencionadas con anterioridad se definen, sin importar la manera en que se aborde las fases de este marco de trabajo. El contenido del proceso es donde se aplican estos de forma temporal.

Según la Fundación Eclipse, estos son algunos de los elementos importantes de OpenUp:

Artefactos: Hace referencia a lo producido, modificado o utilizado por una tarea. Para cada rol se da la responsabilidad de crear y actualizar artefactos. Este es un elemento esencial debido que este son los que se deben usar para guardar información o datos relacionados con el proyecto.

Rol: Hace referencia al individuo que realiza un trabajo determinado. Un rol modifica el comportamiento y añade responsabilidades para los participantes del proyecto. Los roles no hacen referencia a un solo individuo, por lo tanto, un individuo puede abarcar varios roles a la vez.

Tarea: Hace referencia a cómo se realiza un trabajo en concreto. Generalmente, se define la sucesión de pasos que generan el desarrollo de uno o varios productos de trabajo.

Todos los individuos que dependen de su rol llevan a cabo tareas utilizando herramientas donde se obtienen artefactos. Este marco de trabajo dicta un conjunto básicos de roles, tareas y artefactos que son usadas para cada disciplina dentro del desarrollo de software:

Tabla 2

Tareas y Artefactos

Disciplina	Tarea	Artefactos
Entorno	Se dispone el entorno del desarrollo.	Herramientas.

Gestión del proyecto	Se planifica y se evalúan las iteraciones. Se evalúan los resultados.	Listas de riesgos. Lista de ítems de trabajo. Cronograma del proyecto.
Requisitos	Se plantean los requerimientos Se describen los casos de uso.	Glosario. Requerimientos
Arquitectura	Se define la arquitectura. Se diseña la solución.	Cuaderno de arquitectura.
Desarrollo	Desarrollar una solución con sus respectivas pruebas.	Diseño. Construcción. Implementación.
Prueba	Se crean y se ejecutan las pruebas.	Casos, guion y registros de prueba.
Despliegue	Se desarrolla el material (documentos) de apoyo.	Manual de instalación y manuales para cada usuario final

Nota: tomado de Ríos, S., Hinojosa, C., & Delgado, R. (2013).

Riesgos del Proyecto

Dado a que OpenUp se enfoca reducir de manera considerable los riesgos de manera temprana en el ciclo de vida del proyecto, se requieren reuniones frecuentes de revisión de riesgos y donde se proponen soluciones para mitigar estos.

Debido a esto, se desarrolló en conjunto con los miembros del proyecto, un listado de riesgos estimados para el proyecto donde para cada riesgo, se estimaron acciones de mitigación. Cada vez que finalizó una iteración se desarrollaron evaluaciones de *status* para cada uno de los

riesgos considerados con la finalidad de verificar si cuál de estos se presentó y la efectividad de sus respectivas estrategias de mitigación.

Tabla 3

Riesgos del proyecto

COD Riesgo	Riesgo	Descripción	Tipo	Impacto	Probabilidad	Magnitud	Estrategia de mitigación
RI-01	Perdida de información	Posible pérdida de información debido a circunstancias externas	Técnico	4	50%	2	Alojar los datos en Mongo Atlas (Cloud)
RI-02	Robo de información	Posible robo de información sensible por parte de programas de terceros	Indirecto	4	20%	0.8	Desarrollar la aplicación mediante buenas prácticas y adoptar medidas de seguridad y cifrado de contraseñas y uso de tokens
RI-03	Tiempo de desarrollo	Retrasos en el tiempo estimado del desarrollo	Directo	5	70%	3.5	Realizar un cronograma detallado donde se le realice seguimiento semanalmente y se verifiquen los entregables presentados
RI-04	Mala planificación	Planificación errónea que retrase los tiempos de desarrollo	Directo	4	60%	2.4	Según la metodología a usar desarrollar un cronograma con entregables cada semana
COD Riesgo	Riesgo	Descripción	Tipo	Impacto	Probabilidad	Magnitud	Estrategia de mitigación

RI-05	Uso de software no compatible	Mediante el desarrollo hacer uso de tecnologías incompatibles entre ellas	Directo	4	50%	2	Tener presente las tecnologías y sus versionamientos, así como también sus últimas actualizaciones y sus notas del parche
RI-06	Robo/Daño en los equipos de desarrollo	Posible robo o daño en los equipos donde se está llevando a cabo el proyecto y desarrollo	Indirecto	5	20%	1	Hacer uso de tecnologías modernas de alojamiento en la nube donde se sincronizan los cambios de manera automática después de una modificación en los archivos
RI-07	Problemas de rendimiento	Mal funcionamiento de la aplicación que ocasione mal rendimiento por mala optimización u otros factores	Directo	3	30%	0.9	Definir los requisitos básicos del sistema y hacer uso de las buenas prácticas de programación
RI-08	Complejidad del diseño	A medida que avance el desarrollo evidenciar un aumento en la dificultad en el proyecto	Directo	4	60%	1.8	Comunicación constante entre los participantes del proyecto y evaluar al final de cada iteración la iteración pasada y venidera
RI-09	Desastres naturales	Posibles desastres naturales que generen pérdidas de información o retrasos en los tiempos de entrega	Indirecto	5	5%	0.25	Hacer uso del alojamiento en la nube tener equipos secundarios para evitar retrasos en el desarrollo en caso de desastres
COD Riesgo	Riesgo	Descripción	Tipo	Impacto	Probabilidad	Magnitud	Estrategia de mitigación

RI-10	Metodología usada	Hacer uso de una metodología que no sea la más ideal para el proyecto o que genere un grado mayor de dificultad a este	Directo	3	50%	1.5	Realizar un análisis sobre el producto a desarrollar, así como también sus recursos y tiempos de entrega para hacer un estudio y elegir la tecnología que más se acople al proyecto
RI-11	Problemas de compatibilidad	Presentar problemas de compatibilidad entre el hardware y software a usar	Indirecto	3	30%	0.9	Para antes de empezar el desarrollo, evaluar las tecnologías y equipos a usar, así como también sus

Nota: realizado por los autores del presente proyecto.

Plan Del Proyecto

Con el fin de mantener los tiempos de entrega, presentar un prototipo funcional y obtener un mayor entendimiento del problema, se crea un plan de proyecto basado en la metodología OpenUp, la cual dicta que unos de los principales requisitos es el desarrollo de una “*lista de ítems de trabajo*”- para que cada participante del proyecto se basará para realizar las tareas requeridas para la finalización del proyecto.

Con la lista de trabajo antes descrita, cada una de las tareas necesarias, fueron identificadas y agrupadas en las actividades de más alto nivel (WBS). Se desarrolló un cronograma donde se especifican las iteraciones semanales presentando entregables al final de cada una de ellas.

Para garantizar el entendimiento del proyecto en sus fases iniciales y garantizar el desarrollo de la aplicación conforme al problema, se desarrollaron casos de uso y

requerimientos. También, se definieron los stakeholders para identificar todas las partes interesadas en el proyecto.

Finalmente, consultar el *Anexo 1 - Cronograma del proyecto* para generar un mayor entendimiento del proceso sobre el cual fue realizado este proyecto.

Análisis y Diseño del Sistema

Para este capítulo se explica el desarrollo del sistema basados en la metodología OpenUp donde se comenzó realizando un levantamiento inicial de información definiendo todos los interesados para este proyecto y avanzando con la arquitectura propuesta y el diseño para el sistema en su estado y entrega final.

Levantamiento De Información

Dado a que el desarrollo de este proyecto está sujeto a la metodología OpenUp, se siguieron todos los lineamientos que dicta esta metodología. Los miembros del equipo realizaron reuniones para identificar todas las partes interesadas para lograr un mejor entendimiento del proyecto; también, se realizó una encuesta (*ver Anexo 2 - Encuesta*) para ver la aceptación y recepción que posee un grupo de personas sobre la aplicación propuesta.

OpenUp exige un *documento de visión* donde se plasman las distintas necesidades que tienen los usuarios finales de la aplicación. Este documento ofrece una visión general del sistema para entregar una solución acorde a los requerimientos que cumplan con cada uno de los usuarios finales y del mercado.

Demografía Del Mercado

Debido a la enorme cantidad de centros médicos, EPS y redes públicas hospitalarias del país, se analizaron únicamente 10 de estas; con el objetivo de sondear el mercado y analizar cuál es el medio que usan sus pacientes para pedir citas médicas. A su vez, se tendrá en cuenta cuantas de ellas ofrecen el servicio de atención médica a domicilio.

Se analizaron las siguientes entidades de salud:

- Red pública hospitalaria de Bogotá
- EPS Sanitas
- Compensar EPS
- Salud Total EPS
- Nueva EPS
- Metrosalud
- Famisanar EPS
- EPS Sura
- Capital Salud EPS
- Colsanitas

Al analizar estas 10 entidades de salud se encontró que solo 1 ofrece el servicio de atención médica a “domicilio”. Se trata de Famisanar EPS que solo ofrece esta modalidad para sus afiliados de Oncología, donde el servicio médico que se presta a domicilio es a través de la farmacia Cafam. Su vez, la web de Cafam explica que únicamente se accede al servicio domiciliario únicamente para los grupos priorizados y este servicio solo se presta de manera temporal.

Para 8/10 entidades de salud analizadas, todas ofrecían la posibilidad de pedir las citas médicas por diferentes modalidades: línea telefónica, a través de un portal web, o por celular mediante una aplicación web o por una línea de Whatsapp.

Las únicas 2 entidades que aún siguen ofreciendo el pedido de citas a través de líneas telefónicas son la Red pública hospitalaria de Bogotá y Capital Salud EPS.

Análisis De Procesos

Para entender mejor sobre el proceso del pedido de citas, a continuación, se realizará una descripción para este proceso en concreto:

Al momento en que un paciente solicita una cita médica bien sea por teléfono o por alguna aplicación móvil/web, deberá tener a la mano o registrado en la aplicación los siguientes datos

- Sí es mayor de 18 años, tener el original de la cédula de ciudadanía
- Niños menores de 7 años presentar el registro civil
- En caso de ser requerida, presentar orden médica
- En caso de ser requerida, presentar la autorización de ser servicios

Si el paciente cumple algunos de los requisitos anteriores, debe:

- Sí el paciente está afiliado a cualquiera entidad de salud perteneciente al régimen subsidiario: poseer subsidio o afiliación a la entidad promotora de salud u EPS posea un vínculo actual con el centro médico u hospital.

- Sí es población vinculada estar en el nivel I o II del Sisbén y no poseer una afiliación con alguna entidad prestadora de salud.

En el caso de ciudadanos extranjeros:

- Reunir los siguientes documentos y cumplir con las condiciones anteriormente propuestas:

- El original de la cédula de extranjería.
- Sí es necesario, presentar orden médica y autorización de servicios.

Una vez reunidos todos esos requisitos, el pedido de la cita médica ya se podría realizar, sin importar el medio (teléfono/aplicación) presentando los siguientes datos:

- Fecha en que el paciente solicita la cita
- Fecha para la cual se asigna la cita

Al momento de presentarse a la cita, se tendrán en cuenta los siguientes aspectos:

- Presentarse como mínimo con 20 minutos de anticipación
- Si no puede asistir a la cita, se debe realizar la cancelación de mínimo con 24

horas antelación.

La resolución 1552 de 2013 (*ver Anexo 3- resolución 1552*) mejor conocida como ley anti-tramites, exige a las EPS y a su red de prestadores tener disponibilidad para la asignación de citas de medicina en todos los días hábiles del año.

En el momento en que estas entidades reciban la solicitud deben informar al paciente la fecha para la cual la cita médica ha sido agendada sin que les sea permitido negarse a recibir dicha solicitud y fijar la fecha de la consulta requerida.

Resumen Del Problema

En la tabla a continuación se muestra el resumen de la problemática principal que se aborda en este proyecto según varios análisis realizados con anterioridad:

Tabla 4

Resumen del problema

El problema de	Los ciudadanos tengan que salir a la calle para realizar sus controles o citas médicas poniendo en riesgo su salud debido al COVID-19
Afecta a	Los ciudadanos colombianos
Cuyo impacto es	Posibles contagios en la situación más crítica del país desde el inicio de la pandemia con UCI al máximo de su capacidad
Una solución factible es	Una aplicación que permita asignar a los pacientes citas médicas a domicilio en donde no tenga que salir de sus casas y estar expuestos a contagio frente al COVID-19

Nota: realizado por los autores del presente proyecto.

Identificación De Stakeholders

En la siguiente tabla se muestran los Stakeholders identificados para este proyecto:

Tabla 5

Resumen de interesados del sistema

<i>Nombre</i>	<i>Representa a</i>	<i>Responsabilidad</i>
<i>Gerente De Servicios De Salud</i>	Empleado con título profesional en áreas de la salud, económicas o administrativas. Port grado en salud pública o gerencia hospitalaria	Prestación eficiente y eficaz de todos los servicios de salud a la población. Gestión del correcto funcionamiento de un hospital y los servicios que se prestan en este. También son responsables de la gestión de proyectos, personal, equipos e instalaciones.
<i>Médico Jefe De Servicios</i>	Médico que completó toda su	Moral y legal del cuidado de todos los pacientes ingresados a

	formación en el área de post grado con una unidad a su cargo	su respectiva sala dentro de un hospital. Las decisiones de mayor peso, éticas y legales deben ser conocidas por él.
<i>Médico</i>	Médico que ya completaron parte del post grado que interactúan directamente con el paciente	Áreas específicas de la medicina como subespecialidades.
<i>Paciente</i>	Individuo que hace uso de los servicios médicos ofrecidos por el hospital	Medicamentos, información y cualquier cosa relacionada con su salud. Seguir las reglas y regulaciones del hospital, aceptar las obligaciones financieras relacionados a su cuidado.
<i>Constructor Del Sistema</i>	Analista	Conoce los requerimientos, propone soluciones y establece prioridades.
	Arquitecto	Define la arquitectura sobre la cual se desarrolla el sistema; También define restricciones para el sistema.
	Desarrollador	Realiza el desarrollo del sistema a partir de la arquitectura propuesta.

Nota: realizado por los autores del presente proyecto.

Perfiles de Stakeholders

Se describen a continuación los distintos perfiles para cada Stakeholder identificados previamente:

Gerente De Servicios De Salud

Representante	Gerente del hospital
Descripción	Tiene conocimientos en el áreas de la salud, económicas o administrativas
Tipo	Experto del negocio
Responsabilidad	Prestación eficiente y eficaz de los servicios de salud que tiene a su disposición para la población. Gestionan proyectos, recursos e instalaciones

Criterios de éxito	Aprobación y disposición de sus unidades médicas, así como también de sus recursos físicos
Implicación	Ninguna
Entregables	Ninguno

Tabla 1 – Gerente de Servicios de Salud (Stakeholder)

Nota: realizado por los autores del presente proyecto.

Médico Jefe De Servicios

Representante	Médico jefe
Descripción	Médico el cual completo sus estudios de Post grado
Tipo	Jefe de los médicos que usaran la aplicación
Responsabilidad	Áreas específicas de la medicina
Criterios de éxito	Disponibilidad de los médicos a su cargo según la necesidad de los pacientes
Implicación	Ninguna
Entregables	Ninguno

Tabla 2 – Médico Jefe de Servicios (Stakeholder)

Nota: realizado por los autores del presente proyecto.

Médico

Representante	Empleado del hospital
Descripción	Se encarga de atender de manera presencial a los pacientes
Tipo	Usuario final
Responsabilidad	Área según sus respectivos estudios
Criterios de éxito	Poder realizar la cita en la casa del paciente de la misma manera en lo haría en el hospital
Implicación	Provee al sistema su disponibilidad para las citas
Entregables	Ninguno

Tabla 3 – Médico (Stakeholder)

Nota: realizado por los autores del presente proyecto.

Paciente

Representante	Todo usuario el cual haga uno de los servicios médicos que presta el hospital
Descripción	Ninguna
Tipo	Usuario final
Responsabilidad	De todas las cosas relacionadas a su salud y gastos relacionados a ello
Criterios de éxito	Agendar la cita médica a domicilio con éxito donde sea atendido igual que en el hospital
Implicación	Elige la fecha y la dirección para la cita
Entregables	Ninguno

Tabla 4 – Paciente (Stakeholder)

Nota: realizado por los autores del presente proyecto.

Constructor del sistema

Representante	Gabriel Peña – Manuel Morales
Descripción	Se encargan del diseño y desarrollo del sistema
Tipo	Conocimientos en diseño, arquitectura de software con conocimientos técnicos sobre el desarrollo de aplicaciones web
Responsabilidad	Desarrollo del sistema
Criterios de éxito	Crear un sistema donde un paciente pueda agendar una cita médica a domicilio filtrando por especialidades y médicos.
Implicación	Verificar que el desarrollo final del sistema esté acorde a los requerimientos y arquitectura propuesta
Entregables	Diseño y arquitectura del sistema. Desarrollo del Sistema.

Tabla 5 – Constructor del Sistema (Stakeholder)

Nota: realizado por los autores del presente proyecto.

Análisis De Información

Después del levantamiento inicial de información, se inicia el proceso de analizar esta información donde se definen los requerimientos funcionales y no funcionales, las interfaces

que se relacionan con el aspecto final del prototipo (sistema), reglas de negocio y limitaciones asociadas al diseño se definieron para comenzar la fase de diseño.

Requerimientos Funcionales Del Sistema

A continuación, se describen los requerimientos funcionales para el sistema en donde se plasmaron según la necesidad de cada usuario final del sistema:

Id requerimiento:	Requerimiento 1
Nombre:	Ingresar al sistema
Descripción:	El sistema debe solicitar un correo electrónico y una contraseña para poder ingresar al sistema.
Id requerimiento:	Requerimiento 2
Nombre:	Ingresar al sistema por el sistema Google Plus (G+)
Descripción:	El sistema debe disponer de un botón (G+) donde se puede hacer el registro mediante Google Plus seleccionando una cuenta asociada a Gmail o ingresando el correo y la contraseña de Gmail.
Id requerimiento:	Requerimiento 3
Nombre:	Cambio de contraseña
Descripción:	El sistema debe solicitar la contraseña original, la nueva contraseña y la confirmación de la nueva contraseña.
Id requerimiento:	Requerimiento 4
Nombre:	Registro de usuario
Descripción:	El sistema debe solicitar un nombre, un correo electrónico, una contraseña y la confirmación de la contraseña, junto a ello, también solicita aceptar los términos y condiciones.
Id requerimiento:	Requerimiento 5
Nombre:	Modificación de usuario
Descripción:	El sistema debe permitir modificar cualquier dato de los usuarios creados previamente.
Id requerimiento:	Requerimiento 6

Nombre:	Eliminación de usuario
Descripción:	El sistema debe poder eliminar un usuario seleccionado.
Id requerimiento:	Requerimiento 7
Nombre:	Consulta de usuarios
Descripción:	El sistema debe permitir al root (administrador) listar toda información personal para cada usuario en caso de posibles confirmaciones y verificaciones.
Id requerimiento:	Requerimiento 8
Nombre:	Creación de médicos
Descripción:	El sistema debe solicitar un nombre para el médico, una especialidad y un hospital al cual asociar.
Id requerimiento:	Requerimiento 9
Nombre:	Modificación de médicos
Descripción:	El sistema debe permitir modificar cualquier dato de los médicos creados previamente.
Id requerimiento:	Requerimiento 10
Nombre:	Eliminación de médico
Descripción:	El sistema debe poder eliminar un médico seleccionado.
Id requerimiento:	Requerimiento 11
Nombre:	Consulta de médicos
Descripción:	El sistema debe permitir al root (administrador) listar toda información personal de cada usuarios para posibles confirmaciones y verificaciones.
Id requerimiento:	Requerimiento 12
Nombre:	Creación de hospital
Descripción:	El sistema debe solicitar un nombre para el hospital.
Id requerimiento:	Requerimiento 13
Nombre:	Modificación de hospital
Descripción:	El sistema debe permitir modificar el nombre del hospital creado previamente.

Id requerimiento:	Requerimiento 14
Nombre:	Eliminación de hospital
Descripción:	El sistema debe poder eliminar un hospital seleccionado.
Id requerimiento:	Requerimiento 15
Nombre:	Consulta de hospitales
Descripción:	El sistema debe permitir al root (administrado) consultar la información de los hospitales asociados.
Id requerimiento:	Requerimiento 16
Nombre:	Solicitud de citas
Descripción:	El sistema debe permitir solicitar citas seleccionando un servicio, un médico asociado a esa especialidad, la fecha - hora de la cita y una razón por la cual se solicita la cita.
Id requerimiento:	Requerimiento 18
Nombre:	Consulta de citas
Descripción:	El sistema debe poder listar las citas con la información asociada.
Id requerimiento:	Requerimiento 19
Nombre:	Cancelación de citas
Descripción:	El sistema debe permitir cancelar las citas agendadas por los usuarios

Tabla 6 – Requerimientos Funcionales

Nota: realizado por los autores del presente proyecto.

Requerimientos No Funcionales del Sistema: Calidad del Sistema

Según la norma ISO 9126-1 existen características y subcaracterísticas de calidad que estarán relacionadas al proyecto.

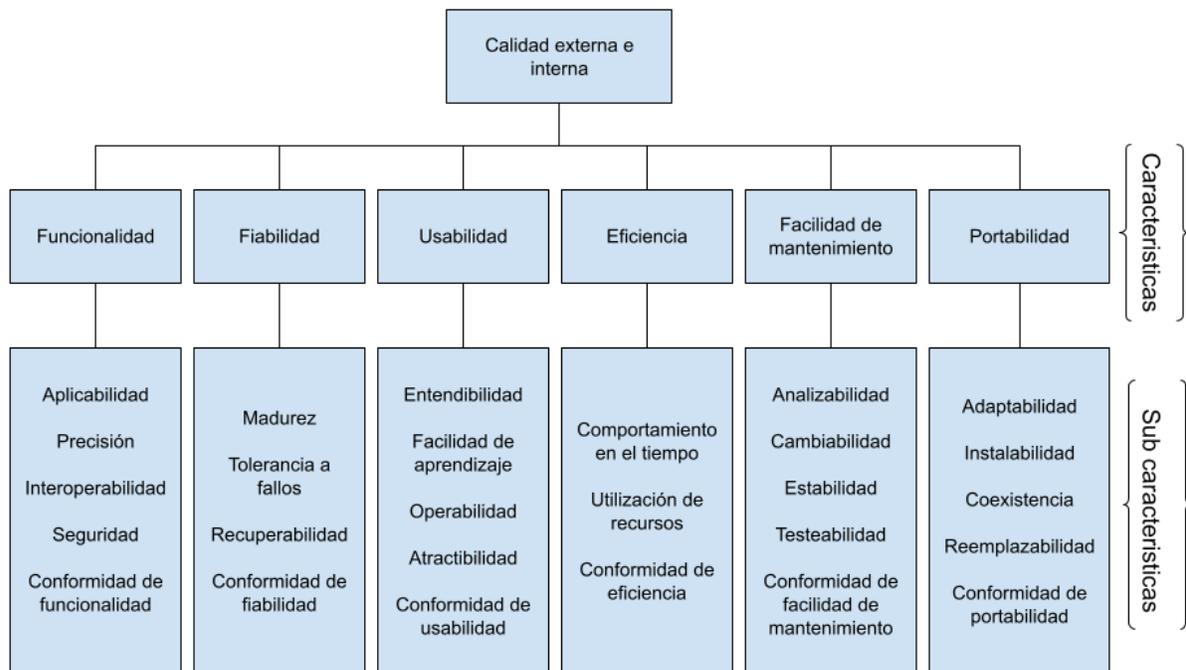


Ilustración 3 – características y subcaracterísticas de Calidad

Nota: tomado de ISO/IEC 25010.

A continuación, se describen las características de calidad expuesta plasmadas en el modelo según la normativa y el rango contemplado para el sistema:

Característica	Definición	Rango
<i>Funcionalidad</i>	La capacidad del producto software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas.	La seguridad será primordial donde se debe cumplir un estándar mínimo como lo es un usuario y contraseña. La funcionalidad debe ser exacta, ya que la información será consultada constantemente y será el núcleo del negocio.
<i>Fiabilidad</i>	La capacidad del producto software para mantener un nivel especificado de funcionamiento cuando se está utilizando bajo condiciones especificadas.	El rendimiento del sistema debe ser confiable a niveles altos de estrés y que siempre esté disponible por ser un sistema médico ya que en cualquier momento se solicitará información sensible.

