

**[AgenDoctor]**

**(DAS) Documento de Arquitetura de Software**

**Versión 1.0**

**Identificación de Documento**

| **Identificación** | Johan Collao - Cristian Quintriqueo |
| --- | --- |
| **Proyecto** | AgenDoctor |
| **Versión** | 1.0 |

| **Documento mantenido por** | Johan Collao - Cristian Quintriqueo |
| --- | --- |
| **Fecha de última revisión** | 15-09-2025 |
| **Fecha de próxima revisión** | Por definir |

| **Documento aprobado por** | Johan Collao - Cristian Quintriqueo |
| --- | --- |
| **Fecha de última aprobación** | 15-09-2025 |

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 03-04-2024 | 1.0 | Versión inicial | Johan Collao,Cristian Quintriqueo |

**Indice**

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

[1.1 Contexto del Problema](#_heading=h.30j0zll) 3

[1.2 Propósito](#_heading=h.1fob9te) 4

[1.3 Definiciones, acrónimos y abreviaciones](#_heading=h.3znysh7) 5

[1.4 Referencias](#_heading=h.2et92p0) 5

[1.5 Resumen ejecutivo](#_heading=h.tyjcwt) 5

[**2. Representación de la Arquitectura**](#_heading=h.hdyo0l5k2p2i) **7**

[**3. Metas y Restricciones de la Arquitectura**](#_heading=h.1t3h5sf) **8**

[3.1 Metas de la arquitectura](#_heading=h.2s8eyo1) 8

[3.2 Restricciones de la Arquitectura](#_heading=h.w1qa8giitztr) 9

[3.3 Otros antecedentes y consideraciones](#_heading=h.3rdcrjn) 9

[**4 Vista de Casos de Uso y Escenarios de Calidad**](#_heading=h.26in1rg) **11**

[4.1 Modelo de Casos de Uso](#_heading=h.lnxbz9) 11

[4.2 Especificación de Casos de Uso Relevantes](#_heading=h.35nkun2) 12

[4.3 Especificación de los Escenarios de Calidad Relevantes](#_heading=h.1ksv4uv) 13

[**5. Vista Lógica**](#_heading=h.qyyj3j9usgv3) **22**

[5.1 Modelo Relaciónal](#_heading=h.2jxsxqh) 22

[**6. Vista de Implementación**](#_heading=h.1y810tw) **23**

[**7. Vista de Procesos**](#_heading=h.fgutlwf3mier) **23**

[**8. Vista de Despliegue**](#_heading=h.4i7ojhp) **24**

[**9. Decisiones de Diseño y Selección de Alternativas**](#_heading=h.keafodjbhrmv) **25**

[**10. Análisis de Reutilización**](#_heading=h.mxhou2rlus2t) **25**

# 

# 

# Introducción

El presente Documento de Arquitectura de Software (DAS) proporciona una visión detallada de la estructura arquitectónica del sistema AgenDoctor. Esta plataforma digital ha sido diseñada para facilitar la interacción entre psicólogos y pacientes, permitiendo la gestión eficiente de citas, seguimiento de sesiones y comunicación segura, respondiendo a la creciente necesidad de soluciones accesibles en salud mental.

## Contexto del Problema

En la sociedad contemporánea, el cuidado de la salud mental ha adquirido una importancia cada vez mayor debido a diversos factores, como el estrés cotidiano, la presión laboral y los desafíos emocionales. Sin embargo, acceder a servicios de atención psicológica puede ser un proceso complejo y a menudo abrumador para muchos individuos. Tradicionalmente, la búsqueda de un psicólogo adecuado, la coordinación de horarios de citas y el seguimiento del progreso del tratamiento han sido tareas que implican tiempo y esfuerzo significativos. En este contexto surge AgenDoctor App, una solución innovadora diseñada para abordar los desafíos inherentes a la búsqueda y acceso a servicios de atención psicológica. Con su enfoque centrado en el usuario y su integración de herramientas tecnológicas avanzadas, AgenDoctor App se posiciona como una plataformaque conecta de manera eficiente a psicólogos y pacientes, ofreciendo beneficios significativos para ambas partes.

Según estudios recientes, aproximadamente el 17% de la población adulta en Chile experimentará algún trastorno de salud mental a lo largo de su vida, evidenciando la magnitud del desafío que AgenDoctor busca abordar.

Los desafíos que enfrentan tanto los psicólogos como los pacientes son diversos y multifacéticos. Por un lado, los pacientes pueden experimentar dificultades para encontrar un psicólogo que se ajuste a sus necesidades específicas, ya sea en términos de especialización, ubicación geográfica o disponibilidad de horarios. Este proceso de búsqueda puede ser frustrante y desalentador, lo que a menudo resulta en una postergación de la búsqueda de ayuda profesional.

Coordinar horarios de citas con pacientes, gestionar registros de sesiones y mantener la confidencialidad de la información son solo algunas de las responsabilidades que enfrentan en su trabajo. Además, la seguridad y la privacidad de los datos de los pacientes son aspectos críticos que deben ser abordados de manera efectiva para garantizar la confianza y la integridad del proceso terapéutico.

En este contexto, AgenDoctor surge como una solución integral que busca simplificar y optimizar el proceso de búsqueda y acceso a servicios de atención psicológica. Al proporcionar una plataforma digital intuitiva y segura, AgenDoctor tiene como objetivo mejorar la experiencia tanto para psicólogos como para pacientes, facilitando la conexión entre ambas partes y promoviendo un enfoque más accesible y eficiente para el cuidado de la salud mental.

Esta aplicación aborda las complejidades inherentes al proceso de búsqueda de servicios psicológicos al ofrecer características como la búsqueda de psicólogos según diferentes criterios, agendamiento de citas, revisión de registro de sesiones y la capacidad de dejar reseñas y calificaciones. Además, proporciona un entorno seguro donde los usuarios pueden acceder a los servicios de manera confiable y mantener la privacidad de sus datos. Con su enfoque centrado en el usuario, AgenDoctor App busca mejorar la accesibilidad y la efectividad de la atención psicológica, brindando una solución integral que se adapta a las necesidades de pacientes y profesionales por igual.

