Se busca desarrollar una herramienta de historia clínica integral que se pueda dividir en dos grandes módulos, el módulo administrativo y el módulo asistencial.

**MODULO ADMINISTRATIVO**

**Contratos:** En este módulo se crearán todos los contratos y tarifas que tenga la IPS, se crearan las EPS con las características de sus contratos, esta información estará ligada a la atención de los usuarios, en este módulo también se realizara la creación de los usuarios y los roles que desempeñaran en el sistema.

**Recursos humanos:** En este módulo pueden crear hojas de vida de los empleados, se tendrá la funcionalidad para la creación de cargos con sus respectivos sueldos.

**MODULO ASISTENCIAL**

**Agenda:** En esta funcionalidad se deben crear las agendas medicas para poder ser asignadas a los médicos, debe permitir la creación de los consultorios

**Creación de pacientes:** El paciente cuando llega al consultorio y solicita un servicio debe ser creado en el sistema, el sistema debe permitir listar los pacientes activos, editar información de los pacientes, pero no debe permitir la eliminación de estos

**Creación de eventos para historia clínica:** una vez creado el paciente se debe crear el evento sobre el cual van a realizársele todas las notas y registros, ya sea historia clínica, evoluciones medicas de observación u hospitalización, aplicación de medicamentos, descripciones quirúrgicas, y cualquier servicio que consuma durante su estancia, cada ingreso a la institución debe tener un evento diferente.

**Creación de historias clínica:** Esta funcionalidad permite el registro del formulario de historia clínica, consultar las historias que pueda tener un paciente en su histórico, editar registros de pacientes activos, pero no debe permitir la eliminación de registros.

**Creación de Evoluciones medicas para pacientes en observación u hospitalizados:** Esta funcionalidad permite que un paciente que está en observación o en hospitalización, se le puedan realizar sus respectivas notas medicas o de enfermería, se debe permitir listar las notas que tenga un evento y editar, pero no eliminar, como funcionalidad adicional en las evoluciones debe tener la facilidad de importar plantillas preestablecidas.

**Creación de descripciones de procedimientos:** Esta funcionalidad permite el registro de todos los procedimientos ya sean invasivos o no invasivos, de enfermería o del médico realizados al paciente, como funcionalidad adicional en la descripción del procedimiento debe tener la facilidad de importar plantillas preestablecidas.

**Creación de ordenes medicas:** cuando el paciente recibe su atención, esta puede generar ordenes medicas o de enfermería tanto internas que son las que se realizarían durante la permanencia del paciente en la IPS, como ordenes externas para que el paciente las ejecute externamente de la IPS.

**Creación de hoja de gastos**: Se busca crear un modulo que permita la creación de una hoja de gastos para que sea diligenciada por el personal a cargo, de esta forma se tendrá un control de todos los insumos y medicamentos consumidos en un procedimiento

**Creación de recepción técnica de medicamentos e insumos:** En este módulo el personal de la IPS realizara el ingreso al sistema de los medicamentos e insumos que se compren con su respectiva factura para posteriormente alimente el inventario.

**Creación de bodegas para manejo de mercancía:** El sistema tendrá la facilidad de crear las diferentes bodegas que necesite el cliente, para almacenar inventario de medicamentos o insumos para luego ser consumidos.

**Consentimientos informados:** en este módulo debe tener la funcionalidad de crear consentimientos informados, estos consentimientos deben ser adicionados al evento en el cual se le esta realizando el procedimiento a autorizar y debe permitir la captura del paciente o acudiente directamente en el sistema.

**Manejo de inventario:** En este módulo podremos administrar todo lo correspondiente al inventario de medicamentos e insumos que tenga la IPS.

