**Taller 1**

Ingeniería de software

**Presentado a:**

Carlos Andrés Parra Acevedo

**Integrantes:**

Johann Fernando Trigos Niño

José Esteban Betín Díaz

Kattya Alexandra Peña Nieto

Johan Miguel Céspedes Ortega

Javier Alfonso Becerra Sánchez

**Versión**

1.0

Pontifica Universidad Javeriana

Facultad de ingeniería

Especialización en arquitectura empresaria de software

Bogotá, Colombia

2020

Tabla de contenido

[Listado de Tablas 3](#_Toc48562393)

[Alcance: 4](#_Toc48562394)

[WBS 4](#_Toc48562395)

[Recursos 4](#_Toc48562396)

[Cronograma 5](#_Toc48562397)

[Presupuesto-Javier 6](#_Toc48562398)

[Riesgos 6](#_Toc48562399)

[Requisitos 7](#_Toc48562400)

[R01- Buscar lugares de atención y Disponibilidad 8](#_Toc48562401)

[R02- Registro de Usuarios 9](#_Toc48562402)

[R03- Autenticación de Usuarios en la Plataforma 11](#_Toc48562403)

[R04- Ubicación del Dispositivo 12](#_Toc48562404)

[R05- Gestión de EPS 12](#_Toc48562405)

[R06- Confirmación de Registro 13](#_Toc48562406)

[R07- Gestión de Lugares de Atención 14](#_Toc48562407)

[R08- Disponibilidad de Lugares de Atención 15](#_Toc48562408)

[R09- Gestión de especialidades 16](#_Toc48562409)

[R10- Reportes 17](#_Toc48562410)

# Listado de Tablas

[Tabla 1 Recursos Humanos del Proyecto 5](#_Toc48497288)

[Tabla 2 Recursos Tecnicos del Proyecto 5](#_Toc48497289)

[Tabla 3 Cronograma 5](#_Toc48497290)

[Tabla 4 Requisitos Principales 7](#_Toc48497291)

[Tabla 5 Disponibilidad de atención 9](#_Toc48497292)

[Tabla 6 Registro de Usuarios 10](#_Toc48497293)

[Tabla 7 Autenticación de Usuarios en la Plataforma 12](#_Toc48497294)

[Tabla 8 Ubicación del Dispositivo 12](#_Toc48497295)

[Tabla 9 13](#_Toc48497296)

[Tabla 10 14](#_Toc48497297)

[Tabla 11 Gestión de Lugares de Atención 15](#_Toc48497298)

[Tabla 12 Disponibilidad de Lugares de Atención 16](#_Toc48497299)

[Tabla 13 16](#_Toc48497300)

[Tabla 14 17](#_Toc48497301)

**Listado de Ilustraciones**

[Ilustración 1 Work Breakdown Structure 4](#_Toc48497318)

# Alcance:

SiMed es una aplicación Web que permite la localización de los centros médicos más cercanos a los usuarios de Bogotá, además contemplando la especialidad requerida y con mayor disponibilidad de atención.

