**[CitaPro]**

**(SAD) Software Architecture Document**

**Versión 1.0**

**Identificación de Documento**

| **Identificación** |  |
| --- | --- |
| **Proyecto** | CitaPro |
| **Versión** | 1.0 |

| **Documento mantenido por** |  |
| --- | --- |
| **Fecha de ultima revisión** |  |
| **Fecha de próxima revisión** |  |

| **Documento aprobado por** |  |
| --- | --- |
| **Fecha de última aprobación** |  |

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Tabla de Contenidos**

[**1**](#_heading=h.30j0zll) **Introducción 3**

[1.1](#_heading=h.1fob9te) Contexto del Problema 3

[1.2](#_heading=h.3znysh7) Propósito 3

[1.3](#_heading=h.2et92p0) Ámbito 3

[1.4](#_heading=h.tyjcwt) Definiciones, acrónimos y abreviaciones 3

[1.5](#_heading=h.3dy6vkm) Referencias 3

[1.6](#_heading=h.1t3h5sf) Resumen ejecutivo 4

[**2**](#_heading=h.2bn6wsx) **Representación de la Arquitectura 5**

[2.1](#_heading=h.4d34og8) Representación 5

[**3**](#_heading=h.2s8eyo1) **Metas y Restricciones de la Arquitectura 6**

[A continuación se revisan las metas y restricciones de la arquitectura. 6](#_heading=h.17dp8vu)

[3.1](#_heading=h.3rdcrjn) Metas de la arquitectura 6

[3.2](#_heading=h.26in1rg) Restricciones de la Arquitectura 6

[3.3](#_heading=h.lnxbz9) Otros antecedentes y consideraciones 6

[**4**](#_heading=h.35nkun2) **Vista de Casos de Uso y Escenarios de Calidad 8**

[4.1](#_heading=h.1ksv4uv) Modelo de Casos de Uso 8

[4.2](#_heading=h.44sinio) Especificación de Casos de Uso Relevantes 8

[4.3](#_heading=h.2jxsxqh) Especificación de los Escenarios de Calidad Relevantes 9

[**Vista Lógica 11**](#_heading=h.z337ya)

[4.4](#_heading=h.3j2qqm3) Parte Estructural 11

[4.5](#_heading=h.1y810tw) Parte Dinámica 11

[**5**](#_heading=h.4i7ojhp) **Vista de Procesos 13**

[**6**](#_heading=h.2xcytpi) **Vista de Implementación 14**

[**7**](#_heading=h.1ci93xb) **Vista de Despliegue 15**

[**8**](#_heading=h.3whwml4) **Decisiones de Diseño y Selección de Alternativas 16**

1. **Introducción**
   1. **Contexto del Problema**

Actualmente, realizar una cita médica puede ser un proceso tedioso y largo. Encontrar a un médico o especialista disponible y cercano es una tarea complicada. CitaPro propone acelerar esta tarea. Permite a sus usuarios buscar citas médicas con especialistas cercanos. Entonces, y sólo entonces, muestra las horas disponibles, ocultando las horas ocupadas y hostigando ofertas tentadoras.

* 1. **Propósito**

El propósito de este documento es describir la arquitectura del sistema denominado CitaPro. Es una guía para entender cómo está compuesto y funciona realmente un sistema, detallando sus componentes de alto nivel, cómo se interconectan, cómo sus módulos interactúan entre sí y proporcionando la información técnica necesaria para comprender cómo se toman ciertas decisiones de diseño.

* 1. **Ámbito**

Este documento comprende la búsqueda de médicos por especialidad y ubicación, la programación de citas médicas y la interacción a través de un mapa que muestra la ubicación del consultorio. Finalmente, considera la integración de un sistema de pago para finalmente confirmar la programación de la cita y hacerse finalmente presente.

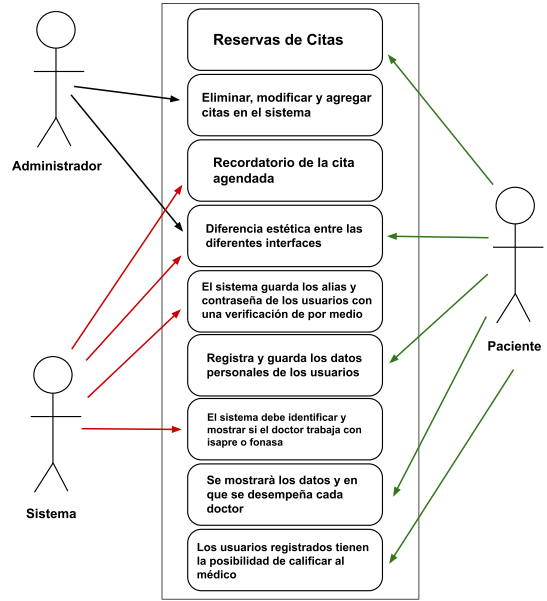
* 1. **Definiciones, acrónimos y abreviaciones**

| **ACRONIMO** | **DESCRIPCION** |
| --- | --- |
| *Cita médica* | Consulta con un médico o un especialista. |
| *Libreta virtual* | Sistema que muestra las horas que tiene un médico. |
| *Especialista* | Médico que se especializa en algo en especifico. |
|  |  |

