

GENOPOLIS: UNA PLATAFORMA WEB PARA LA BÚSQUEDA Y GESTIÓN DE PROTEÍNAS

LINA MARCELA ÁVILA MORENO

SERVICIO NACIONAL DE APRENDIZAJE
TECNÓLOGO EN ANÁLISIS Y DESARROLLO DE SOFTWARE
FASE 1. ANÁLISIS
BOGOTÁ D.C
2024

I. REQUERIMIENTOS DEL SISTEMA: ESPECIFICACIÓN DE REQUISITOS

De acuerdo con la especificación de requisitos según el estándar de IEEE830, se analizaron y revisaron los siguientes requisitos funcionales y no funcionales.

- Requisitos Funcionales

Área general de requisito: Autenticación y Registro de Usuarios

Requisito 1: La aplicación funcionara para usuarios registrados y no registrados.

Perspectiva del producto	La aplicación se diseñará para permitir el acceso a diferentes funcionalidades tanto a usuarios registrados como no registrados, ampliando su usabilidad y alcance.
Funciones del producto	<ul style="list-style-type: none">- Acceso limitado para usuarios no registrados (navegación y búsqueda básica).- Acceso completo a funcionalidades para usuarios registrados (búsquedas avanzadas, historial, favoritos).
Características de los usuarios	<ul style="list-style-type: none">- Usuarios registrados: Personas que han creado una cuenta, lo que les permite acceder a todas las funcionalidades.- Usuarios no registrados: Personas que pueden explorar la aplicación con funcionalidades limitadas, sin la necesidad de registrarse.
Restricciones	No aplica restricciones específicas para este requisito, pero la aplicación asegurará la protección de datos de usuarios registrados.
Requisitos funcionales	<p>Caso de uso 1: Acceso de usuarios no registrados</p> <ul style="list-style-type: none">- Actor: Usuario no registrado- Precondición: El usuario no está autenticado.- Flujo principal: El usuario accede a la aplicación y utiliza funcionalidades limitadas. <p>Caso de uso 2: Acceso de usuarios registrados</p> <ul style="list-style-type: none">- Actor: Usuario registrado- Precondición: El usuario está autenticado.- Flujo principal: El usuario inicia sesión y accede a todas las funcionalidades.
Requisitos no funcionales	<ul style="list-style-type: none">- La aplicación debe ser segura y manejar eficientemente la autenticación.- Debe garantizar un rendimiento aceptable incluso con múltiples usuarios.

Requisito 2: Los usuarios podrán iniciar sesión en la plataforma mediante un nombre de usuario y contraseña.

Perspectiva del producto	La aplicación debe permitir a los usuarios registrados iniciar sesión de forma segura utilizando su nombre de usuario y contraseña, asegurando el acceso a funcionalidades exclusivas.
Funciones del producto	<ul style="list-style-type: none"> - Proporcionar un formulario de inicio de sesión que solicite nombre de usuario y contraseña. - Validar las credenciales ingresadas y otorgar acceso si son correctas.
Características de los usuarios	- Usuarios registrados: Personas que han creado una cuenta con Genopolis y necesitan autenticarse para acceder a las funcionalidades completas.
Restricciones	Se debe implementar mecanismos de seguridad como encriptación de contraseñas y bloqueo de la cuenta tras múltiples intentos fallidos de inicio de sesión.
Requisitos funcionales	<p>Caso de uso: Inicio de sesión de usuarios</p> <ul style="list-style-type: none"> - Actor: Usuario registrado - Precondición: El usuario debe haber creado una cuenta previamente. - Flujo principal: El usuario accede a la pantalla de inicio de sesión. El usuario ingresa su nombre de usuario y contraseña. El sistema valida las credenciales. Si las credenciales son correctas, el sistema otorga acceso a la plataforma; si son incorrectas, se muestra un mensaje de error.
Requisitos no funcionales	<ul style="list-style-type: none"> - La autenticación debe ser rápida y eficiente, permitiendo que los usuarios inicien sesión sin demoras significativas. - Debe garantizar la seguridad de las credenciales mediante encriptación.

Requisito 3: Los usuarios podrán registrarse en el sistema proporcionando un nombre de usuario y definiendo una contraseña.

Perspectiva del producto	La aplicación debe permitir que nuevos usuarios se registren creando una cuenta mediante la introducción de un nombre de usuario único y una contraseña segura, facilitando así el acceso a las funcionalidades de la plataforma.
Funciones del producto	<ul style="list-style-type: none"> - Proporcionar un formulario de registro que solicite un nombre de usuario y una contraseña. - Validar que el nombre de usuario ingresado sea único y que la contraseña cumpla con los requisitos de seguridad. - Almacenar de forma segura la información del usuario en la base de datos.
Características de los usuarios	- Nuevos usuarios: Personas que desean acceder a la plataforma y que no tienen una cuenta previa.
Restricciones	- El sistema debe verificar la unicidad del nombre de usuario antes de permitir el registro.

	- La contraseña debe cumplir con ciertos criterios de complejidad (longitud mínima, uso de caracteres especiales, etc.).
Requisitos funcionales	<p>Caso de uso: Registro de usuario</p> <ul style="list-style-type: none"> - Actor: Nuevo usuario - Precondición: El usuario debe estar en la pantalla de registro. - Flujo principal: <p>El usuario accede a la pantalla de registro. El usuario ingresa un nombre de usuario y una contraseña. El sistema verifica la unicidad del nombre de usuario. Si el nombre de usuario es único y la contraseña es válida, el sistema almacena la información y confirma el registro. Si el nombre de usuario ya está en uso o la contraseña no es válida, se muestra un mensaje de error.</p>
Requisitos no funcionales	<ul style="list-style-type: none"> - El registro debe ser rápido, garantizando que el usuario no experimente demoras significativas al completar el proceso. - Las contraseñas deben ser almacenadas de forma segura (por ejemplo, mediante encriptación).

Área general de requisito: Búsqueda de proteínas

Requisito 4: Tanto los usuarios registrados como los no registrados podrán realizar búsquedas de proteínas utilizando parámetros definidos.