Usabilidad	La capacidad del producto software de ser entendido, aprendido, usado y atractivo al usuario, cuando es usado bajo las condiciones especificadas.	Al saber que los usuarios serán médicos y/o asistentes, la usabilidad debe ser sencilla para que cualquier tipo de persona entienda el funcionamiento del sistema sin tener que analizar el manual. El software está pensado para los usuarios.
Eficiencia	La capacidad del producto software para proveer un desempeño apropiado, de acuerdo con la cantidad de recursos utilizados y bajo las condiciones planteadas.	La información que se deberá obtener debe ser de fácil acceso, haciendo que las consultas sean rápidas y sin limitaciones, incluso que el ingreso de la información sea aún más rápido que la consulta.
Facilidad de mantenimiento	Capacidad del producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, y en requerimientos y especificaciones funcionales.	El sistema será un cascarón sencillo donde se irán incluyendo módulos que lo volverán robusto sin que se vuelva complejo, haciendo que el mantenimiento se pueda hacer módulo. Las modificaciones serán puntuales y en módulos en específico.
Portabilidad	La capacidad del software para ser trasladado de un entorno a otro.	El modelo que se tiene para la instalación es de contenedores para encapsular el código y se permita migrar entre diferentes ambientes de una manera rápida e intuitiva para evitar problemas posteriores con la compatibilidad de ambientes.

Tabla 7 – Rangos del Sistema

Nota: tomado de ISO/IEC 25010.

Prioridad

Con el fin de determinar la prioridad e importancia de las características para cada parte interesada en el sistema, se desarrolló una matriz donde se califica en la escala de 1 a 5 para cada una de las características de calidad; siendo 1 la menor y 5 la de mayor interés.

Característica	Sub- Característica	Gerente hospital	Jefe Médicos	Médico	Paciente	Constructor del sistema	TOTAL	PROMEDIO
Funcionalidad	Aplicabilidad	4	5	5	5	4	19	16.4
	Precisión	4	4	5	6	5	18	
	Interoperabilidad	3	3	4	4	4	14	
	Seguridad	3	3	3	3	3	12	
Fiabilidad	Madurez	5	2	4	3	3	11	13.1
	Tolerancia a fallos	3	3	4	4	5	14	
	Recuperabilidad	5	3	3	5	5	13	
Usabilidad	Entendibilidad	2	2	5	5	5	16	16.8
	Facilidad de aprendizaje	3	3	5	5	5	17	
	Operabilidad	3	3	4	3	5	13	
Eficiencia	Comportamiento en el tiempo	3	4	2	3	5	12	10.6
	Utilización de recursos	3	2	1	1	5	7	
Facilidad de Mantenimiento	Analizabilidad	2	4	0	0	5	6	10.2
	Cambiabilidad	3	4	1	0	5	8	
	Estabilidad	3	5	5	5	5	18	
	Testeabilidad	3	3	1	0	5	7	
Portabilidad	Adaptabilidad	2	4	0	0	4	6	8.9
	Instalabilidad	3	4	2	0	4	9	
	Co-existencia	2	5	0	0	5	7	
	Reemplazabilidad	2	4	1	0	5	7	
TOTAL		58	70	55	51	95		

Tabla 8 – Stakeholders frente y sus Características

Nota: realizado por los autores del presente proyecto.

Según los datos almacenados en la *Tabla 13 – Stakeholders y sus características* se priorizaron cada una de las características de calidad para el sistema a desarrollar, donde 1 es el mayor grado de prioridad y 5 el de menor grado:

Prioridad	Característica	Subprioridad	Sub-Característica
1	Funcionalidad	1	Aplicabilidad
		2	Precisión
		3	Interoperabilidad
		4	Seguridad
2	Facilidad de mantenimiento	1	Estabilidad
		2	Cambiabilidad
		3	Analizabilidad
		4	Testeabilidad
3	Portabilidad	1	Instalabilidad
		2	Co-existencia
		3	Adaptabilidad
		4	Reemplazabilidad
4	Usabilidad	1	Facilidad de aprendizaje
		2	Entendibilidad
		3	Operabilidad
5	Eficiencia	1	Comportamiento en el tiempo
		2	Utilización de recursos
6	Fiabilidad	1	Recuperabilidad
		2	Tolerancia a fallos
		3	Madurez

Tabla 9 – Importancia de Características Para los Stakeholders

Nota: realizado por los autores del presente proyecto.

Requerimientos no Funcionales Para el Sistema

Interfaces Del Sistema. El sistema se debe poder visualizar con ciertas características relacionadas a la interfaz de usuario, para esto debe tener los siguientes criterios:

- Apariencia (interfaz de usuario)

La imagen del cliente se debe ver reflejada en los colores e imágenes que irán presentadas en el sistema.

- Requerimientos de navegación (interfaz de usuario)

La usabilidad en el sistema se debe entender como el concepto fundamental del sistema, haciendo que sea fácil la navegación y el uso. También debe tener un grado muy bajo de complicación y rápida recordación, esto se hará junto a sus menús, botones y textos explicativos.

- Consistencia (interfaz de usuario)

Para tener claro el uso del sistema, se debe usar una terminología básica, muy entendible para que cualquier tipo de usuario pueda completar la tarea que desea realizar.

- Requisitos de búsqueda (interfaz de usuario)

El sistema deberá soportar las búsquedas relacionadas a las citas que es el núcleo del negocio y poder asociar esas citas a los criterios específicos de las citas.

Interfaces De Software. El sistema deberá ser soportado en los diferentes navegadores para interactuar con el usuario y un servidor de bases de datos el cual provee para el cliente que operan bajo MongoDB y NodeJS.

Al ser una aplicación web es necesario recomendar ciertos navegadores para su correcto funcionamiento:

- Google Chrome
- Opera
- Microsoft Edge
- Mozilla Firefox
- Safari

Interfaces De Hardware. Los requisitos mínimos para que el sistema pueda correr sin problemas son:

- Monitor, mouse y teclado.
- Memoria RAM de al menos 1Gb.
- Disco duro con al menos 10Gb de espacio disponible.
- Procesador.
- Tener acceso a internet.
- Tener instalado un navegador web.

Casos de Uso

Se muestra y se describe a continuación los casos de uso más relevantes para cada actor que haga uso del sistema.

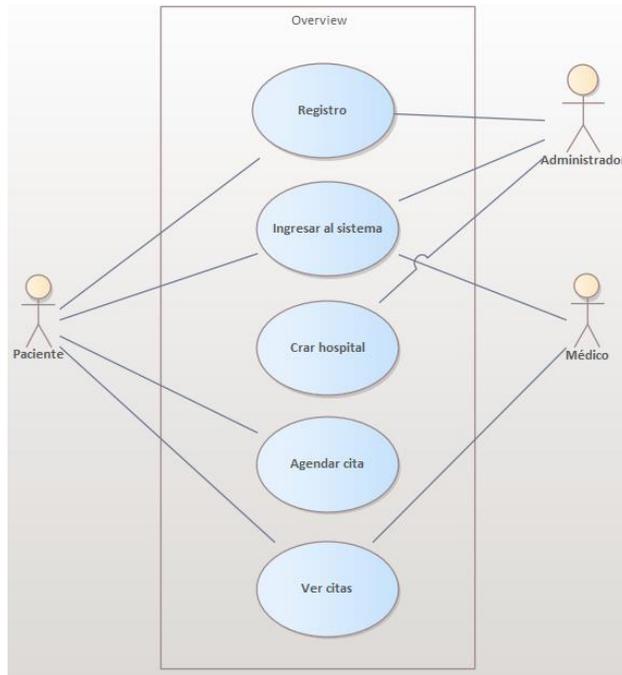


Ilustración 4 – Resumen del Sistema (Caso de Uso)

Nota: realizado por los autores del presente proyecto.

Registro. Se describe los pasos a realizar para registrarse en el sistema según los diferentes actores.

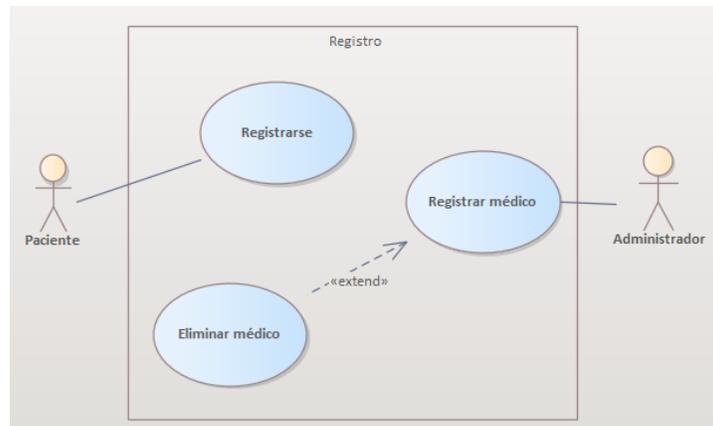


Ilustración 6 – Registro (Caso de Uso)

Nota: realizado por los autores del presente proyecto.

En las siguientes tablas se describe la serie de pasos que se toman para cada caso de uso perteneciente a la *ilustración 5*.

Identificación del caso de uso	RE-01	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Registro		
Proyecto	DESARROLLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Alta	Id requerimiento	RE-04
Descripción	Para este caso de uso se describe como el paciente se registra en el sistema		
Actor(es)	Paciente		
Entrada(s)	El paciente tiene que digitar un nombre, correo electrónico, contraseña y su debida confirmación		
Salida(s)	Poder agendar una cita médica		
Precondiciones	NA		
Postcondiciones	Condición final de éxito:	Registrarse en el sistema	
	Condición final de fallo:	Digitar nuevamente los datos de entrada requeridos	
Flujo de eventos			
No.	Actor	No	Sistema
1	El paciente ingresa los datos de entrada requeridos		
		2	El sistema valida las credenciales
		3	El sistema guarda los datos en la base de datos

Tabla 10 – Registro (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Identificación del caso de uso	RE-02	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Registro médico		
Proyecto	DESARROLLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Alta	Id requerimiento	RE-08
Descripción	Para este caso de uso se describe como el administrador crea un médico y le asigna un hospital		
Actor(es)	Administrador		

Entrada(s)	El administrador ingresa al sistema los datos personales del médico, su especialidad y el hospital al cual relacionar		
Salida(s)	El médico estará disponible para el pedido de una cita médica por parte de un paciente		
Precondiciones	NA		
Postcondiciones	Condición final de éxito:	Registrarse en el sistema	
	Condición final de fallo:	Digitar nuevamente los datos de entrada requeridos	
Flujo de eventos			
No.	Actor	No	Sistema
1	El administrador ingresa los datos de entrada requeridos		
		2	El sistema valida las credenciales
		3	El sistema guarda los datos en la base de datos

Tabla 11 – Registrar Médico (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Identificación del caso de uso	RE-003	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Eliminar médico		
Proyecto	DESARROLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Media	Id requerimiento	RE-10
Descripción	Para este caso de uso se describe como el administrador elimina a un médico del sistema		
Actor(es)	Administrador		
Entrada(s)	NA		
Salida(s)	Médico no disponible para el pedido de una cita médica por parte de paciente		
Precondiciones	Estar registrado en el sistema		
Postcondiciones	Condición final de éxito:	Eliminar médico del sistema	
	Condición final de fallo:	NA	
Flujo de eventos			
No.	Actor	No	Sistema

1	El administrador selecciona el botón de eliminar médico		
		2	El sistema elimina al médico de la base de datos

Tabla 12 – Eliminar Médico (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Ingresar al Sistema. Explica los distintos pasos que los usuarios toman para ingresar al sistema.

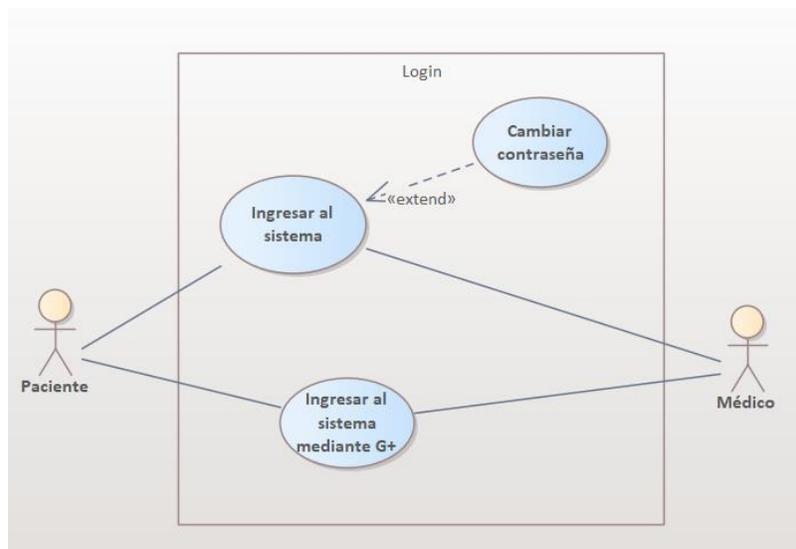


Ilustración 7 – Ingresar al sistema (Caso de Uso)

Nota: realizado por los autores del presente proyecto.

En las siguientes tablas, se describe el proceso exacto para cada caso de uso perteneciente a la *ilustración 6*.

Identificación del caso de uso	LGN-01	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Ingresar al sistema		
Proyecto	DESARROLLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		

Prioridad	Alta	Id requerimiento	RE-01
Descripción	Se describe como el paciente / médico ingresan al sistema un correo electrónico y una contraseña para poder ingresar al sistema		
Actor(es)	Paciente / Médico		
Entrada(s)	Los actores involucrados en este caso de uso tienen que digitar el correo electrónico y la contraseña		
Salida(s)	Poder entrar al sistema y tener la posibilidad de actualizar su contraseña		
Precondiciones	Estar registrado con anterioridad en el sistema		
Postcondiciones	Condición final de éxito:	Ingresar al sistema	
	Condición final de fallo:	Digitar nuevamente las credenciales	
Flujo de eventos			
No.	Actor	No	Sistema
1	El paciente / médico digita las credenciales		
		2	El sistema valida las credenciales
3	El paciente / médico ingresan al sistema		

Tabla 13 – Ingresar al Sistema (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Identificación del caso de uso	LGN-02	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Ingresar al sistema mediante Google Plus (G+)		
Proyecto	DESARROLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Alta	Id requerimiento	RE-02
Descripción	Se describe como el paciente / médico ingresan al sistema mediante G+ haciendo uso de una cuenta existente de correo en Gmail		
Actor(es)	Paciente / Médico		
Entrada(s)	Los actores involucrados en este caso de uso tienen que presionar el botón de G+ el cual hace uso de una cuenta asociada a Gmail o digitando manualmente las credenciales de una cuenta en Gmail		
Salida(s)	Poder ingresar al sistema		
Precondiciones	Poseer una cuenta en Gmail		
Postcondiciones	Condición final de éxito:	Ingresar al sistema	
	Condición final de fallo:	Digitar nuevamente las credenciales	
Flujo de eventos			
No.	Actor	No	Sistema

1	El paciente / médico digita las credenciales		
		2	El sistema valida las credenciales
3	El paciente / médico ingresan al sistema		

Tabla 14 – Ingresar al Sistema Mediante G+ (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Identificación del caso de uso	LGN-03	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Actualizar Contraseña		
Proyecto	DESARROLLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Media	Id requerimiento	RE-03
Descripción	Se describe como el paciente/médico cambia sus credenciales para el ingreso al sistema		
Actor(es):	Paciente / Médico		
Entrada(a)	Los actores involucrados en este caso de uso tienen que digitar la nueva contraseña		
Salida(s)	Tener una clave nueva y poder autenticarse dentro del sistema		
Precondiciones	Estar registrado con anterioridad en el sistema		
Postcondiciones	Condición final de éxito:	Actualización de contraseña exitoso	
	Condición final de fallo:	Digitar nuevamente la contraseña nueva y verificarla	
Flujo de eventos			
No.	Actor	No	Sistema
1	El paciente / médico realiza de forma exitosa el caso de uso LGN-01		
2	El paciente / médico da click en el botón de actualizar contraseña		
3	El paciente / médico escribe la nueva contraseña y la verifica		
		4	El sistema guarda la nueva contraseña en la base de datos

Tabla 15 – Cambio de Contraseña (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Crear Hospital. Explica los pasos que sigue el administrador para registrar un hospital dentro del sistema

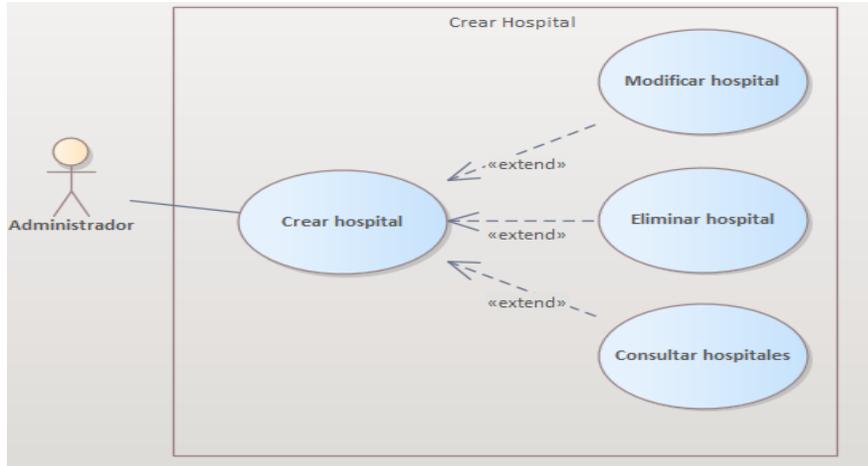


Ilustración 8 – Crear hospital (Caso de Uso)

Nota: realizado por los autores del presente proyecto.

En las siguientes tablas se describe la serie de pasos que se toman para cada caso de uso perteneciente a la *ilustración 7*.

Identificación del caso de uso	REH-01	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Crear hospital		
Proyecto	DESARROLLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Alta	Id requerimiento	RE-12
Descripción	Se describe como el administrador registra un hospital en el sistema		
Actor(es)	Administrador		
Entrada(a)	El administrador digita el nombre del hospital		
Salida(s)	Asociar médicos al hospital recién creado		
Precondiciones	NA		
Postcondiciones	Condición final de éxito:	Hospital registrado	
	Condición final de fallo:	NA	
Flujo de eventos			
No.	Actor	No	Sistema

1	El administrador digita el nombre del hospital		
		2	El sistema guarda el nombre del hospital en la base de datos

Tabla 16 – Crear hospital (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Identificación del caso de uso	REH-02	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Modificar hospital		
Proyecto	DESARROLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Media	Id requerimiento	RE-13
Descripción	Se describe como el administrador modifica el nombre del hospital		
Actor(es)	Administrador		
Entrada(a)	El administrador digita el nombre actualizado del hospital		
Salida(s)	Asociar médicos al hospital recién creado		
Precondiciones	Hospital registrado en el sistema		
Postcondiciones	Condición final de éxito:	Hospital actualizado	
	Condición final de fallo:	Digitar el nombre de un hospital registrado previamente o repetir el nombre	
Flujo de eventos			
No.	Actor	No	Sistema
1	El administrador digita el nuevo nombre del hospital		
		2	El sistema guarda el nombre del hospital en la base de datos

Tabla 17 – Modificar Hospital (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Identificación del caso de uso	REH-03	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Eliminar hospital		
Proyecto	DESARROLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Media	Id requerimiento	RE-14
Descripción	Se describe como el administrador elimina el nombre de un hospital		
Actor(es)	Administrador		
Entrada(a)	NA		
Salida(s)	NA		
Precondiciones	Hospital registrado en el sistema		
Postcondiciones	Condición final de éxito:	Hospital eliminado del sistema	
	Condición final de fallo:	NA	
Flujo de eventos			
No.	Actor	No	Sistema
1	El administrador busca y elimina el nombre de un hospital		
		2	El Sistema elimina el hospital es cuestión de sus registros

Tabla 18 – Eliminar Hospital (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Identificación del caso de uso	REH-04	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Consultar hospitales		
Proyecto	DESARROLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Media	Id requerimiento	RE-15
Descripción	Se describe como el administrador consulta los hospitales creados en el sistema con anterioridad		
Actor(es)	Administrador		
Entrada(a)	NA		
Salida(s)	NA		
Precondiciones	Hospital(es) registrado(s) en el sistema		

Postcondiciones	Condición final de éxito:	Visualizar los hospitales creados	
	Condición final de fallo:	No tener ningún hospital registrado	
Flujo de eventos			
No.	Actor	No	Sistema
		1	El sistema muestra los hospitales creados

Tabla 19 – Consultar Hospitales (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Agendar cita. Explica los pasos que sigue paciente para agendar una cita médica a domicilio según la fecha y especialidad de su necesidad y el médico de su preferencia.

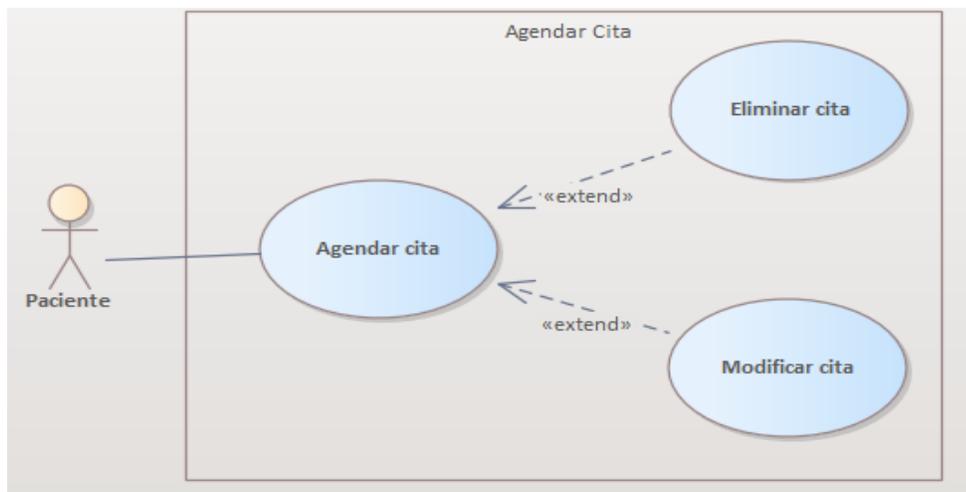


Ilustración 9 – Agendar cita (Caso de Uso)

Nota: realizado por los autores del presente proyecto.

En las siguientes tablas se describe la serie de pasos que se toman para cada caso de uso perteneciente a la *ilustración 8*.

Identificación del caso de uso:	AGC-01	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Agendar cita		

Proyecto	DESARROLLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Alta	Id requerimiento	RE-16
Descripción	Se describe como el paciente realizar la solicitud de una cita médica a domicilio		
Actor(es)	Paciente		
Entrada(a)	El paciente selecciona una especialidad y un médico asociado a dicha especialidad para finalizar con la fecha y razón de la cita		
Salida(s)	El médico elegido ahora tiene una cita que atender		
Precondiciones	NA		
Postcondiciones	Condición final de éxito:	Cita agendada	
	Condición final de fallo:	NA	
Flujo de eventos			
No.	Actor	No	Sistema
1	El paciente elige especialidad, fecha y médico de su gusto		
		2	El sistema guarda los datos y la cita finalmente está solicitada

Tabla 20 – Agendar Cita (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Identificación del caso de uso	AGC-02	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Cancelar cita		
Proyecto	DESARROLLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Media	Id requerimiento	RE-18
Descripción	Se describe como el paciente realiza la cancelación de la cita		
Actor(es)	Paciente		
Entrada(a)	NA		
Salida(s)	El médico elegido ahora tiene una cita menos		
Precondiciones	Tener una cita médica agendada		
Postcondiciones	Condición final de éxito:	Cita cancelada	
	Condición final de fallo:	NA	
Flujo de eventos			
No.	Actor	No	Sistema

1	El paciente selección el botón de cancelar cita		
		2	El sistema guarda los datos y la cita finalmente está cancelada

Tabla 21 – Cancelar Cita (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Ver cita. Se explica cómo los usuario finales de la aplicación tienen la posibilidad de ver las citas médicas agendadas donde ellos participen.