## Propósito

El propósito fundamental de AgenDoctor es abordar las necesidades insatisfechas en el acceso y la gestión de servicios de atención psicológica. En un mundo cada vez más digitalizado, el bienestar mental se ha vuelto una prioridad, y AgenDoctor App surge para ofrecer una solución innovadora y eficiente que facilite el encuentro entre pacientes y psicólogos.

La aplicación tiene como objetivo primordial simplificar y agilizar el proceso de búsqueda, selección y agendamiento de citas con psicólogos, tanto para pacientes como para profesionales de la salud mental. Para los pacientes, AgenDoctor representa una puerta de entrada hacia la atención psicológica, permitiéndoles explorar una amplia variedad de psicólogos según sus necesidades específicas, como ubicación, especialización y disponibilidad de horarios. Además, ofrece herramientas para gestionar citas, acceder a registro de sesiones y dejar reseñas, promoviendo así una experiencia integral y personalizada.

Por otro lado, para los psicólogos, AgenDoctor brinda una plataforma para administrar su práctica de manera eficiente y segura. Desde la gestión de horarios hasta el seguimiento de pacientes y la comunicación, la aplicación ofrece una serie de herramientas diseñadas para optimizar la experiencia tanto para el profesional como para el paciente.

En esencia, AgenDoctor aspira a ser mucho más que una simple herramienta tecnológica; busca ser un catalizador para el cuidado de la salud mental, proporcionando una experiencia digitalizada y centrada en el usuario que promueva el acceso equitativo y la calidad en la atención psicológica. Con este propósito en mente, la aplicación se posiciona como un recurso indispensable para aquellos que buscan mejorar su bienestar emocional y mental, así como para los profesionales comprometidos con brindar un servicio de calidad en el campo de la salud mental.

## Definiciones, acrónimos y abreviaciones

| **ACRÓNIMO** | **DESCRIPCIÓN** |
| --- | --- |
| *DAS* | Documento de Arquitectura de Software |
| *IVA* | Impuesto al Valor Agregado |
| *SSL* | Secure Sockets Layer - protocolo de seguridad |
| *GDPR* | Reglamento General de Protección de Datos |
| *CCPA* | Ley de Privacidad del Consumidor de California |
| *API* | Interfaz de Programación de Aplicaciones |
| *MVC* | Modelo-Vista-Controlador |
| *PC* | Computadora Personal (Personal Computer) |
| *WAF* | Firewall de Aplicación Web (Web Application Firewall) |
| *UI* | Interfaz de Usuario (User Interface) |
| *CRUD* | Crear, Leer, Actualizar y Eliminar (Create, Read, Update, Delete) - operaciones básicas en bases de datos y aplicaciones web |
| *HTTPS* | Hypertext Transfer Protocol Secure - Protocolo seguro de transferencia de hipertexto |
| *JSON* | JavaScript Object Notation - Notación de Objetos de JavaScript |
| *SQL* | Structured Query Language - Lenguaje de Consulta Estructurado |
| *IONIC* | Framework para aplicaciones híbridas |
| *NODE.JS* | Entorno de ejecución para JavaScript en el servidor |
| *AWS* | Amazon Web Services, plataforma de cloud computing |

## Referencias

A continuación, se listan las referencias a otros documentos:

* **Planilla de requerimientos**
* **Escenarios de calidad**
* **Matriz de riesgos**

## Resumen ejecutivo

El proyecto AgenDoctor surge en un contexto donde la salud mental y el bienestar emocional están cobrando cada vez más relevancia en la sociedad moderna. La creciente conciencia sobre la importancia de la salud mental, junto con los avances tecnológicos que han transformado la forma en que nos comunicamos y accedemos a la información, crea una oportunidad única para desarrollar una plataforma digital que facilite el acceso a servicios psicológicos de calidad.

La pandemia de COVID-19 ha exacerbado aún más la necesidad de soluciones innovadoras en el campo de la salud mental, ya que ha generado un aumento significativo en la demanda de atención psicológica debido al estrés, la ansiedad y otros problemas relacionados con la crisis sanitaria. En este contexto, AgenDoctor se presenta como una respuesta eficaz y oportuna para abordar estas necesidades crecientes, proporcionando una plataforma que simplifica y agiliza el proceso de búsqueda, selección y agendamiento de citas psicológicas.

El propósito fundamental de AgenDoctor es mejorar el acceso a la atención psicológica, tanto para pacientes como para profesionales, a través de una interfaz intuitiva y amigable que facilite la conexión entre ambos. Para los pacientes, la plataforma ofrecerá una amplia gama de psicólogos con diversas especialidades y ubicaciones geográficas, permitiendo una búsqueda personalizada que se adapte a las necesidades y preferencias individuales. Además, brindará información detallada sobre los perfiles de los psicólogos, incluyendo sus áreas de experiencia, horarios de disponibilidad y reseñas de otros usuarios, para ayudar a los pacientes a tomar decisiones informadas sobre su atención psicológica.

Por otro lado, para los psicólogos, AgenDoctor ofrecerá herramientas y funcionalidades diseñadas específicamente para simplificar la gestión de su práctica profesional. Esto incluirá la capacidad de administrar su agenda de citas, recibir notificaciones y recordatorios automáticos, acceder al historial de sesiones de los pacientes y generar informes sobre el progreso del tratamiento. Además, la plataforma proporcionará un entorno seguro y confidencial para la comunicación entre pacientes y psicólogos, garantizando la privacidad y la confidencialidad de la información personal y clínica.

En resumen, AgenDoctor tiene como objetivo principal ofrecer una solución integral y centrada en el usuario que facilite el acceso a la atención psicológica, mejore la experiencia de los pacientes y los profesionales, y contribuya a promover el bienestar emocional y la salud mental en la sociedad. Con su enfoque en la innovación tecnológica y la calidad del servicio, AgenDoctor busca posicionarse como una herramienta indispensable en el campo de la salud mental, brindando una experiencia de usuario de calidad y contribuyendo al bienestar de la comunidad.