Perspectiva del producto	La aplicación debe permitir que tanto usuarios registrados como no registrados realicen búsquedas de proteínas utilizando un conjunto de parámetros predefinidos, facilitando el acceso a información sobre proteínas de interés.
Funciones del producto	<ul style="list-style-type: none"> - Proporcionar una interfaz de búsqueda donde los usuarios puedan ingresar parámetros definidos para buscar proteínas. - Mostrar los resultados de la búsqueda en un formato claro y comprensible. - Permitir que los usuarios ajusten los parámetros de búsqueda para refinar los resultados.
Características de los usuarios	<ul style="list-style-type: none"> - Usuarios registrados: Tienen acceso completo a la funcionalidad de búsqueda, pueden guardar búsquedas y seleccionar favoritos para futuras referencias. - Usuarios no registrados: Pueden realizar búsquedas con acceso limitado, sin opción de guardar búsquedas ni favoritos.
Restricciones	<ul style="list-style-type: none"> - La búsqueda debe ser accesible sin necesidad de registro, pero los usuarios no registrados tendrán acceso limitado a ciertas funcionalidades. - El rendimiento de la búsqueda debe ser óptimo, incluso con un gran volumen de datos.
Requisitos funcionales	<p>Caso de uso: Búsqueda de proteínas</p> <ul style="list-style-type: none"> - Actor: Usuario registrado o no registrado - Precondición: El usuario debe estar en la pantalla de búsqueda. - Flujo principal:

	<p>El usuario accede a la pantalla de búsqueda de proteínas.</p> <p>El usuario ingresa parámetros definidos (por ejemplo, nombre de la proteína, categoría, etc.).</p> <p>El sistema procesa la búsqueda y muestra los resultados.</p> <p>El usuario puede refinar la búsqueda utilizando diferentes parámetros si lo desea.</p>
Requisitos no funcionales	<ul style="list-style-type: none"> - La búsqueda debe completarse en un tiempo razonable, incluso con múltiples usuarios realizando búsquedas simultáneamente. - La interfaz debe ser intuitiva y fácil de usar para garantizar una buena experiencia de usuario.

Requisito 5: Se habilitará una búsqueda avanzada para filtrar proteínas basadas en secuencia FASTA, taxonomía y organismo.

Perspectiva del producto	La aplicación proporcionará una funcionalidad de búsqueda avanzada que permitirá a los usuarios filtrar proteínas utilizando criterios específicos como secuencia FASTA, taxonomía y organismo. Esto facilitará la investigación detallada y la identificación de proteínas relevantes.
Funciones del producto	<ul style="list-style-type: none"> - Proporcionar una interfaz de búsqueda avanzada donde los usuarios puedan seleccionar múltiples criterios de filtrado. - Permitir la entrada de secuencias FASTA para búsquedas específicas. - Mostrar resultados que coincidan con los criterios seleccionados de manera clara y ordenada.
Características de los usuarios	<ul style="list-style-type: none"> - Usuarios registrados: Tendrán acceso completo a la búsqueda avanzada y podrán guardar sus filtros como búsquedas frecuentes. - Usuarios no registrados: Podrán utilizar la búsqueda avanzada, pero con opciones limitadas para guardar filtros.
Restricciones	<ul style="list-style-type: none"> - La búsqueda avanzada debe manejar eficientemente múltiples criterios de filtrado sin comprometer el rendimiento. - Los filtros deben ser intuitivos y accesibles para usuarios de diferentes niveles de experiencia.
Requisitos funcionales	<p>Caso de uso: Búsqueda avanzada de proteínas</p> <ul style="list-style-type: none"> - Actor: Usuario registrado o no registrado - Precondición: El usuario debe estar en la pantalla de búsqueda avanzada. - Flujo principal: <p>El usuario accede a la pantalla de búsqueda avanzada.</p> <p>El usuario selecciona los criterios de filtrado (secuencia FASTA, taxonomía y otro).</p> <p>El sistema procesa la búsqueda utilizando los criterios seleccionados.</p> <p>El sistema muestra los resultados filtrados de acuerdo con los criterios especificados.</p>
Requisitos no funcionales	<ul style="list-style-type: none"> - La búsqueda avanzada debe completarse en un tiempo razonable, incluso cuando se aplican múltiples filtros.

	- La interfaz debe ser amigable y fácil de usar, permitiendo una rápida selección de criterios.
--	---

Área general de requisito: Análisis comparativo

Requisito 6: Los usuarios podrán realizar un análisis comparativo entre una secuencia FASTA proporcionada por el usuario y una secuencia almacenada en el sistema.

Perspectiva del producto	La aplicación ofrecerá la funcionalidad de análisis comparativo que permitirá a los usuarios comparar una secuencia FASTA personalizada con secuencias almacenadas en el sistema, facilitando investigaciones detalladas y estudios sobre similitudes y diferencias.
Funciones del producto	<ul style="list-style-type: none"> - Permitir la carga de una secuencia FASTA personalizada por parte del usuario. - Comparar la secuencia cargada con las secuencias almacenadas en el sistema. - Generar un informe de similitudes y diferencias entre las secuencias.
Características de los usuarios	<ul style="list-style-type: none"> - Usuarios registrados: Tendrán acceso completo al análisis comparativo y podrán guardar sus secuencias para comparaciones futuras. - Usuarios no registrados: Podrán realizar comparaciones, pero con limitaciones en el número de secuencias que pueden cargar.
Restricciones	<ul style="list-style-type: none"> - La comparación debe ser precisa y realizarse en un tiempo razonable, incluso para secuencias largas. - Debe haber validación para asegurar que la secuencia FASTA proporcionada cumpla con el formato adecuado.
Requisitos funcionales	<p>Caso de uso: Análisis comparativo de secuencias</p> <ul style="list-style-type: none"> - Actor: Usuario registrado o no registrado - Precondición: El usuario debe estar en la pantalla de análisis comparativo. - Flujo principal: <p>El usuario carga una secuencia FASTA. El sistema valida la secuencia y la almacena temporalmente. El usuario selecciona una secuencia almacenada para comparar. El sistema realiza la comparación y genera un informe detallado. El usuario puede ver el informe de similitudes y diferencias.</p>
Requisitos no funcionales	<ul style="list-style-type: none"> - El análisis comparativo debe completarse en un tiempo razonable, incluso para secuencias largas. - La interfaz de usuario debe ser clara y permitir una fácil carga y selección de secuencia.

Área general de requisito: Histórico y Favoritos

Requisito 7: Los usuarios registrados tendrán acceso a un historial de búsquedas realizadas.

Perspectiva del producto	La aplicación permitirá a los usuarios registrados acceder a un historial de búsquedas realizadas, mejorando la experiencia del usuario al facilitar el acceso rápido a búsquedas anteriores y permitiendo repetir o modificar consultas.
Funciones del producto	<ul style="list-style-type: none">- Mostrar un historial de las búsquedas realizadas por el usuario.- Permitir al usuario volver a ejecutar o modificar búsquedas anteriores.- Ofrecer opciones para limpiar el historial si el usuario lo desea.
Características de los usuarios	<ul style="list-style-type: none">- Usuarios registrados: Tendrán acceso completo al historial de búsquedas, pudiendo gestionar su contenido y utilizarlo para mejorar su eficiencia en la búsqueda de información.
Restricciones	<ul style="list-style-type: none">- Se debe limitar el tamaño del historial para evitar un almacenamiento excesivo.- El historial debe ser accesible solo para el usuario que lo generó.
Requisitos funcionales	<p>Caso de uso: Acceso y gestión del historial de búsquedas</p> <ul style="list-style-type: none">- Actor: Usuario registrado- Precondición: El usuario debe estar autenticado y haber realizado al menos una búsqueda.- Flujo principal: El usuario accede a la sección de historial de búsquedas. El sistema muestra el historial de búsquedas realizadas. El usuario selecciona una búsqueda para volver a ejecutarla o modificarla. El usuario tiene la opción de limpiar su historial.
Requisitos no funcionales	<ul style="list-style-type: none">- El historial debe cargarse rápidamente para una buena experiencia de usuario.- La interfaz debe ser intuitiva, facilitando la navegación y gestión del historial.