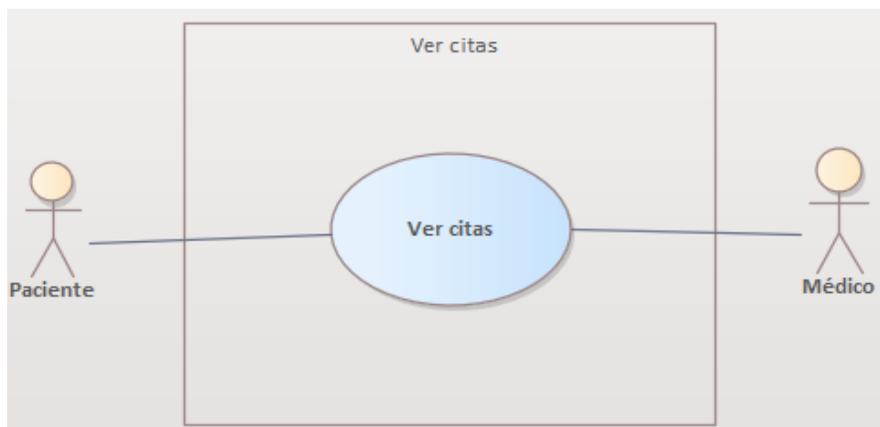


Ilustración 10 – Agendar cita (Caso de Uso)

Nota: realizado por los autores del presente proyecto.

En la siguiente tabla se describe la serie de pasos que es el caso de uso perteneciente a la *ilustración 9*.

Identificación del caso de uso	CC-01	Autor(es)	Juan Gabriel Peña – Manuel Morales
Nombre	Consultar citas		
Proyecto	DESARROLLO DE UNA APLICACIÓN WEB PARA PEDIR CITAS MÉDICAS A DOMICILIO		
Prioridad	Alta	Id requerimiento	RE-17
Descripción	Se describe los actores involucrados para este caso de uso pueden ver las citas médicas de las cuales participen		

Actor(es)	Paciente / Médico		
Entrada(a)	NA		
Salida(s)	NA		
Precondiciones	Tener una cita médica agendada		
Postcondiciones	Condición final de éxito:	Consultar las citas médicas	
	Condición final de fallo:	NA	
Flujo de eventos			
No.	Actor	No	Sistema
		1	El sistema lista todas las citas médicas del interés para los usuarios

Tabla 22 – Consultar Cita (Descripción caso de Uso)

Nota: realizado por los autores del presente proyecto.

Arquitectura

Para esta sección se describe el modelo base para el desarrollo del prototipo. Aquí se explican tanto como dominio y alcance de la arquitectura, así como también conceptos de la propia arquitectura y las decisiones que se tomaron al momento de establecerla.

Dominio de la Arquitectura

El sistema será *-en su versión de prototipo-* está pensada para usarse por una cantidad de usuarios limitados donde cada uno hará peticiones constantes al servidor, ya sea para registrar un ítem u usuario o para agendar una cita médica.

Es necesario tener una información clara para el usuario permitiendo así que el sistema pueda funcionar de la mejor manera para así más adelante realizar una trazabilidad complete de la aplicación.

Alcance

Se desarrollará un prototipo de una aplicación web donde se podrán pedir citas médicas serán realizadas en el domicilio del paciente. El paciente podrá elegir la especialidad de su necesidad, filtrando por un abanico de médicos disponibles y en el horario que el desee. El administrador será el encargado de registrar en el sistema a los distintos hospitales y más adelante, asociar un hospital a varios médicos que también serán registrados por él. El sistema contará con los siguientes módulos:

- Registro de usuarios
- Inicio de sesión
- Creación de citas
- Creación de hospitales
- Visualización de gráficas

El sistema correrá en *localhost* dado a la cantidad de usuarios finales (que son limitados) y dispone de un alto nivel de escalabilidad el cual permite en un futuro una publicación en web.

Restricciones

Teniendo en cuenta los requerimientos y casos de usos presentados y la arquitectura propuesta, se tienen las siguientes restricciones:

- El sistema no permite un chat entre el médico y paciente
- El sistema no permite calcular el costo de los servicios prestados
- El sistema no será un medio de pago para el paciente

Overview de la Aplicación

En la siguiente tabla (*tabla 28*) se describe las características para el sistema brindando un breve listado para cada tecnología a usar.

Característica	Descripción
Tipo de aplicación	Web
Restricciones	<ul style="list-style-type: none"> • Acceso local • Seguridad atada a la seguridad e integridad del servidor
Arquitectura	Se implementará la arquitectura modelo vista controlador
Tecnologías relevantes	<ul style="list-style-type: none"> • MongoDB • JavaScript • HTML • Windows 10 • Node Js

Tabla 23 – Overview del Sistema (Arquitectura)

Nota: realizado por los autores del presente proyecto.

Actores y Roles del Sistema

A continuación, se describen los actores finales asociados a su rol que harán uso del sistema y se muestra un análisis donde se describe qué actividad puede realizar cada uno dentro del sistema:

Stakeholder	Descripción	Funcional	Permisos
Paciente	Individuo que puede acceder al sistema una vez el mismo se registre y pedir una cita eligiendo diferentes parámetros	<ul style="list-style-type: none"> • Ingresar al sistema • Registrarse • Pedir citas • Ver citas • Ver médicos 	<ul style="list-style-type: none"> • Permisos de consulta
Médico	Individuo que accede al sistema mediante la creación de cuenta por parte de un administrador que se sujeta a la demanda por parte de los pacientes	<ul style="list-style-type: none"> • Ingresar al sistema • Ver citas 	<ul style="list-style-type: none"> • Permisos de consulta

Tabla 24 – Actores y Roles

Nota: realizado por los autores del presente proyecto.

Para la relación presentada entre los Stakeholders y el sistema describiendo como ellos hacen uso de la aplicación se puede evidenciar en *ilustración 4 – Resumen del sistema (CU)*

Metas de la Arquitectura

- Permitir escalabilidad de manera sencilla
- Facilidad en mantenimiento o soporte
- Crear diferentes componentes que funcionen entre sí permitiendo hacer modificaciones que no afecte a la aplicación ni a su modelo de negocio
- Funcionar en distintos navegadores
- Crear dependencias en el funcionamiento

Dependencias

Para iniciar la configuración del proyecto se debe considerar instalar las siguientes dependencias de software:

- En el servidor del cliente se debe instalar Mongo Express corriendo en un sistema operativo Windows o macOS donde se van a definir los modelos de datos y el almacenamiento asíncrono con Mongoose ODM (librería object-document mapping)
- La información persistida se encuentra en una base de datos no relacional, para tomar control de esta información es necesario instalar Robo 3T que es la GUI más liviana y sin pago mensual, el cual provee una herramienta MongoDB de código abierto que proporciona una funcionalidad innovadora para satisfacer las necesidades tangibles del proyecto.

- Por otro lado, es necesario instalar el entorno de javascript NodeJS que está orientado a eventos asíncronos, también está diseñado para crear aplicaciones escalables, para el caso de este proyecto, este será el motor de nuestro servidor.

Requisitos Relevantes de la Arquitectura

- El sistema estará soportado en librerías libres (OpenSource).
- El sistema será sostenible para mejoras y evoluciones después de su entrega.

Decisiones de la Arquitectura

La construcción del sistema está dada por módulos, gracias a la versatilidad de AngularJS se puede tener cada componente separado haciendo el mantenimiento más sectorizado. También se reduce bastante el tiempo de elaboración de una funcionalidad al quitarle la complejidad al código.

Al generar la documentación se vuelve lineal debido a la que sintaxis y la lectura del código son menos complejos y a su vez más plana.

Si en algún caso se quisiera reemplazar a los integrantes del equipo para codificar el sistema, no se tendría ningún inconveniente gracias a la documentación de AngularJS al ser un estándar en el mundo.

Para persistir los datos será mediante MongoDB para no tener problemas con las relaciones entre tablas al ser un sistema no relacional, pero a su vez, es un sistema libre y la compatibilidad entre AngularJS y MongoDB es absoluta para no tener ningún problema a la hora de pasar de un sistema a otro.

Arquitectura

Como ya se expresó anteriormente, la arquitectura seleccionada para este proyecto fue Modelo-Vista-Controlador separa entre si todas las distintas capas que componen al sistema (capa de negocio, capa de datos y vistas)

Esta arquitectura de software normalmente es usada por aplicaciones que manejan bastantes datos o transacciones complejas en donde la separación de las distintas capas toma mayor importancia.

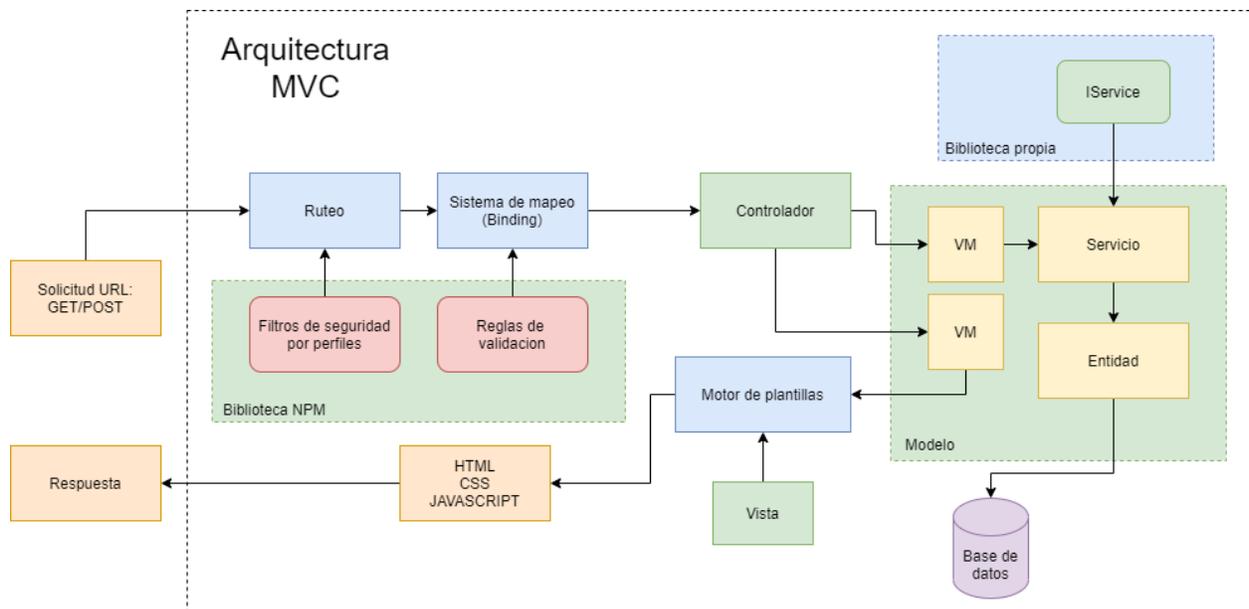


Ilustración 11 - Arquitectura MVC

Nota: realizado por los autores del presente proyecto.

Ventajas del MVC

- Dado que la lógica de negocio se separa del diseño, características como la escalabilidad o funcionalidad del sistema sea mayor

- El modelo ofrece el sistema transferirse de un entorno a otro y la posibilidad de interactuar con agentes externos
- Facilita el mantenimiento en caso de errores
- Es más fácil agregar distintas representaciones para un mismo dato
- El modelo brinda facilidad al implementar métodos y lenguajes de programación entre los distintos miembros del equipo de desarrollo y trabajar en simultaneo
- El modelo brinda maneras más fáciles para probar el correcto uso de la aplicación

Vistas de la arquitectura de software

Se definen las vistas para organizar la arquitectura del sistema, así se establece el orden físico en las carpetas del sistema y del software.

Vista de Desarrollo

El diagrama de componentes muestra los diferentes componentes en la cual está dividido el sistema.

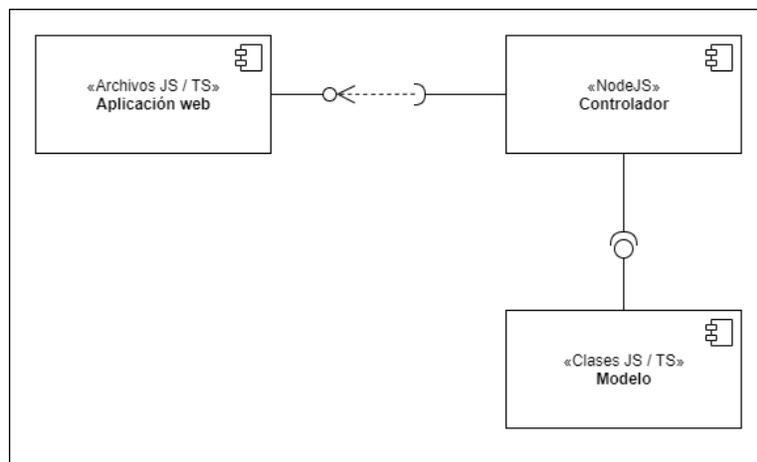


Ilustración 12 – Componente (Diagrama)

Nota: realizado por los autores del presente proyecto.

En las tablas 30, 31, y 32 se describe brevemente cada componente que fue mostrado en la ilustración 9 - Diagrama de componentes.

DC 01 - Aplicación web	
<i>Descripción</i>	Obtiene la información necesaria para dar como punto de inicio del sistema. Abarca globalmente los componentes UI para ser mostrados en los diferentes navegadores. Allí mismo se tienen las siguientes acciones: Inserción, Modificación y Consulta. Incluye otras páginas.
<i>Requerimientos</i>	Cambio de rutas entre los diferentes espacios en la aplicación web.
<i>Interfaces disponibles</i>	app.js, busqueda.js, citas.js, citasConfig.js, hospital.js, imagenes.js, login.js, medico.js, upload.js, usuario.js

Tabla 25 – Aplicación Web (Componentes)

Nota: realizado por los autores del presente proyecto.

DC 02 - Controladores	
<i>Descripción</i>	Son los servicios asociados a cada sección que tenga interacción en el sistema.
<i>Requerimientos</i>	Obtiene la información asociada a cada uno de los servicios para luego tener una respuesta con los datos necesarios para ser mostrados en el sistema.
<i>Interfaces disponibles</i>	usuario.service.ts, service.module.ts, medico.service.ts, hospital.service.ts, admin.guard.ts, login-guards.guard.ts, verificar-token.guard.ts, config-citas.service.ts, appointments.service.ts

Tabla 26 – Controladores (Componente)

Nota: realizado por los autores del presente proyecto.

DC 03 - Modelos	
<i>Descripción</i>	La información debe estar acoplada con los parámetros necesarios y para esto se requiere tener diferentes modelos en relación con los servicios construidos.
<i>Requerimientos</i>	Acopla los datos con los tipos y la estructura para el sistema.
<i>Interfaces disponibles</i>	hospital.model.ts, medico.model.ts, usuario.model.ts

Tabla 27 – Modelos (Componente)

Nota: realizado por los autores del presente proyecto.

Vista Lógica

En las siguientes ilustraciones se muestran los documentos no relaciones para los modelos respectivos:

Citas.

 _id	ObjectId
 doctorId	String
 appointmentDate	String
 name	String
 email	String
 service	String
 doctorName	String
 appointmentReason	String
 appointmentHour	String

Ilustración 13 – Cita (Documento no Relacional)

Nota: realizado por los autores del presente proyecto.

Hospitales.

 _id	ObjectId
 nombre	String
 usuario	ObjectId
 img	String

Ilustración 14 - Hospital (Documento no Relacional)

Nota: realizado por los autores del presente proyecto.

Médicos.

 _id	ObjectId
 nombre	String
 usuario	ObjectId
 hospital	ObjectId
 img	String

Ilustración 15 - Medico (Documento no Relacional)

Nota: realizado por los autores del presente proyecto.

Usuarios.

 _id	ObjectId
 role	String
 google	Boolean
 nombre	String
 email	String
 img	String
 password	String

Ilustración 16 – Usuario (Documento no Relacional)

Nota: realizado por los autores del presente proyecto.

Vista de Despliegue

Existe una relación entre los componentes y el sistema, dado a esto, se muestra la siguiente ilustración.

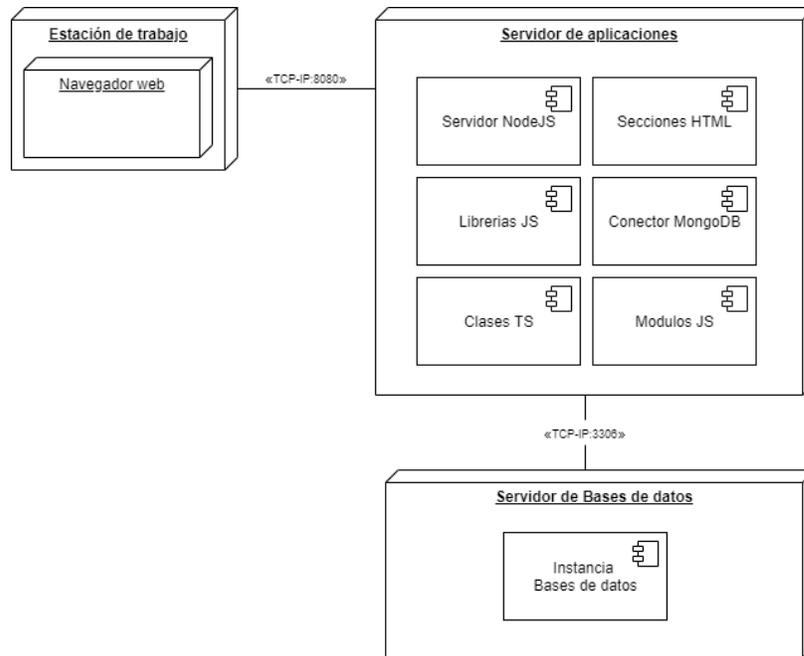


Ilustración 17 - Despliegue (Diagrama)

Nota: realizado por los autores del presente proyecto.

Vista de Procesos

Para culminar con las vistas, se muestra la vista de procesos que muestra la secuencia y relación entre todos los objetos que componen el sistema.

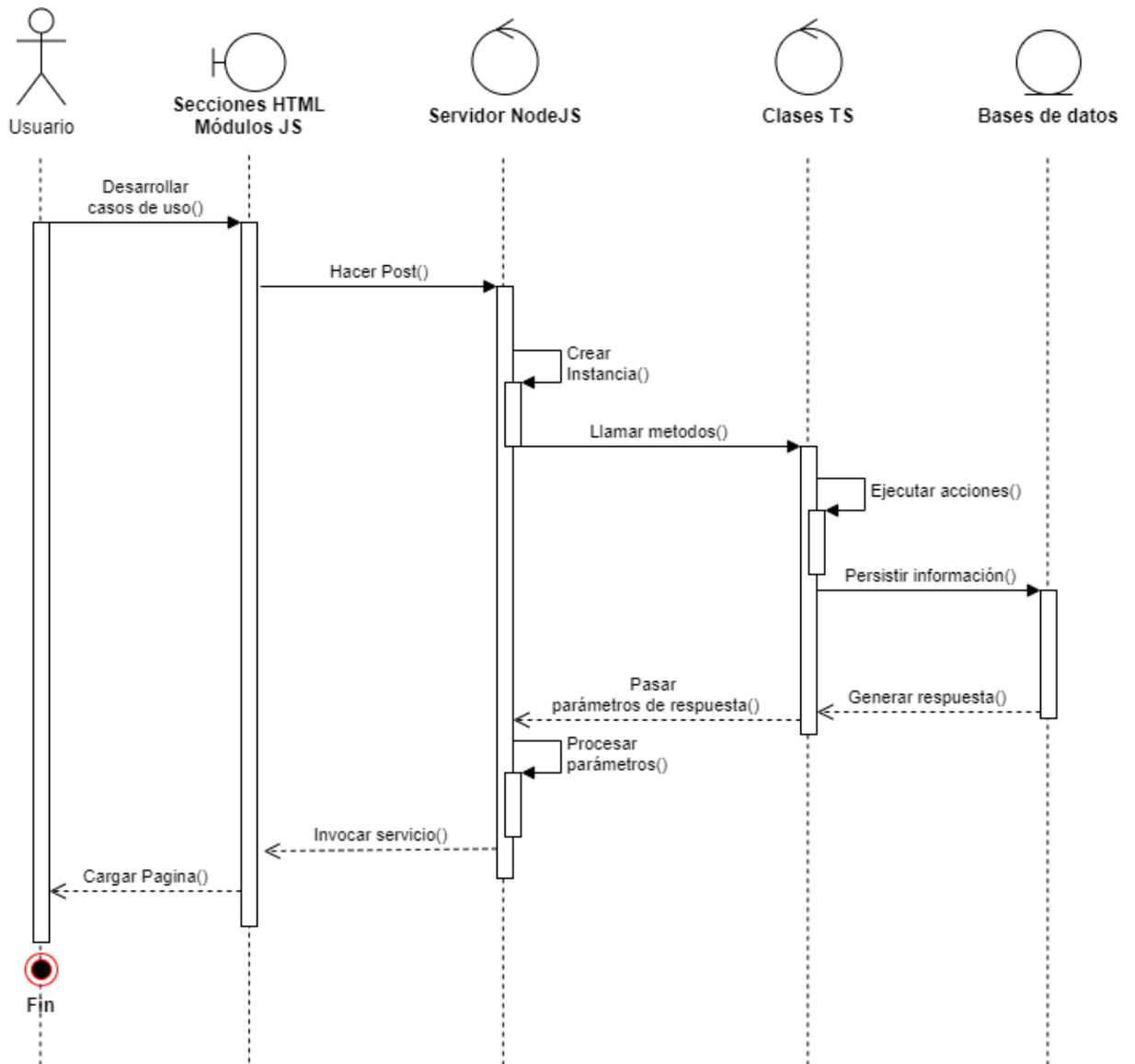


Ilustración 18 – Secuencia (Diagrama)

Nota: realizado por los autores del presente proyecto.

Diseño del sistema

Se realiza la descripción de la operatividad necesaria dentro del sistema en concepto de componentes los cuales sirven para descomponer el código fuente.

Modelo de Datos

Descripción del modelo de datos del sistema

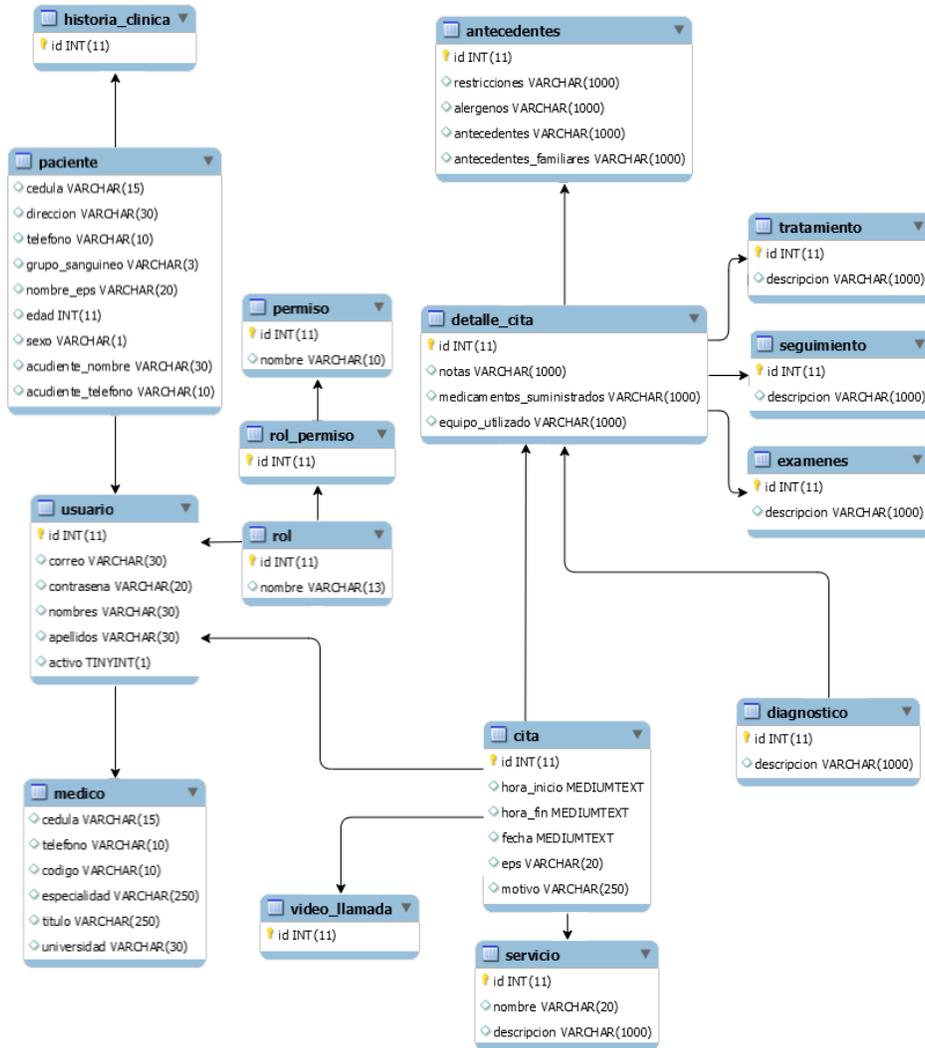


Ilustración 19 - Modelo de datos

Nota: realizado por los autores del presente proyecto.

