**Documento de Especificación**

**de Requisitos de Software**

**Acured**

|  |  |  |
| --- | --- | --- |
| **Asignatura** | **:** | **Capstone** |
| **Sigla** | **:** | **PTY4614** |
| **Profesor** | **:** | **Felix Eduardo Cifuentes Cid** |
|  |  |  |

|  |  |
| --- | --- |
| **Nombre del alumno** | **Correo del alumno** |
| Matias Ignacio Lara Romero | Mat.larar@duocuc.cl |
| Gonzalo Ignacio Lara Romero | Gonz.lara@duocuc.cl |

Contenido

[1. Introducción 6](#_Toc211617900)

[1.1. Propósito del Documento 6](#_Toc211617901)

[1.2. Situación actual del sistema 6](#_Toc211617902)

[1.2.1. Mapa de Procesos AS IS 7](#_Toc211617903)

[1.2.2. Diagrama BPMN AS IS 10](#_Toc211617904)

[1.3. Alcance del Sistema 13](#_Toc211617905)

[1.3.1. Procesos incluidos en el alcance 13](#_Toc211617906)

[1.3.2. Procesos excluidos del alcance 13](#_Toc211617907)

[1.3.3. Actores contemplados en el alcance 14](#_Toc211617908)

[1.3.4. Entradas, procesos y salidas del sistema 14](#_Toc211617909)

[1.3.5. Dependencias y restricciones del alcance 14](#_Toc211617910)

[1.3.6. Impacto esperado 15](#_Toc211617911)

[1.4. Definiciones, Acrónimos y Abreviaturas 15](#_Toc211617912)

[Definiciones 15](#_Toc211617913)

[Acrónimos y Abreviaturas 15](#_Toc211617914)

[1.5. Referencias 16](#_Toc211617915)

[2. Descripción General del Sistema 17](#_Toc211617916)

[2.1. Perspectiva del Producto 17](#_Toc211617917)

[2.1.1. Mapa de Procesos TO BE 18](#_Toc211617918)

[2.1.2. Diagrama BPMN TO BE 20](#_Toc211617919)

[2.2. Funciones del Producto 22](#_Toc211617920)

[2.3. Características de Usuario 23](#_Toc211617921)

[2.4. Restricciones Generales 24](#_Toc211617922)

[3. Requisitos Específicos 25](#_Toc211617923)

[3.1. Requisitos Funcionales 26](#_Toc211617924)

[3.2. Requisitos No Funcionales 27](#_Toc211617925)

[3.3. Otros Requisitos 30](#_Toc211617926)

[4. Interfaces gráficas 31](#_Toc211617927)

[4.1. Interfaces de Usuario 32](#_Toc211617928)

[Pantalla de Inicio de Sesión 32](#_Toc211617929)

[Dashboard del Terapeuta 32](#_Toc211617930)

[Módulo de Gestión de Pacientes 33](#_Toc211617931)

[Agenda del Paciente 33](#_Toc211617932)

[Reportes y Exportación 33](#_Toc211617933)

[4.2. Interfaces de Hardware 35](#_Toc211617934)

[Listado de Hardware Compatible 35](#_Toc211617935)

[Especificaciones Técnicas 35](#_Toc211617936)

[Protocolos de Comunicación 35](#_Toc211617937)

[Requisitos de Datos y Formatos 36](#_Toc211617938)

[Descripción de la Interfaz 36](#_Toc211617939)

[4.3. Interfaces de Software 37](#_Toc211617940)

[Descripción de las Interfaces 37](#_Toc211617941)

[Especificaciones Técnicas 37](#_Toc211617942)

[Requerimientos de Autenticación y Seguridad 37](#_Toc211617943)

[Gestión de Errores y Excepciones 38](#_Toc211617944)

[Dependencias y Limitaciones 38](#_Toc211617945)

[4.4. Interfaces de Comunicación 39](#_Toc211617946)

[Tipos de Comunicación 39](#_Toc211617947)

[Protocolos de Comunicación 39](#_Toc211617948)

[Formatos de Datos 39](#_Toc211617949)

[Interfaces y Servicios Externos 39](#_Toc211617950)

[Requerimientos de Ancho de Banda y Latencia 40](#_Toc211617951)

[Mecanismos de Seguridad y Autenticación 40](#_Toc211617952)

[5. Arquitectura de Software 41](#_Toc211617953)

[5.1. Modelo en Capas 41](#_Toc211617954)

[Descripción de las Capas 41](#_Toc211617955)

[Interacción Entre Capas 41](#_Toc211617956)

[Tecnologías y Frameworks 42](#_Toc211617957)

[Seguridad y Transacciones 42](#_Toc211617958)

[Consideraciones de Performance 42](#_Toc211617959)

[5.1.1. 4.5.6 Diagrama de Arquitectura en Capas 42](#_Toc211617960)

[5.2. Comunicación por Medio de API Rest 43](#_Toc211617961)

[Descripción General de la API 43](#_Toc211617962)

[Endpoints y Métodos HTTP 43](#_Toc211617963)

[Formatos de Mensajes 44](#_Toc211617964)

[Parámetros y Cuerpos de Solicitud: 44](#_Toc211617965)

[Códigos de Estado HTTP 44](#_Toc211617966)

[Seguridad y Autenticación 44](#_Toc211617967)

[Limitaciones y Cuotas 45](#_Toc211617968)

[5.3. Comunicación por Medio de Webservices SOAP (Simple Object Access Protocol) 46](#_Toc211617969)

[Justificación 46](#_Toc211617970)

[Consideraciones Futuras 46](#_Toc211617971)

[6. Requisitos de Software y Hardware 47](#_Toc211617972)

[6.1. Requisitos de Hardware 47](#_Toc211617973)

[Procesador (CPU) 47](#_Toc211617974)

[Memoria (RAM) 47](#_Toc211617975)

[Espacio de Almacenamiento 47](#_Toc211617976)

[Tarjeta Gráfica (GPU) 47](#_Toc211617977)

[Periféricos 47](#_Toc211617978)

[Conectividad 47](#_Toc211617979)

[Sistemas Operativos Compatibles 47](#_Toc211617980)

[6.2. Requisitos de Software 49](#_Toc211617981)

[Compatibilidad de Sistemas Operativos 49](#_Toc211617982)

[Requerimientos de Procesamiento (CPU) 49](#_Toc211617983)

[Requisitos de Memoria RAM 49](#_Toc211617984)

[Espacio de Almacenamiento Necesario 49](#_Toc211617985)

[Requisitos de Hardware Específico 49](#_Toc211617986)

[Dependencias de Software de Terceros 50](#_Toc211617987)

[7. Planificación 51](#_Toc211617988)

[7.1. Definición del Equipo de Trabajo 51](#_Toc211617989)

[Roles y Responsabilidades 51](#_Toc211617990)

[Estructura del Equipo 51](#_Toc211617991)

[Herramientas y Métodos de Comunicación 52](#_Toc211617992)

[Plan de Desarrollo y Capacitación 52](#_Toc211617993)

[Estrategias de Resolución de Conflictos 52](#_Toc211617994)

[Criterios de Éxito del Equipo 52](#_Toc211617995)

[7.2. Definición de Actividades principales del Proyecto 54](#_Toc211617996)

[Lista y Descripción de Actividades 54](#_Toc211617997)

[8. Conclusiones 57](#_Toc211617998)

[9. Referencias Bibliográficas 58](#_Toc211617999)

**Historial de Versiones**

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

**Información del Proyecto**

|  |  |
| --- | --- |
| **Empresa / Organización** |  |
| **Proyecto** |  |
| **Fecha de preparación** |  |
| **Cliente** |  |
| **Patrocinador principal** |  |
| **Gerente / Líder de Proyecto** |  |
| **Gerente / Líder de Análisis de negocio y requerimientos** |  |

**Aprobaciones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introducción

## Propósito del Documento

El presente Documento de Especificación de Requisitos de Software (ERS) tiene como propósito definir de manera clara, completa y precisa los requisitos funcionales y no funcionales del sistema **Acured**. Este documento busca servir como un marco de referencia común entre los distintos actores del proyecto, incluyendo el equipo de desarrollo, la dirección de proyecto, el docente supervisor y los usuarios finales (terapeutas de medicina oriental y pacientes), asegurando que todos los interesados compartan una visión unificada de los objetivos y características del sistema.

La finalidad principal de este ERS es guiar el desarrollo de la plataforma Acured, cuyo objetivo es digitalizar los registros clínicos de los pacientes, mejorar la trazabilidad de los tratamientos y optimizar la gestión de agendamiento de horas. De esta manera, se reemplaza el actual modelo basado en registros manuales en papel, reduciendo riesgos de pérdida de información y limitaciones en el análisis de datos.

Este documento cumplirá las siguientes funciones:

* Base de acuerdo entre clientes, usuarios y desarrolladores sobre el alcance y funcionalidades del sistema.
* Guía técnica de referencia para el equipo de desarrollo durante las fases de diseño, implementación y pruebas.
* Marco para pruebas de calidad, facilitando la verificación y validación de los requisitos especificados.
* Soporte para nuevos integrantes del equipo, entregando una visión clara del proyecto y sus objetivos.
* Medio de comunicación accesible para stakeholders no técnicos, asegurando comprensión de las capacidades y limitaciones del sistema.