# 2. Representación de la Arquitectura

La arquitectura del sistema "AgenDoctor" se basa en el enfoque del framework 4+1 y se implementa como una Arquitectura “Tres Capas”. Si bien el sistema utiliza elementos del patron Modelo-Vista-Controlador(MVC) en diferentes capas, la estructura general sigue un enfoque basado en servicios RESTful y componentes modulares.

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

Describe los casos de uso principales del sistema (agendamiento, gestión de usuarios, validación de profesionales) y cómo se asignan a los diferentes componentes de la arquitectura. Incluye escenarios de calidad críticos como: tiempo de respuesta < 2 segundos, disponibilidad del 99%, y protección de datos sensibles mediante autenticación JWT y cifrado.

1. **Vista Lógica**

Representa la descomposición del sistema en componentes modulares:

* Capa de Presentación: Aplicación híbrida Ionic con componentes Angular/React (Vista).
* Capa de Lógica de Negocio: API REST con Node.js/Express (Controladores/Servicios).
* Capa de Datos: Modelos MySQL gestionados mediante ORM.
* La separación sigue principios de alta cohesión y bajo acoplamiento.

1. **Vista de Procesos**

Describe el comportamiento dinámico del sistema durante la ejecución de procesos clave:

* Flujo de autenticación y autorización
* Proceso de agendamiento de citas
* Validación de credenciales profesionales
* Integración con servicios externos(pagos, notificaciones)

1. **Vista de Implementación**

Detalla la distribución física de los componentes:

* Frontend: Aplicacion web responsibe + Aplicación movil(Ionic)
* Backend: Node.js desplegados en AWS EC2
* Base de datos: MySQL
* Seguridad: API Gateway, SSL/TLS, Grupos de seguridad

Esta representación multi-vista proporciona una comprensión completa de la arquitectura de "AgenDoctor", facilitando el desarrollo, el mantenimiento y la evolución del sistema.

# 3. Metas y Restricciones de la Arquitectura

## 3.1 Metas de la arquitectura

Las metas de la arquitectura para el sistema "AgenDoctor" están diseñadas para garantizar un diseño sólido, eficiente y escalable que cumpla con los requisitos funcionales y no funcionales del proyecto. Estas metas guiarán en las decisiones técnicas y garantizarán la calidad del producto final.

* **Escalabilidad Horizontal** : La debe permitir el crecimiento del sistema para soportar incrementos en la cantidad de usuarios concurrentes, volumen de citas gestionadas y datos almacenados, mediante la adición de recursos distribuidos en   
  AWS sin requerir cambios significativos y en la base del código.
* **Mantenibilidad y Evolutividad**: El sistema debe facilitar la incorporación de nuevas funcionalidades, corrección de errores y adaptación a cambios en requisitos, mediante una estructura modular, código limpio, documentación técnica completa y pruebas automatizadas que permitan ciclos de desarrollo ágiles.
* **Seguridad**: Garantizar la confidencialidad, integridad y disponibilidad de la información sensible de usuarios y profesionales, implementando autenticación JWT, cifrado de datos en tránsito y en reposo, control de acceso basado en roles y cumplimiento de normativas de protección de datos.
* **Confiabilidad y tolerancia a fallos**: El sistema debe mantener su disponibilidad y funcionalidad ante fallos parciales, implementando estrategias de replicación, backups automáticos, manejo elegante de errores y monitoreo proactivo que garantice una disponibilidad objetivo del 99.5%.
* **Portabilidad multiplataforma**: Garantizar una experiencia consistente y funcional across diferentes dispositivos y navegadores mediante el uso de Ionic Framework, asegurando que la aplicación se comporte de manera uniforme en versión web y móvil.
* **Interoperabilidad con sistemas externos**: Proporcionar una interfaz que satisfaga las necesidades tanto de pacientes como de profesionales de la salud mental, reduciendo la curva de aprendizaje y maximizando la eficiencia en las operaciones diarias.

Estas metas arquitectónicas serán la base para la evaluación de las decisiones de diseño y servirán como criterio para validar la efectividad de la arquitectura implementada a lo largo del ciclo de vida del proyecto.

## 3.2 Restricciones de la Arquitectura

Las restricciones de la arquitectura para el sistema "AgenDoctor" están determinadas por diversos factores técnicos, normativos y operativos que deben ser considerados durante el diseño y desarrollo del sistema. Algunas de las restricciones más relevantes incluyen:

* **Tecnológicas**: Las tecnologías seleccionadas deben ser compatibles con los requisitos del sistema y con las capacidades del equipo de desarrollo. Por ejemplo, la plataforma móvil podría requerir el uso de tecnologías específicas como Ionic Framework para el desarrollo de la aplicación, mientras que el backend podría basarse en Node.js para garantizar la eficiencia y escalabilidad.
* **Normativas y Legales**: El sistema debe cumplir con las regulaciones y leyes de protección de datos, privacidad y seguridad de la información en Chile (Ley 19.628 sobre protección de la vida privada). Por otra parte también se debe considerar la validación de credenciales profesionales mediante el registro en superintendencia de salud y consentimiento digital para tratamiento de datos de salud.
* **Interoperabilidad**: El sistema debe ser capaz de integrarse con sistemas externos y plataformas de terceros, como servicios de pago y sistemas de verificación de psicólogos. Esto puede requerir el uso de APIs y estándares de interoperabilidad reconocidos para facilitar la comunicación y el intercambio de datos entre sistemas heterogéneos, formatos de intercambio JSON, codificación UTF-8 y protocolos HTTPS/TLS para las comunicaciones.
* **Rendimiento y Escalabilidad**: La arquitectura debe ser capaz de manejar cargas de trabajo variables y un crecimiento en el número de usuarios y transacciones sin comprometer el rendimiento del sistema. Esto implica el diseño de una infraestructura robusta y escalable, así como la optimización de consultas y operaciones para minimizar el tiempo de respuesta y maximizar la disponibilidad del sistema.
* **Costo y Recursos**: Los recursos financieros y humanos disponibles pueden imponer limitaciones en términos de tecnologías y herramientas que pueden ser utilizadas, así como en el tiempo y esfuerzo dedicados al desarrollo y mantenimiento del sistema. Es importante considerar estas restricciones al seleccionar las tecnologías y planificar la implementación del proyecto.