* 1. **Referencias**

A continuación se listan las referencias a otros documentos:

* **Casos de Uso**



* 1. **Resumen ejecutivo**

“CitaPro” es una plataforma que permite a los pacientes concertar citas médicas de forma sencilla, ofreciendo una experiencia fácil y rápida para los pacientes y para los administradores del centro médico. El sistema puede buscar doctores cercanos, programar citas, verificar si el médico se encuentra afiliado a planes de salud tales como Isapre o Fonasa, con recordatorios automáticos para el día de la cita. Además, se puede calificar al médico tras la consulta.

* 1. **Representación**

La arquitectura del sistema <<Nombre del proyecto y/o Aplicación>> está representada siguiendo el enfoque de del framework 4+1 y las recomendaciones del proceso unificado. Las vistas incluidas en esta versión del documento son:

* **Vista de Casos de Uso y Escenarios de Calidad**: Describe los casos de uso más significativos, presenta los actores y una descripción de sus casos de uso asociados. De igual forma describe los escenarios de calidad más relevantes para la arquitectura.
* **Vista de Metas y Restricciones**: Describe restricciones tecnológicas, normativas, estándares, etc., los cuales influyen sobre las decisiones arquitectónicas, del producto y del proceso de desarrollo.
* **Vista Lógica**: Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias. Usa el view type Módulos para representar la estructura lógica y el view type Componentes y Conectores para representar el comportamiento.
* **Vista de Procesos**: Describe los procesos involucrados para darle sentido a la ejecución del sistema, así como sus relaciones de comunicación y sincronización.
* **Vista de Implementación**: Describe los componentes de deployment construidos y sus dependencias.

1. **Metas y Restricciones de la Arquitectura**

A continuación se revisan las metas y restricciones de la arquitectura.

* 1. **Metas de la arquitectura**

De acuerdo a las reuniones y al análisis de los requerimientos, se listan los principales conductores iniciales de la arquitectura los cuales corresponden a las metas arquitectónicas iniciales:

* **Desempeño**:
* **Tolerancia a fallos**:
* **Seguridad**:
* **Modificabilidad/Reuso**:
* **Operatividad**:
  1. **Restricciones de la Arquitectura**

Existen restricciones que han sido levantadas con los stakeholders, las cuales se presentan a continuación:

* **Tiempo de construcción**: se cuenta con un plazo estrecho de tiempo para su construcción, 4 semanas según la planificación.
* **Infraestructura**: se cuenta con servidores de aplicación replicados y con balanceadores de carga, asimismo, con una base de datos en estructura de cluster.
* **Otros componentes de software**: no se considera la adquisición y licenciamiento de otros componentes de software.
  1. **Otros antecedentes y consideraciones**

La empresa desarrolladora cuenta con un framework que considera los siguientes componentes que permiten satisfacer los requerimientos arquitectónicos:

* Framework de inyección de dependencias, con esto se soporta la encapsulación y modularización de componentes para facilitar la mantenibilidad del sistema. Asimismo, privilegia el performance en tiempo de ejecución dado que es un framework liviano.
* Framework de seguridad, con esto se soporta la meta de seguridad.

1. **Vista de Casos de Uso y Escenarios de Calidad**

Esta sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad en el análisis. Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

* 1. **Modelo de Casos de Uso**

El modelo de casos de uso puede ser encontrado en el documento “Casos de Uso”.

* 1. **Especificación de Casos de Uso Relevantes**

Los casos de uso considerados los más relevantes para el desarrollo de la arquitectura fueron determinados. Los criterios usados para dicha determinación fueron:

* Su implementación implica varios nodos de la vista de despliegue.
* Su implementación es de alto riesgo.
* Incluye muchos conceptos y relaciones del dominio.
* Incluye posibles escenarios críticos de calidad.

A continuación se listan los casos de uso relevantes, los cuales pueden ser encontrados con su especificación detallada en el documento “Casos de Uso”.

| **Código** | **Nombre** | **Actores** | **Prioridad** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* 1. **Especificación de los Escenarios de Calidad Relevantes**