Requisito 8: Los usuarios registrados podrán mantener un listado de proteínas favoritas.

Perspectiva del producto	La aplicación permitirá a los usuarios registrados crear y gestionar una lista de proteínas favoritas, facilitando el acceso rápido a las proteínas que les interesan. Esta funcionalidad mejorará la experiencia del usuario al permitir un manejo eficiente de sus preferencias.
Funciones del producto	<ul style="list-style-type: none">- Agregar proteínas a la lista de favoritas.- Eliminar proteínas de la lista de favoritas.- Visualizar la lista de proteínas favoritas.- Limitar la cantidad de proteínas en la lista.
Características de los usuarios	<ul style="list-style-type: none">- Usuarios registrados: Tendrán la capacidad de personalizar su experiencia al guardar proteínas de interés, lo que les permitirá acceder a ellas de manera rápida y sencilla.

Restricciones	<ul style="list-style-type: none"> - El sistema debe establecer un límite máximo en la cantidad de proteínas que se pueden guardar en la lista de favoritas para evitar un almacenamiento excesivo. - Las operaciones de agregar y eliminar proteínas deben ser eficientes para no afectar la experiencia del usuario.
Requisitos funcionales	<p>Caso de uso: Gestión de lista de proteínas favoritas</p> <ul style="list-style-type: none"> - Actor: Usuario registrado - Precondición: El usuario debe estar autenticado en la aplicación. - Flujo principal: El usuario selecciona una proteína para agregar a su lista de favoritas. El sistema verifica si se ha alcanzado el límite de favoritos. Si no se ha alcanzado el límite, el sistema agrega la proteína a la lista y confirma la acción. El usuario puede ver y gestionar su lista de favoritas.
Requisitos no funcionales	<ul style="list-style-type: none"> - El sistema debe asegurar la integridad y seguridad de los datos de la lista de favoritas. - Las operaciones de agregar y eliminar deben completarse en un tiempo razonable para no afectar la interacción del usuario con la aplicación.

Área general de requisito: Integraciones

Requisito g: Integración con bases de datos externas como NCBI y PubChem.

Perspectiva del producto	La aplicación se integrará con bases de datos externas como NCBI y PubChem para proporcionar a los usuarios información actualizada y completa sobre proteínas. Esta integración permitirá enriquecer las búsquedas y comparaciones de proteínas con datos adicionales provenientes de fuentes confiables.
Funciones del producto	<ul style="list-style-type: none"> - Consultar y obtener información de proteínas desde las bases de datos externas. - Presentar los datos obtenidos en la interfaz de la aplicación. - Sincronizar la información de manera eficiente para garantizar su actualización.
Características de los usuarios	- Usuarios registrados y no registrados: Ambos tipos de usuarios podrán acceder a información adicional sobre proteínas gracias a la integración, mejorando la calidad de sus búsquedas y análisis.
Restricciones	- La aplicación debe gestionar adecuadamente la carga de datos para no afectar el rendimiento del sistema.
Requisitos funcionales	<p>Caso de uso: Integración con bases de datos externas</p> <ul style="list-style-type: none"> - Actor: Sistema - Precondición: El sistema debe estar conectado a Internet y tener acceso a las APIs de NCBI y PubChem. - Flujo principal: El sistema realiza una solicitud a la base de datos externa (NCBI o PubChem).

	La base de datos externa devuelve los datos solicitados sobre una proteína. El sistema presenta los datos al usuario en la interfaz.
Requisitos no funcionales	<ul style="list-style-type: none"> - La integración debe ser robusta y manejar errores de conexión de manera adecuada. - El tiempo de respuesta para obtener datos de las bases de datos externas debe ser aceptable para no afectar la experiencia del usuario.

Área general de requisito: Descarga de resultados (Opcional)

Requisito 10: Los usuarios tendrán la opción de descargar los resultados de búsqueda en formato CSV.

Perspectiva del producto	La aplicación ofrecerá a los usuarios la posibilidad de descargar los resultados de búsqueda en formato CSV, facilitando el almacenamiento y análisis de datos fuera de la plataforma. Esta funcionalidad permitirá a los usuarios integrar los datos con otras herramientas de análisis y procesamiento.
Funciones del producto	<ul style="list-style-type: none"> - Permitir a los usuarios descargar los resultados de búsqueda en un archivo CSV. - Se buscaría asegurar que el formato del archivo CSV sea compatible con herramientas comunes de análisis de datos.
Características de los usuarios	- Usuarios registrados y no registrados: Ambos tipos de usuarios podrán descargar los resultados de búsqueda, lo que les permitirá manipular los datos según sus necesidades.
Restricciones	<ul style="list-style-type: none"> - El sistema debe gestionar correctamente los datos para asegurarse de que se exporten de manera estructurada y precisa. - Se debe garantizar que la funcionalidad de descarga no afecte el rendimiento general de la aplicación.
Requisitos funcionales	<p>Caso de uso: Descarga de resultados en formato CSV</p> <ul style="list-style-type: none"> - Actor: Usuario registrado o no registrado - Precondición: El usuario debe haber realizado una búsqueda que arroje resultados. - Flujo principal: <p>El usuario realiza una búsqueda y obtiene resultados. El usuario selecciona la opción de descargar los resultados en formato CSV. El sistema genera el archivo CSV con los resultados de la búsqueda. El usuario descarga el archivo CSV en su dispositivo.</p>
Requisitos no funcionales	<ul style="list-style-type: none"> - La generación y descarga del archivo CSV deben realizarse en un tiempo razonable para no afectar la experiencia del usuario. - El sistema debe asegurar la integridad de los datos exportados en el archivo CSV.

- **Requisitos No Funcionales**

Área general de requisito: Seguridad

Requisito 11: Al autenticarse, los usuarios recibirán un JWT (JSON Web Token) que deberán usar para identificarse a través de los microservicios.

Perspectiva del producto	El sistema proporcionará un JWT a los usuarios autenticados, lo que permitirá su identificación segura y su uso en las diferentes peticiones a los microservicios. Este mecanismo mejorará la seguridad de la plataforma y la escalabilidad de los servicios.
Funciones del producto	<ul style="list-style-type: none">- Generar un JWT al momento de la autenticación del usuario.- Permitir que los usuarios autenticados usen el JWT para acceder a los microservicios.- Validar el JWT en cada petición a los microservicios para garantizar la autenticidad y validez del usuario.
Características de los usuarios	<ul style="list-style-type: none">- Usuarios registrados: Solo los usuarios que se autentifiquen correctamente recibirán un JWT que deberán utilizar para interactuar con los servicios protegidos del sistema.
Restricciones	<ul style="list-style-type: none">- El JWT deberá tener una caducidad establecida para reforzar la seguridad.- Los tokens deben estar encriptados y ser validados por el servidor antes de acceder a cualquier microservicio.- Debe haber un mecanismo de revocación en caso de que se comprometa el token.
Requisitos funcionales	<p>Caso de uso: Autenticación con JWT</p> <ul style="list-style-type: none">- Actor: Usuario registrado- Precondición: El usuario debe estar registrado y debe autenticarse con su nombre de usuario y contraseña.- Flujo principal: El usuario ingresa su nombre de usuario y contraseña para iniciar sesión. El sistema valida las credenciales y genera un JWT para el usuario autenticado. El usuario recibe el JWT y lo incluye en las cabeceras de las solicitudes a los microservicios. Los microservicios verifican la validez del JWT antes de procesar cada solicitud. El sistema permite o niega el acceso a los servicios en función de la validez del token.