## 3.3 Otros antecedentes y consideraciones

Los atributos de calidad utilizados para el sistema "AgenDoctor", basado en una arquitectura monolítica, se detallan a continuación:

* **Desempeño**: Se prioriza un rendimiento óptimo para garantizar una respuesta rápida a las solicitudes de los usuarios. Se establece como objetivo que cada carga no supere los 1 segundo, lo que contribuirá a una experiencia de usuario fluida y una alta tasa de conversión. Esto es especialmente importante en un sistema de gestión de citas donde la rapidez en la reserva y confirmación de citas es fundamental para la satisfacción del usuario.
* **Tolerancia a fallos**: La arquitectura monolítica debe ser tolerante a fallos, lo que significa que debe ser capaz de manejar situaciones inesperadas y errores de manera adecuada. En caso de fallos, se implementarán mensajes de alerta para comunicar efectivamente cualquier problema al usuario, brindando transparencia y confianza en el sistema.
* **Seguridad**:

**Control de Acceso:**

* Autenticación multi-factor requerida para profesionales de la salud
* Roles de usuario predefinidos: paciente, psicólogo, coordinador, administrador
* Sesiones automáticas expiradas a los 30 minutos de inactividad

**Protección de Datos:**

* Cifrado AES-256 para datos en reposo
* TLS 1.3 para datos en tránsito
* Logs de auditoría retenidos por 2 años
* Backup automático diario con retención de 30 días

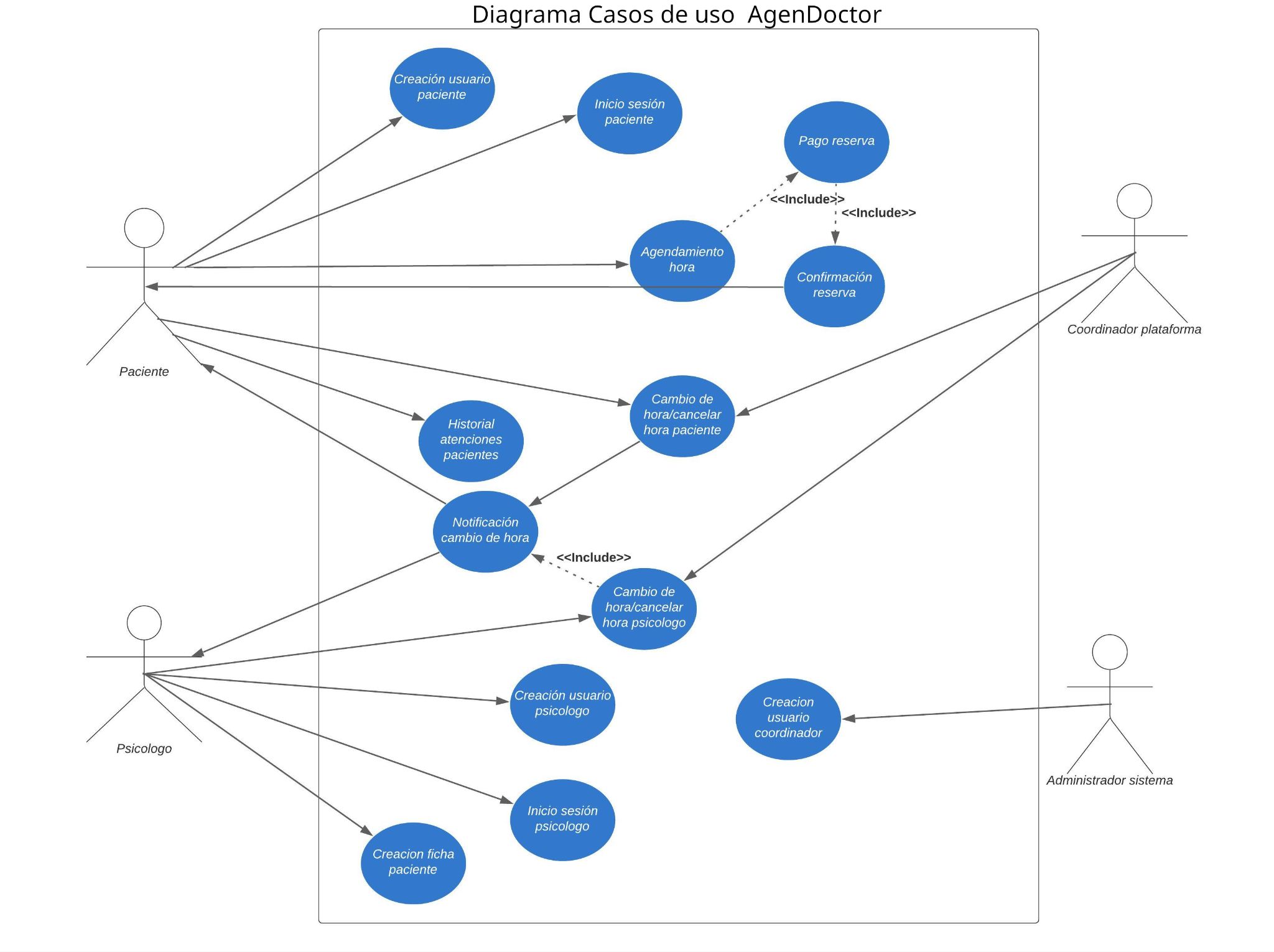
Además, se considera que la empresa desarrolladora cuenta con un framework que incluye componentes como el framework de inyección de dependencias y el framework de seguridad. Estos componentes permiten satisfacer los requerimientos arquitectónicos establecidos y se alinean con la elección de la arquitectura que busca simplificar la implementación y garantizar la seguridad del sistema "AgenDoctor".