En conclusión, este ERS constituye el documento formal que garantizará que el sistema **Acured** se desarrolle conforme a las necesidades identificadas, satisfaciendo tanto las expectativas de los usuarios como los objetivos académicos y técnicos del proyecto.

## Situación actual del sistema

En la actualidad, los terapeutas de medicina oriental gestionan la información clínica de sus pacientes mediante fichas en papel, cuadernos físicos o planillas simples en hojas de cálculo. Este enfoque presenta múltiples limitaciones que impactan directamente en la calidad de la gestión clínica y en la experiencia tanto de terapeutas como de pacientes.

Entre las principales problemáticas identificadas se destacan:

* Falta de centralización de la información: los registros clínicos se encuentran dispersos, sin un repositorio digital único que permita un acceso rápido y organizado.
* Riesgo de pérdida de datos: el uso de documentos físicos expone la información a deterioro, extravío o acceso no autorizado.
* Dificultad en la trazabilidad clínica: La ausencia de un sistema informatizado dificulta dar seguimiento a los tratamientos realizados y evaluar la evolución de los pacientes.
* Limitaciones en el análisis de información: Al no existir una base de datos digital estructurada, resulta imposible generar reportes confiables, aplicar herramientas de inteligencia de negocios o realizar estudios clínicos sistemáticos.
* Gestión manual de agendamiento: La coordinación de horas de atención entre pacientes y terapeutas se realiza por medios informales (agenda física o comunicación verbal), lo que genera descoordinaciones y pérdida de tiempo.

El escenario descrito refleja la necesidad de modernizar los procesos actuales mediante la implementación de un sistema digital centralizado. La plataforma Acured surge como respuesta a esta problemática, proponiendo una solución que permitirá digitalizar los registros clínicos, mejorar la trazabilidad de los tratamientos y automatizar el agendamiento de citas, garantizando mayor seguridad, eficiencia y capacidad de análisis en la gestión de la información clínica.

### Mapa de Procesos AS IS

|  |
| --- |
| **Mapa de procesos AS IS** |
| Diagrama  El contenido generado por IA puede ser incorrecto. |
| **Figura 1: Mapa de Procesos AI IS. Esta imagen puede que no se vea completa debido a su tamaño, pero puede revisar el mapa completo en el archivo “1.2.1 Mapa de Procesos AS IS.jpg”.** |

### Diagrama BPMN AS IS

|  |
| --- |
| **Diagrama BPMN AS IS** |
| Diagrama, Esquemático  El contenido generado por IA puede ser incorrecto. |
| **Figura 2: Diagrama BPMN AI IS. Esta imagen puede que no se vea completa debido a su tamaño, pero puede revisar el diagrama completo en el archivo “1.2.2 Diagrama BPMN AS IS.bpm.”.** |

## Alcance del Sistema

El sistema **Acured** abarcará el desarrollo e implementación de una plataforma web modular, basada en arquitectura de microservicios, cuyo objetivo es digitalizar y optimizar la gestión clínica de terapeutas de medicina oriental, con énfasis en la práctica de acupuntura.

El alcance incluye la definición de requerimientos funcionales, diseño de la arquitectura, implementación de módulos principales, integración de componentes de frontend y backend, despliegue en un entorno académico controlado, pruebas de calidad y entrega final de un producto funcional.

### Procesos incluidos en el alcance

1. **Gestión de pacientes**
   1. Registro, edición y eliminación de datos personales y clínicos.
   2. Consulta del historial clínico digital.
   3. Control de accesos diferenciados (terapeutas vs. pacientes).
2. **Gestión clínica y de tratamientos**
   1. Creación y actualización de fichas clínicas digitales.
   2. Registro de tratamientos aplicados por sesión.
   3. Seguimiento de evolución por paciente.
   4. Generación de reportes individuales y grupales.
3. **Gestión de agenda**
   1. Disponibilización de horarios por parte de terapeutas.
   2. Reserva de horas por parte de pacientes.
   3. Confirmación, modificación y cancelación de citas.
   4. Notificación básica a pacientes (correo o sistema interno).
4. **Seguridad y administración del sistema**
   1. Autenticación de usuarios mediante credenciales.
   2. Encriptación y validación de datos sensibles.
   3. Gestión básica de roles (paciente / terapeuta / administrador académico).
5. **Reportes y análisis de información**
   1. Reportes básicos de citas, asistencia y tratamientos.
   2. Exportación de información clínica en formatos estándar (CSV/PDF).
   3. Preparación de datos para análisis en futuras fases (ej. BI).

### Procesos excluidos del alcance

* Integración con sistemas de facturación, contabilidad o ERP.
* Interoperabilidad con plataformas de salud pública o privada.
* Implementación de módulos de inteligencia artificial.
* Aplicaciones móviles nativas (solo versión web responsiva).
* Soporte multilingüe (en esta primera versión solo estará en español).
* Servicios en la nube con escalabilidad a nivel productivo (se usará despliegue local/académico).

### Actores contemplados en el alcance

* **Pacientes:** usuarios que registran sus datos y gestionan sus citas.
* **Terapeutas de acupuntura:** responsables de registrar fichas clínicas, actualizar tratamientos y gestionar disponibilidad de atención.
* **Administrador del sistema / docente supervisor:** encargado de validar pruebas, monitorear el correcto funcionamiento y recibir reportes académicos del avance del proyecto.

### Entradas, procesos y salidas del sistema

* **Entradas:**
  + Datos clínicos del paciente.
  + Solicitudes de citas.
  + Disponibilidad horaria de terapeutas.
* **Procesos:**
  + Registro y validación de la información ingresada.
  + Procesamiento de citas (crear, modificar, cancelar).
  + Registro de evolución clínica y generación de fichas digitales.
  + Encriptación y almacenamiento en BD.
* **Salidas:**
  + Confirmaciones de citas.
  + Fichas clínicas digitalizadas.
  + Reportes clínicos y administrativos.
  + Historial de tratamientos y seguimiento.

### Dependencias y restricciones del alcance

* **Dependencias:**
  + Conexión a internet estable para acceso al sistema.
  + Disponibilidad de computadores personales para usuarios.
  + Uso de herramientas de desarrollo acordadas: React, Spring Boot, PostgreSQL.
* **Restricciones:**
  + El desarrollo debe completarse dentro del semestre académico establecido.
  + Se utilizarán exclusivamente herramientas open source y gratuitas.
  + El sistema no manejará datos de pacientes reales, sino casos de prueba simulados, por motivos de privacidad.

### Impacto esperado

* Optimización de la trazabilidad clínica.
* Reducción de errores asociados a registros manuales.
* Mayor eficiencia en la coordinación de citas entre terapeutas y pacientes.
* Disponibilidad de información centralizada y estructurada para análisis futuro.
* Mejora de la experiencia del paciente al ofrecer un servicio digitalizado.

## Definiciones, Acrónimos y Abreviaturas

A continuación, se presentan las principales definiciones, acrónimos y abreviaturas utilizadas en este documento, con el fin de asegurar un entendimiento común entre todos los actores del proyecto.

### Definiciones

* **Acured:** Sistema en desarrollo para clínicas de medicina oriental, orientado a digitalizar fichas clínicas, gestionar tratamientos y administrar agendas de atención.
* **Centro de Práctica:** Entidad colaboradora (Triskeledu SpA) donde el estudiante desarrolla sus funciones aplicando tecnologías y metodologías en el marco del proyecto Acured

Matías Lara Validación Centro P…

.

* **Ficha clínica digital:** Registro electrónico de datos personales, diagnósticos y tratamientos de pacientes.
* **Agenda digital:** Módulo del sistema que permite reservar y administrar citas entre pacientes y terapeutas.
* **Arquitectura de microservicios:** Diseño de software en el cual las funcionalidades del sistema se implementan como servicios independientes.
* **Práctica profesional/laboral:** Proceso formativo supervisado donde el estudiante aplica competencias adquiridas en su carrera.

### Acrónimos y Abreviaturas

* **API:** *Application Programming Interface*. Interfaz que permite comunicación entre frontend y backend.
* **AS IS:** Representación de los procesos actuales antes de la digitalización.
* **BD:** Base de Datos.
* **BPMN:** *Business Process Model and Notation*. Notación estándar para diagramas de procesos.
* **CRUD:** *Create, Read, Update, Delete*. Operaciones básicas sobre datos en sistemas de información.
* **DTO:** *Data Transfer Object*. Objeto usado para transportar datos entre capas de la aplicación (ej. en Spring Boot).
* **ERS:** Especificación de Requisitos de Software (*Software Requirements Specification*).
* **Figma:** Herramienta de prototipado y diseño de interfaces UI/UX.
* **Formik / React Hook Form:** Bibliotecas de validación de formularios en React.
* **Frontend:** Componente visual de la aplicación, desarrollado en React.
* **Backend:** Componente encargado de la lógica de negocio, implementado en Spring Boot.
* **IDE:** *Integrated Development Environment*. Entorno de desarrollo (ej. IntelliJ IDEA, VS Code).
* **JUnit / Mockito:** Frameworks de pruebas unitarias en Java, usados en el backend.
* **OpenAPI / Swagger:** Estándar para documentar servicios REST, utilizado en la práctica.
* **PDF:** *Portable Document Format*. Formato usado para exportar reportes.
* **PostgreSQL:** Sistema de gestión de bases de datos relacional.
* **REST:** *Representational State Transfer*. Estilo de arquitectura de APIs.
* **SQL:** *Structured Query Language*. Lenguaje de consultas para bases de datos.
* **TO BE:** Representación de los procesos futuros, con el sistema digital implementado.
* **UI:** *User Interface*. Interfaz de usuario.
* **UML:** *Unified Modeling Language*. Lenguaje estándar para modelado de sistemas.
* **UX:** *User Experience*. Experiencia de usuario.
* **VS Code:** Editor de código multiplataforma usado en el desarrollo frontend.