Área general de requisito: Despliegue y accesibilidad

Requisito 12: Las APIs deberán estar desplegadas en la nube y ser accesibles desde Internet.

Perspectiva del producto	El sistema deberá permitir que todas las APIs estén desplegadas en la nube y sean accesibles públicamente a través de Internet, asegurando que las funcionalidades de la aplicación estén disponibles desde cualquier lugar y dispositivo.
Funciones del producto	<ul style="list-style-type: none">- Desplegar las APIs en un servicio de cloud computing que soporte alta disponibilidad y escalabilidad.- Hacer que las APIs sean accesibles mediante URLs públicas.- Implementar mecanismos de seguridad para controlar el acceso a las APIs desde Internet, como HTTPS y autenticación basada en tokens.
Características de los usuarios	<ul style="list-style-type: none">- Usuarios registrados y no registrados: Podrán acceder a las APIs a través de solicitudes HTTP o HTTPS según sus niveles de permisos.
Restricciones	<ul style="list-style-type: none">- Las APIs deben estar protegidas para asegurar la comunicación a través de HTTPS.- Las APIs deben estar alojadas en AWS que garantice alta disponibilidad y escalabilidad.- Se deben implementar políticas de control de acceso y limitación de tasa.
Requisitos funcionales	<p>Caso de uso: Acceso a las APIs desde la nube</p> <ul style="list-style-type: none">- Actor: Usuario o sistema externo- Precondición: El usuario debe tener conexión a Internet y acceso a la API a través de las credenciales o permisos adecuados.- Flujo principal: El usuario o sistema envía una solicitud a la API a través de Internet utilizando la URL pública. El sistema valida el acceso y el mecanismo de autenticación JWT. El sistema responde con los datos solicitados o un mensaje de error si el acceso no es válido. La API maneja múltiples solicitudes concurrentes sin perder rendimiento o disponibilidad.

Área general de requisito: Portabilidad

Requisito 13: La plataforma debe ser compatible con navegadores web y sistemas operativos definidos.

Perspectiva del producto	La plataforma deberá ser compatible con una lista de navegadores web y sistemas operativos previamente definidos, asegurando que los usuarios puedan acceder y utilizar todas las funcionalidades sin problemas desde diferentes dispositivos y configuraciones.
Funciones del producto	<ul style="list-style-type: none">- La plataforma debe funcionar de manera óptima en Google Chrome.- El sistema será accesible con Windows.

Características de los usuarios	- Usuarios finales: Utilizarán la plataforma navegadores web.
Restricciones	- Las compatibilidades que puedan existir con el navegador o sistemas operativos.
Requisitos funcionales	<p>Caso de uso: Compatibilidad</p> <p>- Actor: Usuario final</p> <p>- Precondición: El usuario debe estar utilizando el navegador y sistema operativo seleccionado.</p> <p>- Flujo principal:</p> <p>El usuario accede a la plataforma desde un navegador web en su dispositivo.</p> <p>La plataforma se carga correctamente y permite la navegación completa sin errores.</p> <p>El usuario utiliza las funcionalidades disponibles, incluyendo formularios, búsquedas, y visualización de datos.</p> <p>La plataforma mantiene el mismo comportamiento y apariencia en todos los navegadores y sistemas operativos compatibles.</p>

Requisito 14: (Opcional) La aplicación debe estar conteneirizada utilizando Docker.

Perspectiva del producto	La aplicación debe ser conteneirizada utilizando Docker para garantizar que pueda ser desplegada y ejecutada de manera uniforme en diferentes entornos de desarrollo, prueba, y producción. La conteneirización permitirá una mayor flexibilidad, portabilidad y eficiencia en la gestión de la infraestructura.
Funciones del producto	<p>- La aplicación debe ser empaquetada en contenedores Docker, lo que incluirá todas las dependencias, configuraciones y librerías necesarias para su funcionamiento.</p> <p>- Docker permitirán que la aplicación se despliegue fácilmente en cualquier servidor compatible, eliminando problemas de configuración de entorno.</p> <p>- La aplicación podrá ejecutarse tanto en entornos locales como en servicios de nube compatibles con Docker.</p>
Características de los usuarios	<p>- Desarrolladores: Necesitarán conteneirizar la aplicación utilizando Docker para garantizar consistencia en los entornos de desarrollo, prueba y producción.</p> <p>- Administradores del sistema: Se beneficiarán de una gestión más eficiente de la infraestructura mediante la conteneirización.</p>
Restricciones	<p>- Se debe asegurar que todos los componentes de la aplicación (front-end, back-end, bases de datos) estén correctamente integrados en los contenedores.</p> <p>- El Dockerfile debe estar optimizado para evitar un consumo innecesario de recursos y garantizar la eficiencia.</p>
Requisitos funcionales	<p>Caso de uso: Conteneirización de la aplicación</p> <p>- Actor: Desarrollador</p> <p>- Precondición: La aplicación debe estar lista para empaquetarse en contenedores.</p>

	<ul style="list-style-type: none"> - Flujo principal: El desarrollador configura el Dockerfile para conteneirizar la aplicación. La aplicación se empaqueta en un contenedor Docker junto con todas sus dependencias. El contenedor Docker se ejecuta en un entorno local o en un servidor en la nube. La aplicación se ejecuta correctamente dentro del contenedor y está disponible para los usuarios.
--	---

Área general de requisito: Estándares de APIs

Requisito 15: Las APIs deberán cumplir con el estándar RESTful o HATEOAS.

Perspectiva del producto	Las APIs que proporcionará la plataforma deberán cumplir con los principios del estándar RESTful o HATEOAS para asegurar la interoperabilidad, eficiencia y facilidad de uso en la comunicación entre cliente y servidor. Estas APIs permitirán la interacción con las funcionalidades del sistema, como búsqueda de proteínas, autenticación de usuarios, y manejo de favoritos.
Funciones del producto	<ul style="list-style-type: none"> - Deben seguir los principios de REST, como el uso de verbos HTTP (GET, POST, PUT, DELETE), representación de recursos y operaciones. - En caso de aplicar HATEOAS, cada respuesta API debe incluir enlaces que indiquen las acciones disponibles según el estado actual del recurso, facilitando la navegabilidad dinámica para los usuarios.
Características de los usuarios	<ul style="list-style-type: none"> - Front-End: Consumirán las APIs para integrar las funcionalidades de la aplicación web.
Restricciones	<ul style="list-style-type: none"> - Las APIs deben seguir las convenciones de URI legibles y ser consistentes en su estructura para facilitar su uso. - Se debe garantizar que los códigos de estado HTTP se utilicen correctamente para reflejar el resultado de cada solicitud. - Las APIs deben ser eficientes y minimizar la latencia.
Requisitos funcionales	<p>Caso de uso: Implementación de APIs RESTful/HATEOAS</p> <ul style="list-style-type: none"> - Actor: Desarrollador - Precondición: Los servicios backend de la plataforma deben estar implementados. - Flujo principal: El desarrollador diseña y crea las APIs siguiendo los principios de REST o HATEOAS. El cliente realiza una solicitud HTTP a la API. El servidor responde con el recurso solicitado o con un error en caso de que la operación falle. En caso de HATEOAS, la respuesta incluye enlaces a otras acciones disponibles sobre el recurso.