## WBS

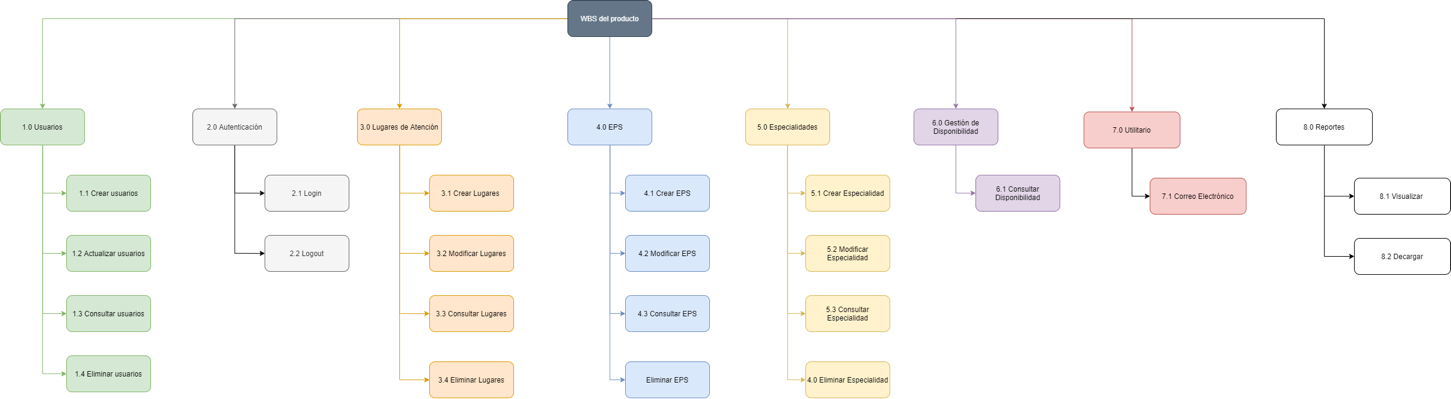


Ilustración 1 Work Breakdown Structure

## Recursos

Este proceso de gestión del talento que acompañará el proceso de ejecución del proyecto debe incluir los procesos que los organizan y quienes realizan el proceso de dirección y control del proceso, para esto se tienen en cuenta las fases del proyecto donde es necesario la definición de roles, que interrelación tienen entre ellos y las actividades de competencia que debe realizar cada rol, además de ello es necesario tener en cuenta el tiempo que se invertirá por hora/labor del proceso.

|  |  |  |  |
| --- | --- | --- | --- |
| **NOMBRE** | **ID** | **PERFIL** | **RESPONSABLE** |
| Gerente del proyecto | GP | Líder nato que debe tener una alta capacidad de comunicación y motivación para llevar un liderazgo con el equipo de ejecución del proyecto, y además evaluar las actividades que comprenden el trabajo en equipo con los interesados. | José Betín |
| Líder de desarrollo | LD | Capacidad de coordinar y organizar actividades de los integrantes de desarrollo, tiene que llevar a cabo procesos de liderazgo, uso de recursos disponibles, realizar seguimiento y retroalimentación de las actividades. | Johann Trigos |
| Líder de configuración | LC | Rol encargado de la habilitación de las herramientas tecnológicas necesarias para todo el ciclo de vida de desarrollo de la aplicación. | Johann Trigos |
| Arquitecto de software | AS | Como liderazgo técnico, este debe tener la capacidad de tomar las óptimas decisiones tomando en cuenta los requisitos no funcionales del proyecto, este debe tener en cuenta la definición de tecnología y su mejora continua. | Johan Céspedes |
| Analista QA Senior | AQA | Para lograr la gestión de la calidad del proyecto, es necesario responsables en revisar flujos desde la perspectiva del usuario final cuestionando que podría pasar si se realiza una acción en especifico. Debe tener un alto nivel de abstracción y capacidad de rastrear amenazas. | Kattya |
| Desarrollador Front-End | FED | Expertos técnicos en el desarrollo de software del lado de la vista, responsable de la capa de presentación que implementan técnicas a base de decisiones del arquitecto, actuando con eficiencia y alto nivel de compromiso, este es un recurso muy importante debido que es el que recrea la idea principal de visualización del Software. | Javier Becerra  José Betín |
| Desarrollador Back-End | BED | Expertos técnicos en el desarrollo de software, responsable de la lógica del sistema implementando las técnicas sujetas de las decisiones del arquitecto, actuando con eficiencia y alto nivel de compromiso, este es un recurso muy importante debido que quien desarrolla la lógica del aplicativo y permite que la vista presente la información requerida por el usuario. | Kattya  Johan Céspedes |

Tabla 1 Recursos humanos del proyecto

Otros recursos necesarios para el desarrollo del proyecto que van a ser tenidos en cuenta para adquisición son los siguientes:

|  |  |  |
| --- | --- | --- |
| CONCEPTO | CARACTERÍSTICAS | CANTIDAD |
| Computador | Portátil o escritorio mínimo procesador Core i7 4ta generación +, SO a elección. | 5 |
| Internet | Hasta 30 Megas bajada y hasta 6 Megas subida de Internet. | 5 |

Tabla 2 Recursos técnicos del proyecto

## Cronograma

A continuación se presenta el cronograma general para el proyecto SiMed:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Actividad\Semana | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Definición proyecto |  |  |  |  |  |  |  |  |
| Conformación equipo |  |  |  |  |  |  |  |  |
| Iteración 1 |  |  |  |  |  |  |  |  |
| Análisis |  |  |  |  |  |  |  |  |
| Diseño |  |  |  |  |  |  |  |  |
| Implementación |  |  |  |  |  |  |  |  |
| Pruebas |  |  |  |  |  |  |  |  |
| Primer DEMO |  |  |  |  |  |  |  |  |
| Iteración 2 |  |  |  |  |  |  |  |  |
| Análisis |  |  |  |  |  |  |  |  |
| Diseño |  |  |  |  |  |  |  |  |
| Implementación |  |  |  |  |  |  |  |  |
| Pruebas |  |  |  |  |  |  |  |  |
| Producción |  |  |  |  |  |  |  |  |
| Refinamiento |  |  |  |  |  |  |  |  |
| Segundo DEMO |  |  |  |  |  |  |  |  |
| Postmortem |  |  |  |  |  |  |  |  |