Diccionario de Datos

Con este diccionario de datos se especifica el tipo y particularidades de los datos para cada documento del modelo de datos que se describió anteriormente. (Ilustración 18)

La descripción se observa en el *Anexo 4 - Diccionario de datos* que se haya adjunto en este documento.

Mapa de Navegación del Sistema

Se describe el mapa de navegación con las rutas que se presentarán en el sistema.

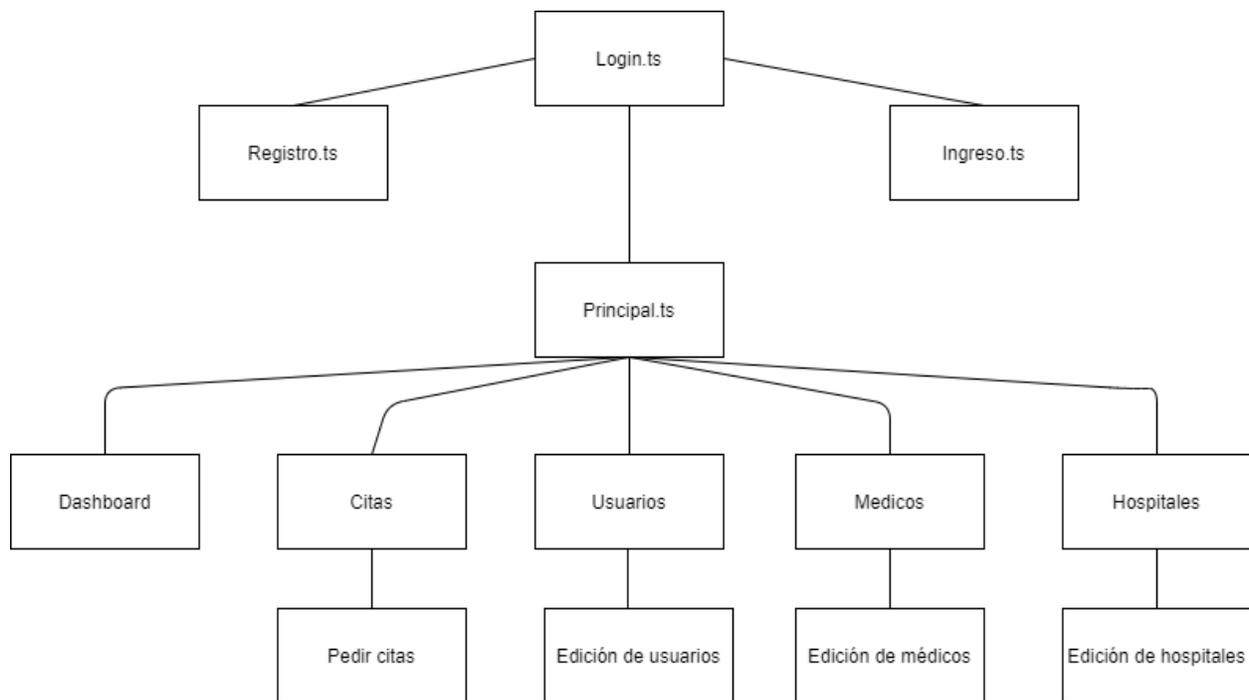


Ilustración 20 – Mapa Mind de Navegación del Sistema

Nota: realizado por los autores del presente proyecto.

Wireframes

Los Wireframes sirven para representar de forma visual la estructura y funcionalidad del sistema a un nivel muy básico.

Pantalla de inicio de sesión (login).

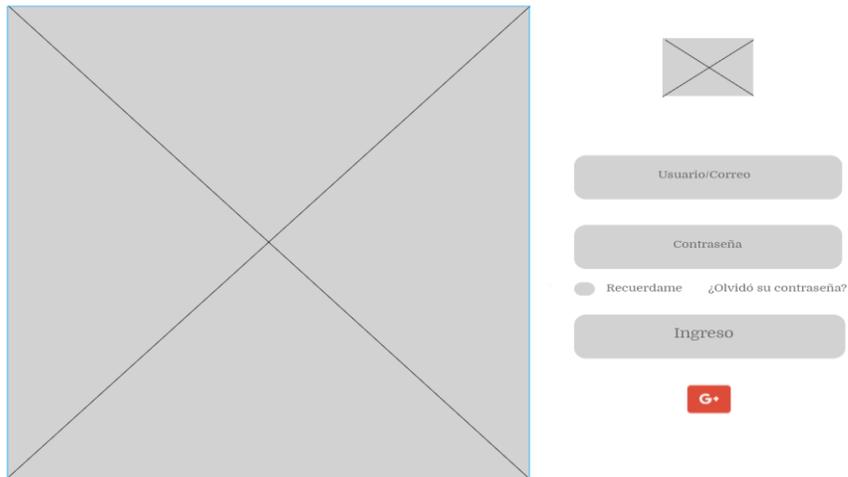


Ilustración 21 – Login (Wireframe)

Nota: realizado por los autores del presente proyecto.

Pantalla de Registro (paciente).

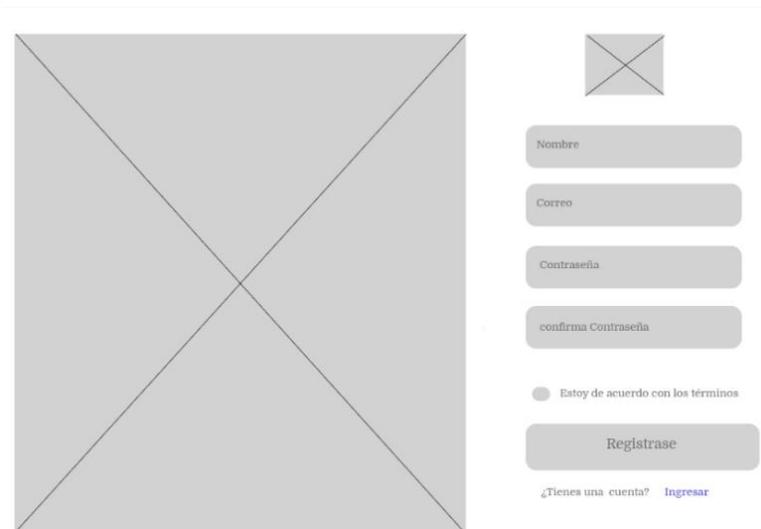


Ilustración 22– Registro paciente (Wireframe)

Nota: realizado por los autores del presente proyecto.

Pantalla de Solicitud de Cita Médica.

Wireframe of the Medical Appointment Request screen. The layout includes a sidebar on the left with a logo and navigation items, a main content area with a title and two paragraphs of placeholder text, and a right-hand sidebar with several dropdown menus for service, doctor, date, and time, plus a 'Reason for appointment' and 'Direction' field, and a 'Request appointment' button.

Ilustración 23 – Solicitud de Cita (Wireframe)

Nota: realizado por los autores del presente proyecto.

Pantalla de Consulta de Citas (login).

Wireframe of the Appointment List screen. The layout includes a sidebar on the left with a logo and navigation items, a main content area with a title, a search bar, and a table with columns for Date, Time, Name, Consultation, Edit, and Cancel. The table contains two rows of placeholder data.

Fecha	Hora	Nombre	Consulta	Editar	Cancelar
24/02/2021	15:30	Lorem ipsum	Lorem ipsum	Editar	Cancelar
10/05/2021	8:30	Lorem ipsum	Lorem ipsum	Editar	Cancelar

Ilustración 24 – Listado de citas (Wireframe)

Nota: realizado por los autores del presente proyecto.

Pantalla de Listado de Usuarios (paciente).



Ilustración 25 – Lista de usuarios (Wireframe)

Nota: realizado por los autores del presente proyecto.

Pantalla de Listado de Hospitales.



Ilustración 26 – Lista de Hospitales (Wireframe)

Nota: realizado por los autores del presente proyecto.

Pantalla de Creación de Hospitales.



Ilustración 27 – Lista de Hospitales (Wireframe)

Nota: realizado por los autores del presente proyecto.

Pantalla de Listado de Usuarios (Médico).

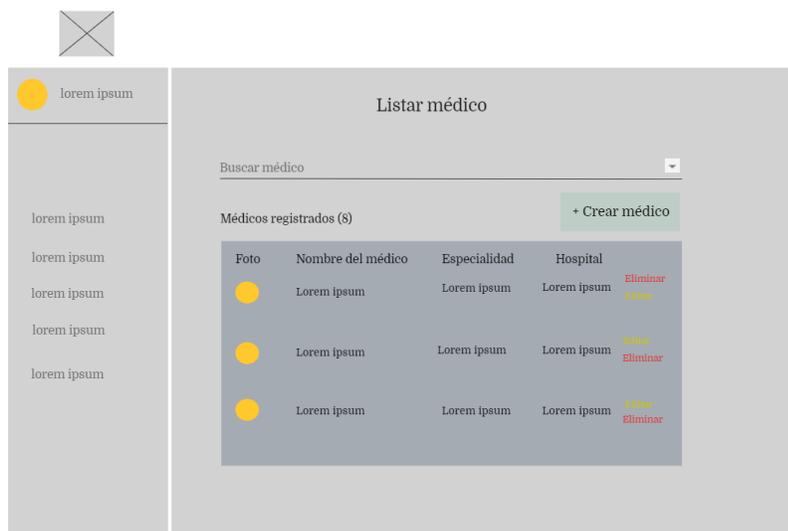


Ilustración 28– Lista de Médicos (Wireframe)

Nota: realizado por los autores del presente proyecto.

Pantalla de Registro (Médico).

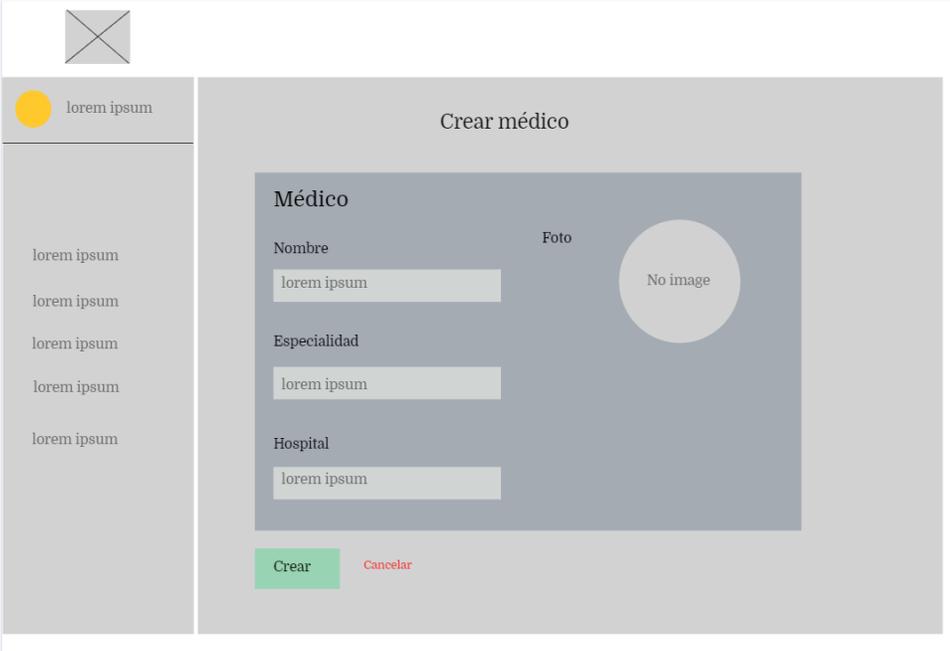


Ilustración 29 – Registro (Wireframe)

Nota: realizado por los autores del presente proyecto.

Desarrollo Del Sistema

Una vez desarrollado el sistema de acuerdo con los requerimientos, casos de uso y arquitectura planteada, se muestran los casos de prueba que se realizaron y se mostrará pantallazos del sistema para el uso de los usuarios finales del sistema.

Casos de Prueba

Se presentan los siguientes casos de prueba para corroborar el funcionamiento del sistema en cada pantalla:

<i>Caso de prueba 01</i>				
Nombre	CP-01 Registro de usuario	Caso de uso relacionado	LG-001 Ingresar datos de registro de un usuario	
Descripción	Con este caso se desea comprobar el ingreso de los datos de un usuario en el formulario de registro.			
Condiciones de ejecución	El usuario no debe haber sido registrado previamente.			
Pasos de prueba			P	F
1	Al ingresar a la vista principal, se debe hacer clic en el botón “Registrarse”.			
2	En el momento en que se muestra el formulario de registro, se espera que el usuario digite los datos de registro.			
Resultado esperado	El resultado esperado es que el usuario pueda ingresar sus datos, aceptar los términos y condiciones y al momento de hacer clic en el botón de “Registrarse”			
Evaluación de la Prueba	Se ingresó a la pantalla principal, hizo clic en el botón “Registrarse”, se completó el formulario, luego se aceptaron los términos y condiciones y por último se hizo clic en el botón “Registrarse”.			

Tabla 28 – Registro de Usuario (Casos de Prueba)

Nota: realizado por los autores del presente proyecto.

Caso de prueba 02				
Nombre	CP-02 Ingreso de usuario	Caso de uso relacionado	LOGIN 02 - Ingresar datos de un usuario para ingresar al sistema.	
Descripción	Con este caso se desea comprobar el ingreso de un usuario en el sistema.			
Condiciones de ejecución	El usuario debe haber sido registrado previamente.			
Pasos de prueba			P	F
1	Al ingresar a la vista principal, se deben llenar los campos de usuario y contraseña.			
2	Se debe hacer clic en el botón “Ingreso”.			
Resultado esperado	El resultado esperado es que el usuario pueda ingresar al sistema digitando los datos correctos en el formulario de ingreso y haga clic en el botón de “Ingreso”.			
Evaluación de la Prueba	Se ingresó a la pantalla principal, se completó el formulario de ingreso y luego se hizo clic en el botón “Ingreso”.			

Tabla 29 – Ingreso de Usuario (Casos de Prueba)

Nota: realizado por los autores del presente proyecto.

Caso de prueba 03				
Nombre	CP-03 Ingreso incorrecto de usuario	Caso de uso relacionado	LG-003 Ingresar incorrectamente los datos de un usuario para ingresar al sistema.	
Descripción	Con este caso se desea comprobar el ingreso incorrecto de los datos de un usuario en el sistema.			
Condiciones de ejecución	El usuario debe haber sido registrado previamente.			
Pasos de prueba			P	F
1	Al ingresar a la vista principal, se deben llenar los campos de usuario y contraseña incorrectos.			
2	Se debe hacer clic en el botón “Ingreso”.			

Resultado esperado	El resultado esperado es que el usuario pueda ingresar al sistema digitando los datos correctos en el formulario de ingreso y haga clic en el botón de “Ingreso”.
Evaluación de la Prueba	Se ingresó a la pantalla principal, se completó incorrectamente el formulario de ingreso y luego se hizo clic en el botón “Ingreso”.

Tabla 30 – Ingreso incorrecto de Usuario (Casos de Prueba)

Nota: realizado por los autores del presente proyecto.

Caso de prueba 04				
Nombre	CP-04 Solicitud de cita médica	Caso de uso relacionado	LOGIN 04 Ingresar datos para solicitar una cita médica.	
Descripción	Con este caso se desea comprobar el ingreso de los datos para solicitar una cita médica.			
Condiciones de ejecución	El usuario debe haber sido registrado previamente. El usuario debió haber ingresado a la pantalla “appointment”.			
Pasos de prueba			P	F
1	Al momento de acceder a la vista principal, se deben llenar los campos de usuario y contraseña. Se presiona en el botón “Ingreso”.			
2	En la vista inicial, se dirige al menú lateral y se hace clic en “Pedir citas”			
3	Se completa el formulario de solicitud de citas médicas.			
4	Hacer clic en el botón “Solicitar cita”.			
Resultado esperado	El resultado esperado es que el usuario pueda solicitar una cita médica.			
Evaluación de la Prueba	Se ingresó a la pantalla principal, se completó el formulario de solicitud de citas médicas y luego se hizo clic en el botón “Solicitar cita”.			

Tabla 31 – Solicitud de Cita Médica (Casos de Prueba)

Nota: realizado por los autores del presente proyecto.

Caso de prueba 05				
Nombre	CP-05 Visualizar citas médicas	Caso de uso relacionado	LOGIN 05 Listar las citas médicas que tiene un paciente.	
Descripción	Con este caso se desea comprobar el listado de citas médicas que puede tener un paciente.			
Condiciones de ejecución	El usuario debe haber sido registrado previamente. El usuario debió haber ingresado a la pantalla “appointment”. El usuario debió haber creado por lo menos una cita médica previamente.			
Pasos de prueba			P	F
1	Al momento de acceder a la vista principal, se deben llenar los campos de usuario y contraseña. Hacer click en el botón “Ingreso”.			
2	En la vista inicial, se dirige al menú lateral y se hace clic en “Pedir citas”.			
3	Se completa el formulario de solicitud de citas médicas.			
4	Hacer clic en el botón “Solicitar cita”.			
5	Hacer clic en el botón “Citas” del menú lateral.			
6	Visualizar el listado de citas médicas previamente creadas.			
Resultado esperado	El resultado esperado es que el usuario pueda visualizar el listado de citas médicas creadas previamente.			
Evaluación de la Prueba	Se ingresó a la pantalla principal, se completó el formulario de solicitud de citas médicas y luego se hizo clic en el botón “Solicitar cita”. Posteriormente se hace clic en el botón “Citas” del menú lateral y se visualiza el listado de citas médicas previamente creadas.			

Tabla 32 – Visualizar Citas médicas (Casos de Prueba)

Nota: realizado por los autores del presente proyecto.

Caso de prueba 06			
Nombre	CP-04 Creación de hospital	Caso de uso relacionado	LOGIN 04 Ingresar los datos para la creación de un hospital.

Descripción	Con este caso se desea comprobar el ingreso de los datos para crear un hospital dentro del sistema.		
Condiciones de ejecución	El usuario debe haber sido registrado previamente. El usuario debe hacer clic en el botón de “Mantenimiento” del menú lateral El usuario debe hacer clic en el botón “Hospitales” El usuario debe hacer clic en el botón “Crear hospital”.		
Pasos de prueba		P	F
1	Al momento de acceder a la vista principal, se deben llenar los campos de usuario y contraseña. Hacer click en el botón “Ingreso”.		
2	En la pantalla vista inicial, se dirige al menú lateral y se hace clic en “Mantenimiento”		
3	En la pantalla inicial, se dirige al menú lateral y se hace clic en “Hospitales”		
4	Hacer clic en el botón “Crear hospital”.		
5	Se ingresa el nombre del hospital a crear.		
6	Se hace clic en el botón “Aceptar” para crear el hospital.		
Resultado esperado	El resultado esperado es que el usuario pueda crear un hospital.		
Evaluación de la Prueba	Se ingresó a la pantalla principal, en la pantalla inicial, se dirige al menú lateral y se hace clic en “Mantenimiento”, se hizo clic en el botón “Crear hospital”. Se ingresó el nombre del hospital a crear y por último se hizo clic en el botón “Aceptar” para crear el hospital.		

Tabla 33 – Creación de Hospital (Casos de Prueba)

Nota: realizado por los autores del presente proyecto.

Sistema Desarrollado

Para esta última sección, se muestran brevemente las pantallas en su estado final tras su desarrollo, donde se describen brevemente.

Ingreso al Sistema

En esta pantalla, los diferentes usuarios acceden al sistema, ya sea por nombre de usuario y contraseña, o haciendo uso de algún correo Gmail.



Ilustración 30 – Pantalla de Ingreso al Sistema

Registro

En esta pantalla el paciente se registra en el sistema llenando el formulario.



Ilustración 31 – Pantalla de Registro

Solicitud de Citas

En esta pantalla se solicita una cita médica llenado el formulario.

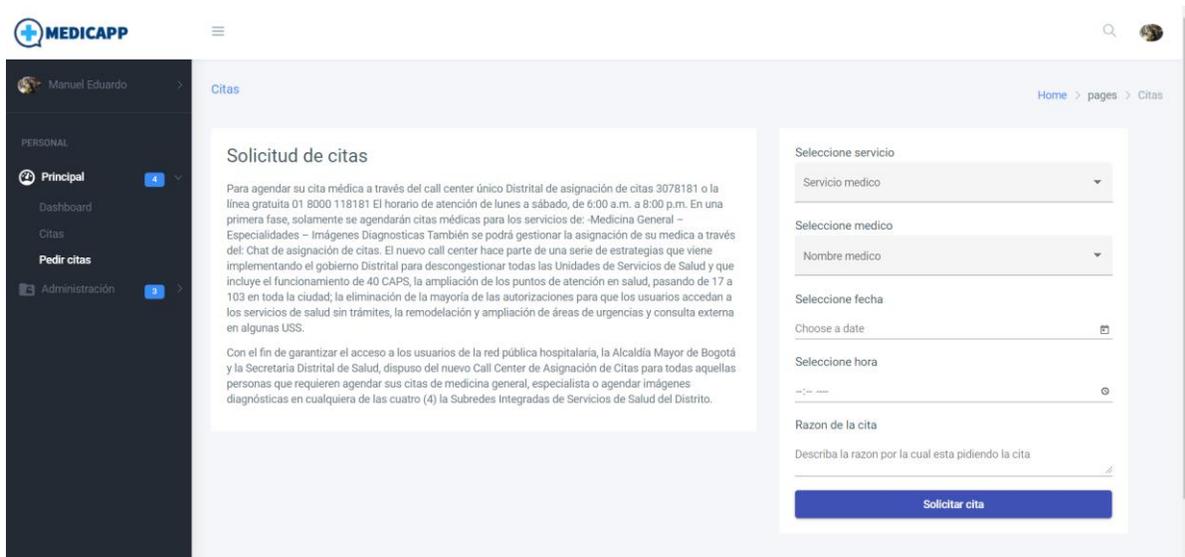


Ilustración 32- Pantalla de Solicitud de Citas

Listado de Citas

En esta pantalla se ven todas las citas relacionadas a una determinada cuenta de usuario.

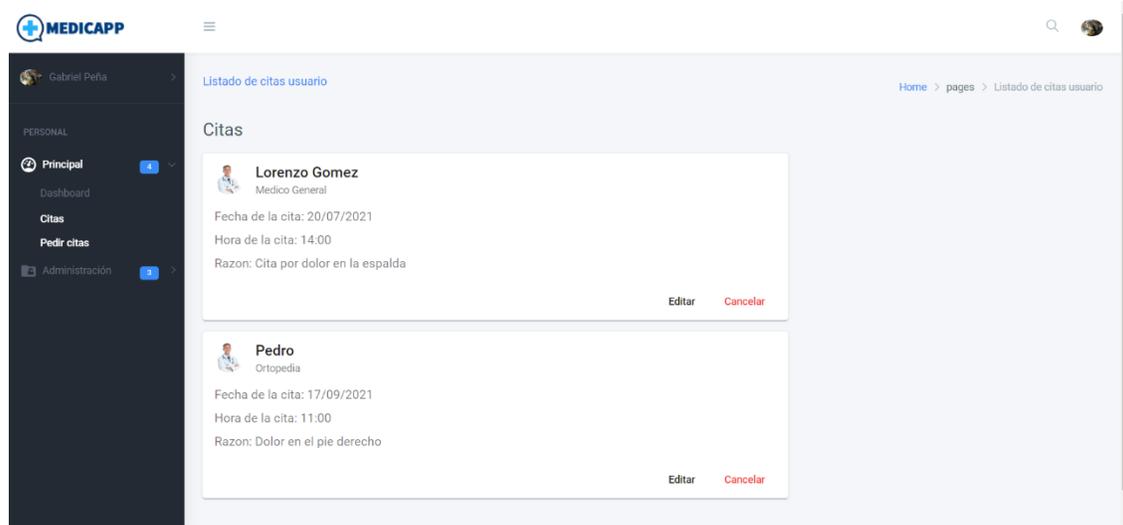


Ilustración 33 – Pantalla de Listado de Citas

Listado de Usuarios

En esta pantalla se consultan los diferentes usuarios registrados en la aplicación.