## Referencias

* IEEE. (1998). *IEEE Recommended Practice for Software Requirements Specifications* (IEEE Std 830-1998). Institute of Electrical and Electronics Engineers.
* Object Management Group (OMG). (2011). *Business Process Model and Notation (BPMN) Version 2.0*. Recuperado de: <https://www.omg.org/spec/BPMN/2.0/>
* Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.). Addison-Wesley.
* Sommerville, I. (2011). *Ingeniería del Software* (9ª ed.). Pearson Educación.
* Wiegers, K., & Beatty, J. (2013). *Software Requirements* (3rd ed.). Microsoft Press.
* React. (2025). *React Documentation*. Recuperado de: <https://react.dev>
* Spring. (2025). *Spring Boot Documentation*. Recuperado de: <https://spring.io/projects/spring-boot>
* PostgreSQL. (2025). *PostgreSQL Documentation*. Recuperado de: <https://www.postgresql.org/docs/>
* OpenAPI Initiative. (2025). *Swagger / OpenAPI Specification*. Recuperado de: <https://swagger.io/specification/>
* JUnit. (2025). *JUnit 5 User Guide*. Recuperado de: <https://junit.org/junit5/>
* Mockito. (2025). *Mockito Framework Documentation*. Recuperado de: <https://site.mockito.org/>
* Formik. (2025). *Formik Documentation*. Recuperado de: <https://formik.org>
* React Hook Form. (2025). *React Hook Form Documentation*. Recuperado de: <https://react-hook-form.com>
* Axios. (2025). *Axios HTTP Client Documentation*. Recuperado de: <https://axios-http.com>
* GitHub. (2025). *GitHub Documentation*. Recuperado de: <https://docs.github.com>
* Figma. (2025). *Figma Design Tool*. Recuperado de: <https://www.figma.com>
* Lucidchart. (2025). *Diagramming and Visual Collaboration Tool*. Recuperado de: <https://www.lucidchart.com>
* Bizagi. (2025). *Modeler for BPMN Diagrams*. Recuperado de: <https://www.bizagi.com>
* Duoc UC. (2025). *Políticas y normativas académicas*. Recuperado de: [https://www.duoc.cl](https://www.duoc.cl/)

# Descripción General del Sistema

## Perspectiva del Producto

El sistema **Acured** se concibe como una plataforma web modular orientada a la digitalización de los procesos clínicos en consultas de medicina oriental, específicamente en el área de acupuntura. Este producto está siendo desarrollado en el marco de la asignatura Capstone y en el contexto del Centro de Práctica Triskeledu SpA, empresa dedicada a la capacitación, desarrollo de software y consultorías informáticas.

Actualmente, los terapeutas gestionan la información de pacientes en fichas de papel y agendas manuales, lo que limita la trazabilidad, genera riesgos de pérdida de datos y dificulta el análisis clínico. El sistema propuesto responde a esta problemática proporcionando un entorno digital seguro, escalable y accesible.

Características principales del producto:

* Centralización de datos clínicos en una base de datos relacional PostgreSQL, garantizando integridad y persistencia de la información.
* Desarrollo de un frontend en React, enfocado en usabilidad y experiencia de usuario.
* Implementación de un backend en Spring Boot, bajo arquitectura de microservicios, que asegura modularidad y escalabilidad.
* Uso de herramientas y prácticas de la industria (Swagger/OpenAPI para documentación de APIs, pruebas unitarias con JUnit y Mockito, validaciones en DTOs y formularios con bibliotecas como Formik o React Hook Form).
* Posibilidad de extenderse en el futuro mediante APIs REST, facilitando integración con otros sistemas.

Relación con el entorno:

* Usuarios finales: terapeutas de medicina oriental y pacientes, quienes accederán al sistema a través de un navegador web responsivo.
* Contexto académico y empresarial: el desarrollo se enmarca en la práctica profesional en Triskeledu SpA, aplicando metodologías y estándares de documentación (IEEE 830, UML, ERD, BPMN) para asegurar la calidad del producto.
* Dependencias técnicas: requiere un entorno de desarrollo basado en IDEs como IntelliJ y VS Code, repositorios en GitHub y despliegue en servidores locales o en nube académica.
* Limitaciones actuales: en su primera versión no incluye interoperabilidad con sistemas hospitalarios ni facturación, centrándose en la gestión clínica básica y agendamiento de horas.

En conclusión, Acured se presenta como un producto que transforma el actual modelo manual en un sistema digital confiable, alineado con las buenas prácticas de la industria del software y con impacto directo en la eficiencia operativa y en la calidad del servicio de atención clínica.

### Mapa de Procesos TO BE

|  |
| --- |
| **Mapa de procesos TO BE** |
|  |
| **Figura 3: Mapa de Procesos TO BE. Esta imagen puede que no se vea completa debido a su tamaño, pero puede revisar el mapa completo en el archivo “2.1.1 Mapa de Procesos TO BE.jpg”.** |

### Diagrama BPMN TO BE

|  |
| --- |
| **Diagrama BPMN TO BE** |
|  |
| **Figura 4: Diagrama BPMN AI IS. Esta imagen puede que no se vea completa debido a su tamaño, pero puede revisar el diagrama completo en el archivo “2.1.2 Diagrama BPMN TO BE.bpm.”.** |

## Funciones del Producto

El sistema **Acured** contempla las siguientes funciones principales:

1. **Gestión de pacientes:**
   1. Registrar nuevos pacientes con información personal y clínica.
   2. Consultar y actualizar datos de pacientes existentes.
   3. Acceder al historial clínico de atenciones y tratamientos realizados.
2. **Gestión de fichas clínicas digitales:**
   1. Crear fichas clínicas electrónicas para cada sesión de tratamiento.
   2. Registrar diagnósticos, procedimientos y observaciones del terapeuta.
   3. Almacenar la información en una base de datos relacional (**PostgreSQL**) para garantizar persistencia y trazabilidad.
3. **Gestión de tratamientos:**
   1. Registrar los tratamientos aplicados a cada paciente en las distintas sesiones.
   2. Permitir la consulta de la evolución clínica a lo largo del tiempo.
   3. Generar reportes de diagnóstico y de evolución para uso del terapeuta.
4. **Agendamiento de horas:**
   1. Disponibilizar horarios por parte de los terapeutas.
   2. Permitir a los pacientes reservar citas en línea a través de la plataforma.
   3. Confirmar, modificar o cancelar citas registradas en el sistema.
   4. Notificar a los usuarios sobre el estado de sus reservas.
5. **Gestión de usuarios y seguridad:**
   1. Autenticación de usuarios mediante credenciales.
   2. Control de accesos basado en roles (paciente, terapeuta, administrador académico).
   3. Validación de datos de entrada para garantizar consistencia e integridad.
6. **Reportes clínicos y administrativos:**
   1. Generar reportes básicos de pacientes, tratamientos y citas agendadas.
   2. Exportar información en formatos estándar (PDF o CSV).
   3. Permitir la preparación de datos para análisis posteriores (ej. Business Intelligence).
7. **Documentación y soporte técnico:**
   1. Documentar servicios REST mediante **Swagger/OpenAPI**, para facilitar integración futura con otros sistemas.
   2. Entregar manuales técnicos y de usuario como parte del producto final.

## Características de Usuario

El sistema Acured está diseñado para tres tipos principales de usuarios, cada uno con diferentes perfiles y necesidades:

1. Pacientes
   1. Perfil: Personas que solicitan atención en medicina oriental (acupuntura u otros tratamientos).
   2. Funciones principales en el sistema:
      1. Registro inicial de sus datos personales.
      2. Solicitud, confirmación y cancelación de horas.
      3. Consulta de su historial de citas y tratamientos registrados.
   3. Nivel técnico esperado: Bajo. Se espera que el paciente utilice el sistema de manera intuitiva mediante un navegador web, por lo que la interfaz debe ser amigable, clara y accesible.
2. Terapeutas de medicina oriental
   1. Perfil**:** Profesionales responsables de realizar los diagnósticos y registrar los tratamientos.
   2. Funciones principales en el sistema:
      1. Gestión de fichas clínicas digitales de sus pacientes.
      2. Registro de tratamientos aplicados en cada sesión.
      3. Administración de disponibilidad horaria y gestión de citas.
      4. Generación de reportes clínicos básicos.
   3. Nivel técnico esperado: Medio. Se requiere conocimiento básico de herramientas digitales, navegación web y manejo de sistemas de gestión clínica.
3. Administrador académico / Supervisor
   1. Perfil: Actor responsable del seguimiento del proyecto en el marco académico (docente o supervisor designado).
   2. Funciones principales en el sistema:
      1. Monitoreo de la correcta utilización del sistema.
      2. Validación del cumplimiento de requisitos técnicos y funcionales.
      3. Acceso a reportes de uso y pruebas para fines de evaluación académica.
   3. Nivel técnico esperado: Medio – alto. Familiaridad con conceptos técnicos y capacidad de revisar documentación, pruebas y reportes generados por el sistema.