Tabla 3 Cronograma

## Presupuesto

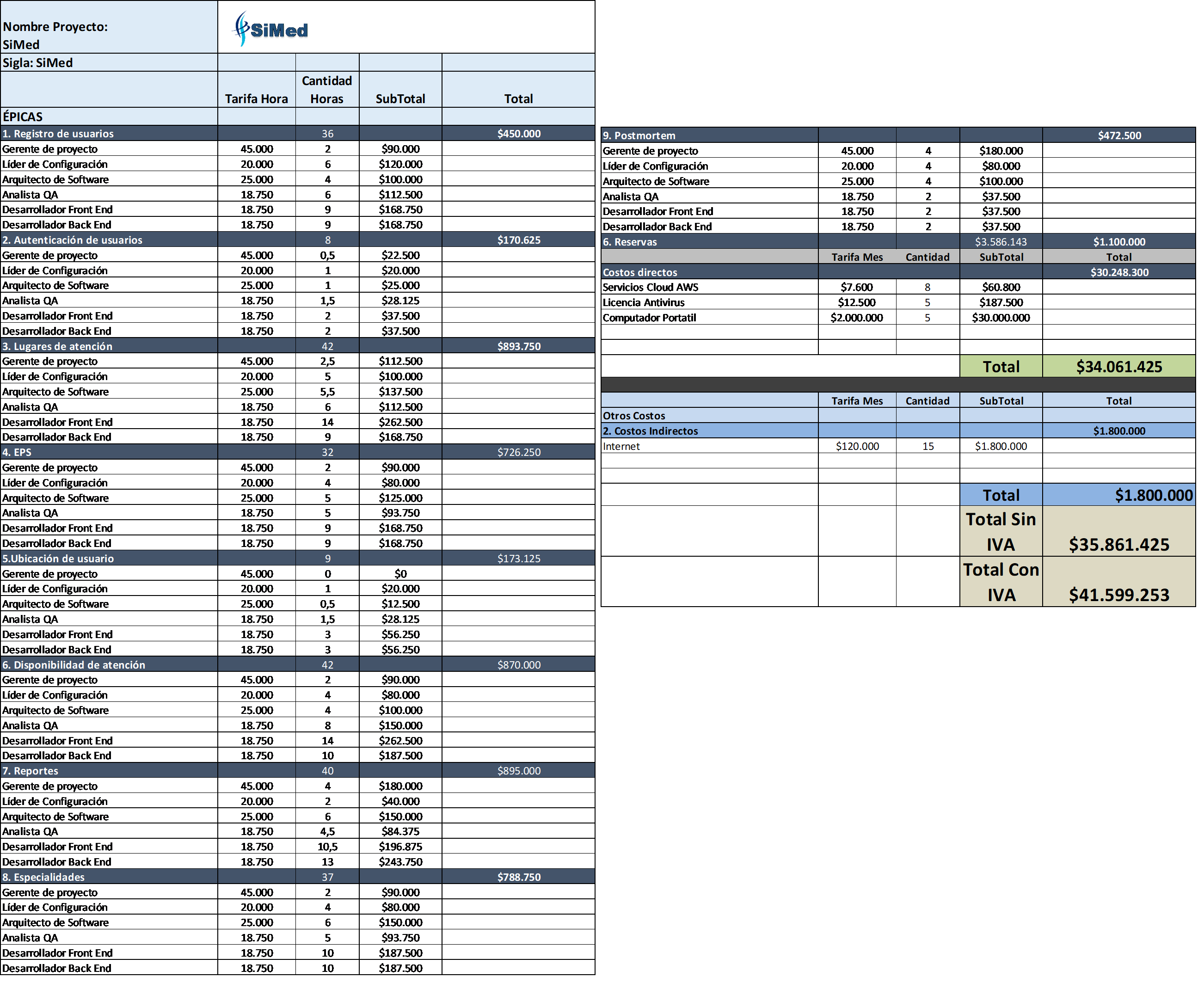


Ilustración 2 Costos

## Riesgos

## 6.1 Positivos

|  |  |  |
| --- | --- | --- |
| **Riesgo** | **Impacto** | **Probabilidad de ocurrencia** |
| Mayor uso de la aplicación por aumento de enfermedades en Bogotá | Alto | Media |
| Necesidad de ubicar un centro de atención con baja aglomeración | Alto | Media |
| Tecnología ya conocida por el equipo | Alto | Alto |
| Disponibilidad de recursos | Alto | Alto |
| Experiencia técnica | Alto | Alto |
| Posibilidad de apoyo de instituciones del estado | Alto | Bajo |
| Necesidad del producto en el mercado | Alto | Alto |

Tabla 4 Riesgos positivos

## 6.2 Negativos

|  |  |  |  |
| --- | --- | --- | --- |
| Riesgo | Impacto | Probabilidad de ocurrencia | Mitigación |
| Falta de capacidad tecnológica de las EPS para integrarse con la aplicación | Alto | Media | Trabajar con los datos públicos de los lugares de atención asociados a las EPS cargando la información a la plataforma. |
| Posibilidad de entrada de competidores con el mismo mercado objetivo | Medio | Media | Establecer estrategias de diferenciación que permitan tener una ventaja competitiva, esto acompañado de una arquitectura que permita hacer cambios en pocos días para dar respuesta ante cualquier cambio en el mercado |
| La aplicación no tenga buena acogida | Alto | Media | Implementar estrategias de mercadeo e imagen para impulsar el uso de la aplicación |
| La normativa colombiana restrinja el uso y distribución de la aplicación | Medio | Baja | Permitir configurar la información que puede ser mostrada en caso de que se restrinja el uso de la información |
| Cambio en la estructura de las EPS y lugares de atención | Alto | Baja | Plantear un modelo de relación de los componentes flexible |

Tabla 5 Riesgos negativos

# Requisitos funcionales

Los requisitos funcionales se plantean a través de épicas e historias de usuario:

|  |  |
| --- | --- |
| **Épica** | **Descripción** |
| Disponibilidad de atención | El sistema debe permitir al usuario visualizar la disponibilidad de los lugares de atencion que fueron consultados de acuerdo a la especialidad y su ubicación, este requisito permitira al usuario elegir y llegar al lugar con mayor disponibilidad de atencion para sus necesidades medicas |
| Registro de usuarios | El sistema debe permitir registrar usuarios para el acceso a los servicios de la plataforma, durante el registro se diligencian algunos parametros y son persistidos en la base de datos |