	- Postcondición: Las APIs proporcionan una interfaz consistente y eficiente para que los clientes interactúen con los recursos de la plataforma.
--	--

Requisito 16: Las rutas (URI) de las APIs deben seguir las convenciones de naming de RESTful.

Perspectiva del producto	Las rutas de las APIs deben seguir las convenciones de naming de RESTful, lo que significa que las URIs deben ser claras, legibles y representar adecuadamente los recursos y las operaciones que se realizan sobre ellos.
Funciones del producto	<ul style="list-style-type: none"> - Las rutas deben utilizar sustantivos en plural para representar recursos (/proteins /users). - Las operaciones sobre los recursos deben realizarse a través de verbos HTTP como GET, POST, PUT, DELETE (GET /proteins, POST /users).
Características de los usuarios	- Front-End: Consumirán estas rutas para realizar solicitudes desde la interfaz web.
Restricciones	- Las rutas deben seguir las convenciones REST estrictamente para asegurar que la API sea predecible, coherente y fácil de utilizar.
Requisitos funcionales	<p>Caso de uso: Naming de URIs para APIs RESTful</p> <ul style="list-style-type: none"> - Actor: Desarrollador - Precondición: La API debe estar diseñada para seguir el estándar RESTful. - Flujo principal: El desarrollador define las rutas para los recursos del sistema siguiendo las convenciones RESTful. Cada ruta URI representa un recurso, como una lista de proteínas, un usuario, o una búsqueda. Las operaciones sobre los recursos (lectura, creación, actualización, eliminación) se ejecutan a través de los verbos HTTP adecuados. - Postcondición: Las URIs son consistentes y siguen las mejores prácticas de RESTful, facilitando su uso e integración.

Área general de requisito: Eficiencia

Requisito 17: La plataforma deberá ser eficiente en términos de tiempo de respuesta y uso de recursos.

Perspectiva del producto	La plataforma debe garantizar una alta eficiencia en cuanto a tiempos de respuesta y uso óptimo de recursos (memoria, CPU, y almacenamiento) para ofrecer una experiencia de usuario fluida y garantizar que pueda manejar una gran cantidad de solicitudes simultáneas sin degradar el rendimiento.
Funciones del producto	- Las peticiones a la plataforma deben procesarse de forma rápida y eficiente, reduciendo la latencia de las operaciones.

	- El sistema debe estar diseñado para aprovechar al máximo los recursos disponibles, minimizando el uso innecesario de memoria y procesamiento.
Características de los usuarios	- Usuarios finales: Esperan tiempos de carga rápidos y una experiencia sin interrupciones.
Restricciones	- El sistema debe estar optimizado para un tiempo de respuesta bajo para la mayoría de las solicitudes comunes.
Requisitos funcionales	<p>Caso de uso: Optimización de tiempo de respuesta y uso de recursos</p> <ul style="list-style-type: none"> - Actor: Usuario final - Precondición: El sistema debe estar disponible y activo. - Flujo principal: El usuario realiza una solicitud a la plataforma (como una búsqueda o autenticación). El sistema procesa la solicitud dentro de los límites de tiempo aceptables (menos de 1 segundo en promedio). La plataforma utiliza de forma óptima los recursos del servidor para completar la operación. - Postcondición: La operación se completa con éxito sin interrupciones y el sistema sigue funcionando eficientemente.

Área general de requisito: Mantenibilidad

Requisito 18: El código debe seguir buenas prácticas de desarrollo para facilitar su comprensión y mantenimiento.

Perspectiva del producto	El código de la plataforma debe ser comprensible, mantenible y escalable, siguiendo buenas prácticas de desarrollo.
Funciones del producto	<ul style="list-style-type: none"> - El código debe ser legible, con una estructura clara y siguiendo convenciones establecidas (como naming conventions y indentación). - El código debe estar organizado en módulos o clases que cumplan con el principio de responsabilidad única, facilitando su reutilización y prueba. - El código debe ser probado unitariamente y seguir un flujo de integración continua para asegurar su correcta funcionalidad.
Características de los usuarios	- Desarrolladores: Necesitan que el código sea fácil de leer y comprender para facilitar el trabajo colaborativo y el mantenimiento futuro.
Restricciones	- El código debe cumplir con los estándares SOLID en desarrollo orientado a objetos.
Requisitos funcionales	<p>Caso de uso: Aplicación de buenas prácticas en el desarrollo de código</p> <ul style="list-style-type: none"> - Actor: Desarrollador - Precondición: El desarrollador debe estar trabajando en una funcionalidad del sistema. - Flujo principal: El desarrollador sigue las buenas prácticas para escribir código modular, reutilizable y mantenible.

	<p>Se documenta el código con comentarios y se agregan las pruebas necesarias para asegurar la calidad.</p> <ul style="list-style-type: none"> - Postcondición: El código es aprobado, documentado y está listo para su integración en el sistema principal.
--	---

Requisito 19: La documentación del código deberá ser completa, legible y precisa.

Perspectiva del producto	La documentación del código es fundamental para garantizar que cualquier desarrollador pueda entender rápidamente la lógica y funcionamiento del sistema. La documentación debe ser completa, legible y precisa, permitiendo que los futuros mantenedores puedan modificar o extender el código sin dificultades.
Funciones del producto	<ul style="list-style-type: none"> - La documentación debe incluir comentarios claros y concisos sobre las funciones, clases y métodos más importantes. - Cada vez que se realicen cambios significativos en el código, la documentación debe actualizarse en consecuencia.
Características de los usuarios	<ul style="list-style-type: none"> - La documentación esté alineada con el código para poder comprender el propósito y uso de cada función, clase, o módulo.
Restricciones	<ul style="list-style-type: none"> - El sistema debe cumplir con los estándares de documentación del lenguaje y del equipo de desarrollo.
Requisitos funcionales	<p>Caso de uso: Documentación del código</p> <ul style="list-style-type: none"> - Actor: Desarrollador - Precondición: El desarrollador ha creado o modificado partes del código. - Flujo principal: El desarrollador crea o actualiza la documentación, proporcionando descripciones precisas de cada función, clase, o módulo. Se agregan ejemplos de uso y detalles de los parámetros y retorno de funciones. Se incluye la documentación externa en el repositorio, siguiendo las guías del proyecto. - Postcondición: La documentación está lista y es revisada junto con el código.