## Restricciones Generales

El desarrollo del sistemaAcured está sujeto a las siguientes restricciones:

1. Restricciones temporales y académicas
   1. El proyecto debe completarse dentro del período definido por la asignatura Capstone, ajustándose al cronograma de avance establecido.
   2. El alcance funcional se limita a los módulos de gestión clínica y agendamiento definidos en el Acta Kick Off, excluyendo funcionalidades adicionales como facturación o interoperabilidad hospitalaria.
2. Restricciones técnicas
   1. El sistema se desarrollará bajo un stack tecnológico previamente definido: React (frontend), Spring Boot (backend), yPostgreSQL (base de datos).
   2. El sistema será accesible únicamente desde navegadores web modernos, no contemplando aplicaciones móviles nativas en esta versión.
   3. El despliegue inicial se realizará en entornos de prueba o servidores locales, sin disponibilidad garantizada en entornos de producción empresarial.
   4. Se utilizarán exclusivamente herramientas de software libre o gratuitas (ej. GitHub para control de versiones, Swagger/OpenAPI para documentación de APIs, Figma para diseño UI/UX).
3. Restricciones de seguridad y datos
   1. Los datos utilizados durante el desarrollo y pruebas serán ficticios o simulados, sin incluir información clínica real de pacientes.
   2. El acceso al sistema estará restringido a usuarios autenticados, con control de roles (paciente, terapeuta, administrador académico).
   3. No se contempla la implementación de mecanismos avanzados de cifrado de datos (ej. SSL en producción), dado que el proyecto se limita a un entorno académico.
4. Restricciones organizacionales
   1. El desarrollo se enmarca en el contexto de práctica profesional en Triskeledu SpA, aplicando metodologías y estándares de documentación acordados (IEEE 830, UML, BPMN, ERD).
   2. La gestión del proyecto se regirá por la metodología **en cascada**, definida en el Acta Kick Off, sin aplicar marcos ágiles como Scrum o Kanban en esta etapa.

# Requisitos Específicos

## Requisitos Funcionales

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Número de Requerimiento | Nombre del Requerimiento | Tipo Requerimiento | Actores Relacionados | Descripción corta del requisito | Criterio de Aceptación | Estado |
| RF-01 | Registro de Pacientes | Funcional | Terapeuta | El sistema permitirá registrar, consultar, actualizar y eliminar datos de paciente | Se podrá crear un paciente, consultarlo en el listado, actualizarlo y eliminarlo con éxito. | Aprobado |
| RF-02 | Gestión de Fichas Clínicas | Funcional | Terapeuta | El sistema permitirá crear fichas clínicas digitales vinculadas a cada paciente. | Una ficha clínica debe poder crearse, almacenarse y visualizarse correctamente. | Solicitado |
| RF-03 | Registro de Tratamientos | Funcional | Terapeuta | El terapeuta podrá registrar los tratamientos aplicados en cada sesión. | Al guardar un tratamiento, debe aparecer en el historial del paciente. | Solicitado |
| RF-04 | Historial Clínico | Funcional | Terapeuta, Paciente | El sistema permitirá consultar la evolución clínica de un paciente. | El historial deberá mostrar de manera ordenada todas las fichas registradas. | Aprobado |
| RF-05 | Agendamiento de Citas | Funcional | Paciente, Terapeuta | El paciente podrá reservar, modificar o cancelar citas en línea. | Una cita registrada debe aparecer en la agenda del terapeuta y notificarse al paciente. | Aprobado |
| RF-06 | Gestión de Agenda | Funcional | Terapeuta | El terapeuta podra administrar su disponibilidad horaria. | Los cambios en disponibilidad deben reflejarse inmediatamente en la agenda. | Solicitado |
| RF-07 | Notificaciones Básicas | Funcional | Paciente, Terapeuta | El sistema enviara confirmaciones de citas y avisos de modificaciones | El paciente debe recibir una notificación cada vez que se cambie el estado de su cita. | Diferido |
| RF-08 | Autenticación de Usuarios | Funcional | Paciente, Terapeuta, Administrador | El sistema deberá autenticar usuarios con credenciales. | Solo usuarios con credenciales válidas podrán acceder al sistema. | Aprobado |
| RF-09 | Control de Accesos por Rol | Funcional | Administrador, Terapeuta, Paciente | El sistema restringirá funciones en base al rol del usuario. | Cada rol verá únicamente las opciones que le corresponden. | Aprobado |
| RF-10 | Reportes Clínicos y Administrativos | Funcional | Terapeuta, Administrador | El sistema permitirá generar y exportar reportes en PDF y CSV | Los reportes exportados deben contener la información registrada sin pérdida de datos. | Solicitado |

## Requisitos No Funcionales

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Número de requerimiento | Nombre del Requerimiento | Tipo de Requerimiento | Actores Relacionados | Descripción corta del requisito | Criterios de Aceptación | Estado |
| RNF-01 | Tiempo de Respuesta del Sistema | No Funcional | Todos los Usuarios | El sistema debe responder en menos de 2 segundos para operaciones críticas (login, registro, consulta de ficha clínica). | Las páginas principales (login, registro, agenda) deben cargar en ≤ 2 segundos en pruebas de laboratorio. | En Desarrollo |
| RNF-02 | Escalabilidad | No Funcional | Administrador | El sistema debe ser capaz de soportar un aumento del 50% en usuarios sin afectar el rendimiento. | Pruebas de carga deberán demostrar que el sistema sigue respondiendo en ≤ 2 segundos con un 50% más de usuarios concurrentes. | Pendiente |
| RNF-03 | Seguridad de Datos | No Funcional | Paciente, terapeuta, administrador | El sistema debe garantizar la integridad y confidencialidad de los datos clínicos. | Solo usuarios autenticados acceden a datos, con validaciones de entrada y restricciones por rol. | Aprobado |
| RNF-04 | Privacidad de Información | No Funcional | Paciente, Administrador | El sistema debe cumplir con normativas locales de protección de datos y permitir a los pacientes gestionar su información personal. | El paciente podrá acceder, modificar o solicitar la eliminación de sus datos. | Solicitado |
| RNF-05 | Fiabilidad del Sistema | No Funcional | Todos los Usuarios | El sistema debe funcionar sin fallos críticos durante periodos de uso prolongado. | El sistema deberá alcanzar al menos un 99% de disponibilidad durante pruebas de estabilidad. | En Evaluación |
| RNF-06 | Disponibilidad | No Funcional | Todos los Usuarios | El sistema debe estar disponible 24/7 en el entorno académico de pruebas. | El sistema deberá estar operativo de manera continua durante un mes de evaluación. | Pendiente |
| RNF-07 | Mantenibilidad | No Funcional | Administrador, Equipo de Desarrollo | El código debe ser modular, documentado y versionado en GitHub, siguiendo estándares de la industria. | Se validará la existencia de documentación técnica, versionamiento y cumplimiento de estándares de codificación. | Aprobado |
| RNF-08 | Portabilidad | No Funcional | Paciente, Terapeuta | El sistema debe ejecutarse correctamente en los principales navegadores (Chrome, Firefox, Edge). | Pruebas de compatibilidad deberán mostrar que todas las funciones operan en dichos navegadores. | En Progreso |
| RNF-09 | Documentación | No Funcional | Administrador, Equipo de Desarrollo | El sistema deberá contar con documentación técnica (IEEE 830, UML, ERD, BPMN) y manuales de usuario. | Entregables deberán incluir manual de instalación, manual de usuario y documentación técnica. | Solicitado |
| RNF-10 | Pruebas de Seguridad | No Funcional | Administrador | El sistema deberá someterse a pruebas periódicas para identificar vulnerabilidades. | Deben registrarse y corregirse vulnerabilidades encontradas en pruebas internas. | Diferido |

## Otros Requisitos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Número de Requerimiento | Nombre del Requerimiento | Tipo de Requerimiento | Actores Relacionados | Descripción corta del requisito | Criterios de Aceptación | Estado |
| OR-01 | Cumplimiento Normativo | Otro | Administrador, Equipo de Desarrollo | El sistema y su documentación deberán elaborarse siguiendo el estándar IEEE 830 y normativas académicas de Duoc UC. | La entrega final incluirá un ERS estructurado según IEEE 830 y revisado por el docente supervisor. | Aprobado |
| |  | | --- | | OR-02 | |  | | Documentación Técnica y de Usuario | Otro | Administrador, Equipo de Desarrollo, Terapeutas | Se deberán entregar manuales de usuario, manual de instalación y documentación técnica (UML, ERD, BPMN, APIs REST con Swagger). | Los manuales y diagramas deberán estar disponibles junto al entregable final en formato PDF. | En Progreso |
| OR-03 | Plan de Pruebas | Otro | Administrador, Equipo de desarrollo | El sistema deberá incluir un plan de pruebas que cubra casos unitarios, integración, validación de formularios y APIs REST. | La ejecución de las pruebas deberá estar documentada y ser verificable por el docente supervisor. | |  | | --- | | Pendiente | |  | |
| OR-04 | Plan de Implantación Académico | Otro | Administrador, Docente Supervisor | El sistema deberá contemplar un plan de despliegue en entorno académico (servidor local o nube de prueba), indicando pasos de instalación. | El plan de instalación deberá ejecutarse con éxito en el entorno de práctica designado. | Solicitado |
| OR-05 | Restricción de Datos Reales | Otro | Administrador, Pacientes, Terapeutas | Durante el desarrollo y pruebas, no se manejarán datos clínicos reales, solo casos simulados, para resguardar la privacidad. | Verificación de que las bases de datos usadas contienen información ficticia y no sensible. | Aprobado |

# Interfaces gráficas

## Interfaces de Usuario

Las interfaces de usuario del sistema **Acured** fueron diseñadas en React, siguiendo principios de usabilidad, accesibilidad y responsividad. A continuación, se describen las principales pantallas del sistema, su layout, interacción esperada y lineamientos visuales.