**Módulo de reportes:** en este módulo se generarán todos los reportes reglamentarios por ley que da lugar en la atención de pacientes

El proyecto se desarrollará en 3 fases

Fase 1: Para salir a producción el sistema debe contar con los siguientes desarrollos:

Contratos

Agenda

Creación de pacientes

Reporte de paciente

Creación de historias clínica

Reporte de historia clinica

Creación de eventos para historia clínica

Reporte del los eventos

Creación de procedimientos

Creación de descripciones de procedimientos

Creación de ordenes medicas

Creación de los usr

Login

Fase 2: la Fase 2 tendrá la implementación de

Creación de Evoluciones medicas para pacientes en observación u hospitalizados:

Módulo de reportes

Consentimiento informado

Creación de recepción técnica de medicamentos e insumos

Creación de bodegas para manejo de mercancía

Creación de hojas de gastos

Manejo de inventario

Fase 3:

Recursos humanos

RECOMENDACIONES PASO A PASO:

Para implementar la Fase 1 del proyecto de la herramienta de historia clínica integral en Spring Boot, que incluye los siguientes desarrollos: Contratos, Agenda, Creación de pacientes, Creación de eventos para historia clínica, Creación de historias clínicas, Creación de descripciones de procedimientos y Creación de órdenes médicas, puedes seguir estos pasos generales:

Paso 1: Configuración del Entorno

Asegúrate de tener Java y Spring Boot configurados en tu entorno de desarrollo.

Paso 2: Crear la Base de Datos

Diseña y crea la base de datos que respalde tu aplicación de historia clínica. Deberás crear tablas para contratos, pacientes, eventos, historias clínicas, descripciones de procedimientos, órdenes médicas, etc., según los requerimientos del módulo.

Paso 3: Crear Modelos de Datos

Define las clases de modelo de datos de Java que representen las entidades en tu base de datos. Utiliza anotaciones de JPA para mapear estas clases a las tablas de la base de datos.

Paso 4: Desarrollar Controladores

Crea controladores de Spring Boot para cada uno de los módulos mencionados en la Fase 1. Los controladores se encargarán de manejar las solicitudes HTTP y llamarán a los servicios correspondientes.

Paso 5: Desarrollar Servicios

Implementa servicios que contengan la lógica de negocio de cada módulo. Por ejemplo, un servicio para crear pacientes, otro para crear eventos de historia clínica, etc.

Paso 6: Crear Repositorios

Define repositorios de Spring Data JPA para cada entidad. Estos repositorios simplificarán la interacción con la base de datos.

Paso 7: Implementar la Lógica de Negocio

Desarrolla la lógica de negocio para cada módulo, asegurándote de que cumpla con los requisitos del proyecto. Esto incluye la validación de datos, la gestión de eventos, la creación de historias clínicas y otros procesos definidos en los requerimientos.

Paso 8: Configurar la Seguridad (si es necesario)

Si la aplicación requiere autenticación y autorización, configura la seguridad de Spring Boot para proteger las rutas y los recursos.

Paso 9: Desarrollar Vistas (si es necesario)

Si es necesario, crea las vistas (páginas HTML o plantillas) para que los usuarios interactúen con la aplicación a través de un navegador.

Paso 10: Pruebas

Realiza pruebas exhaustivas de cada módulo para asegurarte de que funcionan según lo esperado. Esto incluye pruebas unitarias y pruebas de integración.

Paso 11: Documentación

Documenta tu código y la arquitectura de la aplicación para facilitar el mantenimiento y la comprensión del sistema por parte de otros desarrolladores.

Paso 12: Despliegue

Despliega la aplicación en el entorno de producción siguiendo las prácticas recomendadas para servidores Spring Boot.

Paso 13: Monitoreo y Mantenimiento

Implementa herramientas de monitoreo y realiza un seguimiento constante del sistema en producción para asegurarte de que funcione sin problemas y realice mantenimiento cuando sea necesario.

Recuerda que estos son pasos generales y que la implementación real puede variar según tus necesidades específicas y los detalles de tu proyecto. Además, es importante involucrar a otros miembros del equipo, como desarrolladores front-end, diseñadores de UI/UX, y profesionales de QA para asegurarte de que el proyecto se desarrolle de manera integral y cumpla con los requisitos del cliente.