Área general de requisito: Comprobabilidad

Requisito 20: Se deben realizar pruebas exhaustivas para asegurar el buen funcionamiento de la plataforma.

Perspectiva del producto	El objetivo de este requisito es garantizar que la plataforma funcione de manera eficiente, sin errores, y cumpla con los requisitos establecidos. Para ello, es esencial realizar pruebas exhaustivas que cubran tanto aspectos funcionales como no funcionales, permitiendo detectar y corregir cualquier posible fallo.
Funciones del producto	<ul style="list-style-type: none"> - Pruebas unitarias: Verificar que cada módulo o función del sistema funcione correctamente de manera individual.

	<ul style="list-style-type: none"> - Pruebas de integración: Comprobar que los módulos interactúan de manera correcta entre sí cuando se integran. - Pruebas funcionales: Asegurar que la plataforma cumple con todos los requisitos funcionales definidos. - Pruebas de seguridad: Identificar vulnerabilidades y asegurar que los datos del sistema están protegidos.
Características de los usuarios	- Desarrollador: Necesita implementar y ejecutar una suite de pruebas exhaustiva para validar el correcto funcionamiento de la plataforma en diferentes escenarios.
Restricciones	- Las pruebas deben cubrir todos los requisitos definidos y ejecutarse en todos los módulos y funcionalidades de la plataforma.
Requisitos funcionales	<p>Caso de uso: Realizar pruebas exhaustivas</p> <ul style="list-style-type: none"> - Actor: Desarrolladores - Precondición: La plataforma o una parte de ésta debe estar lista para ser evaluada. - Flujo principal: El equipo de QA ejecuta pruebas unitarias para cada módulo de la plataforma. Se realizan pruebas de integración para validar que los módulos interactúan correctamente. Se realizan pruebas de seguridad para identificar y corregir vulnerabilidades. - Postcondición: La plataforma ha sido probada y cualquier defecto ha sido corregido.

La gestión de requisitos se aborda desde una perspectiva ágil, empleando historias de usuario para priorizar y planificar el desarrollo. Esta metodología facilita la agilización del proceso, permitiendo una estimación precisa del esfuerzo mediante la técnica de puntos de historia. Se utiliza la secuencia de Fibonacci (1, 2, 3, 5, 8) para asignar un valor a cada historia, considerando los siguientes criterios:

- 1 punto: Tarea muy sencilla y familiar, similar a otras ya realizadas.
- 2 puntos: Tarea fácil, pero que puede requerir cierta investigación.
- 3 puntos: Tarea de complejidad moderada, con un esfuerzo estimado.
- 5 puntos: Tarea relativamente compleja, que demanda mayor tiempo y esfuerzo.
- 8 puntos: Tarea compleja, que probablemente requiera colaboración o habilidades especializadas."

Requisitos Funcionales

Área de interés: Autenticación y Registro de Usuarios

Requisito 1: La aplicación funcionara para usuarios registrados y no registrados.

Número de historia: 1

Nombre de la historia: Acceso para usuarios no registrados

Usuario: Usuario no registrado

Puntos estimados de esfuerzo: 3

Descripción de la historia de usuario: Como usuario no registrado, quiero acceder a la aplicación y realizar búsquedas básicas para explorar las proteínas disponibles sin necesidad de registrarme.

Observaciones: Asegurar que el acceso no registrado no comprometa la seguridad del sistema.

Criterios de aceptación:

- El usuario puede acceder a la aplicación.
- El usuario puede realizar búsquedas básicas de proteínas.

Número de historia: 2

Nombre de la historia: Acceso para usuarios registrados

Usuario: Usuario registrado

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Como usuario registrado, quiero iniciar sesión en la aplicación para acceder a todas las funcionalidades disponibles.

Observaciones: Incluir medidas de seguridad para proteger la información del usuario.

Criterios de aceptación:

- El usuario puede iniciar sesión con su nombre de usuario y contraseña.
- El usuario tiene acceso a funcionalidades completas tras iniciar sesión.

Requisito 2: Los usuarios podrán iniciar sesión en la plataforma mediante un nombre de usuario y contraseña.

Número de historia: 3

Nombre de la historia: Inicio de sesión de usuarios

Usuario: Usuario registrado

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Como usuario registrado, quiero iniciar sesión en la plataforma utilizando mi nombre de usuario y contraseña para acceder a las funcionalidades completas.

Observaciones: Asegurar que la sesión se mantenga activa durante un período razonable y que haya opciones para recuperar la contraseña en caso de olvido.

Criterios de aceptación:

- El usuario puede acceder a la pantalla de inicio de sesión.
- El usuario puede ingresar su nombre de usuario y contraseña.

- Si las credenciales son correctas, el usuario es redirigido a la página principal de la aplicación.
- Si las credenciales son incorrectas, se muestra un mensaje de error claro.

Requisito 3: Los usuarios podrán registrarse en el sistema proporcionando un nombre de usuario y definiendo una contraseña.

Número de historia: 4

Nombre de la historia: Registro de usuarios

Usuario: Nuevo usuario

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Como nuevo usuario, quiero registrarme en la plataforma proporcionando un nombre de usuario y una contraseña, para poder acceder a las funcionalidades de la aplicación.

Observaciones: Asegurar que se brinden mensajes claros sobre la complejidad de la contraseña y la disponibilidad del nombre de usuario.

Criterios de aceptación:

- El usuario puede acceder a la pantalla de registro.
- El usuario puede ingresar un nombre de usuario y una contraseña.
- El sistema verifica que el nombre de usuario es único y que la contraseña cumple con los requisitos.
- Si el registro es exitoso, el usuario recibe una confirmación de que su cuenta ha sido creada.
- Si hay un error (nombre de usuario ya en uso o contraseña no válida), se muestra un mensaje de error claro.

Área de interés: Búsqueda de proteínas

Requisito 4: Tanto los usuarios registrados como los no registrados podrán realizar búsquedas de proteínas utilizando parámetros definidos.

Número de historia: 5

Nombre de la historia: Búsqueda de proteínas

Usuario: Usuario registrado o no registrado

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Como usuario, quiero poder realizar búsquedas de proteínas utilizando parámetros definidos, para acceder a información relevante sobre proteínas de mi interés.

Observaciones: Asegurar que los resultados sean presentados de manera clara y fácil de interpretar.

Criterios de aceptación:

- El usuario puede acceder a la pantalla de búsqueda.
- El usuario puede ingresar parámetros de búsqueda.
- El sistema procesa la búsqueda y muestra resultados relevantes.
- Los resultados son presentados de manera organizada y fácil de entender.

- Los usuarios no registrados pueden realizar búsquedas, pero con acceso limitado a ciertos resultados.

Requisito 5: Se habilitará una búsqueda avanzada para filtrar proteínas basadas en secuencia FASTA, taxonomía y organismo.