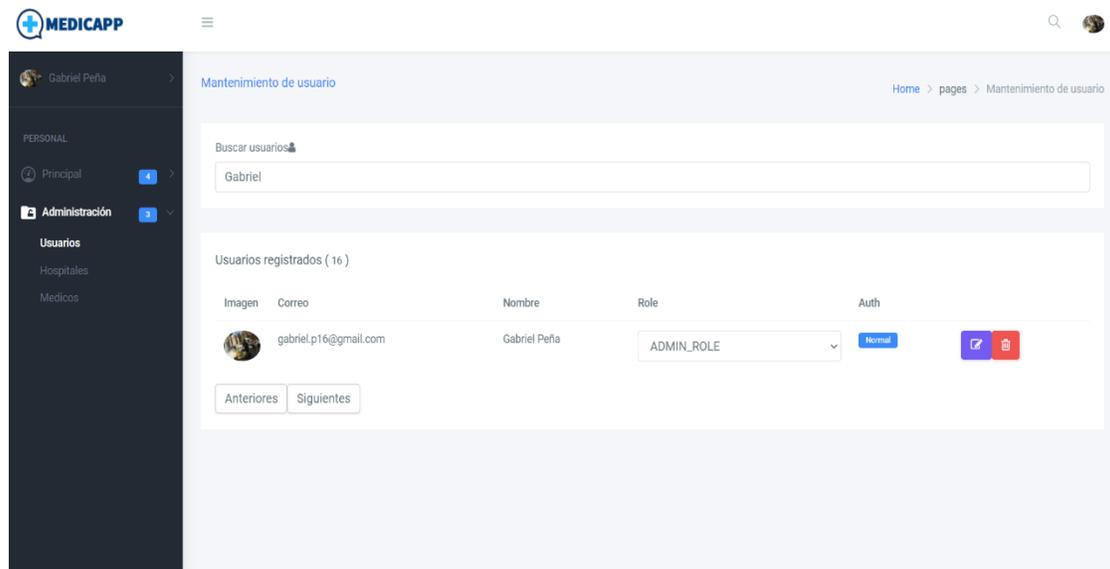


Ilustración 34 – Pantalla de Listado de Usuarios

Listado de Hospitales

En esta pantalla se consultan los diferentes hospitales registrados en el sistema.

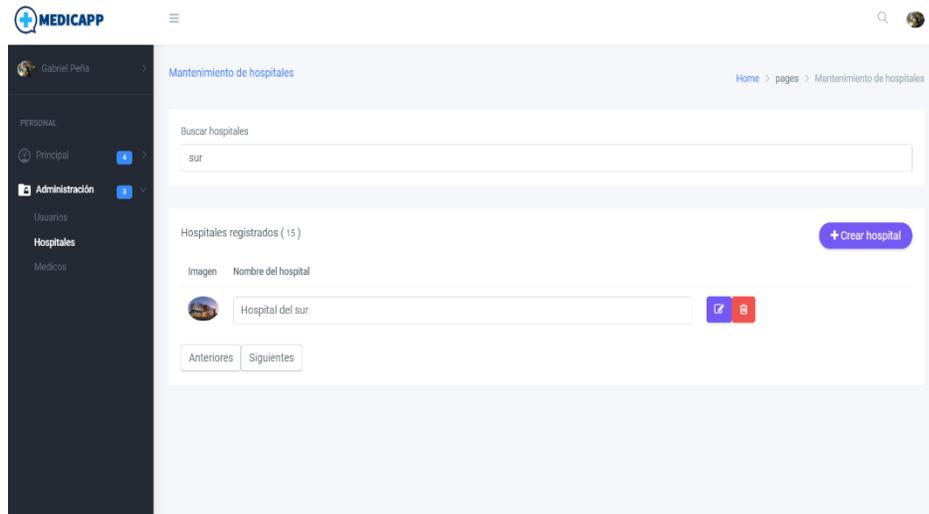


Ilustración 35 – Pantalla de Listado de Hospitales

Creación de Hospitales

En esta pantalla se registran los hospitales para luego poder asociar médicos a dicho hospital.

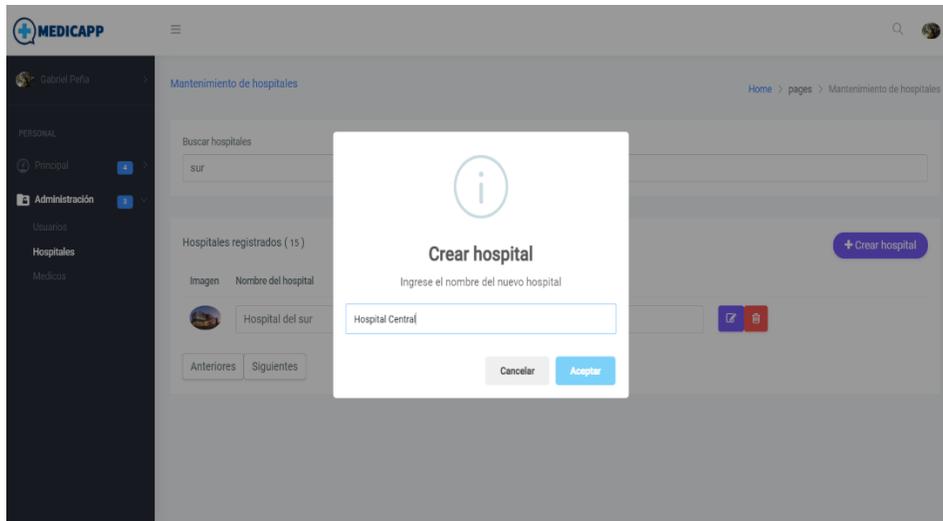


Ilustración 36 – Pantalla de Creación de Hospital

Consulta de Médicos

En esta pantalla se consultan los médicos registrados en el sistema.

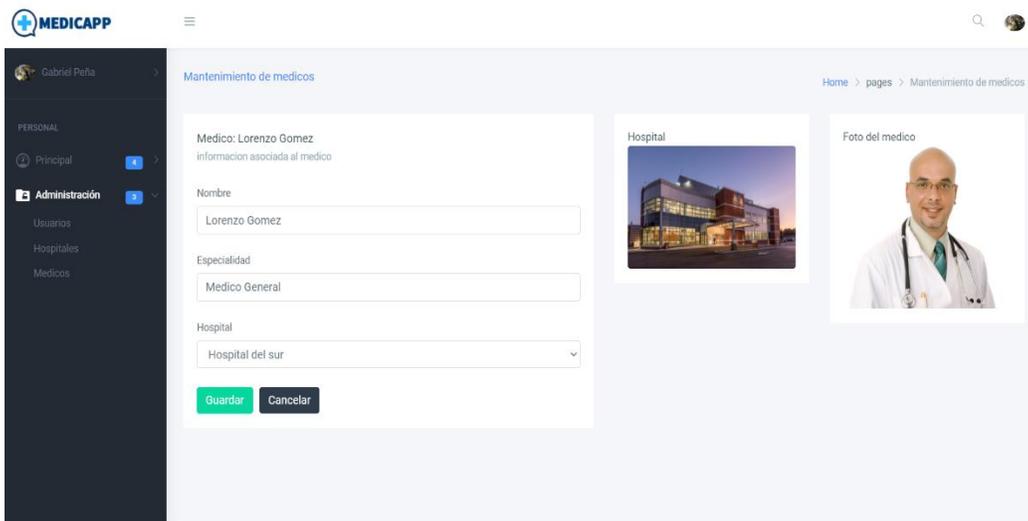


Ilustración 37 – Pantalla de Consulta de Médicos

Registro de Médicos

En esta pantalla se registran los médicos para más tarde asociarlos a un hospital.

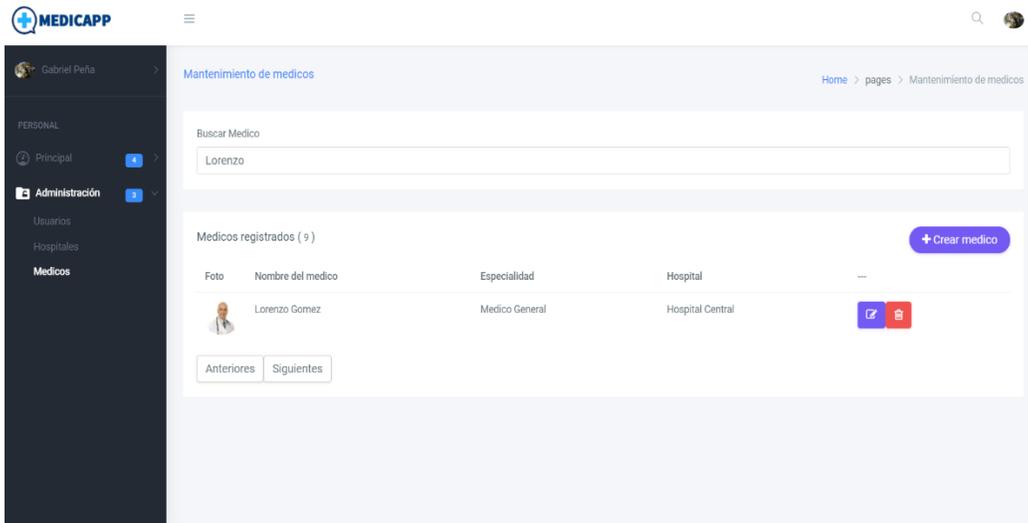


Ilustración 38 – Pantalla de Registro de Médicos

Visualización de Gráficas

En esta pantalla se muestra un dashboard con información relevante con respecto a las citas médicas.

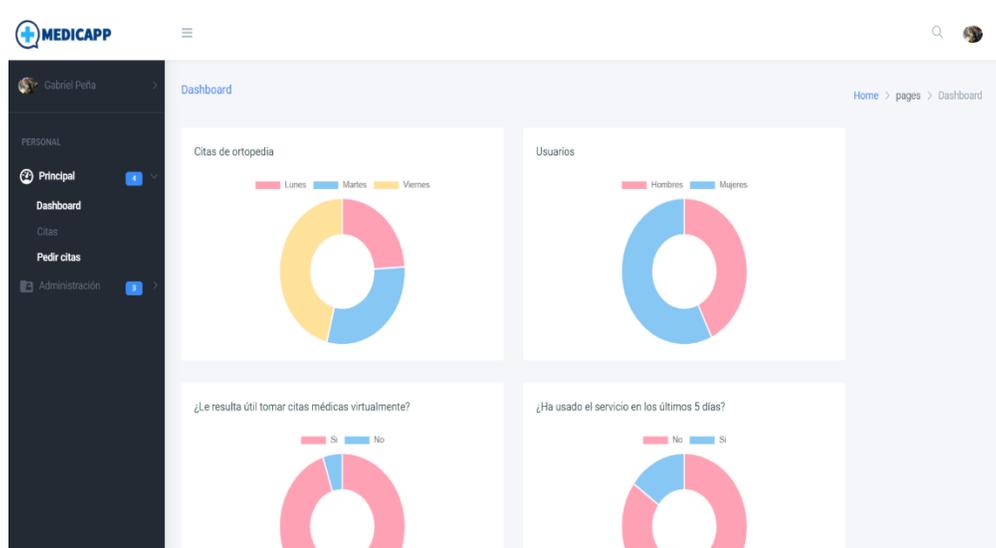


Ilustración 39 – Pantalla de Visualización de Gráficas

Glosario

Términos de Negocio Relevantes

EPS

Entidad promotora de salud la cual se encarga de promover la afiliación al sistema de seguridad social. Se refiere al servicio administrativo y comercial, no al servicio médico como tal.

Redes Públicas Hospitalarias

Redes encargadas de gestionar la prestación de los servicios de la salud de manera oportuna, eficiente y de calidad destinada a la población pobre a los cuales no los cubre el subsidio.

Centros Médicos

Es un conjunto de servicios complejos, áreas y departamentos que se ocupan y preocupan del cuidado de los ciudadanos distinguidos por su nivel de gravedad y especialidad.

Orden Médica

Es un documento escrito donde el médico prescribe servicios y/o tratamientos para el paciente en cuestión.

Régimen Subsidiario

Hace referencia al mecanismo por el cual la población más pobre del país y sin capacidad de pago, tienen acceso a los servicios de la salud mediante un subsidio ofrecido por el estado.

Población Vinculada

Es aquella población que se encuentra con puntajes por debajo de los niveles 1 y 2 del Sisbén y a su vez, no pueden ingresar a una EPS del régimen subsidiado.

Sisbén

Es el canal por el cual el Estado conoce las necesidades y capacidades de los hogares y con base a ello, determinan si un individuo puede tener acceso a un determinado beneficio.

Términos Técnicos Relevantes

Metodología ágil

Buscan distribuir de forma permanente sistemas de software en funcionamiento diseñados en iteraciones rápidas. Proporcionan diferentes maneras de pensar en colaboración y los flujos de trabajo definen valores los cuales guían las decisiones con respecto a lo que se tiene que hacer y en la manera en que se tiene que hacer.

WBS

Work Breakdown Structure, es un documento que descompone el alcance de un producto resultante de un proyecto en paquetes de trabajo en forma individual incluyendo a su vez, relativos a la gestión del proyecto en cuestión generando como resultado la muestra de todos los paquetes de trabajo que conforman el paquete del nivel superior. Al tratarse de paquetes de trabajo (no de tareas), la WBS está orientada a entregables.

Stakeholder

Son todas las personas u organizaciones de interés para una empresa o proyecto que permiten su completo funcionamiento.

Sistema

Es una clasificación que se otorga al software que corre en un determinado dispositivo.

Componente

Los componentes de un sistema hacen referencia a las entradas y salidas de un sistema. Las entradas son los elementos que permiten que el sistema pueda disponer para su propio beneficio. Las salidas son los objetivos procesados por el sistema.

Patrones

Son una solución aplicable a todo tipo de proyectos de diseño de software. Se tratan de plantillas las cuales identifican los problemas para el sistema y proporcionan soluciones justas a problemas de carácter general a los cuales se enfrenta los desarrolladores del sistema a través del desarrollo de este.

Modelos de Calidad de Software

Permiten a cualquier organización o equipo de trabajo, realizar una autoevaluación a través de una revisión sistemática de las estrategias y prácticas de gestión para asegurar la calidad del producto final.

Código libre

Es un modelo de desarrollo de software basado en la colaboración abierta que se enfocan en los beneficios prácticos que en cuestiones éticas o de libertad. A menudo se relaciona este término con la de adquirir un software de manera gratuita.

Conclusiones

Con la información recopilada sobre las nuevas tecnologías, se tomó la decisión de realizar la aplicación en AngularJS para ahorrar tiempo creando la arquitectura del proyecto y la organización que se desea tener. Por lo que una de las conclusiones a las que se llegó fue que el tipo de lenguaje que se usó es compatible con diferentes navegadores y fácil de modificar, haciendo que por cualquier motivo un desarrollador externo al proyecto pueda leer el código y continuar con la codificación.

De la misma manera, por medio del análisis inicial y conociendo a los integrantes y recursos del proyecto, se escogió la metodología adecuada para llevar a cabo el proyecto con la filosofía ágil, encaminada a llevar los procesos colaborativos y con alta cohesión en los procesos iterativos.

Por otro lado, gracias a los mockups que se realizaron previamente, se llegó a la conclusión de que estos fueron funcionales para constatar que la navegación de la aplicación fuera la esperada, y en conjunto a los estudios con varios usuarios blindando el uso y la navegación para que no se presentaran tropiezos durante el flujo.

Asimismo, por medio del patrón MVC utilizado para el desarrollo, fueron creados los modelos asociados para llevar la información a través de toda la aplicación, mientras que el controlador indicó cuáles servicios serían utilizados para persistir la información y posterior a esto, se pudiera mostrar la experiencia de usuario en las vistas.

Por su parte, las tablas de calidad ofrecieron el estado idóneo para eliminar las anomalías que se pudieron presentar en el futuro, haciendo que el código fuera limpio,

ordenado y de fácil lectura por si en cualquier momento se tuviera que asociar un desarrollador externo al proyecto para crear nuevos componentes.

Igualmente, el framework de Angular ayudó a conocer que los componentes se debían crear en un estado de desacople para que no interfirieran entre ellos, y que la manipulación entre estos fuera transparente logrando un desarrollo sustentable y expandible.

Finalmente, se trabajó en una base de datos no relacional para lograr flexibilidad al momento de crear esquemas de datos, generando una opción para almacenar y gestionar los datos no estructurados.

Referencias

- Balduino, R. (2007). *Introduction to OpenUp*.
- Cantillo, E., Rueda, M., & Fuquene, O. (2007). Diseño e implementación de un sistema de información para la asignación de citas de consulta externa en las áreas de medicina general, odontología y psicología. Bogotá.
- Ciudades y Gobiernos Locales Unidos. (2020). *Tecnologías digitales y la pandemia de COVID-19*. Barcelona. Obtenido de https://www.uclg.org/sites/default/files/eng_briefing_technology_es.pdf
- Cotino, L. (2020). Inteligencia artificial, big data y aplicaciones contra la COVID-19: privacidad y protección de datos. *Revista de Internet, Derecho y Política*.
- Franco, A. (2013). Aplicación web para la administración online de citas médicas en el centro médico de Orientación y Planificación Familiar CEMOPLAF-OTAVALO; Utilizando el patrón de arquitectura MVC en PHP.
- Gimson, L. (2012). *Metodologías ágiles y desarrollo basado en conocimiento*. La Plata: Doctoral dissertation.
- INS. (2021). *INS*. Obtenido de <https://www.ins.gov.co/Noticias/paginas/coronavirus.aspx>
- Nolasco, Y. (2019). Desarrollo de una aplicación web para el control de citas médicas del centro de salud de San Jerónimo- Andahuaylas. Apurímac, Perú.
- Noriega, R. (2015). *Procesos de desarrollo de software*. IT Campus Academy.
- OMS. (2021). *WHO*. Obtenido de <https://www.who.int/es>
- Organización Mundial de la Salud. (2019). *WHO*. Obtenido de <https://www.who.int/es/news-room/q-a-detail/coronavirus-disease-covid-19>

- Ríos, S., Hinojosa, C., & Delgado, R. (2013). *Aplicación de la metodología OpenUp en el desarrollo del sistema de difusión de gestión del conocimiento de la ESPE*. Ecuador: Escuela Politécnica del Ejercicio. Obtenido de <http://repositorio.espe.edu.ec/bitstream/21000/6316/1/AC-SISTEMAS-ESPE-047042.pdf>
- Semana. (2021). Ocupación de camas UCI en Colombia: estas son las ciudades y departamentos con la situación más crítica. *Semana*. Obtenido de <https://www.semana.com/coronavirus/articulo/ocupacion-de-camas-uci-en-colombia-estas-son-la-ciudades-y-departamentos-con-la-situacion-mas-critica/202152/>
- Sommerville, I. (2005). *Ingeniería del Software*. Alicante: Pearson.
- Tolentino, E. (2018). Aplicación Web basada en el patrón MVC para mejorar la gestión de citas médicas en la clínica Belén. Juan Pablo II, Chimbote. Perú.
- Vidal, C., Schmal, R., Rivero, S., & Villarroel, R. (2012). Vidal, C. L., Schmal, R. F., Rivero, S., & Villarroel, R. H. (2012). Extensión del Diagrama de Secuencias UML (Lenguaje de Modelado Unificado) para el Modelado Orientado a Aspectos. *Información tecnológica*, 23(6), 51-62. Obtenido de <https://scielo.conicyt.cl/pdf/infotec/v23n6/art07.pdf>

Anexos

Anexo 1 – Cronograma del Proyecto

		Junio				Julio				Agosto				Septiembre				Octubre				Noviembre				Diciembre							
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Disciplina	Ítem de trabajo	Concepción				Elaboración				Construcción				Construcción				Transición															
Gerencia de proyecto	PLAN DE PROYECTO																																
Gerencia de proyecto	Lista de riesgos																																
Gerencia de proyecto	Plan del proyecto																																
Requerimientos	GLOSARIO																																
Requerimientos	Capturas vocabulario técnico y del negocio																																
Requerimientos	Vision																																
Requerimientos	Identificación de los stakeholders																																
Requerimientos	Recolectar requerimientos de los Stakeholders																																
Requerimientos	Definir límites del sistema																																
Requerimientos	Definir características del sistema																																
Requerimientos	CASOS DE USO																																
Requerimientos	Recolectar información																																
Requerimientos	Capturar requerimientos																																
Requerimientos	Capturar casos de uso y actores																																
Requerimientos	REQUERIMIENTOS GENERALES																																
Requerimientos	Detallar requerimientos generales																																
Requerimientos	Requisitos de interfaz de usuario																																
Requerimientos	Reglas del negocio																																
Requerimientos	Limitaciones del sistema																																
Requerimientos	Cumplimiento del sistema																																
		Junio				Julio				Agosto				Septiembre				Octubre				Noviembre				Diciembre							
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Arquitectura	CUADERNO DE ARQUITECTURA	Concepción				Elaboración				Construcción				Construcción				Transición															
Arquitectura	Identificar metas arquitectónicas																																
Arquitectura	Identificar restricciones arquitectónicas																																
Arquitectura	Mecanismos arquitectónicos																																
Desarrollo	DISEÑO																																
Desarrollo	Modelo de datos																																
Desarrollo	Diccionario de datos																																
Desarrollo	Diseño de pantallas																																
Desarrollo	Refinar diseño de pantallas																																
Desarrollo	Mapa de navegación																																
Desarrollo	CONSTRUCCIÓN																																
Desarrollo	Hacer un prototipo de la arquitectura (Prueba de concepto)																																
Desarrollo	Escribir código de arquitectura base																																
Desarrollo	Escribir código fuente (Requisitos de alta prioridad)																																
Desarrollo	RE-04 Registro de Usuario																																
Desarrollo	RE-08 Creacion de médicos																																
Desarrollo	RE-16 Solicitud de Citas																																
Desarrollo	RE-17 Cancelacion de Citas																																
Desarrollo	RE-09 Modificación de médicos																																
Desarrollo	Escribir código fuente (Requisitos de prioridad normal)																																
Desarrollo	RE-01 Ingresar al sistema																																
Desarrollo	RE-05 Modificacion de Usuarios																																
		Junio				Julio				Agosto				Septiembre				Octubre				Noviembre				Diciembre							
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Test	CASOS, GUIÓN Y REGISTROS DE PRUEBAS	Concepción				Elaboración				Construcción				Construcción				Transición															
Test	Crear los casos de prueba																																
Test	Probar el prototipo de arquitectura																																
Test	Probar arquitectura de referencia (probar que los requisitos de alta prioridad funcionan con la arquitectura)																																

Anexo 2 – Encuesta

Medicaap

16 respuestas

[Publicar análisis](#)

¿Cuenta usted con un dispositivo móvil o computadora con acceso a internet?

16 respuestas



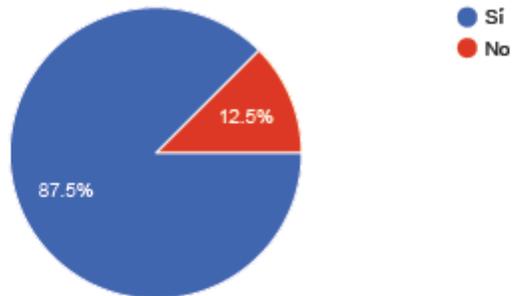
¿Está afiliado ahora mismo a un plan médico?

16 respuestas



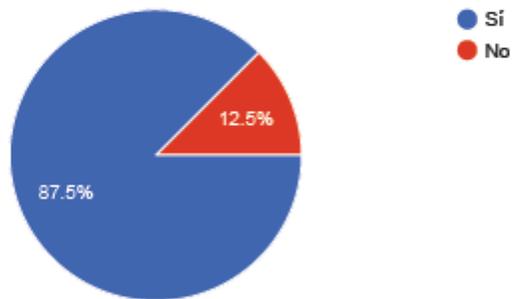
Sí su respuesta en la pregunta anterior fue afirmativa, ¿está usted satisfecho con el servicio al cual está afiliado?

16 respuestas



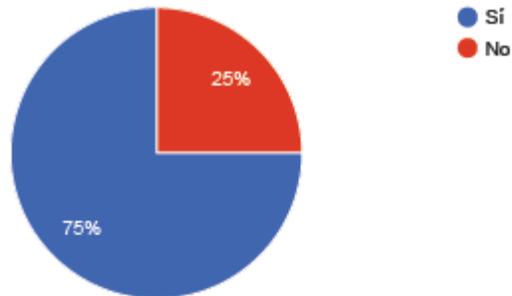
¿Usted considera útil poder realizar una cita o control médico domiciliar debido a la pandemia?