# 4 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.

## 4.1 Modelo de Casos de Uso

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



## 4.2 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”.



## 4.3 Especificación de los Escenarios de Calidad Relevantes

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

| **Escenario de Calidad N° 1** | |
| --- | --- |
| **Atributo de Calidad Asociado (Característica): Mantenibilidad**  **Sub Característica: Modularidad** | |
| **Descripción: El sistema debe permitir la incorporación de nuevas funcionalidades o cambios en el diseño, sin afectar negativamente a las partes existentes del sistema.** | |
|  | |
| **Fuente del Estímulo:** | Desarrolladores del sistema. |
| **Estímulo:** | Se requiere agregar una nueva funcionalidad del sistema. |
| **Artefacto:** | Código Fuente |
| **Ambiente:** | Entorno del desarrollo del sistema |
| **Respuesta:** | El sistema permite la integración de la nueva funcionalidad de manera modular, sin causar fallas en otras partes del sistema |
| **Medida de Respuesta:** | La nueva funcionalidad se integra sin necesidad de modificar partes existentes del código |

| **Escenario de Calidad N° 2** | |
| --- | --- |
| **Atributo de Calidad Asociado (Característica): Usabilidad**  **Sub Característica: Eficiencia** | |
| **Descripción: Los usuarios deben poder acceder y utilizar el sistema de manera eficiente, desde dispositivos móviles, tablets y computadoras.** | |
|  | |
| **Fuente del Estímulo:** | Usuarios |
| **Estímulo:** | Acceso al sistema desde diferentes dispositivos |
| **Artefacto:** | Interfaz de usuario y funcionalidad del sistema |
| **Ambiente:** | Dispositivos de uso del sistema |
| **Respuesta:** | El sistema se adapta de manera eficiente a la pantalla y capacidades de los dispositivos utilizados, permitiendo la navegación fluida |
| **Medida de Respuesta:** | Los usuarios pueden realizar las tareas con la misma eficiencia independiente de los dispositivos que utilicen |

| **Escenario de Calidad N° 5** | |
| --- | --- |
| **Atributo de Calidad Asociado (Característica): Rendimiento**  **Sub Característica: Capacidad de respuesta** | |
| **Descripción: El sistema debe ser capaz de manejar simultáneamente entre 50 y 120 usuarios activos sin degradación significativa del rendimiento.** | |
|  | |
| **Fuente del Estímulo:** | Usuarios concurrentes del sistema |
| **Estímulo:** | Acceso y uso del sistema de forma simultánea por parte de múltiples usuarios. |
| **Artefacto:** | Recursos del sistema y gestión de sesiones. |
| **Ambiente:** | Entorno de acceso al sistema con usuarios concurrentes. |
| **Respuesta:** | El sistema mantiene un rendimiento aceptable incluso con la carga máxima de usuarios concurrentes. |
| **Medida de Respuesta:** | Se monitorea el rendimiento del sistema durante las pruebas de carga y se verifica que se mantenga dentro de los límites especificados. |

| **Escenario de Calidad N° 6** | |
| --- | --- |
| **Atributo de Calidad Asociado (Característica): Seguridad**  **Sub Característica: Confidencialidad** | |
| **Descripción: El sistema debe garantizar que los datos del cliente, como la información de contacto y los datos de pago, estén protegidos y sean confidenciales.** | |
|  | |
| **Fuente del Estímulo:** | Intentos de acceso no autorizado |
| **Estímulo:** | Intento de acceso a datos de clientes por parte de usuarios no autorizados |
| **Artefacto:** | Mecanismos de seguridad y acceso a datos |
| **Ambiente:** | Entorno de acceso al sistema |
| **Respuesta:** | El sistema previene el acceso no autorizado a los datos del cliente y garantiza su confidencialidad |
| **Medida de Respuesta:** | Se registran intentos de acceso no autorizado y se verifica que sean bloqueados |

| **Escenario de Calidad N° 7** | |
| --- | --- |
| **Atributo de Calidad Asociado (Característica): Fiabilidad**  **Sub Característica: Disponibilidad** | |
| **Descripción: Se debe mantener un respaldo diario de los datos en un medio físico extraíble y en la nube para evitar la pérdida de información crítica.** | |
|  | |
| **Fuente del Estímulo:** | Pérdida de datos debido a fallos del sistema |
| **Estímulo:** | Fallo del sistema o pérdida de datos |
| **Artefacto:** | Proceso de respaldo y medios de almacenamiento |
| **Ambiente:** | Entorno de almacenamiento de datos |
| **Respuesta:** | Los datos se respaldan diariamente en un medio físico extraíble y en la nube, asegurando la disponibilidad de la |
| **Medida de Respuesta:** | Se verifica la ejecución exitosa de los procesos de respaldo y la disponibilidad de los datos respaldados |

| **Escenario de Calidad N° 8** | |
| --- | --- |
| **Atributo de Calidad Asociado (Característica): Fiabilidad**  **Sub Característica: Integridad de los datos** | |
| **Descripción: Se deben registrar y almacenar todas las acciones realizadas por los usuarios, como las agendas de citas, modificaciones de datos y operaciones de administrador.** | |
|  | |
| **Fuente del Estímulo:** | Acciones de los usuarios en el sistema |
| **Estímulo:** | Realización de acciones por parte de los usuarios |
| **Artefacto:** | Registro de acciones y base de datos |
| **Ambiente:** | Entorno de acceso al sistema |
| **Respuesta:** | Todas las acciones de los usuarios se registran y almacenan de manera íntegra y precisa |
| **Medida de Respuesta:** | Se verifica la integridad de los registros de acciones y su coherencia con las acciones realizadas |

| **Escenario de Calidad N° 9** | |
| --- | --- |
| **Atributo de Calidad Asociado (Característica): Escalabilidad - Rendimiento**  **Subcaracterística: Capacidad de Crecimiento - Capacidad de Respuesta** | |
| **Descripción: El sistema debe ser escalable para adaptarse al crecimiento continuo de la empresa, tanto en términos de cantidad de productos como de usuarios.** | |
|  | |
| **Fuente del Estímulo:** | Crecimiento de la empresa y aumento de usuarios y productos |
| **Estímulo:** | Aumento en la cantidad de usuarios activos |
| **Artefacto:** | Recursos del sistema y capacidad de almacenamiento |
| **Ambiente:** | Entorno de crecimiento de la empresa |
| **Respuesta:** | El sistema se adapta de manera eficiente al aumento de psicologos y usuarios pacientes sin degradación del rendimiento |
| **Medida de Respuesta:** | Se evalúa el rendimiento del sistema bajo condiciones de crecimiento y se verifica que se mantenga dentro de los |