Número de historia: 6

Nombre de la historia: Búsqueda avanzada de proteínas

Usuario: Usuario registrado o no registrado

Puntos estimados de esfuerzo: 8

Descripción de la historia de usuario: Como usuario, quiero poder realizar búsquedas avanzadas de proteínas utilizando criterios como secuencia FASTA, taxonomía y organismo, para encontrar proteínas específicas que se ajusten a mis necesidades de investigación.

Observaciones: Asegurar que la interfaz de selección de filtros sea intuitiva y que los resultados se presenten de manera clara.

Criterios de aceptación:

- El usuario puede acceder a la pantalla de búsqueda avanzada.
- El usuario puede seleccionar múltiples criterios de filtrado.
- El sistema procesa la búsqueda y muestra resultados que coinciden con los criterios.
- Los resultados se presentan de manera ordenada y fácil de interpretar.
- La búsqueda se completa en un tiempo razonable, incluso con múltiples filtros aplicados.

Área de interés: Análisis Comparativo

Requisito 6: Los usuarios podrán realizar un análisis comparativo entre una secuencia FASTA proporcionada por el usuario y una secuencia almacenada en el sistema.

Número de historia: 7

Nombre de la historia: Análisis comparativo de secuencias

Usuario: Usuario registrado o no registrado

Puntos estimados de esfuerzo: 8

Descripción de la historia de usuario: Como usuario, quiero poder realizar un análisis comparativo entre una secuencia FASTA que proporciono y una secuencia almacenada en el sistema, para entender las similitudes y diferencias entre ellas en mi investigación.

Observaciones: Asegurar que la validación de la secuencia FASTA sea efectiva y que el informe sea claro y detallado.

Criterios de aceptación:

- El usuario puede cargar una secuencia FASTA que cumple con el formato requerido.
- El sistema valida la secuencia cargada.
- El usuario puede seleccionar una secuencia almacenada para la comparación.
- El sistema genera un informe detallado de similitudes y diferencias entre las secuencias.
- El análisis se completa en un tiempo razonable y el informe es fácil de interpretar.

Área de interés: Histórico y favoritos

Requisito 7: Los usuarios registrados tendrán acceso a un historial de búsquedas realizadas.

Número de historia: 8

Nombre de la historia: Acceso al historial de búsquedas

Usuario: Usuario registrado

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Como usuario registrado, quiero tener acceso a un historial de búsquedas realizadas para poder revisar mis consultas anteriores y facilitar mi investigación.

Observaciones: Es importante que el historial sea fácil de gestionar y visualizar.

Criterios de aceptación:

- El usuario puede ver un listado de búsquedas realizadas.
- El sistema permite al usuario seleccionar una búsqueda para volver a ejecutarla.
- El usuario puede limpiar su historial de búsquedas.
- El historial se carga rápidamente y es fácil de navegar.

Requisito 8: Los usuarios registrados podrán mantener un listado de proteínas favoritas.

Número de historia: 9

Nombre de la historia: Mantenimiento de listado de proteínas favoritas

Usuario: Usuario registrado

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Como usuario registrado, quiero poder mantener un listado de proteínas favoritas para acceder rápidamente a las que me interesan y gestionar fácilmente mis preferencias.

Observaciones: Es importante que la gestión de la lista sea intuitiva y que el límite de proteínas sea claramente comunicado al usuario.

Criterios de aceptación:

- El usuario puede agregar proteínas a su lista de favoritas.
- El usuario puede eliminar proteínas de su lista de favoritas.
- Se informa al usuario si se ha alcanzado el límite de proteínas en la lista.
- La lista de favoritas se muestra de manera clara y accesible en la interfaz del usuario.

Área de interés: Integraciones

Requisito 9: Integración con bases de datos externas como NCBI y PubChem.

Número de historia: 10

Nombre de la historia: Integración con bases de datos externas

Usuario: Usuario registrado y no registrado

Puntos estimados de esfuerzo: 8

Descripción de la historia de usuario: Como usuario de la plataforma, quiero acceder a información de proteínas desde bases de datos externas como NCBI y PubChem, para obtener datos actualizados y completos que enriquezcan mis búsquedas.

Observaciones: Es crucial que la integración sea estable y eficiente para mantener la calidad del servicio.

Criterios de aceptación:

- El sistema puede realizar consultas a NCBI y PubChem.
- La información obtenida se presenta de manera clara en la aplicación.
- Se manejan adecuadamente los errores de conexión y se informa al usuario en caso de fallos.
- Los datos mostrados están actualizados y son relevantes para la búsqueda realizada.
- Descarga de resultados

Área de interés: Descarga de resultados (Opcional)

Requisito 10: Los usuarios tendrán la opción de descargar los resultados de búsqueda en formato CSV.

Número de historia: 11

Nombre de la historia: Descarga de resultados de búsqueda en formato CSV

Usuario: Usuario registrado o no registrado

Puntos estimados de esfuerzo: 3

Descripción de la historia de usuario: Como usuario, quiero tener la opción de descargar los resultados de búsqueda en formato CSV para poder almacenar y analizar los datos fuera de la plataforma.

Observaciones: Es importante que el archivo CSV sea legible y estructurado para facilitar su uso en otras herramientas.

Criterios de aceptación:

- El usuario puede descargar los resultados de búsqueda en formato CSV.
- El archivo CSV generado debe contener todos los datos relevantes de los resultados de búsqueda.
- La descarga debe completarse sin demoras significativas en la respuesta de la aplicación.
- Se proporciona una notificación al usuario cuando la descarga se completa con éxito.

Requisitos No Funcionales

Área de interés: Seguridad

Requisito 11: Al autenticarse, los usuarios recibirán un JWT (JSON Web Token) que deberán usar para identificarse a través de los microservicios.

Número de historia: 12

Nombre de la historia: Autenticación con JWT para identificación en microservicios

Usuario: Usuario registrado

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Como usuario registrado, quiero recibir un JWT después de iniciar sesión para poder identificarme de forma segura en las interacciones con los microservicios de la aplicación.

Observaciones: El JWT debe tener una caducidad adecuada para minimizar riesgos de seguridad y debe ser fácil de integrar en las solicitudes de los microservicios.

Criterios de aceptación:

- El usuario autenticado recibe un JWT válido tras iniciar sesión.
- El JWT es aceptado por todos los microservicios.
- El sistema rechaza las solicitudes si el JWT es inválido o ha caducado.
- El sistema notifica al usuario si su token ha caducado y necesita autenticarse nuevamente.
- El JWT está encriptado de manera segura y cumple con los estándares de seguridad aplicables.

Área de interés: Despliegue y Accesibilidad

Requisito 12: Las APIs deberán estar desplegadas en la nube y ser accesibles desde Internet.

Número de historia: 13

Nombre de la historia: Despliegue de APIs en la nube accesibles desde Internet

Usuario: Usuario o sistema externo

Puntos estimados de esfuerzo: 8

Descripción de la historia de usuario: Como usuario o sistema externo, quiero que las APIs estén desplegadas en la nube y accesibles desde Internet, para que pueda interactuar con las funcionalidades del sistema de forma remota y segura.