### Pantalla de Inicio de Sesión

* **Diseño y Layout:**
  + Formulario centrado con campos: correo electrónico, contraseña y botón *Iniciar Sesión*.
  + Enlace inferior “¿Olvidaste tu contraseña?”.
* **Interacción del Usuario:**
  + El usuario ingresa credenciales y presiona *Iniciar Sesión*.
  + En caso de error, se muestra mensaje en rojo sobre el formulario.
* **Estilos y Temas:**
  + Fondo claro (#f8f9fa), formulario con bordes redondeados y botón principal en azul (#007bff).
* **Adaptabilidad y Responsividad:**
  + En móviles, el formulario ocupa el 90% del ancho de pantalla.
* **Accesibilidad:**
  + Campos con etiquetas <label> asociadas, soporte de navegación con teclado.
* **Mensajes y Notificaciones:**
  + Errores de autenticación en rojo.
  + Confirmación de cierre de sesión en verde.

### Dashboard del Terapeuta

* **Diseño y Layout:**
  + Barra lateral izquierda con accesos: *Pacientes, Agenda, Reportes*.
  + Sección principal con calendario de citas y listado de pacientes recientes.
* **Interacción del Usuario:**
  + Al seleccionar una cita en el calendario, se abre la ficha clínica correspondiente.
  + El terapeuta puede cambiar entre vistas de *semana* y *mes*.
* **Estilos y Temas:**
  + Colores neutros con acentos en azul (#007bff).
  + Tipografía: *Roboto* tamaño 14 px.
* **Adaptabilidad y Responsividad:**
  + En móviles, la barra lateral se convierte en menú desplegable.
* **Accesibilidad:**
  + Iconografía con descripciones textuales para usuarios de lectores de pantalla.
* **Mensajes y Notificaciones:**
  + Notificación emergente en verde al registrar un tratamiento exitoso.

### Módulo de Gestión de Pacientes

* **Diseño y Layout:**
  + Tabla con listado de pacientes, buscador en la parte superior y botón *Agregar Paciente*.
  + Formulario modal para ingreso/edición de datos.
* **Interacción del Usuario:**
  + El terapeuta puede buscar por nombre o RUT.
  + Los botones *Editar* y *Eliminar* aparecen en cada fila.
* **Estilos y Temas:**
  + Tabla con filas alternadas en gris claro (#f1f1f1).
* **Adaptabilidad y Responsividad:**
  + En móviles, la tabla se convierte en lista de tarjetas (cards).
* **Accesibilidad:**
  + Formularios con validación de campos obligatorios y mensajes en lenguaje claro.
* **Mensajes y Notificaciones:**
  + Confirmación al crear o modificar un paciente.
  + Advertencia al eliminar un registro.

### Agenda del Paciente

* **Diseño y Layout:**
  + Vista de calendario con citas disponibles.
  + Botón *Reservar Cita* junto a cada disponibilidad.
* **Interacción del Usuario:**
  + El paciente selecciona fecha y hora, confirma, y recibe un aviso de confirmación.
* **Estilos y Temas:**
  + Uso de tarjetas para mostrar citas con colores diferenciados: azul (disponible), gris (ocupado).
* **Adaptabilidad y Responsividad:**
  + En dispositivos pequeños, el calendario se simplifica a lista de horarios.
* **Accesibilidad:**
  + Textos claros y botones grandes para pacientes con dificultades visuales.
* **Mensajes y Notificaciones:**
  + Mensaje emergente en verde tras confirmar la reserva.
  + Notificación en rojo si el horario ya no está disponible.

### Reportes y Exportación

* **Diseño y Layout:**
  + Panel con filtros (por paciente, por rango de fechas, por terapeuta).
  + Botón *Generar Reporte* con opción de exportar en PDF o CSV.
* **Interacción del Usuario:**
  + Al generar un reporte, se abre vista previa antes de exportar.
* **Estilos y Temas:**
  + Botón de exportación en verde (#28a745).
* **Adaptabilidad y Responsividad:**
  + Los filtros se convierten en menú desplegable en pantallas pequeñas.
* **Accesibilidad:**
  + Opción de descargar reportes con nombres descriptivos para fácil identificación.
* **Mensajes y Notificaciones:**
  + Confirmación de descarga exitosa.
  + Mensaje de error si no hay datos disponibles para el rango elegido.

## Interfaces de Hardware

El sistema Acured está diseñado para funcionar en entornos de hardware estándar, accesibles en un contexto académico y clínico, sin necesidad de dispositivos especializados. A continuación, se detallan las interfaces de hardware con las que el sistema debe ser compatible:

### Listado de Hardware Compatible

* **Computadores de Escritorio y Portátiles**
  + Procesador Intel i5 o superior / AMD Ryzen 5 o superior.
  + Memoria RAM mínima: 8 GB (recomendado 16 GB).
  + Almacenamiento: 256 GB SSD (mínimo).
  + Tarjeta de red con soporte Ethernet y Wi-Fi (802.11ac o superior).
* **Dispositivos Móviles y Tablets**
  + Android versión 10 o superior.
  + iOS versión 13 o superior.
  + Resolución mínima soportada: 1280 x 720 px.
* **Servidores de Pruebas/Académicos**
  + Procesador Intel Xeon o equivalente.
  + RAM: mínimo 16 GB.
  + Almacenamiento: 500 GB SSD.
  + Sistema Operativo: Linux Ubuntu Server 22.04 LTS.

### Especificaciones Técnicas

* **Conectividad:**
  + Puerto de red Ethernet (10/100/1000 Mbps).
  + Wi-Fi 802.11ac para uso en dispositivos móviles.
* **Pantallas:**
  + Monitores Full HD (1920x1080 px) recomendados para estaciones de trabajo.
  + Compatibilidad con pantallas táctiles en dispositivos móviles.

### Protocolos de Comunicación

* **HTTP/HTTPS** = Para la interacción entre frontend (React) y backend (Spring Boot).
* **REST API (JSON sobre HTTPS)** = Para la transmisión de datos entre servicios.
* **TCP/IP** = Para toda la comunicación cliente-servidor en red local o internet.

### Requisitos de Datos y Formatos

* **JSON** = Formato de intercambio de datos entre cliente y servidor.
* **CSV/PDF** = Formato de exportación de reportes.
* **UTF-8** = Codificación estándar para almacenamiento y transmisión de texto.

### Descripción de la Interfaz

El sistema interactúa con el hardware de la siguiente manera:

1. El usuario accede al sistema desde un navegador web en PC, notebook, tablet o smartphone.
2. El navegador se comunica con el servidor de aplicaciones mediante **HTTPS**.
3. El servidor procesa la solicitud y responde en formato **JSON**.
4. En caso de exportar reportes, el servidor genera archivos **PDF o CSV**, los cuales se envían al dispositivo cliente para su descarga.
5. El hardware del cliente (pantalla, teclado, mouse o pantalla táctil) permite la interacción del usuario con el sistema.

## Interfaces de Software

El sistema Acured se desarrollará bajo una arquitectura **web** utilizando React (frontend), Spring Boot (backend) y PostgreSQL (base de datos). Para garantizar su interoperabilidad, se establecen las siguientes interfaces de software:

### Descripción de las Interfaces

* **Frontend – Backend**
  + Interfaz entre React y Spring Boot mediante **APIs REST**.
  + Flujo de datos en formato **JSON**.
* **Backend – Base de Datos**
  + Conexión entre Spring Boot y PostgreSQL mediante **JDBC (Java Database Connectivity)**.
  + Consultas SQL parametrizadas para garantizar seguridad y rendimiento.
* **API de Documentación**
  + Generación automática de documentación de endpoints utilizando **Swagger/OpenAPI**.
* **Servicios de Exportación**
  + Generación de reportes en **PDF y CSV** mediante librerías estándar de Java.
* **Control de Versiones**
  + Repositorio en **GitHub** para integración del equipo y manejo de cambios.

### Especificaciones Técnicas

* **APIs REST:**
  + Método: HTTP/HTTPS.
  + Versionado: v1.0 inicial.
  + Formato de datos: JSON (UTF-8).
* **Base de Datos:**
  + PostgreSQL v15.
  + Lenguaje SQL estandarizado.
* **Documentación:**
  + Swagger UI para visualización interactiva.
* **Exportación de Datos:**
  + Reportes PDF: librería iText.
  + Reportes CSV: librería Apache Commons CSV.