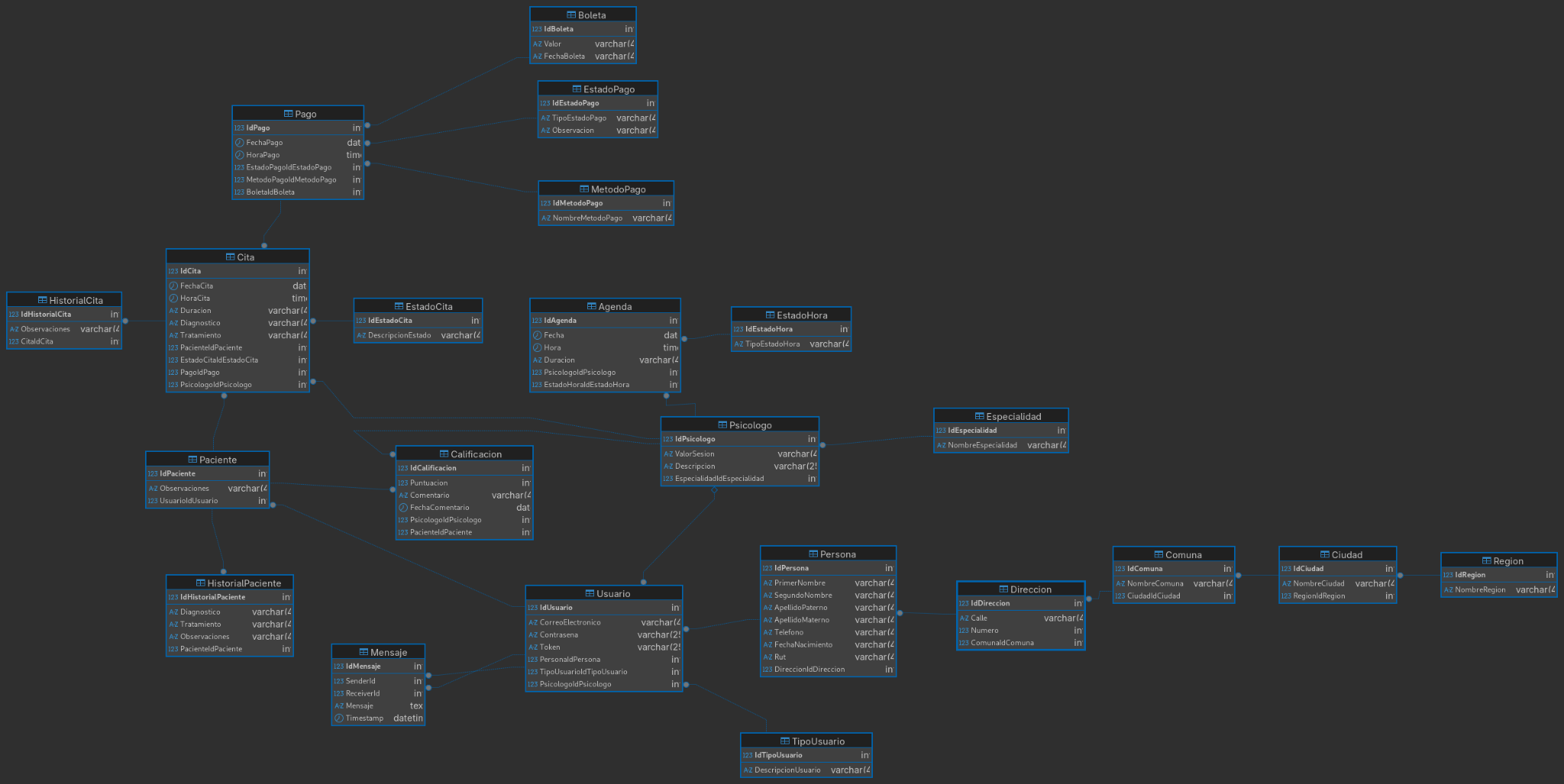
| **Escenario de Calidad N° 10** | |
| --- | --- |
| **Atributo de Calidad Asociado (Característica): Seguridad**  **Sub Característica: Autorización** | |
| **Descripción: El módulo de autenticación del sistema debe ser seguro y permitir el acceso adecuado de acuerdo con los perfiles de administrador y cliente.** | |
|  | |
| **Fuente del Estímulo:** | Usuarios que intentan acceder al sistema |
| **Estímulo:** | Intento de acceso al sistema por parte de administradores y clientes |
| **Artefacto:** | Módulo de autenticación y permisos de acceso |
| **Ambiente:** | Entorno de acceso al sistema |
| **Respuesta:** | El módulo de autenticación garantiza que solo los usuarios autorizados puedan acceder a las funciones correspondientes. |
| **Medida de Respuesta:** | Se verifica que los usuarios solo tengan acceso a las funciones para las cuales están autorizados. |

| **Escenario de Completitud** | |
| --- | --- |
| **Atributo de Calidad Asociado (Característica): Funcionalidad**  **Subcaracteristica: Completitud** | |
| **Descripción: verificar que el sistema AgenDoctor App cumple con todas las funcionalidades y requisitos especificados en su diseño y documentación.** | |
|  | |
| **Fuente del Estímulo:** | Usuario del sistema (administradores, Psicólogos y Pacientes) |
| **Estímulo:** | Los usuarios interactúan con el sistema realizando una serie de acciones y operaciones, cómo administrar usuarios, gestionar productos, registrar ventas, valorar productos, realizar pagos en línea, etc. Estas acciones representan las funcionalidades clave del sistema. |
| **Artefacto:** | Sistema completo |
| **Ambiente:** | Entorno de producción o un entorno de pruebas que simule las condiciones de producción. |
| **Respuesta:** | El sistema responde a las acciones de los usuarios ejecutando las operaciones solicitadas de acuerdo con su diseño y especificaciones.. |
| **Medida de Respuesta:** | Se verifica que todas las operaciones realizadas por los usuarios se completen correctamente sin errores ni omisiones. |