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

**ID: QS1**

**Nombre:** Desempeño: Tiempo de respuesta en la generaicón de los archivos PDA.

**Sinopsis**: Tiempos de respuesta en la generación de archivos para la PDA que no afecten el actual proceso.

**Entorno**: Proceso normal de operación del sistema.

**Cambio en el entorno**: ejecución de proceso de carga PDA.

**Comportamiento esperado**: El proceso genera los archivos de carga en el file system.

**Medida**: debe demorar menos de 2 minutos por cada PDA cargada.

**Prioridad Arquitectónica**: Alta

**Aplicación**: Global

**ID: QS2**

**Nombre:** Tolerancia a fallos: Recuperación del sistema ante un fallo.

**Sinopsis**: Sistema debe poder reestablecerse en un tiempo razonable.

**Entorno**: El sistemsa trabajando en su carga normal.

**Cambio en el entorno**: El sistema sufre un crash.

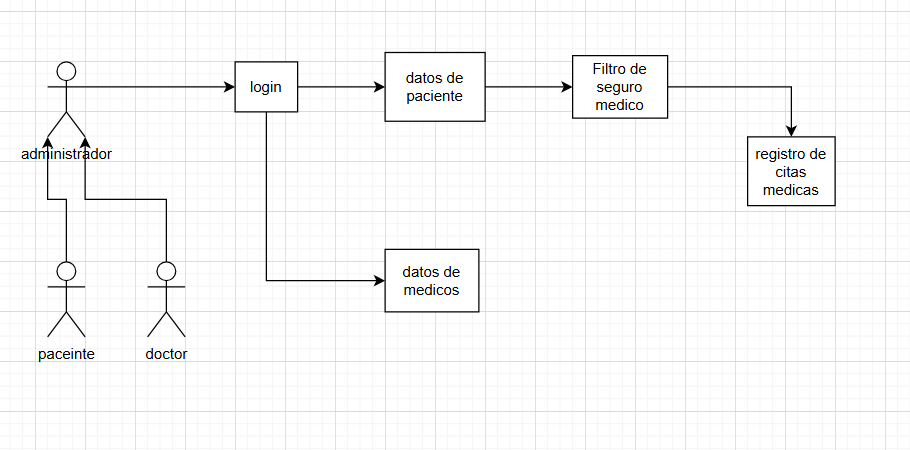
**Comportamiento esperado**: Se debe poder reestablecer el sistema para seguir con la operación normal.

**Medida**: debe demorar menos de 30 minutos en reestablecer el sistema.

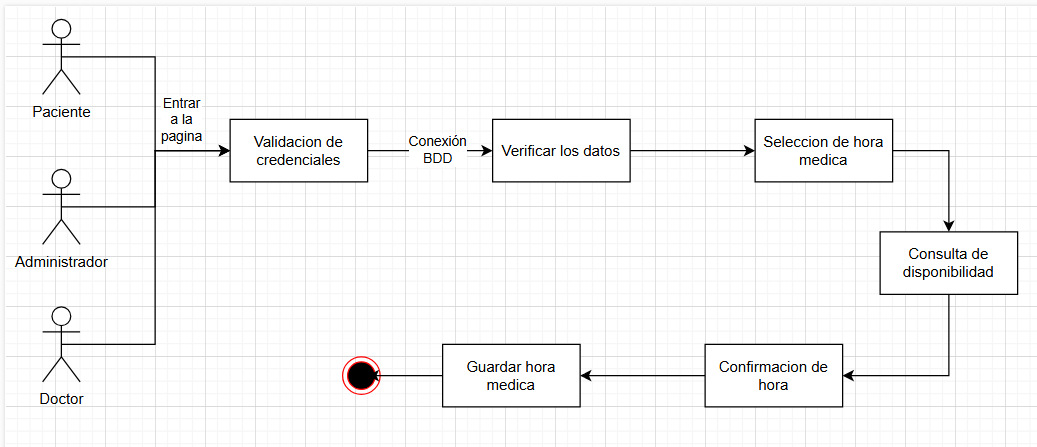
**Prioridad Arquitectónica**: Alta

**Aplicación**: Local

**Vista Lógica**



1. **Vista de Procesos**



1. **Vista de Implementación**

En esta vista se aprecia que existirán dos módulos principales que contendrán distintas funcionalidades de la aplicación. A continuación se describen:

**Ilustración 4: Vista de Implementación**

1. **Vista de Despliegue**

En esta vista se despliegan los nodos que participan con el sistema. Los nodos principales son los nodos Servidor de Integración. Características a continuación:

**Ilustración 6: Diagrama de Despliegue**

1. **Decisiones de Diseño y Selección de Alternativas**

Dado el poco tiempo que tenemos para el desarrollo de este proyecto decidimos usar Laravel un framework bastante robusto para estas situaciones y ampliamente utilizado, esto nos permite acelerar ampliamente el desarrollo de la aplicación.  
El software fue creado para estar dividido en diferentes módulos, son 3 los principales, el primero es el de gestión de citas médicas, el siguiente es la interfaz del usuario y por último la Base de Datos.  
Restricción: Infraestructura  
Se adoptó una arquitectura activa-activa para el servidor web, equilibrando la carga en horarios pico, y una base de datos activa-pasiva con réplica en tiempo real para garantizar continuidad en caso de fallos.

Escenario de Calidad: Mantenibilidad  
Se utilizó el principio de separación de preocupaciones y el patrón Provider para modularizar el sistema, facilitando su evolución e integración futura con aseguradoras externas como ISAPRE y FONASA.

Escenario de Calidad: Tolerancia a Fallos  
Se emplearon temporizadores (Quartz) para monitorear colas de mensajes y archivos de entrada, asegurando resiliencia y recuperación eficiente ante fallos externos o internos.