### Requerimientos de Autenticación y Seguridad

* Autenticación de usuarios mediante **JWT (JSON Web Tokens)**.
* Encriptación de contraseñas utilizando **bcrypt**.
* Todas las solicitudes a las APIs deben realizarse bajo **HTTPS** para evitar ataques de interceptación.
* Control de roles y permisos a nivel de backend (paciente, terapeuta, administrador).

### Gestión de Errores y Excepciones

* Los errores de comunicación entre frontend y backend retornarán códigos de estado HTTP:
  + 200 éxito.
  + 400 solicitud incorrecta.
  + 401 no autorizado.
  + 500 error interno del servidor.
* En caso de errores con la base de datos, se implementará un sistema de **reintentos automáticos** hasta 3 veces antes de registrar la falla.
* Todos los errores se registrarán en logs centralizados mediante **Spring Boot Actuator**.

### Dependencias y Limitaciones

* Dependencia de la conexión a internet para el funcionamiento del sistema web.
* Dependencia de disponibilidad de **PostgreSQL** en el servidor.
* Limitación: en esta versión no se contempla integración con sistemas externos de salud (HL7, FHIR).
* El sistema está limitado a entornos académicos de pruebas, no garantizando soporte en producción empresarial.

## Interfaces de Comunicación

El sistema Acured interactúa con sus componentes internos y externos a través de interfaces de comunicación basadas en protocolos estándar de la web, asegurando interoperabilidad, seguridad y eficiencia en el intercambio de información.

### Tipos de Comunicación

* **Comunicación síncrona:**
  + El frontend (React) se comunica con el backend (Spring Boot) mediante solicitudes HTTP/HTTPS en tiempo real.
* **Comunicación asíncrona:**
  + Procesos de exportación de reportes (PDF/CSV) que se ejecutan en segundo plano.
* **Comunicación cliente-servidor:**
  + El sistema sigue un modelo clásico donde el cliente solicita información y el servidor responde de forma inmediata.

### Protocolos de Comunicación

* **HTTP/1.1 y HTTPS (TLS 1.2 o superior):** para solicitudes y respuestas entre cliente y servidor.
* **TCP/IP:** protocolo base de toda comunicación en red.
* **JDBC:** para la interacción entre el backend (Spring Boot) y la base de datos (PostgreSQL).

### Formatos de Datos

* **JSON:** formato estándar para intercambio de información entre frontend y backend.
* **CSV/PDF:** formatos de exportación de reportes administrativos y clínicos.
* **UTF-8:** codificación utilizada en todo el sistema para asegurar compatibilidad internacional.

### Interfaces y Servicios Externos

En esta primera versión del sistema no se contempla integración con sistemas hospitalarios externos (HL7, FHIR). Sin embargo, el sistema sí interactúa con:

* **Swagger/OpenAPI:** para documentar y consumir los endpoints REST.
* **Repositorio GitHub:** como sistema de integración continua y control de versiones.

### Requerimientos de Ancho de Banda y Latencia

* **Ancho de banda mínimo:** 1 Mbps por cliente conectado para asegurar la correcta visualización de interfaces y consultas a la base de datos.
* **Latencia esperada:** menor a 200 ms en consultas a la base de datos y menor a 2 segundos en la carga de vistas del frontend.

### Mecanismos de Seguridad y Autenticación

* **JWT (JSON Web Tokens):** para autenticación y autorización de usuarios en las APIs REST.
* **HTTPS obligatorio:** todas las comunicaciones deben realizarse bajo TLS 1.2 o superior.
* **Control de roles:** el backend restringe el acceso a funcionalidades según perfil de usuario (paciente, terapeuta, administrador).
* **Encriptación de contraseñas:** utilizando algoritmos seguros como bcrypt antes de almacenar en la base de datos.

# Arquitectura de Software

## Modelo en Capas

El sistema Acured se estructura siguiendo un modelo en capas que permite separar responsabilidades, mejorar la mantenibilidad y facilitar la escalabilidad. Esta organización asegura una clara división entre presentación, lógica de negocio, servicios, persistencia y almacenamiento de datos.

### Descripción de las Capas

* **Capa de Presentación (Frontend)**
  + Implementada en React, con diseño responsivo y accesible.
  + Gestiona la interacción con el usuario final (paciente, terapeuta y administrador).
  + Contiene formularios de autenticación, gestión de pacientes, agendamiento y reportes.
* **Capa de Negocio (Business Layer)**
  + Implementada en Spring Boot (Java).
  + Contiene la lógica de negocio: validación de datos, gestión de citas, control de fichas clínicas y reportes.
  + Aplica reglas de negocio definidas en los requisitos funcionales (ej. restricción de roles, disponibilidad de agenda).
* **Capa de Servicios (API REST)**
  + Define los puntos de integración entre la capa de presentación y la de negocio.
  + Gestionada mediante RESTful APIs documentadas con Swagger/OpenAPI.
  + Utiliza formatoJSON para el intercambio de datos.
* **Capa de Acceso a Datos (Persistence Layer)**
  + Se conecta a la base de datos relacional utilizando **JPA/Hibernate** a través de JDBC.
  + Gestiona las operaciones CRUD (crear, leer, actualizar y eliminar) sobre las entidades del sistema.
* **Capa de Datos (Database Layer)**
  + Almacenamiento en PostgreSQL v18.
  + Base de datos relacional estructurada con entidades: *Pacientes, Fichas Clínicas, Citas, Usuarios, Roles, Reportes*.

### Interacción Entre Capas

* La **Capa de Presentación** envía solicitudes a la Capa de Servicios mediante llamadas HTTP/HTTPS.
* La **Capa de Servicios** interpreta las solicitudes y las redirige a la Capa de Negocio.
* La **Capa de Negocio** procesa la lógica, solicita datos a la Capa de Acceso a Datos, y retorna la respuesta.
* La **Capa de Acceso a Datos** se comunica con laCapa de Datos (PostgreSQL) para persistir o consultar información.
* Finalmente, la respuesta fluye de regreso hasta la Capa de Presentación, que la muestra al usuario.

### Tecnologías y Frameworks

* **Presentación:** React, TailwindCSS para UI responsiva.
* **Negocio:** Spring Boot (Java 17), Maven para gestión de dependencias.
* **Servicios:** REST APIs con Swagger/OpenAPI.
* **Acceso a Datos:** Hibernate, JPA, JDBC.
* **Datos:** PostgreSQL v15, SQL estándar.

### Seguridad y Transacciones

* Autenticación y autorización mediante JWT.
* Encriptación de contraseñas con bcrypt.
* Todas las transacciones cumplen con principios **ACID** (Atomicidad, Consistencia, Aislamiento, Durabilidad).
* Conexión cifrada bajo HTTPS (TLS 1.2 o superior).

### Consideraciones de Performance

* Uso de caché en capa de negocio para consultas frecuentes (ej. disponibilidad de agenda).
* Optimización de consultas SQL para reducir latencia.
* Estrategia de lazy loading en Hibernate para evitar sobrecarga en la memoria.
* Escalabilidad horizontal mediante despliegue en contenedores (Docker opcional en etapas posteriores).

### 4.5.6 Diagrama de Arquitectura en Capas

## Comunicación por Medio de API Rest

El sistema **Acured** utiliza un conjunto de **APIs RESTful** para la comunicación entre su capa de presentación (React) y su capa de negocio (Spring Boot), garantizando la interoperabilidad, modularidad y seguridad en el intercambio de datos. Estas APIs también permiten la futura integración con sistemas externos de salud si fuese necesario.

### Descripción General de la API

La API REST de **Acured** gestiona los principales recursos del sistema:

* **Usuarios** (pacientes, terapeutas y administradores).
* **Pacientes** (datos clínicos y personales).
* **Fichas Clínicas** (diagnósticos, tratamientos y evolución).
* **Citas y Agenda** (reserva, modificación, cancelación).
* **Reportes** (generación y exportación en PDF/CSV).

Cada recurso está disponible a través de endpoints específicos, cumpliendo con los principios de **RESTful APIs**.

### Endpoints y Métodos HTTP

* **Usuarios**
  + POST /api/usuarios → Crear usuario.
  + GET /api/usuarios/{id} → Consultar usuario.
  + PUT /api/usuarios/{id} → Actualizar usuario.
  + DELETE /api/usuarios/{id} → Eliminar usuario.
* **Pacientes**
  + GET /api/pacientes → Listar pacientes.
  + POST /api/pacientes → Registrar paciente.
  + PUT /api/pacientes/{id} → Actualizar paciente.
  + DELETE /api/pacientes/{id} → Eliminar paciente.
* **Fichas Clínicas**
  + GET /api/fichas/{pacienteId} → Consultar historial clínico.
  + POST /api/fichas → Crear ficha clínica.
* **Citas y Agenda**
  + GET /api/citas → Listar citas.
  + POST /api/citas → Reservar cita.
  + PUT /api/citas/{id} → Modificar cita.
  + DELETE /api/citas/{id} → Cancelar cita.
* **Reportes**
  + GET /api/reportes → Generar reporte (filtros por fecha, paciente, terapeuta).
  + GET /api/reportes/export?tipo=pdf → Exportar reporte en PDF.
  + GET /api/reportes/export?tipo=csv → Exportar reporte en CSV.