# 5. Vista Lógica

A continuación, se presenta una vista lógica de la aplicación expresado en el diagrama entidad relación.

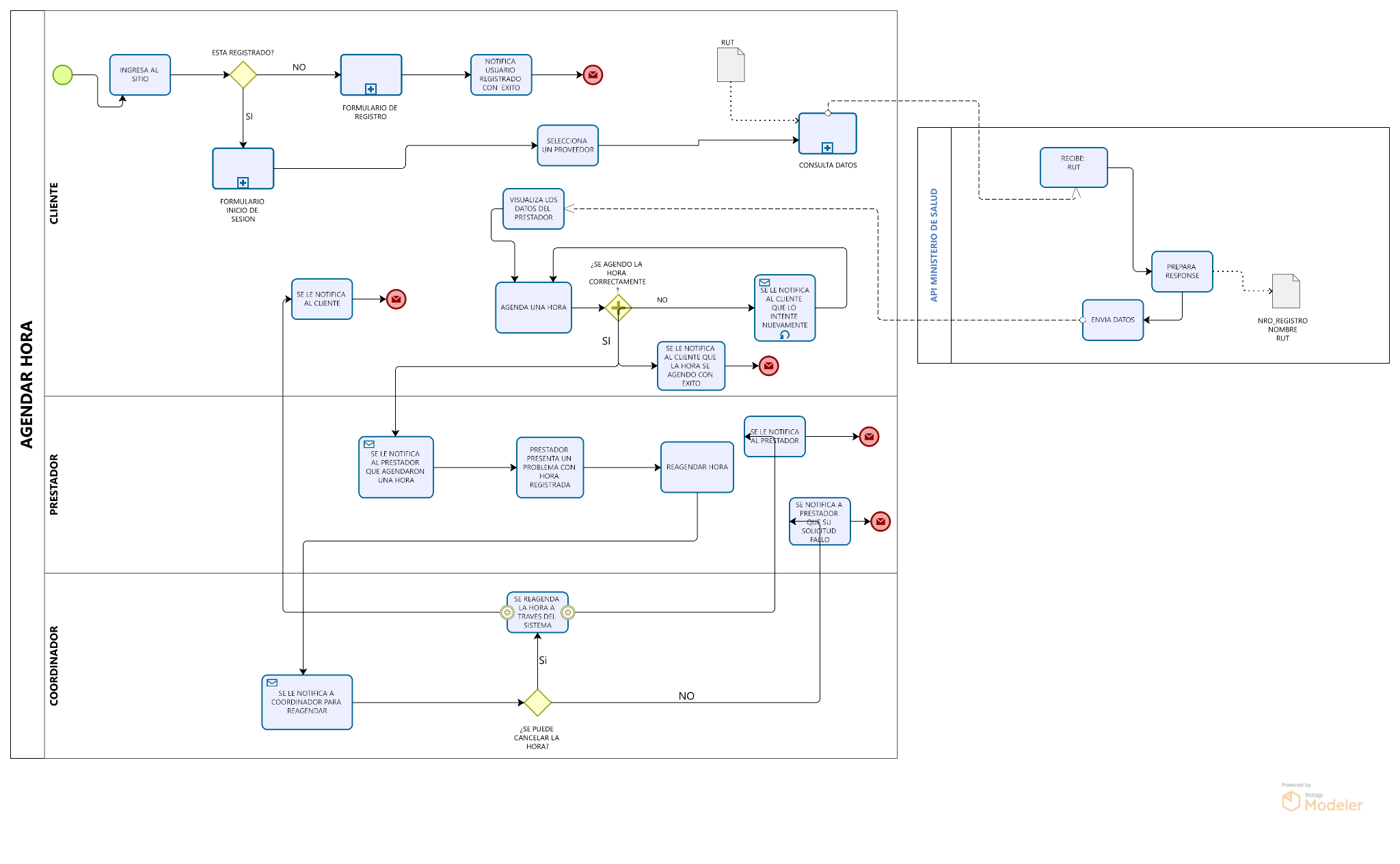
## 5.1 Modelo Relaciónal



Este modelo entidad-relación refleja las entidades clave y las relaciones entre ellas para gestionar la información según los requisitos de nuestro sistema. Puede servir como punto de partida para diseñar la base de datos y desarrollar la aplicación.

# 6. Vista de Implementación

# 7. Vista de Procesos



# 8. 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:

En esta vista de despliegue:

* Los usuarios pueden conectarse a la página web desde dispositivos como PC, notebook, smartphone o tablet.
* Antes de llegar al "Servidor de Aplicación Monolítica", se implementa un "Firewall" (Firewall de Aplicación Web o WAF) que actúa como un componente de seguridad. Este firewall verifica y asegura que el acceso proviene de direcciones IP confiables y protege la aplicación contra amenazas comunes, como ataques de seguridad web.
* El "Servidor de Aplicación Monolítica" es el núcleo de la aplicación, donde se encuentra la lógica de la aplicación, la base de datos, la seguridad, la autenticación y otras funciones. Aquí se pueden realizar compras, agregar productos y administrar la aplicación.
* La "Base de Datos" almacena toda la información del sistema y es accesible desde el servidor de aplicaciones.
* El "Módulo de Reportes" permite a los administradores generar informes a partir de los datos almacenados en la base de datos.
* El "Servicio de Autenticación" se integra en el servidor de aplicaciones y se encarga de gestionar la autenticación de los usuarios.
* El "Servidor de Respaldo" se encarga de realizar copias de seguridad de la base de datos y otros datos importantes.