| Autenticación de usuarios | El sistema debe permitir la autenticación de usuarios a la plataforma. |
| Ubicación del usuario | El sistema debe permitir encontrar la ubicación del usuario. |
| EPS | El sistema debe permitir realizar la gestión de las EPS |
| Lugares de atención | El sistema debe permitir realizar la gestión de los lugares de atención |
| Especialidades | El sistema debe permitir CRUD las especialidades que tienen los centros de atención |
| Reportes | El sistema debe permitir visualizar reportes de uso del producto |

El detalle de las historias de usuario de cada épica se puede ver en el siguiente archivo:

[Anexo historias de usuario](https://livejaverianaedu-my.sharepoint.com/:x:/g/personal/johann_trigos_javeriana_edu_co/EQ44vtYY6mZKudLVaO2wd7EBoBtaPCalgaMHes_RNMQtAw?e=xaayfR)

# Requisitos no funcionales

1. Adecuación funcional.
   * Completitud funcional: el sistema debe cubrir todas las tareas y objetivos del producto planteado
2. Usabilidad.
   * Estética de la interfaz de usuario: el sistema debe agradar y satisfacer la interacción con el usuario en cualquier dispositivo por el que se conecte.
   * Protección contra errores de usuario: el sistema debe guiar la navegación del usuario con el fin de proteger a los usuarios de errores.
   * Capacidad de ser usado: el sistema debe permitir al usuario operar y controlar la aplicación con facilidad.
3. Fiabilidad.
   * Disponibilidad: el sistema debe ser operativo y accesible al usuario cuando este lo requiera
4. Portabilidad.
   * Adaptabilidad: La aplicación web debe poseer un diseño “Responsive” para garantizar una adecuada visualización en navegadores de diferentes dispositivos (computadores personales, dispositivos tableta y teléfonos inteligentes).
   * Se debe confirmar el correcto funcionamiento en el navegador Chrome.
5. Seguridad
   * En la interacción entre el Back-end y Front-end debe intermediar un JWT
   * Los datos de contraseñas deberán ser encriptados con algoritmo Sha 256
6. Mantenibilidad
   * El aplicativo estará dividido en componentes para poder tener una modificación más rápida, efectiva y eficiente.
   * El sistema se dividirá en componentes reutilizables.
7. Portabilidad
   * El aplicativo inicialmente se instalará en la nube de AWS, sin embargo, debe permitir ser migrado a otras Nubes o un data center on-premise.
8. Redundancia o inconsistencia
   * Se espera que el sistema evite la duplicidad de datos, los datos almacenados deben conserven su integridad.