Formatos de Mensajes

* Lo obligatorio es **especificar qué formato usa tu API (JSON, XML, PDF, CSV, etc.)**.
* El ejemplo en JSON que yo puse (con nombre, rut, etc.) **no es estrictamente obligatorio**, pero **sí es recomendable** porque deja claro cómo se verá una respuesta real.

### Parámetros y Cuerpos de Solicitud:

* Lo obligatorio es **indicar qué parámetros acepta cada endpoint**.
* Los ejemplos de JSON que incluí (para crear un usuario o una cita) son **opcionales**.
* Sin embargo, ayudan a que un lector que no sea técnico entienda qué datos se esperan y cómo viajan.

### Códigos de Estado HTTP

* **200 OK** → Operación exitosa.
* **201 Created** → Recurso creado con éxito.
* **400 Bad Request** → Datos inválidos en la solicitud.
* **401 Unauthorized** → Usuario no autenticado.
* **403 Forbidden** → Usuario sin permisos suficientes.
* **404 Not Found** → Recurso no encontrado.
* **500 Internal Server Error** → Error interno del servidor.

### Seguridad y Autenticación

* Todas las solicitudes deben realizarse mediante **HTTPS (TLS 1.2 o superior)**.
* Autenticación con **JWT (JSON Web Token)** enviado en el encabezado:

Authorization: Bearer <token>

* Control de roles: Paciente, Terapeuta y Administrador con accesos diferenciados.

### Limitaciones y Cuotas

* Para garantizar el rendimiento, cada usuario autenticado tendrá un límite de **1000 solicitudes por hora**.
* Los reportes en PDF/CSV se generarán bajo demanda, pudiendo requerir hasta 5 segundos de procesamiento según volumen de datos.

## Comunicación por Medio de Webservices SOAP (Simple Object Access Protocol)

El sistema **Acured** se encuentra diseñado bajo una arquitectura moderna basada en APIs RESTful, utilizando JSON como formato de intercambio de datos y protocolos de seguridad como HTTPS y JWT. Por lo tanto, no se contempla el uso deWebservices SOAP como mecanismo de comunicación principal dentro del sistema.

### Justificación

Si bien SOAP fue un protocolo ampliamente utilizado en sistemas legados para la integración empresarial, en el caso de Acured no resulta necesario debido a que:

* La interoperabilidad con servicios externos (como Webpay y otros futuros) se resuelve mediante RESTful APIs.
* SOAP implicaría una complejidad innecesaria considerando que la mayoría de los proveedores actuales operan con REST.
* La arquitectura definida prioriza la ligereza, escalabilidad y simplicidad propias de REST.

### Consideraciones Futuras

En caso de que en un escenario posterior el sistema deba integrarse con plataformas externas que aún operen bajo SOAP, se evaluará la creación de un **módulo adaptador** que consuma los WSDL correspondientes y garantice compatibilidad. Dicho módulo aplicará los estándares de seguridad establecidos por **WS-Security** (encriptación, firmas digitales y certificados X.509) para asegurar la confidencialidad e integridad de los mensajes.

# Requisitos de Software y Hardware

## Requisitos de Hardware

Para el correcto funcionamiento del sistema **Acured**, tanto en su entorno de desarrollo como en producción, se han definido los siguientes requisitos mínimos y recomendados de hardware:

### Procesador (CPU)

* **Mínimo:** Intel Core i5 de 4 núcleos o equivalente (2.5 GHz).
* **Recomendado:** Intel Core i7 de 6 núcleos o superior / AMD Ryzen 7 (3.0 GHz o más).

### Memoria (RAM)

* **Mínimo:** 8 GB.
* **Recomendado:** 16 GB para entornos de producción con múltiples usuarios concurrentes.

### Espacio de Almacenamiento

* **Mínimo:** 256 GB SSD para instalación del sistema y base de datos.
* **Recomendado:** 512 GB SSD o superior, con almacenamiento adicional para respaldos periódicos.

### Tarjeta Gráfica (GPU)

* No es requerida una GPU dedicada. Sin embargo, se recomienda **GPU integrada Intel UHD o AMD Radeon Vega** para mejorar el rendimiento visual en entornos de escritorio.

### Periféricos

* Monitor con resolución mínima de **Full HD (1920x1080)**.
* Teclado y mouse estándar.
* Impresora compatible con **PCL6 o PDF** para la emisión de reportes clínicos.

### Conectividad

* Puerto **Ethernet (1 Gbps)** para servidores.
* WiFi 802.11ac como mínimo para estaciones de trabajo.
* Conectividad segura mediante HTTPS para acceso remoto.

### Sistemas Operativos Compatibles

* **Servidores:** Linux (Ubuntu Server 20.04 o superior) y Windows Server 2019 o superior.
* **Estaciones de trabajo:** Windows 10/11 Pro, macOS Catalina o superior.

## Requisitos de Software

Para garantizar el correcto funcionamiento del sistema **Acured**, se establecen los siguientes requisitos de software mínimos y recomendados. Estos abarcan sistemas operativos compatibles, requerimientos de procesamiento, memoria, almacenamiento, dependencias de terceros y compatibilidad con hardware definido en el punto 6.1.

### Compatibilidad de Sistemas Operativos

* **Servidores:**
  + Linux: Ubuntu Server 20.04 LTS o superior.
  + Windows Server 2019 o superior.
* **Estaciones de Trabajo / Desarrollo:**
  + Windows 10/11 Pro.
  + macOS Catalina 10.15 o superior.

### Requerimientos de Procesamiento (CPU)

* **Mínimo:** Procesador Intel Core i5 o AMD Ryzen 5, 4 núcleos, velocidad ≥ 2.0 GHz.
* **Recomendado:** Procesador Intel Xeon o AMD EPYC, 6 núcleos o más, velocidad ≥ 2.5 GHz, para entornos de producción con múltiples usuarios concurrentes.

### Requisitos de Memoria RAM

* **Mínimo:** 8 GB para servidores en pruebas y entornos de desarrollo.
* **Recomendado:** 16 GB o superior para entornos de producción, considerando la ejecución simultánea de servicios backend, frontend, base de datos y pasarelas de pago.

### Espacio de Almacenamiento Necesario

* **Mínimo:** 20 GB disponibles en disco SSD para instalación del sistema y datos iniciales.
* **Recomendado:** 100 GB en SSD de alto rendimiento, con almacenamiento adicional dedicado para respaldos de fichas clínicas, reportes y logs históricos.

### Requisitos de Hardware Específico

* No se requiere GPU dedicada.
* Tarjeta de red con conectividad mínima de **1 Gbps Ethernet** o WiFi 802.11ac para estaciones de trabajo.
* Periféricos básicos: teclado, mouse y monitor con resolución mínima **Full HD (1920x1080)**.

### Dependencias de Software de Terceros

* **Java JDK 17** =9- requerido para la ejecución del backend en **Spring Boot**.
* **Node.js 18+ y NPM** = requeridos para la construcción y ejecución del frontend en **React**.
* **PostgreSQL 14 o superior** = gestor de base de datos principal.
* **SQL Server 2019 o superior** = para integraciones mediante procedimientos almacenados.
* **Swagger / OpenAPI** = documentación de servicios RESTful.
* **Docker (opcional)** = para despliegue en contenedores.
* **Servidor de Aplicaciones Tomcat embebido** = incluido en Spring Boot.

# Planificación

## Definición del Equipo de Trabajo

El desarrollo del sistema **Acured** será llevado a cabo por el equipo de la empresa **Creadora Isidora Gómez**, cuya estructura y funciones han sido definidas para garantizar el cumplimiento de los objetivos del proyecto.

### Roles y Responsabilidades

* **Gerente Comercial**
  + Responsable de la relación con los clientes y stakeholders externos.
  + Define los lineamientos estratégicos del proyecto desde la perspectiva del negocio.
  + Supervisa el cumplimiento de objetivos comerciales y la viabilidad del producto.
* **Gerente en Informática**
  + Responsable de la supervisión tecnológica global.
  + Define la arquitectura tecnológica del sistema y la selección de herramientas de desarrollo.
  + Coordina con el Jefe de Proyecto y los programadores para garantizar la calidad técnica.
* **Jefe de Proyecto**
  + Planifica, organiza y controla el avance del proyecto.
  + Gestiona riesgos, cronograma y recursos.
  + Es el nexo principal entre el equipo técnico y la gerencia.
* **Programadores Frontend (2)**
  + Desarrollan la interfaz de usuario en React, asegurando la usabilidad, responsividad y accesibilidad.
  + Implementan la comunicación con la API REST y las funcionalidades visuales.
  + Corrigen errores de interfaz y optimizan la experiencia del usuario final.
* **Programadores Backend (2)**
  + Implementan la lógica de negocio en **Spring Boot**.
  + Gestionan la base de datos principal en **PostgreSQL** y las integraciones con **SQL Server**.
  + Desarrollan los servicios REST, asegurando la seguridad y rendimiento del sistema.