Esta vista de despliegue ilustra la seguridad proporcionada por el firewall antes de que los usuarios accedan al servidor de aplicaciones y cómo se gestionan los datos a través de la base de datos, el servicio de autenticación y el servidor de respaldo. Esto asegura que los usuarios accedan a un entorno seguro y puedan interactuar con la aplicación de manera protegida.

# 9. Decisiones de Diseño y Selección de Alternativas

**Adopción de una arquitectura monolítica conocida y de bajo riesgo**: Dado el tiempo ajustado para la implementación de AgenDoctor, se optó por privilegiar una arquitectura monolítica, la cual es ampliamente conocida y presenta un bajo riesgo en su implementación. Esta decisión se tomó para garantizar una rápida entrega del producto sin comprometer su calidad ni estabilidad.

1. **Modularización de la arquitectura para facilitar la construcción y testeabilidad**: Se decidió modularizar la arquitectura de AgenDoctor con el objetivo de separar las preocupaciones y permitir la paralelización en la construcción de los diferentes componentes. Esta modularización permitirá que cada módulo sea testeable unitariamente, utilizando herramientas como JUnit para garantizar una baja tasa de fallos en cada pieza del sistema.
2. **Arquitectura activa-activa para la aplicación web y activa-pasiva para la aplicación de servicios**: Considerando las restricciones de infraestructura y el escenario de calidad relacionado con la tolerancia a fallos, se decidió modularizar la aplicación en una arquitectura activa-activa para la aplicación web y activa-pasiva para la aplicación de servicios. Esto garantizará la disponibilidad del sistema incluso en caso de fallos en uno de los nodos, asegurando así una experiencia de usuario continua y sin interrupciones.
3. **Enfoque en Mantenibilidad y Extensibilidad**: Para abordar el escenario de calidad relacionado con la mantenibilidad, se decidió separar las preocupaciones de los componentes y utilizar el patrón provider para delegar los requerimientos de información hacia sistemas externos. Esto permitirá una mayor extensibilidad del sistema en el futuro, ya que cada componente estará desacoplado y podrá ser fácilmente reemplazado o actualizado sin afectar al resto del sistema.
4. **Implementación de servicios asíncronos utilizando un temporizador (quartz)**: Para aquellos servicios asíncronos de entrada, se optó por implementarlos mediante un temporizador que levantará los procesos encargados de verificar la llegada de información a las colas de entrada. Esta estrategia simple y efectiva ayudará a garantizar la fiabilidad y la eficiencia en el manejo de los datos, minimizando el riesgo de errores y asegurando un flujo de trabajo sin interrupciones en AgenDoctor.

Estas decisiones de diseño y selección de alternativas se alinean con los objetivos y requisitos de AgenDoctor, garantizando una implementación eficiente y confiable del sistema dentro del tiempo establecido y sin comprometer la calidad ni la seguridad del producto final.

# 10. Análisis de Reutilización

En el caso de AgenDoctor, se ha llevado a cabo un análisis detallado para determinar qué componentes se re-utilizan y cuáles se desarrollarán desde cero. Esta estrategia se fundamenta en criterios específicos de necesidad, calidad y eficiencia del proyecto:

**Componentes a Reutilizar:**

* **Framework de Seguridad:** Dado que la confidencialidad de la información de los usuarios es una prioridad para AgenDoctor, se ha decidido reutilizar un framework de seguridad reconocido y confiable. Este componente proporcionará capas de protección para los datos del cliente y garantizará la integridad de la información sensible.
* **Servicio de Procesamiento de Pagos:** Para asegurar una gestión eficiente de las transacciones financieras en línea, se optará por utilizar un servicio de procesamiento de pagos establecido y probado en lugar de desarrollar uno desde cero. Esto garantizará la seguridad y la confiabilidad en las transacciones monetarias realizadas a través de la plataforma.

**Componentes a Desarrollar:**

* **Módulo de Gestión de Citas Personalizado:** Dado que AgenDoctor tiene requisitos específicos para la gestión de citas entre pacientes y psicólogos, se desarrollará un módulo personalizado para administrar este proceso de manera eficiente. Este módulo permitirá gestionar horarios, disponibilidad de psicólogos y reservas de citas de forma óptima.
* **Sistema de Evaluación de Sesiones:** Como parte de la experiencia del usuario, se desarrollará un sistema de evaluación de sesiones personalizado que permita a los pacientes calificar y revisar sus experiencias con los psicólogos. Esto ayudará a mejorar la calidad del servicio y proporcionará retroalimentación valiosa para la mejora continua.
* **Interfaz de Usuario (UI) Personalizada:** Se diseñará una interfaz de usuario personalizada que refleje la identidad y la marca de AgenDoctor. Esta interfaz garantizará una experiencia de usuario coherente en todos los dispositivos, lo que mejorará la usabilidad y la satisfacción del usuario.

**Razonamiento de la Reutilización:**

La elección de reutilizar componentes clave como el framework de seguridad y el servicio de procesamiento de pagos se basa en la necesidad de contar con soluciones probadas y confiables en áreas críticas para el proyecto. Esto permitirá ahorrar tiempo y recursos, además de garantizar la seguridad y la eficiencia en aspectos fundamentales de la plataforma.

Por otro lado, el desarrollo de componentes personalizados como el módulo de gestión de citas, el sistema de evaluación de sesiones y la interfaz de usuario personalizada se justifica por la necesidad de adaptar estas partes del sistema para que se ajusten perfectamente a la estrategia y la identidad de AgenDoctor. Esto permitirá ofrecer una experiencia de usuario única y satisfactoria, al tiempo que se cumplen con los requisitos específicos del negocio y las expectativas de los usuarios.