MODELO DE HU PARA TRABAJAR  
  
**Nombre de la Historia de Usuario**: [Un nombre descriptivo y breve que identifica la funcionalidad]

**ID de la Historia de Usuario**: [Un identificador único, si lo utilizas]

**Descripción**:

[Una breve descripción de la funcionalidad que se está solicitando. Debe ser lo suficientemente clara para que el programador entienda el propósito de la historia de usuario]

**Criterios de Aceptación**:

[Una lista de condiciones que la funcionalidad debe cumplir para que se considere completa y aceptable. Cada criterio debe ser específico y medible.]

[Ejemplo: El usuario debe poder iniciar sesión con su dirección de correo electrónico y contraseña.]

[Ejemplo: Después del inicio de sesión exitoso, el usuario debe ver su panel de control con la información relevante.]

**Requisitos Técnicos**:

[Si hay requisitos técnicos específicos, como tecnologías, lenguajes de programación, bases de datos, o bibliotecas que deben utilizarse, menciónalos aquí.]

**Mockups o Diseños (si corresponde)**:

[Si tienes diseños o mockups de la interfaz de usuario, inclúyelos aquí para ayudar a visualizar la funcionalidad.]

**Notas Adicionales**:

[Cualquier otra información relevante o consideraciones especiales que el programador deba conocer.]

# RECOMENDACIONES GENERALES:

**Documenta tus ideas**: Comienza por escribir todo lo que tengas en mente. Haz un documento donde describas los objetivos, funcionalidades, requerimientos, y cualquier detalle importante del proyecto. Esto te ayudará a organizar tus pensamientos y aclarar tus ideas.

**Crea un esquema o diagrama de flujo**: Utiliza diagramas o esquemas para visualizar la estructura y el flujo del proyecto. Esto puede incluir diagramas de flujo, diagramas de arquitectura, diagramas de base de datos, entre otros. Estas representaciones visuales pueden ayudarte a entender mejor cómo se relacionan las partes del proyecto.

**Crea un plan detallado**: Desarrolla un plan detallado que incluya fechas límite, recursos necesarios y un cronograma. Esto te ayudará a tener una visión clara de cómo avanzar en el proyecto y a establecer metas realistas.

**Habla con otros**: Comparte tus ideas con colegas, amigos o mentores en el campo relacionado. Obtener retroalimentación y diferentes perspectivas puede ayudarte a refinar y mejorar tu proyecto.

**Investiga y aprende**: Si hay aspectos técnicos o tecnológicos que no dominas, dedica tiempo a investigar y aprender. Puedes encontrar recursos en línea, tutoriales y cursos que te ayudarán a adquirir las habilidades necesarias.

**Prototipa y valida**: Crea prototipos o maquetas de tu proyecto para tener una representación visual y funcional. Luego, muestra estos prototipos a usuarios reales o a personas de confianza para obtener retroalimentación y validar tus ideas.

**Divide y conquista**: En lugar de intentar construir todo el proyecto a la vez, divídelo en partes más pequeñas y manejables. Esto facilita el progreso gradual y te permite concentrarte en una parte a la vez.

**Mantén registros y documentación**: A medida que avances en el proyecto, mantén registros detallados de lo que has hecho. Esto te ayudará a rastrear el progreso y a tener documentación útil para futuras referencias.

**No temas hacer cambios**: A medida que trabajas en el proyecto, es posible que surjan nuevas ideas o que necesites ajustar el enfoque. No tengas miedo de hacer cambios en tu plan si crees que son necesarios para mejorar el resultado final.

Recuerda que plasmar un proyecto en algo tangible lleva tiempo y esfuerzo, pero con una planificación adecuada y pasos metodológicos, puedes convertir tus ideas en una realidad. ¡Persistencia y determinación son clave!