16 respuestas



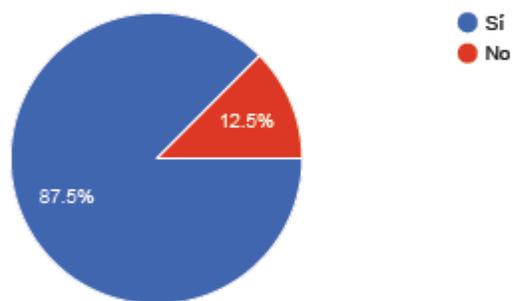
¿Cuenta usted con la posibilidad de pedir una cita o control médico domiciliada al servicio al cual está afiliado?

16 respuestas



¿Estaría usted dispuesto a recibir atención médica virtual?

16 respuestas



Google no creó ni aprobó este contenido. [Denunciar abuso](#) - [Condiciones del Servicio](#) - [Política de Privacidad](#)

Google Formularios

Anexo 3 - Resolución 1552

REPÚBLICA DE COLOMBIA



MINISTERIO DE SALUD Y PROTECCIÓN SOCIAL

RESOLUCIÓN NÚMERO 10001552 DE 2013

(14 MAY 2013)

Por medio de la cual se reglamentan parcialmente los artículos 123 y 124 del Decreto - Ley 019 de 2012 y se dictan otras disposiciones

EL MINISTRO DE SALUD Y PROTECCIÓN SOCIAL

En ejercicio de las facultades legales, en especial, de las conferidas en los artículos 123 y 124 del Decreto - Ley 019 de 2012, y

CONSIDERANDO

Que el artículo 123 del Decreto - Ley 019 de 2012 establece que "Las Entidades Promotoras de Salud, EPS, deberán garantizar la asignación de citas de medicina general u odontología general, sin necesidad de hacer la solicitud de forma presencial y sin exigir requisitos no previstos en la Ley. La asignación de estas citas no podrá exceder los tres (3) días hábiles contados a partir de la solicitud. De igual forma, las EPS contarán con sistemas de evaluación y seguimiento a los tiempos de otorgamiento de citas que deberán reportarse a la Superintendencia Nacional de Salud y publicarse periódicamente en medios masivos de comunicación. (...)".

Que así mismo, el artículo 124, ibidem, señala que: "La asignación de citas médicas con especialistas deberá ser otorgada por las Empresas Promotoras de Salud en el término que señale el Ministerio de Salud y Protección Social, la cual será adoptada en forma gradual, atendiendo la disponibilidad de oferta por especialidades en cada región del país, la carga de la enfermedad de la población, la condición médica del paciente, los perfiles epidemiológicos y demás factores que incidan en la demanda de prestación del servicio de salud por parte de la población colombiana. (...)".

Que las precitadas disposiciones facultaron a este Ministerio para reglamentar lo pertinente y especialmente, el artículo 124 ya citado, se pronunció sobre la necesidad de que su reglamentación obedezca a criterios de gradualidad, así como a las condiciones y factores allí establecidos para la prestación del servicio de salud.

En mérito de lo expuesto,

RESUELVE:

Artículo 1.- Agendas abiertas para asignación de citas. Las Entidades Promotoras de Salud – EPS de ambos regímenes, directamente o a través de la red de prestadores que definan, deberán tener agendas abiertas para la asignación de citas de medicina especializada la totalidad de días hábiles del año. Dichas entidades en el momento en que reciban la solicitud, informarán al usuario la fecha para la cual se asigna la cita, sin que les sea permitido negarse a recibir la solicitud y a fijar la fecha de la consulta requerida.

Parágrafo 1. En los casos en que la cita por medicina especializada requiera autorización previa por parte de la Entidad Promotora de Salud - EPS, ésta deberá dar respuesta sin exceder los cinco (5) días hábiles, contados a partir de la solicitud.

14 MAY 2013

RESOLUCIÓN NÚMERO 0001552 DE 2013

HOJA No 2 de 3

Continuación de la resolución "Por medio de la cual se reglamentan parcialmente los artículos 123 y 124 del Decreto - Ley 019 de 2012"

Parágrafo 2. Cuando por la condición clínica del paciente, especialmente, tratándose de gestantes y de pacientes que presenten diagnóstico presuntivo o confirmado de cáncer, el profesional tratante defina un término para la consulta especializada, la Entidad Promotora de Salud – EPS gestionará la cita, buscando que la misma sea asignada, en lo posible, dentro del término establecido por dicho profesional.

Parágrafo 3. La asignación de las citas de odontología general y medicina general, no podrá exceder los tres (3) días hábiles, contados a partir de la solicitud, salvo que el paciente las solicite de manera expresa para un plazo diferente. Dichas entidades en el momento en que reciban la solicitud, informarán al usuario la fecha para la cual se asigna la cita, sin que les sea permitido negarse a recibir la solicitud y a fijar la fecha de la consulta requerida.

Artículo 2.- Obligación de registro. Las Entidades Promotoras de Salud – EPS de ambos regímenes, directamente o a través de la red de prestadores, estarán obligadas a disponer de un sistema de información para las citas de odontología, medicina general y medicina especializada, en el que se registren los siguientes datos: (i) la identificación del usuario y datos de contacto, (ii) la fecha en que el usuario solicita la cita, (iii) la fecha en que el usuario solicita le sea asignada la cita; (iv) la fecha para la cual se asigna la cita y (v) institución prestadora de servicios de salud donde se asigna la cita, identificándola con el código del Registro Especial de Prestadores de Servicios de Salud.

Parágrafo. Las Entidades Promotoras de Salud – EPS de ambos regímenes, directamente o a través de la red de prestadores, deberán contar con mecanismos no presenciales para recibir la solicitud y asignar las citas de odontología, medicina general y medicina especializada e incrementar su cobertura de manera progresiva.

Artículo 3.- Medición de la oportunidad de citas. Con base en el registro de que trata el artículo 2 de la presente resolución, las Entidades Promotoras de Salud - EPS de ambos regímenes, mensualmente deberán cuantificar la siguiente información para las citas de odontología, medicina general y medicina especializada, asignadas en el mes anterior a la cuantificación, discriminada por tipo, especialidad y departamento o distrito del domicilio del solicitante, así:

1. El número total de citas asignadas.
2. Sumatoria de la diferencia de días entre la fecha para la cual se asignó la cita y la fecha en la cual el usuario solicitó la cita.
3. Sumatoria de la diferencia de días entre la fecha para la cual se asignó la cita y la fecha para la cual el usuario solicitó le fuera asignada.
4. Tiempo promedio de espera según fecha en que se solicita la cita: sumatoria de la diferencia de días entre la fecha en que se asignó la cita y la fecha en la cual el usuario la solicitó (3.2) / Número de citas asignadas (3.1).
5. Tiempo promedio de espera según fecha para la cual se solicita la cita: sumatoria de la diferencia de días entre la fecha para la cual se asignó la cita y la fecha para la cual el usuario solicitó le fuera asignada (3.3) / Número de citas asignadas (3.1).
6. Teniendo en cuenta los datos utilizados para el tiempo promedio de espera, se deberá cuantificar el número mínimo y máximo de días de espera para las citas asignadas durante el mes anterior a la cuantificación, discriminado por tipo de especialidad.

Continuación de la resolución "Por medio de la cual se reglamentan parcialmente los artículos 123 y 124 del Decreto - Ley 019 de 2012"

7. Número de horas-especialista, contratadas o disponibles para cada especialidad en el mes anterior a la cuantificación.

Artículo 4.- Publicación de tiempos de espera. La información de tiempos de espera de medicina especializada, cuantificada con la metodología definida en el artículo 3 de la presente resolución, deberá mantenerse actualizada y publicarse dentro de los diez (10) primeros días hábiles de cada mes en la página web de la respectiva Entidad Promotora de Salud - EPS. Esta información deberá estar disponible para ser consultada a través de diferentes canales de información.

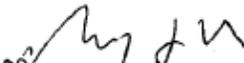
Artículo 5.- Optimización de la oportunidad de la consulta médica especializada. Las Entidades Promotoras de Salud - EPS de ambos regímenes y las Instituciones Prestadoras de Servicios de Salud - IPS, incluirán en su Programa de Auditoría para el Mejoramiento de la Calidad, el análisis periódico de la información generada con aplicación de lo dispuesto en la presente resolución, así como de la pertinencia de la remisión a consulta especializada y de la contra remisión oportuna al médico general, para proceder a adecuar su red y optimizar la oportunidad de la consulta médica especializada. Esta información podrá ser solicitada en cualquier momento tanto por el Ministerio de Salud y Protección Social, como por la Superintendencia Nacional de Salud.

Artículo 6.- Vigilancia y control. El cumplimiento de lo dispuesto en la presente resolución estará a cargo de la Superintendencia Nacional de Salud y demás organismos de inspección, vigilancia y control, según las competencias definidas en las normas vigentes.

Artículo 7.- Vigencia y derogatorias. La presente resolución entrará en vigencia a partir de su publicación, salvo lo dispuesto en los artículos 2, 3, 4 y 5, que entrarán a regir cuatro (4) meses después, término éste a partir del cual, las Entidades Promotoras de Salud - EPS de ambos regímenes, deberán dar aplicación a lo allí dispuesto. Igualmente, deroga las disposiciones que le sean contrarias.

PUBLÍQUESE Y CÚMPLASE

Dada en Bogotá, D.C., a los 14 MAY 2013


ALEJANDRO GAVIRIA URIBE
Ministro de Salud y Protección Social

Anexo 4 - Diccionario de datos

Las siguientes tablas explican los documentos para la base de datos en MongoDB:

USUARIO			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	ObjectId	No	Identificador del usuario
correo	String	No	Correo electrónico del usuario
contrasena	String	No	Contraseña para ingresar al sistema
nombres	String	No	Nombres del usuario
apellidos	String	No	Apellidos del usuario
activo	Boolean	Si	Estado activo del usuario

PACIENTE			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
cedula	String	No	Cedula del paciente
direccion	String	Si	Dirección del paciente
telefono	String	Si	Teléfono del paciente
grupo_sanguineo	String	No	Grupo sanguíneo del paciente
nombre_eps	String	No	Nombre de la EPS a la que está asociado el paciente
edad	Int	No	Edad del paciente
sexo	String	Si	Sexo del paciente
acudiente_nombre	String	Si	Nombre del acudiente que cuidara al paciente
acudiente_telefono	String	Si	Teléfono del acudiente que cuidara al paciente

MEDICO			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
cedula	String	No	Cedula del paciente

telefono	String	Si	Teléfono del paciente
codigo	String	No	Código de identificación del médico profesional
especialidad	String	No	Especialidad médica
titulo	String	Si	Título universitario / Especialización
universidad	String	Si	Nombre de la universidad de egreso

HISTORIA_CLINICA			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	String	No	Identificador del historial clínico

ROL			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador del rol
nombre	String	No	Nombres del rol

ROL_PERMISO			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador del rol y su permiso

PERMISO			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador del permiso
nombre	String	No	Nombres del permiso

CITA			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador de la cita

hora_inicio	String	Si	Hora de inicio de la cita
hora_fin	String	Si	Hora de finalización de la cita
fecha	String	Si	Fecha de la cita
eps	String	No	Nombre de la EPS
motivo	String	Si	Motivo por el cual se solicita la cita

VIDEO_LLAMADA			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador de la video llamada

SERVICIO			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador del servicio
nombre	String	No	Nombres del servicio
descripcion	String	Si	Descripción del servicio médico que se va a prestar

DETALLE_CITA			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador del servicio
notas	String	No	Nombres del servicio
medicamentos	String	Si	Descripción del servicio médico que se va a prestar
equipo_utilizado	String	Si	Equipo médico que se utilizó en la cita

ANTECEDENTES			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador de los antecedentes
restricciones	String	No	Restricciones que puede tener el paciente

alergenos	String	Si	Alérgenos que no soporta el paciente
antecedentes	String	Si	Antecedentes del paciente
antecedentes_familiares	String	Si	Antecedentes familiares

TRATAMIENTO			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador del tratamiento
descripcion	String	No	Descripción del tratamiento

SEGUIMIENTO			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador del seguimiento
descripcion	String	No	Descripción del seguimiento

EXAMENES			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador del examen
descripcion	String	No	Descripción del examen

DIAGNOSTICO			
Nombre del campo	Tipo de dato	Acepta NULO	Descripción
id	Int	No	Identificador del diagnóstico
descripcion	String	No	Descripción del diagnóstico

Anexo 5 – Manual de Instalación

Inicialmente es necesario descargar NodeJs desde la página oficial, <https://nodejs.org/es/download/>

A screenshot of a Google search for "node js download". The search results show the official Node.js download page as the top result. The page title is "Descarga | Node.js" and the description mentions the current version is 14.17.3. There are links for "Actual Últimas características" and "Versiones Anteriores". Below the search results, there are "Preguntas relacionadas" (related questions) such as "How do I download node JS?" and "What is the latest node js version?".

Se selecciona la versión LTS para obtener la versión más estable.

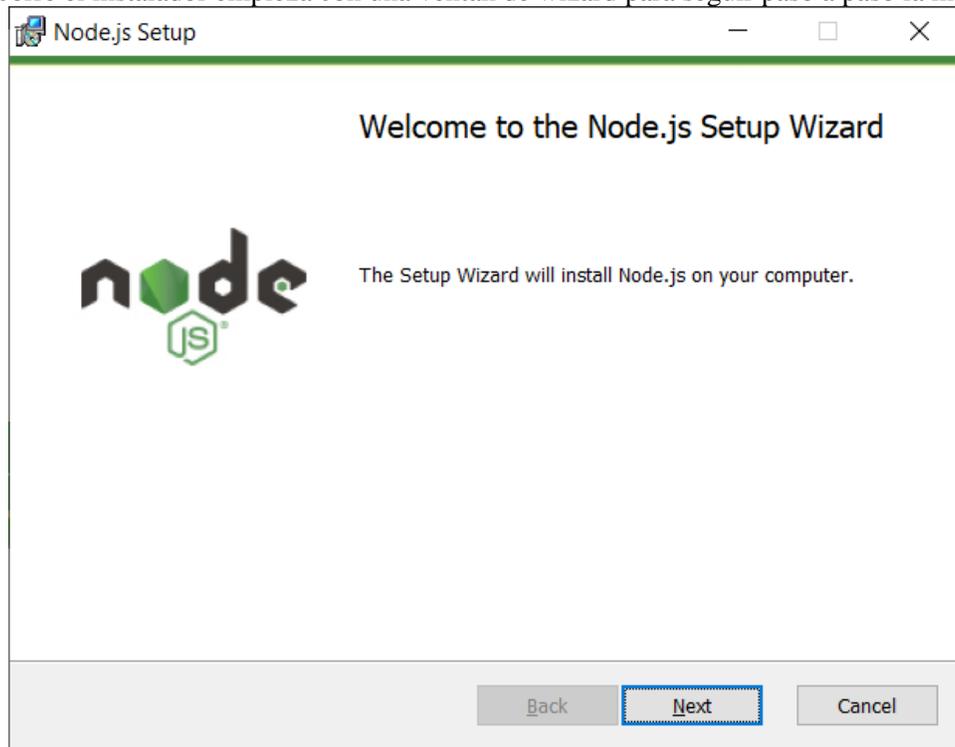
A screenshot of the Node.js download page. The page features the Node.js logo at the top, followed by navigation links: INICIO, ACERCA, DESCARGAS, DOCUMENTACIÓN, PARTICIPE, SEGURIDAD, NOTICIAS, and CERTIFICATION. The main heading is "Descargas" with the current version "14.17.3 (includes npm 6.14.13)". Below this, there is a section for "LTS" (Recommended for most) and "Actual" (Latest features). Under "LTS", there are links for "Instalador Windows" (node-v14.17.3-x64.msi) and "Instalador macOS" (node-v14.17.3.pkg). Under "Actual", there is a link for "Código Fuente" (node-v14.17.3.tar.gz). A table below lists the available binaries for Windows and Linux (x64) in 32-bit and 64-bit formats.

	32-bit	64-bit
Instalador Windows (.msi)	32-bit	64-bit
Binario Windows (.zip)	32-bit	64-bit
Instalador macOS (.pkg)		64-bit
Binario macOS (.tar.gz)		64-bit
Binario Linux (x64)		64-bit

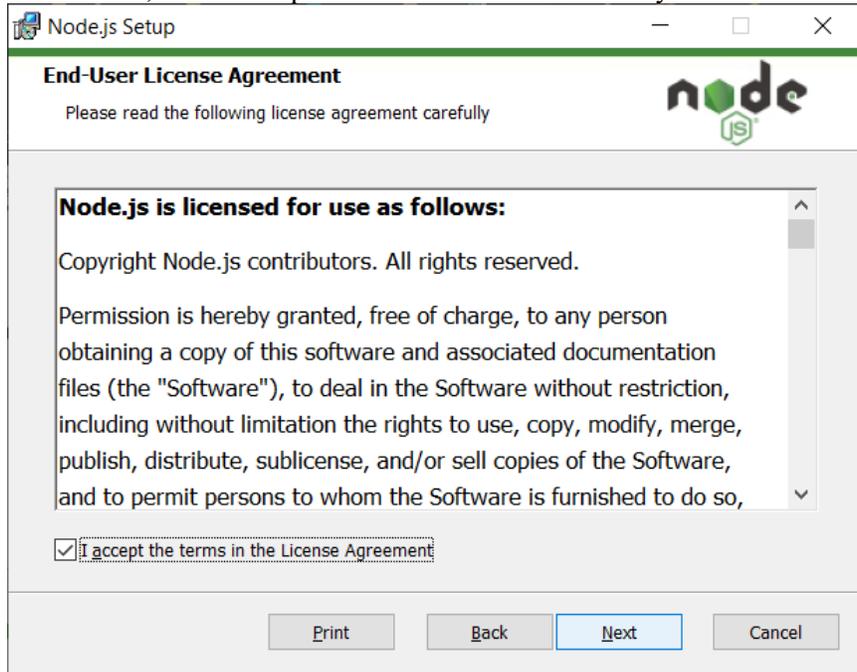
Se descarga la versión MSI para el sistema operativo Windows de X64.

The screenshot shows the Node.js website's download page. At the top, there is a navigation bar with links: INICIO, ACERCA, DESCARGAS, DOCUMENTACIÓN, PARTICIPE, SEGURIDAD, NOTICIAS, and CERTIFICATION. The main heading is "Descargas" (Downloads), with the current version noted as "Versión actual: 14.17.3 (includes npm 6.14.13)". A sub-heading says "Descargue el código fuente de Node.js o un instalador pre-compilado para su plataforma, y comience a desarrollar hoy." Below this, there are two main sections: "LTS" (Recommended for the majority) and "Actual" (Latest features). Under "LTS", there is a button for "Instalador Windows" (node-v14.17.3-x64.msi). Under "Actual", there are buttons for "Instalador macOS" (node-v14.17.3.pkg) and "Código Fuente" (node-v14.17.3.tar.gz). A table below lists download options for Windows and macOS, with columns for 32-bit and 64-bit. A browser address bar at the bottom shows the URL: https://nodejs.org/dist/v14.17.3/node-v14.17.3-x64.msi.

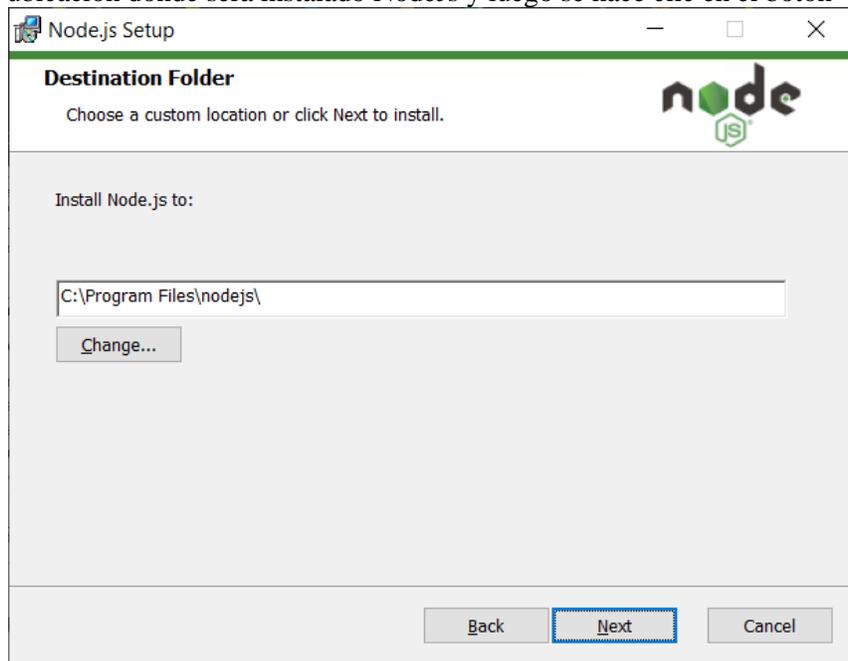
Cuando se corre el instalador empieza con una ventana de wizard para seguir paso a paso la instalación



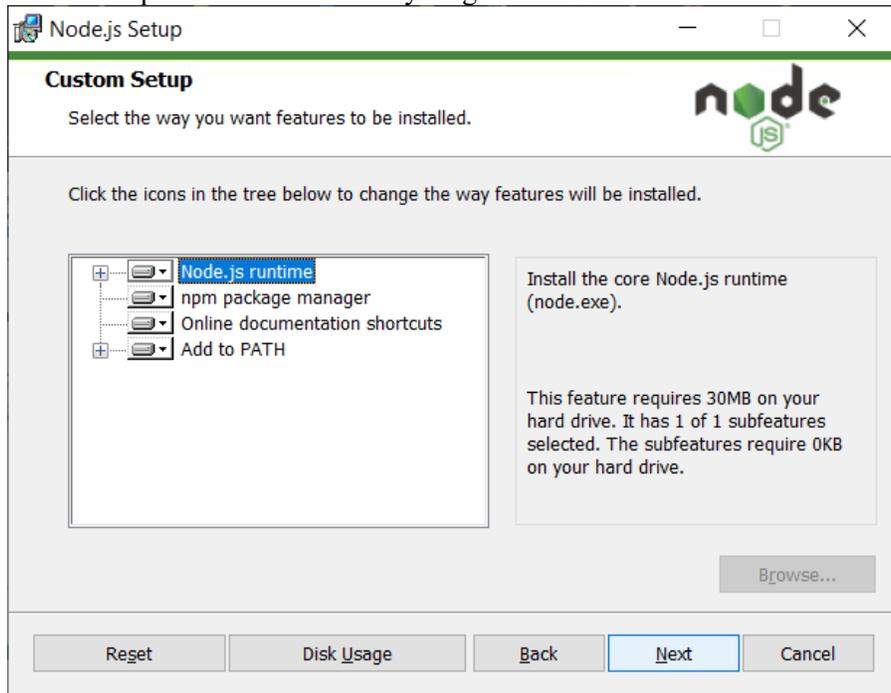
Al dar inicio a la instalación, solicita aceptar los términos de la licencia y hacer clic en el botón “Next.”



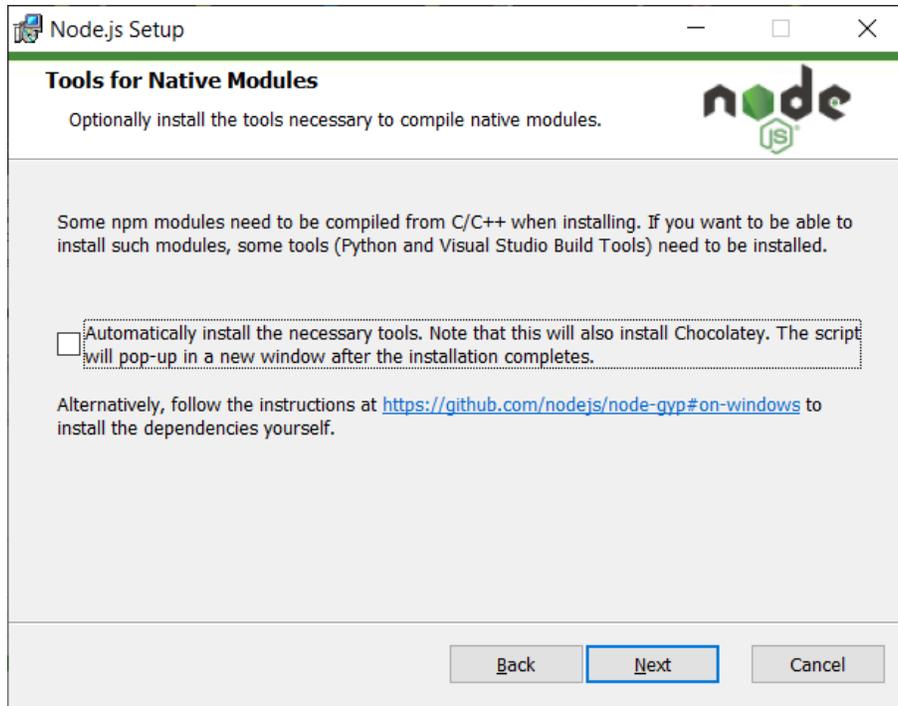
Se selecciona la ubicación donde será instalado NodeJs y luego se hace clic en el botón “Next”.