### Estructura del Equipo

La estructura de trabajo seguirá un modelo jerárquico con apoyo matricial:

* El **Gerente Comercial** supervisa la alineación del proyecto con los objetivos del negocio.
* El **Gerente en Informática** asegura la calidad técnica y toma decisiones sobre arquitectura.
* El **Jefe de Proyecto** coordina las actividades diarias, controlando avances y entregables.
* Los **programadores (frontend y backend)** trabajan en conjunto bajo la dirección del Jefe de Proyecto y con lineamientos técnicos del Gerente en Informática.

### Herramientas y Métodos de Comunicación

* **Comunicación diaria**: Slack / Microsoft Teams.
* **Gestión de proyectos**: JIRA para tareas e incidencias.
* **Control de versiones**: GitHub.
* **Reuniones formales**: Semanales vía Google Meet para revisión de hitos.

### Plan de Desarrollo y Capacitación

El equipo contará con instancias de actualización en:

* Buenas prácticas en **desarrollo ágil**.
* Capacitación en **seguridad de aplicaciones web** y normativas aplicables en salud.
* Talleres de actualización en **React y Spring Boot**.

### Estrategias de Resolución de Conflictos

Los conflictos internos serán resueltos en primera instancia por el **Jefe de Proyecto**.  
 En caso de no encontrar solución, se escalarán al **Gerente en Informática** o al **Gerente Comercial**, según corresponda al ámbito del problema (técnico o estratégico).

### Criterios de Éxito del Equipo

* Cumplimiento de plazos establecidos en el cronograma.
* Entregables alineados a los requerimientos funcionales y no funcionales.
* Alta calidad técnica del software (mínima cantidad de errores críticos).
* Satisfacción del cliente final y de los usuarios durante la validación del sistema.

## Definición de Actividades principales del Proyecto

El desarrollo del sistema Acured se organizará en actividades principales, distribuidas en fases del ciclo de vida del desarrollo de software. Cada actividad está definida con su objetivo, recursos asignados, dependencias y criterios de éxito, asegurando una planificación clara y un control adecuado del proyecto.

### Lista y Descripción de Actividades

1. **Análisis de Requisitos**
   1. **Descripción:** Recolección, documentación y validación de requisitos funcionales y no funcionales en conjunto con stakeholders.
   2. **Recursos asignados:** Jefe de Proyecto, Analista Funcional, Gerente Comercial.
   3. **Dependencias:** Actividad inicial del proyecto.
   4. **Criterio de éxito:** Documento ERS aprobado por el cliente y equipo técnico.
2. **Diseño de Arquitectura del Sistema**
   1. **Descripción:** Definición de la arquitectura en capas, diagramas UML, modelos de datos y selección de tecnologías (React, Spring Boot, PostgreSQL).
   2. **Recursos asignados:** Gerente en Informática, Líder Técnico, Programadores.
   3. **Dependencias:** Finalización del análisis de requisitos.
   4. **Criterio de éxito:** Diagramas y modelos validados y aceptados por el equipo.
3. **Diseño de Interfaces de Usuario (UI/UX)**
   1. **Descripción:** Creación de la estructura de pantallas, flujos de interacción, lineamientos de usabilidad y accesibilidad.
   2. **Recursos asignados:** Programadores Frontend, Jefe de Proyecto.
   3. **Dependencias:** Análisis de requisitos.
   4. **Criterio de éxito:** Prototipos y guías de estilo validadas por usuarios finales.
4. **Implementación de Backend**
   1. **Descripción:** Desarrollo de la lógica de negocio en **Spring Boot**, APIs RESTful, integración con Webpay y procedimientos almacenados en SQL Server.
   2. **Recursos asignados:** Programadores Backend, Gerente en Informática.
   3. **Dependencias:** Diseño de arquitectura y base de datos.
   4. **Criterio de éxito:** Módulos backend probados en entorno de desarrollo.
5. **Implementación de Frontend**
   1. **Descripción:** Desarrollo de las interfaces de usuario en **React**, comunicación con API REST y validaciones de usabilidad.
   2. **Recursos asignados:** Programadores Frontend.
   3. **Dependencias:** Diseño de interfaces de usuario y APIs disponibles.
   4. **Criterio de éxito:** Interfaces operativas que reflejen los flujos definidos.
6. **Pruebas Unitarias y de Integración**
   1. **Descripción:** Verificación de cada módulo en forma aislada y luego en conjunto, asegurando que el sistema opere de manera integrada.
   2. **Recursos asignados:** QA Tester, Programadores.
   3. **Dependencias:** Implementación de frontend y backend.
   4. **Criterio de éxito:** Superar pruebas con ≥95% de casos aprobados.
7. **Pruebas de Aceptación de Usuario (UAT)**
   1. **Descripción:** Validación por parte de los usuarios finales en un entorno de prueba controlado.
   2. **Recursos asignados:** Clientes, Jefe de Proyecto, QA Tester.
   3. **Dependencias:** Pruebas unitarias e integración completas.
   4. **Criterio de éxito:** Aprobación del sistema sin errores críticos reportados.
8. **Despliegue en Producción**
   1. **Descripción:** Instalación del sistema en el entorno de producción, configuraciones finales y habilitación para uso real.
   2. **Recursos asignados:** Especialista DevOps, Gerente en Informática.
   3. **Dependencias:** Pruebas de aceptación aprobadas.
   4. **Criterio de éxito:** Sistema operativo en producción sin errores críticos durante 48 horas.
9. **Capacitación y Transferencia de Conocimiento**
   1. **Descripción:** Entrenamiento a los usuarios y entrega de manuales técnicos y de usuario.
   2. **Recursos asignados:** Jefe de Proyecto, Programadores, QA Tester.
   3. **Dependencias:** Despliegue en producción.
   4. **Criterio de éxito:** Usuarios capacitados y documentación entregada.
10. **Mantenimiento Inicial y Soporte Post-Implementación**

* **Descripción:** Corrección de errores, optimizaciones y soporte técnico en los primeros meses de operación.
* **Recursos asignados:** Programadores, QA Tester, Jefe de Proyecto.
* **Dependencias:** Despliegue completado.
* **Criterio de éxito:** Reducción progresiva de incidencias reportadas y estabilidad del sistema.

# Conclusiones

El presente documento de Especificación de Requisitos de Software (ERS) ha permitido establecer de manera clara y estructurada los objetivos, requerimientos y lineamientos necesarios para el desarrollo del sistema Acured, orientado a mejorar la gestión clínica y administrativa de pacientes, terapeutas y administradores en un entorno digital centralizado.

Se definieron los objetivos principales del proyecto, los cuales se enfocan en:

* Digitalizar los procesos de registro y control de pacientes.
* Optimizar la programación y seguimiento de citas mediante un sistema ágil e intuitivo.
* Mejorar la seguridad y disponibilidad de la información mediante tecnologías modernas.
* Integrar funcionalidades de pago en línea a través de Webpay.

En cuanto a los requisitos críticos del sistema, se destacan:

* Requisitos funcionales que garantizan el correcto manejo de usuarios, citas, fichas clínicas y reportes.
* Requisitos no funcionales asociados a seguridad de datos (confidencialidad e integridad), rendimiento (tiempos de respuesta menores a 3 segundos) y disponibilidad (operación 24/7).
* Compatibilidad con infraestructuras modernas (Linux/Windows en servidores, React + Spring Boot en desarrollo, PostgreSQL y SQL Server en bases de datos).

Respecto al **alcance del proyecto**, se establece que Acured cubrirá los procesos esenciales de gestión clínica, administrativa y de pagos en línea, priorizando la interoperabilidad mediante APIs RESTful. Sin embargo, se reconoce como limitación que la integración con sistemas legados dependerá de las condiciones técnicas externas, pudiendo requerir adaptadores específicos en caso de ser necesario.

Finalmente, lospróximos pasos se orientan hacia:

* El diseño detallado de la arquitectura de software y sus componentes.
* La implementación iterativa del sistema en fases controladas.
* La validación constante con usuarios finales para garantizar que el producto satisfaga las necesidades reales.
* El despliegue en producción acompañado de un plan de capacitación y soporte post-implementación.

En conclusión, el documento sienta las bases necesarias para asegurar la correcta planificación, desarrollo e implementación del sistema **Acured**, alineando a todos los miembros del equipo y stakeholders en torno a los objetivos del proyecto y asegurando un marco de trabajo ordenado, escalable y sostenible.

# Referencias Bibliográficas

* Bass, L., Clements, P., & Kazman, R. (2013). *Software architecture in practice* (3rd ed.). Addison-Wesley Professional.
* Sommerville, I. (2011). *Ingeniería del software* (9ª ed.). Pearson Educación.
* Fowler, M. (2003). *Patterns of enterprise application architecture*. Addison-Wesley.
* Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill.
* Fielding, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Doctoral dissertation, University of California, Irvine).
* Oracle. (2025). *Spring Boot Documentation*. Recuperado de <https://spring.io/projects/spring-boot>
* Meta. (2025). *React Documentation*. Recuperado de <https://react.dev/>
* PostgreSQL Global Development Group. (2025). *PostgreSQL Documentation*. Recuperado de <https://www.postgresql.org/docs/>
* Transbank. (2025). *Webpay Plus REST API Documentation*. Recuperado de <https://transbankdevelopers.cl/>