# Estilo de Arquitectura

La arquitectura en capas es un estilo arquitectónico que nos permite promover la descomposición de tareas en grupos y responsabilidades, cada uno de estos grupos tiene un nivel de abstracción de acuerdo con su tarea, de tal manera que sea individualizar tareas y responsabilidades dentro de SiMed.

Así mismo es importante resaltar que se empleará el patrón cliente servidor, este patrón nos permitirá ejercer mayor control en la etapa inicial del proyecto, así mismo se espera evitar situaciones como la redundancia o inconsistencia de información en las bases de datos, por lo cual para un comienzo resulta benéfico emplear este tipo de patrones.

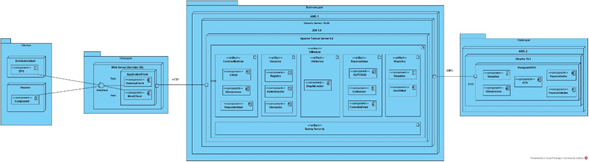
Tomando en cuenta el alcance de la aplicación a desarrollar definimos Elementos, relaciones, responsabilidades y colaboración para la implementación de SiMed, la implementación permite hacer una descomposición entre capas de vista, negocio y modelo, esto mejora en mantenibilidad, disminuye el nivel de complejidad de la implementación y favorece los tiempos de desarrollo por cuanto se espera hacer la entrega del aplicativo con un máximo de dos meses dando cumplimiento con los requisitos funcionales y no funcionales propuestos por el equipo de trabajo.

# Herramientas a utilizar

Para la implementación de SiMed emplearan las siguientes herramientas que permitirían realizar lo planteado en la arquitectura.

|  |  |
| --- | --- |
| **Herramienta** | **Descripción** |
| RAML | Para el desarrollo de los contratos, este ofrece una herramienta de código abierto para su diseño. |
| Node.js | Node.js es un entorno JavaScript de lado de servidor que utiliza un modelo asíncrono y dirigido por eventos. Soporta protocolos TCP, DNS y HTTP, importantes para el desarrollo de la SiMed. |
| Angular | Framework de desarrollo para JavaScript creado por Google. La finalidad de Angular es facilitarnos el desarrollo de SiMed, al tratarse de una aplicación web, proporciona herramientas para trabajar con los elementos de una web de una manera más sencilla y óptima. |
| Postgres | Sistema de gestión de bases de datos relacionales, su licencia es de código abierto por lo cual es soportado por una comunidad de desarrolladores. |
| Spring Boot | Framework de desarrollo, el cual será empleado para el desarrollo de apis con servicios REST no lo permite hacer de manera sencilla mediante sus anotaciones. Además de ello permite hacer uso de clientes REST y SOAP. |
| Java | Lenguaje de programación orientado a objetos, especialmente utilizado para el desarrollo de aplicaciones empresariales. |

**Diagrama de Despliegue Arquitectura Candidata**



*Diagrama de arquitectura en Capas SiMed*

**Capa de presentación:** Se empleará un Servidor AWS 0, con un Sistema Operativo Ubuntu Server 18.04, allí se alojará, el frontend de la aplicación, a futuro se pretende desplegar el servicio de aplicación móvil para usuarios finales.

**Capa de Negocio:** Sera implementada en un Servidor AWS 1 con Sistema Operativo Ubuntu Server 18.04, JDK 1.8, servidor embebido de aplicaciones TomCat 8.0, se empaquetan servicios de aplicación, divididos en módulos Centros médicos, Usuarios, Utilitarios, Especialistas y Reportes, por último, se implementa Capa de Seguridad de Spring (Spring Security).

**Capa de Datos:** Será implementada en un Servidor AWS 2 Sistema Operativo Ubuntu Server 18.04, allí se alojará base de datos PostgresDB12, para la persistencia de datos de aplicación.