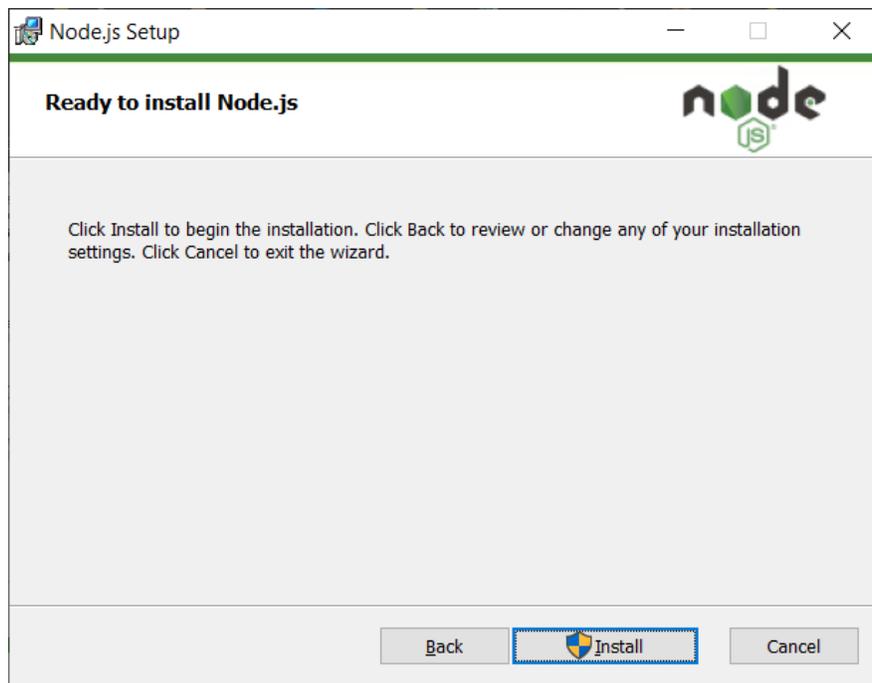
Se seleccionan los componentes que son necesarios para continuar con la instalación, en este caso es necesario tener todas las opciones seleccionadas y luego hacer clic en el botón “Next”



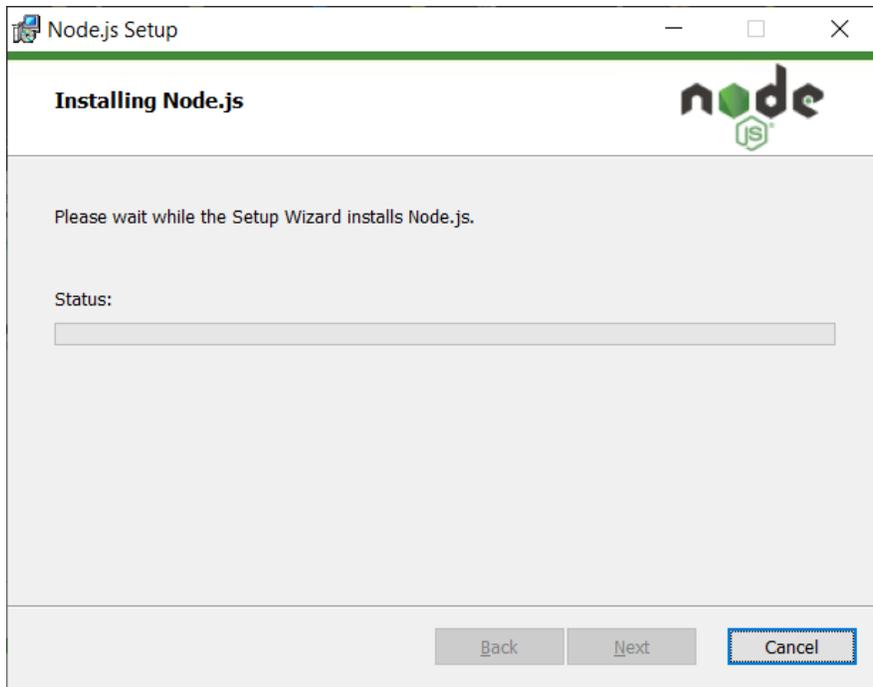
En esta pantalla se busca asociar la instalación al componente “Chocolatey” pero por este momento no es necesario tenerlo instalado, entonces se hace clic en el botón “Next”



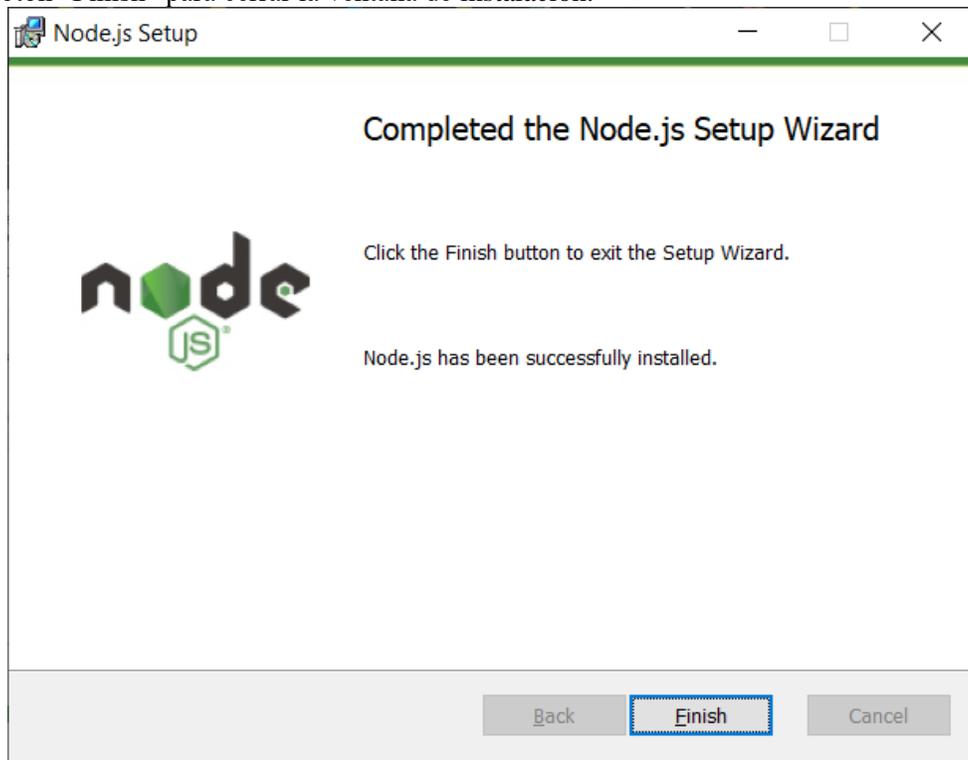
Con las opciones ya configuradas, se da inicio a la instalación de NodeJs haciendo clic en el botón “Install”.



Se inicia el proceso de instalación hasta que la barra de progreso se complete.



Se culmina la instalación de NodeJs y aparece el mensaje que fue instalado correctamente. Luego se hace clic en el botón "Finish" para cerrar la ventana de instalación.



En la consola de Git se verifica la versión de NodeJs con el comando “node -v”.

```
User@DESKTOP-LTIS9MP MINGW64 ~/Desktop/Dropbox/Gabito/USBB0G/Semestre 10/Tesis/medic-app/api
$ node -v
v14.17.3
```

Luego se instala el gestor de paquetes desarrollado íntegramente en JavaScript con la versión más reciente que se encuentra en la red.

```
User@DESKTOP-LTIS9MP MINGW64 ~/Desktop/Dropbox/Gabito/USBB0G/Semestre 10/Tesis/medic-app/api
$ npm install npm@latest -g
C:\Users\User\AppData\Roaming\npm\npm -> C:\Users\User\AppData\Roaming\npm\node_modules\npm\bin\npm-cli.js
C:\Users\User\AppData\Roaming\npm\npx -> C:\Users\User\AppData\Roaming\npm\node_modules\npm\bin\npx-cli.js
+ npm@7.19.1
added 62 packages from 26 contributors, removed 241 packages and updated 194 packages in 14.254s
```

Se crea un documento donde se instancian las dependencias necesarias para la instalación del proyecto.

```
"dependencies": {
  "@angular/animations": "^11.2.11",
  "@angular/cdk": "^11.2.10",
  "@angular/common": "^11.2.11",
  "@angular/compiler": "^11.2.11",
  "@angular/core": "^11.2.11",
  "@angular/forms": "^11.2.11",
  "@angular/material": "^11.2.10",
  "@angular/platform-browser": "^11.2.11",
  "@angular/platform-browser-dynamic": "^11.2.11",
  "@angular/router": "^11.2.11",
  "@kolkov/angular-editor": "^1.1.4",
  "chart.js": "^2.9.4",
  "file-saver": "^2.0.5",
  "html-docx-js": "^0.3.1",
  "html2pdf.js": "^0.9.3",
  "jspdf": "^1.5.3",
  "ng2-charts": "^2.4.2",
  "rxjs": "~6.5.5",
  "rxjs-compat": "^6.6.7",
  "sweetalert": "^2.1.2",
  "tslib": "^2.0.0",
  "websocket-extensions": "^0.1.4",
  "zone.js": "~0.10.2"
},
```

También se asocian las dependencias de desarrollo para el proyecto

```
"devDependencies": {
  "@angular-devkit/build-angular": "^0.1102.10",
  "@angular/cli": "^11.2.10",
  "@angular/compiler-cli": "^11.2.11",
  "@angular/language-service": "^11.2.11",
  "@types/jasmine": "~3.6.0",
  "@types/jasminewd2": "~2.0.3",
  "@types/node": "^12.20.11",
  "html-docx-js-typescript": "^0.1.5",
  "jasmine-core": "~3.6.0",
  "jasmine-spec-reporter": "~5.0.0",
  "karma": "~5.0.0",
  "karma-chrome-launcher": "~3.1.0",
  "karma-coverage-istanbul-reporter": "~3.0.2",
  "karma-jasmine": "~4.0.0",
  "karma-jasmine-html-reporter": "^1.5.0",
  "protractor": "~7.0.0",
  "ts-node": "~7.0.0",
  "tslint": "~6.1.0",
  "typescript": "~4.1.5"
}
```

Con el comando “npm run start” en la ubicación del proyecto con el nombre “api” se corren los servicios del servidor y se abre la base de datos para persistir la información.

```
User@DESKTOP-LTIS9MP MINGW64 ~/Desktop/Dropbox/Gabito/USBB0G/Semestre 10/Tesis/medic-app/api
$ npm run start

> angularmean_backend@1.0.0 start
> nodemon app.js

[nodemon] 2.0.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
(node:12720) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:12720) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Express server corriendo en el puerto 3000: online
(node:12720) DeprecationWarning: collection.ensureIndex is deprecated. Use createIndexes instead.
Base de datos: online
█
```

Con el comando “npm run start” en la ubicación del proyecto con el nombre “app” se corre la compilación del proyecto y se abre un puerto para ser visualizado en el navegador con la dirección “http://localhost:4200/#”

```
User@DESKTOP-LTIS9MP MINGW64 ~/Desktop/Dropbox/Gabito/USBBOG/Semestre 10/Tesis/medic-app/app
$ npm run start
> adminproc@0.0.0 start
> ng serve

✓ Browser application bundle generation complete.

Initial Chunk Files | Names          | Size
vendor.js           | vendor         | 4.77 MB
polyfills.js        | polyfills     | 487.52 kB
styles.css, styles.js | styles        | 422.47 kB
main.js             | main          | 139.44 kB
runtime.js          | runtime       | 9.01 kB

| Initial Total | 5.81 MB

Lazy Chunk Files | Names          | Size
pages-pages-module.js | pages-pages-module | 4.35 MB

Build at: 2021-07-07T04:21:16.537Z - Hash: 786ad08a91fd87650959 - Time: 16399ms

Warning: C:\Users\User\Desktop\Dropbox\Gabito\USBBOG\Semestre 10\Tesis\medic-app\app\src\app\pages\impresiones\impresiones.component.ts depends on 'html2pdf.js'. CommonJS or AMD dependencies can cause optimization bailouts.
For more info see: https://angular.io/guide/build#configuring-commonjs-dependencies

Warning: C:\Users\User\Desktop\Dropbox\Gabito\USBBOG\Semestre 10\Tesis\medic-app\app\src\app\pages\rxjs\rxjs.component.ts depends on 'rxjs/Observable'. CommonJS or AMD dependencies can cause optimization bailouts.
For more info see: https://angular.io/guide/build#configuring-commonjs-dependencies

Warning: C:\Users\User\Desktop\Dropbox\Gabito\USBBOG\Semestre 10\Tesis\medic-app\app\src\app\pages\impresiones\impresiones.component.ts depends on 'file-saver'. CommonJS or AMD dependencies can cause optimization bailouts.
For more info see: https://angular.io/guide/build#configuring-commonjs-dependencies

Warning: C:\Users\User\Desktop\Dropbox\Gabito\USBBOG\Semestre 10\Tesis\medic-app\app\src\app\services\usuario\usuario.service.ts depends on 'rxjs/add/operator/catch'. CommonJS or AMD dependencies can cause optimization bailouts.
For more info see: https://angular.io/guide/build#configuring-commonjs-dependencies

Warning: C:\Users\User\Desktop\Dropbox\Gabito\USBBOG\Semestre 10\Tesis\medic-app\app\src\app\services\usuario\usuario.service.ts depends on 'rxjs/add/operator/map'. CommonJS or AMD dependencies can cause optimization bailouts.
For more info see: https://angular.io/guide/build#configuring-commonjs-dependencies

Warning: C:\Users\User\Desktop\Dropbox\Gabito\USBBOG\Semestre 10\Tesis\medic-app\app\node_modules\ng2-charts\_ivy_ngcc_\fesm2015\ng2-charts.js depends on 'chart.js'. CommonJS or AMD dependencies can cause optimization bailouts.
For more info see: https://angular.io/guide/build#configuring-commonjs-dependencies

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
```

Anexo 6 - Manual de Usuario (Paciente)

Al ingresar a la aplicación, el usuario debe hacer clic en el link “Registrarse” para crear un nuevo usuario.



El usuario debe completar el formulario de registro, aceptar los “Términos” y luego hacer clic en el botón “REGISTRARSE”



El usuario hará clic en el botón de “G+” para ingresar a la aplicación para generar el registro del usuario



Correo _____

Contraseña _____

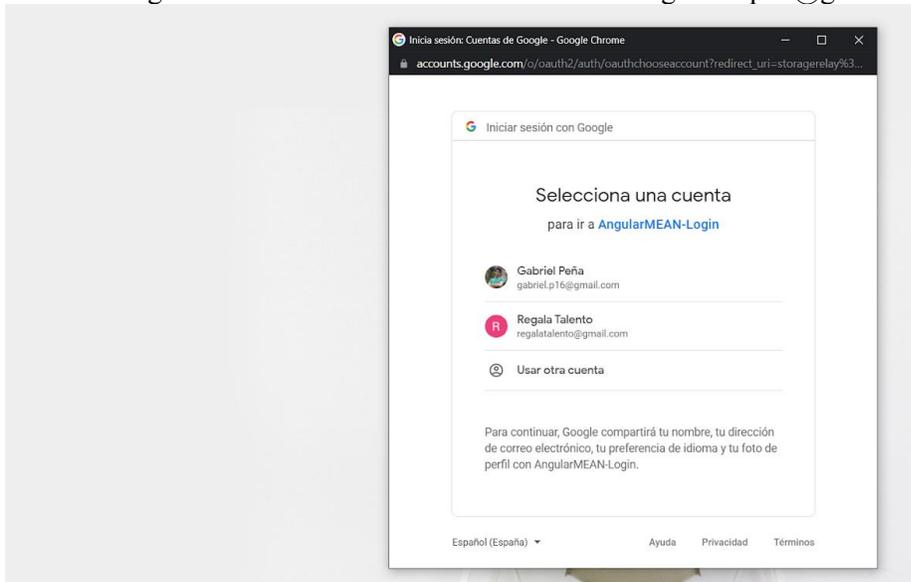
Recuerdame Olvido contraseña?

INGRESO



No tienes cuenta? [Regístrate](#)

Aparecerá una ventana donde se escogerá el usuario con el cual se desea ingresar a la aplicación. En este caso se escoge el usuario “Gabriel Peña” con el correo “gabriel.p16@gmail.com”.



Correo _____

Contraseña _____

Recuerdame Olvido contraseña?

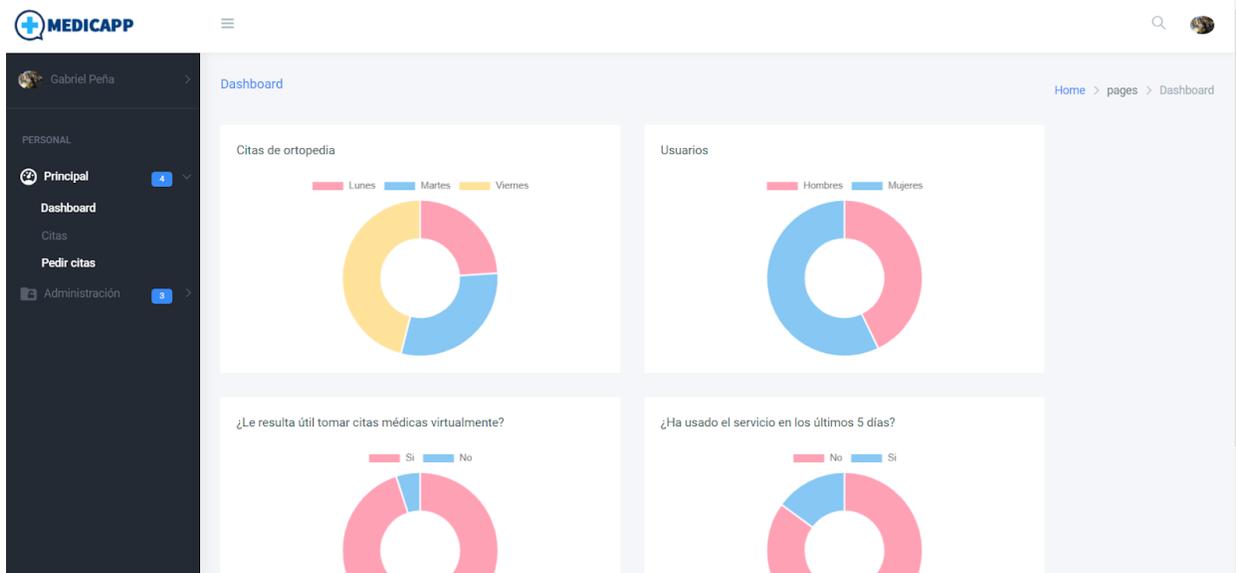
INGRESO

Iniciar sesión con Google



No tienes cuenta? [Regístrate](#)

Al ingresar con usuario y contraseña correctos, la navegación será dirigida a la pantalla del “Dashboard”, donde se tienen gráficas asociadas a las citas médicas y a la encuesta que se encuentra en el documento.



Al Hacer clic sobre el link “Pedir cita” que se encuentra en el menú lateral se despliega la siguiente pantalla “Solicitud de citas”, donde se puede coordinar la creación de una cita médica, para ello es necesario completar el formulario con la información necesaria y luego hacer clic en el botón “Solicitar cita”.

The 'Solicitud de citas' form contains the following elements:

- Header:** MEDICAPP logo and navigation icons.
- Sidebar:** Menu items including 'Principal', 'Dashboard', 'Citas', 'Pedir citas', and 'Administración'.
- Main Content:**
 - Title:** Solicitud de citas
 - Text:** Para agendar su cita médica a través del call center único Distrital de asignación de citas 3078181 o la línea gratuita 01 8000 118181 El horario de atención de lunes a sábado, de 6:00 a.m. a 8:00 p.m. En una primera fase, solamente se agendarán citas médicas para los servicios de: -Medicina General - Especialidades - Imágenes Diagnosticas También se podrá gestionar la asignación de su medica a través del: Chat de asignación de citas. El nuevo call center hace parte de una serie de estrategias que viene implementando el gobierno Distrital para descongestionar todas las Unidades de Servicios de Salud y que incluye el funcionamiento de 40 CAPS, la ampliación de los puntos de atención en salud, pasando de 17 a 103 en toda la ciudad; la eliminación de la mayoría de las autorizaciones para que los usuarios accedan a los servicios de salud sin trámites, la remodelación y ampliación de áreas de urgencias y consulta externa en algunas USS.
 - Text:** Con el fin de garantizar el acceso a los usuarios de la red pública hospitalaria, la Alcaldía Mayor de Bogotá y la Secretaría Distrital de Salud, dispuso del nuevo Call Center de Asignación de Citas para todas aquellas personas que requieren agendar sus citas de medicina general, especialista o agendar imágenes diagnósticas en cualquiera de las cuatro (4) la Subredes Integradas de Servicios de Salud del Distrito.
 - Form Fields:**
 - Selecciones servicio:** Servicio medico (dropdown)
 - Selecciones medico:** Nombre medico (dropdown)
 - Selecciones fecha:** Choose a date (calendar icon)
 - Selecciones hora:** --:-- (clock icon)
 - Razon de la cita:** Describa la razon por la cual esta pidiendo la cita (text area)
 - Button:** Solicitar cita

Una vez creadas las citas necesarias, se podrán visualizar en un listado en forma de tarjetas, como se puede evidenciar en la siguiente pantalla. En cada una de las tarjetas está la opción de “Editar” por si se desea cambiar algún parámetro de la cita médica y “Cancelar” por si se desea cancelar la cita médica.

PERSONAL

👤 Principal 4

Dashboard

Citas

Pedir citas

👤 Administración 3

Citas



Lorenzo Gomez
Medico General

Fecha de la cita: 20/07/2021

Hora de la cita: 14:00

Razon: Cita por dolor en la espalda

Editar Cancelar



Pedro
Ortopedia

Fecha de la cita: 17/09/2021

Hora de la cita: 11:00

Razon: Dolor en el pie derecho

Editar Cancelar

Anexo 7 – Manual de Usuario (Médico administrador)

Al ingresar a la aplicación, el usuario debe hacer clic en el link “Registrarse” para crear un nuevo usuario.