MODELO DE HU PARA TRABAJAR  
  
**Nombre de la Historia de Usuario**: Login por parte del back

**ID de la Historia de Usuario**: 01

**Descripción**:

Como PO del proyecto de plataforma de salud, necesito que se cree por aprte del backend un endpoint el cual me gestione el login a la plataforma

**Criterios de Aceptación**:

1. El endpoint debe recibir por método post en el body la siguiente estructura de datos:
   1. {“email”:”<email del usr>”,”pass”:”<pass sin encriptar>”}
2. Al recibir esta información se debe consultar el email en bd para corroborar que exista un usr con esa información
   1. Si la información no existe debe devolver la siguiente respuesta
      1. {“message”:”user not found”,”success”:false}
      2. El servidor debe devolver un status 500
   2. Si la información existe el app deberá encriptar el valor del campo pass y cotejarlo con su correspondiente valor en la bd
      1. Se debe crear un JWT donde contenga toda la información del usr anexando permisos e información del mismo exceptuando el pass
      2. Dicho JWT deberá tener un tiempo de vida correspondiente a 24hrs
      3. {“message”:”user authentic successfully”, “success”:true, “Data”:{“token”:”<token generado>”, “time”:86400}}
      4. El servidor debe devolver status 200
   3. El endpoint debe escuchar por la ruta-path < dominio >/api/login
   4. Si la respuesta es exitosa, se debe actualizar el campo en la tabla xx donde llevamos el histprial de los inicios de sesión

**Requisitos Técnicos**:

1. El backend debe estar construido en Sprintboot
2. La información siempre debe recibirse a través del body de la petición y su estructura siempre será un JSON

**Mockups o Diseños (si corresponde)**:

[Si tienes diseños o mockups de la interfaz de usuario, inclúyelos aquí para ayudar a visualizar la funcionalidad.]

**Notas Adicionales**:

[Cualquier otra información relevante o consideraciones especiales que el programador deba conocer.]

**Nombre de la Historia de Usuario**: Login por parte del back

**ID de la Historia de Usuario**: 01

**Descripción**:

Como PO del proyecto de plataforma de salud, necesito que se cree por aprte del backend un endpoint el cual me gestione el login a la plataforma

**Criterios de Aceptación**:

1. El endpoint debe recibir por método post en el body la siguiente estructura de datos:
   1. {“email”:”<email del usr>”,”pass”:”<pass sin encriptar>”}
2. Al recibir esta información se debe consultar el email en bd para corroborar que exista un usr con esa información
   1. Si la información no existe debe devolver la siguiente respuesta
      1. {“message”:”user not found”,”success”:false}
      2. El servidor debe devolver un status 500
   2. Si la información existe el app deberá encriptar el valor del campo pass y cotejarlo con su correspondiente valor en la bd
      1. Se debe crear un JWT donde contenga toda la información del usr anexando permisos e información del mismo exceptuando el pass
      2. Dicho JWT deberá tener un tiempo de vida correspondiente a 24hrs
      3. {“message”:”user authentic successfully”, “success”:true, “Data”:{“token”:”<token generado>”, “time”:86400}}
      4. El servidor debe devolver status 200
   3. El endpoint debe escuchar por la ruta-path < dominio >/api/login
   4. Si la respuesta es exitosa, se debe actualizar el campo en la tabla xx donde llevamos el histprial de los inicios de sesión

**Requisitos Técnicos**:

1. El backend debe estar construido en Sprintboot
2. La información siempre debe recibirse a través del body de la petición y su estructura siempre será un JSON

**Mockups o Diseños (si corresponde)**:

[Si tienes diseños o mockups de la interfaz de usuario, inclúyelos aquí para ayudar a visualizar la funcionalidad.]

**Notas Adicionales**:

[Cualquier otra información relevante o consideraciones especiales que el programador deba conocer.]