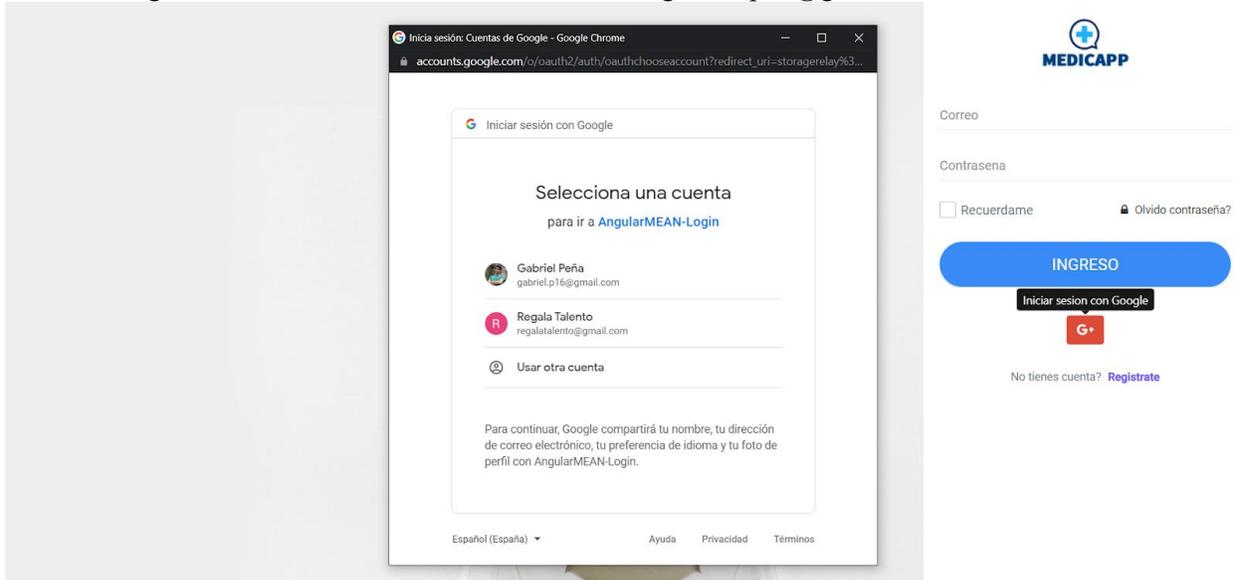
El usuario debe completar el formulario de registro, aceptar los “Términos” y luego hacer clic en el botón “REGISTRARSE”



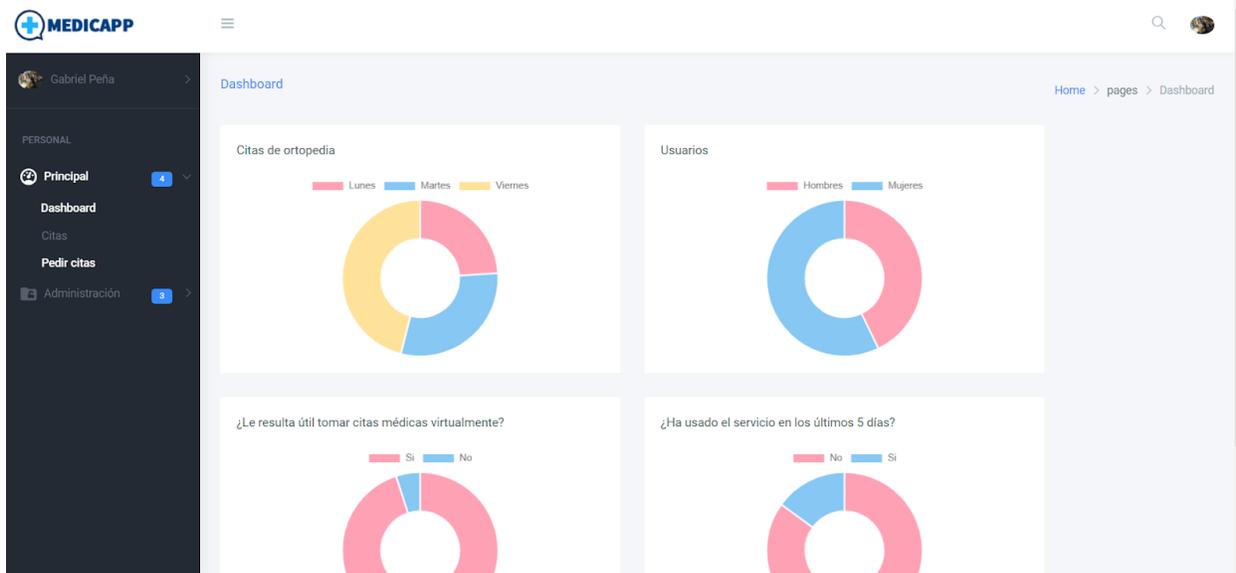
El usuario hará clic en el botón de “G+” para ingresar a la aplicación para generar el registro del usuario



Aparecerá una ventana donde se escogerá el usuario con el cual se desea ingresar a la aplicación. En este caso se escoge el usuario “Gabriel Peña” con el correo “gabriel.p16@gmail.com”.



Al ingresar con usuario y contraseña correctos, la navegación será dirigida a la pantalla del “Dashboard”, donde se tienen gráficas asociadas a las citas médicas y a la encuesta que se encuentra en el documento.

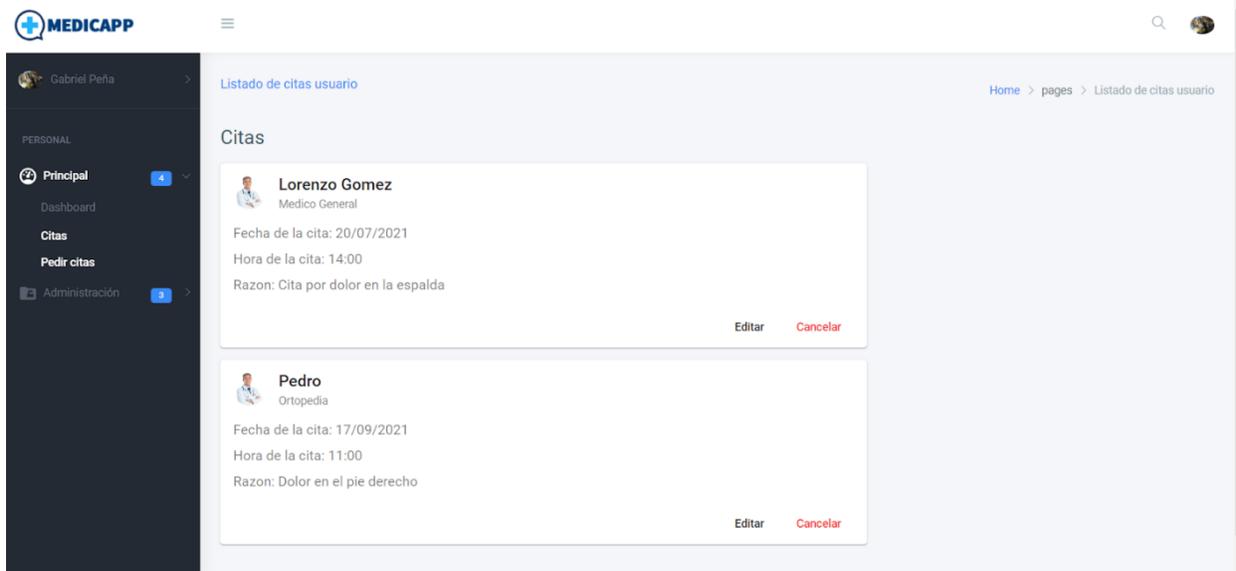


Al Hacer clic sobre el link “Pedir cita” que se encuentra en el menú lateral se despliega la siguiente pantalla “Solicitud de citas”, donde se puede coordinar la creación de una cita médica, para ello es necesario completar el formulario con la información necesaria y luego hacer clic en el botón “Solicitar cita”.

The 'Solicitud de citas' form contains the following elements:

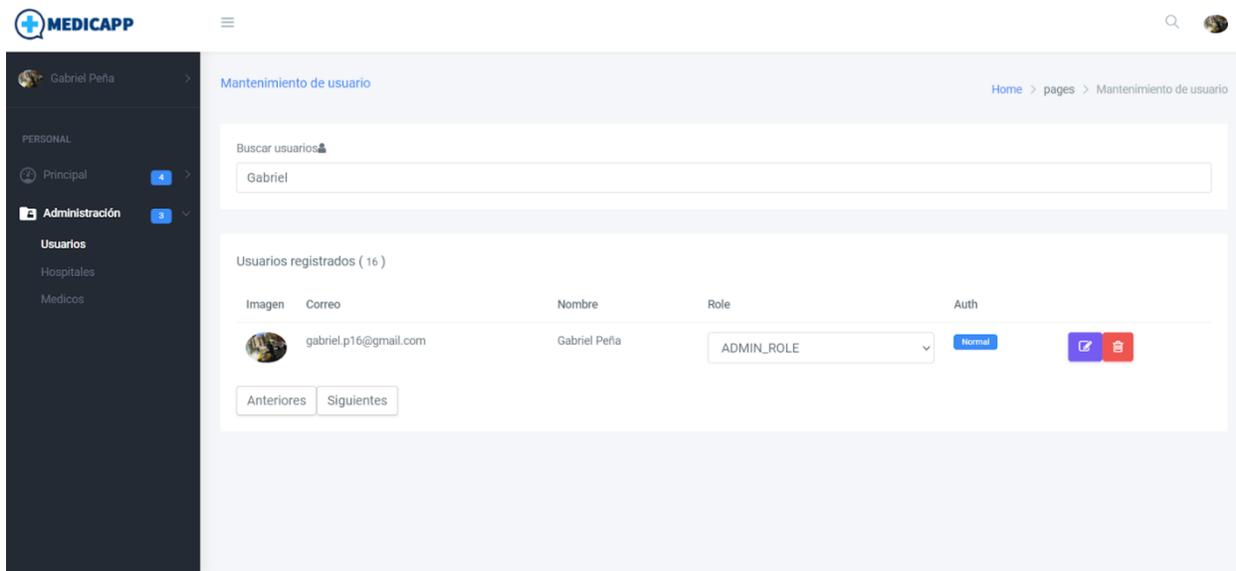
- Header:** MEDICAPP logo and navigation icons.
- Sidebar:** Menu items including 'Principal', 'Dashboard', 'Citas', 'Pedir citas', and 'Administración'.
- Main Content:**
 - Title:** Solicitud de citas
 - Text:** Para agendar su cita médica a través del call center único Distrital de asignación de citas 3078181 o la línea gratuita 01 8000 118181 El horario de atención de lunes a sábado, de 6:00 a.m. a 8:00 p.m. En una primera fase, solamente se agendarán citas médicas para los servicios de: -Medicina General - Especialidades - Imágenes Diagnosticas También se podrá gestionar la asignación de su medica a través del: Chat de asignación de citas. El nuevo call center hace parte de una serie de estrategias que viene implementando el gobierno Distrital para descongestionar todas las Unidades de Servicios de Salud y que incluye el funcionamiento de 40 CAPS, la ampliación de los puntos de atención en salud, pasando de 17 a 103 en toda la ciudad; la eliminación de la mayoría de las autorizaciones para que los usuarios accedan a los servicios de salud sin trámites, la remodelación y ampliación de áreas de urgencias y consulta externa en algunas USS.
 - Text:** Con el fin de garantizar el acceso a los usuarios de la red pública hospitalaria, la Alcaldía Mayor de Bogotá y la Secretaría Distrital de Salud, dispuso del nuevo Call Center de Asignación de Citas para todas aquellas personas que requieren agendar sus citas de medicina general, especialista o agendar imágenes diagnósticas en cualquiera de las cuatro (4) la Subredes Integradas de Servicios de Salud del Distrito.
 - Form Fields:**
 - Selecciones servicio:** Servicio medico (dropdown)
 - Selecciones medico:** Nombre medico (dropdown)
 - Selecciones fecha:** Choose a date (calendar icon)
 - Selecciones hora:** --:-- (clock icon)
 - Razon de la cita:** Describa la razon por la cual esta pidiendo la cita (text area)
 - Button:** Solicitar cita

Una vez creadas las citas necesarias, se podrán visualizar en un listado en forma de tarjetas, como se puede evidenciar en la siguiente pantalla. En cada una de las tarjetas está la opción de “Editar” por si se desea cambiar algún parámetro de la cita médica y “Cancelar” por si se desea cancelar la cita médica.



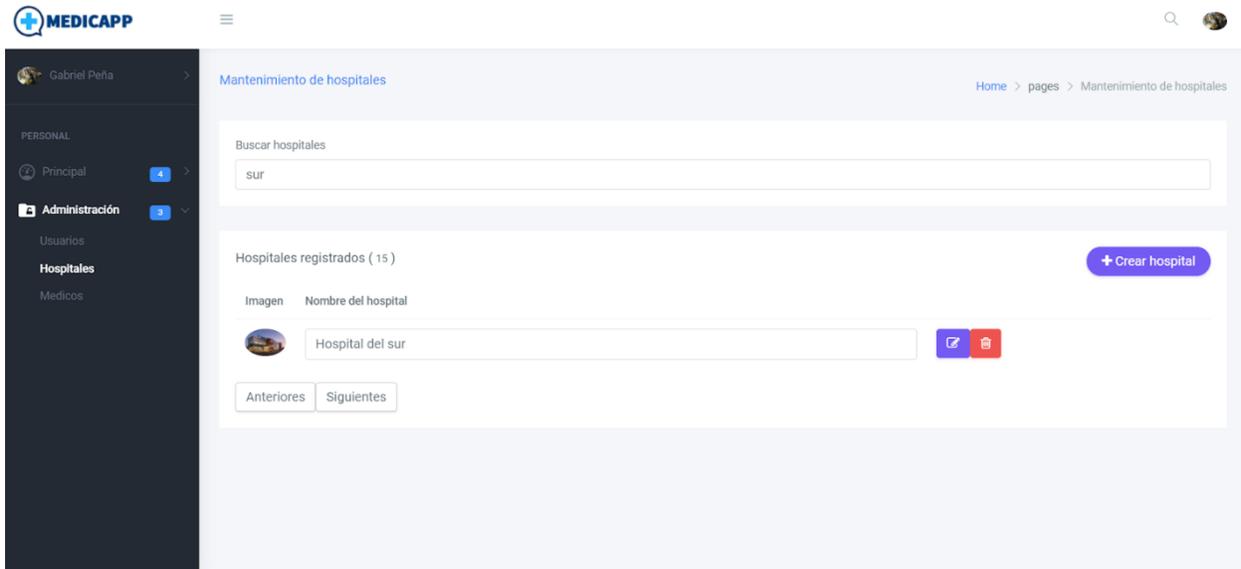
Para listar a los usuarios que previamente se han creado es necesario desplegar el menú lateral “Administración” y luego hacer clic en el link “Usuarios”. Se mostrará el listado de usuarios con una imagen del usuario, correo, nombre completo, rol que ejerce dentro de la aplicación, el tipo de autenticación con el que fue creado y dos botones de acción, el primero es “Actualizar” para cambiar el rol que puede tener el usuario y el segundo es el botón “Eliminar” que está dado para eliminar un usuario seleccionado.

En esta pantalla se puede hacer una búsqueda por nombre del usuario. También se tiene generada una paginación para mostrar un listado de 5 usuarios por pantalla.

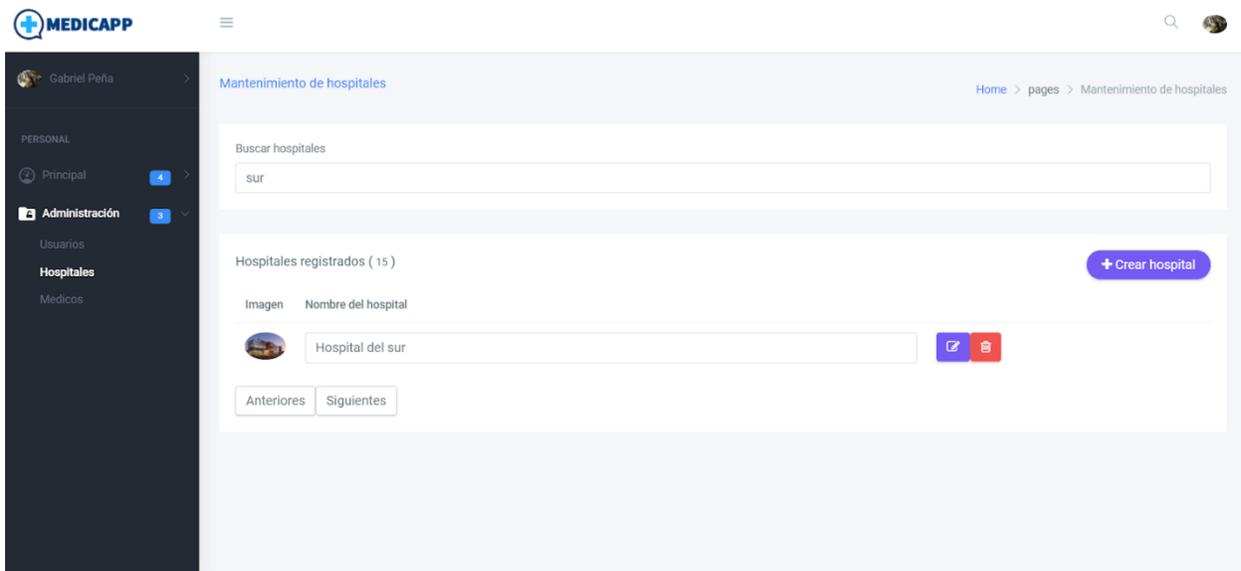


Para listar los hospitales que previamente se han creado es necesario desplegar el menú lateral “Administración” y luego hacer clic en el link “Hospitales”. Se mostrará el listado de hospitales con una imagen del hospital, nombre completo y dos botones de acción, el primero es “Actualizar” para cambiar el nombre que puede tener el hospital y el segundo es el botón “Eliminar” que está dado para eliminar un hospital seleccionado.

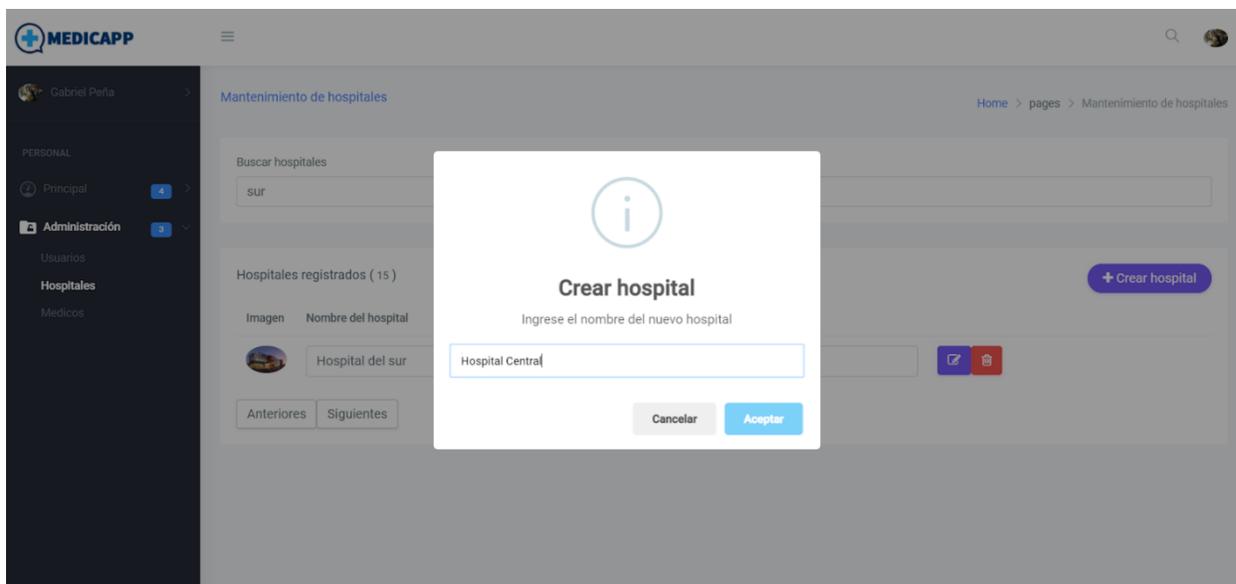
En esta pantalla se puede hacer una búsqueda por nombre del hospital. También se tiene generada una paginación para mostrar un listado de 5 hospitales por pantalla.



En la pantalla “Mantenimiento de hospitales” se tiene un botón “Crear hospital” donde al hacer clic se puede crear un nuevo hospital.

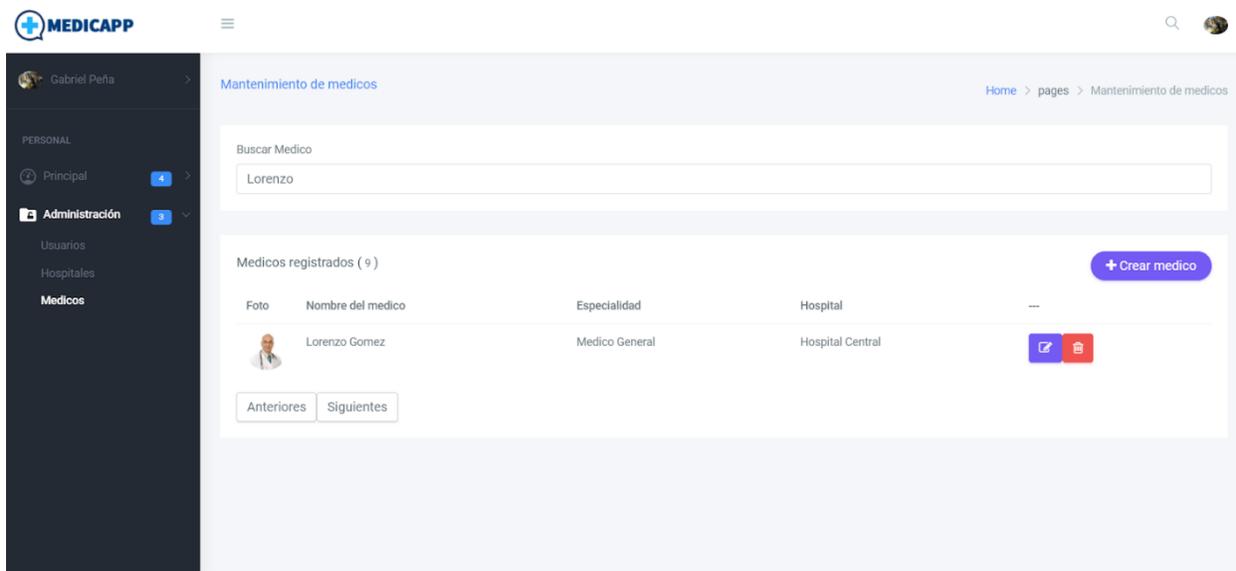


Se despliega una ventana tipo “pop-up” donde se podrá crear un hospital llenando el campo “Nombre del hospital” y haciendo clic en el botón “Aceptar”. Si se desea salir de la ventana es necesario hacer clic en el botón “Cancelar”.

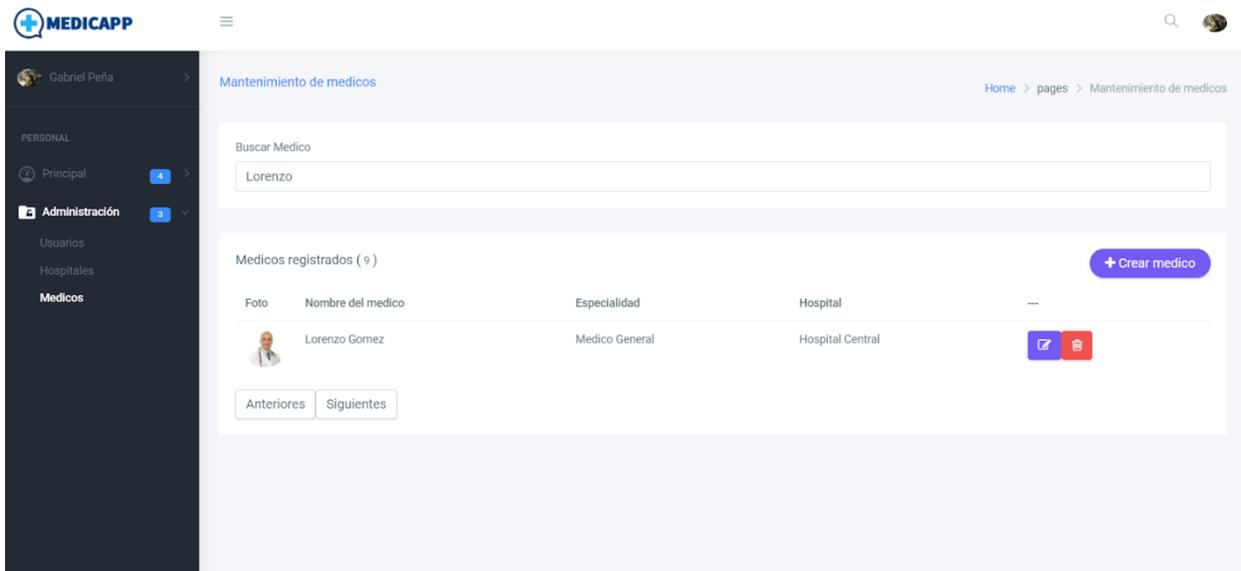


Para listar los médicos que previamente se han creado es necesario desplegar el menú lateral “Administración” y luego hacer clic en el link “Médicos”. Se mostrará el listado de médicos con una imagen del médico, nombre completo, Especialidad, hospital asociado y dos botones de acción, el primero es “Actualizar” para cambiar el nombre que puede tener el médico y el segundo es el botón “Eliminar” que está dado para eliminar un médico seleccionado.

En esta pantalla se puede hacer una búsqueda por nombre del médico. También se tiene generada una paginación para mostrar un listado de 5 médicos por pantalla.



En la pantalla “Mantenimiento de médicos” se tiene un botón “Crear médico” donde al hacer clic se puede crear un nuevo médico.



Se despliega una pantalla donde se podrá crear un médico llenando el campo “Nombre”, seleccionar “Especialidad”, seleccionar “Hospital” al cual se encuentra vinculado y haciendo clic en el botón “Aceptar”. Si se desea salir de la ventana es necesario hacer clic en el botón “Cancelar”.