Observaciones: Es esencial que las APIs estén aseguradas y optimizadas para manejar cargas elevadas sin comprometer el rendimiento o la seguridad.

Criterios de aceptación:

- Las APIs están accesibles mediante una URL pública desde cualquier lugar con conexión a Internet.
- El acceso a las APIs se realiza exclusivamente a través de HTTPS para garantizar la seguridad.
- Las solicitudes a las APIs son autenticadas y verificadas correctamente.

- El sistema puede manejar múltiples usuarios o sistemas interactuando con las APIs simultáneamente sin problemas de rendimiento.
- Las APIs están desplegadas en AWS.

Área de interés: Portabilidad

Requisito 13: La plataforma debe ser compatible con Google Chrome y Windows.

Número de historia: 14

Nombre de la historia: Compatibilidad de la plataforma con navegadores y sistemas operativos

Usuario: Usuario final

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Como usuario final, quiero que la plataforma sea compatible con Google Chrome y Windows, para poder utilizar todas las funcionalidades sin problemas técnicos.

Observaciones: Se deben realizar pruebas de calidad para garantizar que no haya incompatibilidades.

Criterios de aceptación:

- La plataforma es completamente funcional en Google Chrome y Windows.
- No hay errores visuales ni de funcionalidad cuando se utiliza desde diferentes dispositivos.
- El diseño de la plataforma es responsivo y se adapta a diferentes resoluciones y tamaños de pantalla.
- Las pruebas de compatibilidad confirman que la plataforma cumple con los requisitos en todos los navegadores y sistemas operativos soportados.

Requisito 14: (Opcional) La aplicación debe estar conteneirizada utilizando Docker.

Número de historia: 15

Nombre de la historia: Conteneirización de la aplicación con Docker

Usuario: Desarrollador

Puntos estimados de esfuerzo: 8

Descripción de la historia de usuario: Como desarrollador, quiero conteneirizar la aplicación utilizando Docker para asegurar que pueda ser desplegada en diferentes entornos de manera eficiente y sin problemas de compatibilidad.

Observaciones: Se debe garantizar que todos los componentes de la aplicación se empaqueten correctamente y que se realicen pruebas para asegurar que el contenedor funcione correctamente en todos los entornos.

Criterios de aceptación:

- La aplicación se empaqueta correctamente en un contenedor Docker.
- El contenedor Docker puede desplegarse y ejecutarse sin errores en entornos locales y en la nube.
- El sistema dentro del contenedor responde a las solicitudes y cumple con los requisitos funcionales

Área de interés: Estándares de APIs

Requisito 15: Las APIs deberán cumplir con el estándar RESTful o HATEOAS.

Número de historia: 16

Nombre de la historia: Implementación de APIs RESTful o HATEOAS

Usuario: Desarrollador

Puntos estimados de esfuerzo: 8

Descripción de la historia de usuario: Como desarrollador, quiero que las APIs sigan el estándar RESTful o HATEOAS para que la interacción con los servicios backend sea eficiente, clara y compatible con otros sistemas.

Observaciones: Se deben considerar los principios de diseño RESTful para todas las APIs, y en los casos donde sea necesario, implementar HATEOAS para mejorar la navegabilidad de los recursos.

Criterios de aceptación:

- Las APIs deben exponer los recursos mediante verbos HTTP estándar.
- Cada respuesta API debe devolver un código de estado HTTP adecuado.
- Si se utiliza HATEOAS, las respuestas deben incluir hipervínculos relevantes que guíen a los usuarios a las posibles acciones disponibles.
- El sistema debe cumplir con los principios de seguridad y ser accesible solo para usuarios autorizados.

Requisito 16: Las rutas (URI) de las APIs deben seguir las convenciones de naming de RESTful.

Nombre de la historia: Definición de rutas RESTful consistentes

Usuario: Desarrollador

Puntos estimados de esfuerzo: 3

Descripción de la historia de usuario: Como desarrollador, quiero que las rutas de las APIs sigan las convenciones de RESTful, utilizando URIs claras y bien estructuradas, para que la API sea más fácil de usar y mantener.

Observaciones: Todas las rutas deben usar sustantivos en plural, evitar incluir verbos, y los verbos HTTP estándar deben manejar las acciones sobre los recursos.

Criterios de aceptación:

- Todas las URIs deben utilizar nombres de recursos en plural.
- Las operaciones se deben realizar utilizando verbos HTTP (GET, POST, PUT, DELETE).
- Las URIs deben estar bien documentadas.
- Las rutas deben ser legibles y seguir una estructura clara, como /proteins, /users/{id}.

Área de interés: Eficiencia

Requisito 17: La plataforma deberá ser eficiente en términos de tiempo de respuesta y uso de recursos.

Número de historia: 18

Nombre de la historia: Eficiencia de la plataforma en tiempo de respuesta y uso de recursos

Usuario: Usuario final y administrador del sistema

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Como usuario final, quiero que la plataforma sea eficiente en cuanto al tiempo de respuesta, para que mis solicitudes se completen rápidamente y la experiencia sea fluida. Como administrador del sistema, quiero que la plataforma utilice los recursos de forma eficiente, para garantizar que el rendimiento se mantenga bajo condiciones de carga.

Observaciones: Es crucial que el sistema se pruebe bajo condiciones de carga para verificar el rendimiento y la eficiencia en el uso de recursos.

Criterios de aceptación:

- La plataforma debe ser capaz de manejar incrementos en la carga sin una degradación significativa en el rendimiento.
- El uso de recursos (memoria, CPU) debe estar optimizado y monitoreado regularmente.
- Se deben implementar mecanismos de optimización, como el uso de caché y la gestión de conexiones de base de datos, para asegurar eficiencia.

Área de interés: Mantenibilidad

Requisito 18: El código debe seguir buenas prácticas de desarrollo para facilitar su comprensión y mantenimiento.

Número de historia: 19

Nombre de la historia: Código siguiendo buenas prácticas de desarrollo

Usuario: Desarrollador

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Como desarrollador, quiero que el código siga buenas prácticas de desarrollo, para que sea fácil de entender, mantener y mejorar a lo largo del tiempo sin causar deuda técnica.

Observaciones: Se deben seguir convenciones de código específicas del lenguaje (Java y Python).

Criterios de aceptación:

- El código debe seguir un formato y estilo coherente.
- El código debe estar comentado donde sea necesario, especialmente en métodos complejos.
- El código debe ser probado unitariamente.
- El código debe cumplir con los principios SOLID, DRY, y KISS.
- El código debe ser modular y seguir el principio de responsabilidad única.

Requisito 19: La documentación del código deberá ser completa, legible y precisa.

Número de historia: 20

Nombre de la historia: Documentación del código completa y precisa

Usuario: Desarrollador

Puntos estimados de esfuerzo: 3

Descripción de la historia de usuario: Como desarrollador, quiero que la documentación del código sea completa, legible y precisa, para que cualquier miembro del equipo pueda entender rápidamente el propósito, funcionamiento y uso del código.

Observaciones: Se deben documentar tanto las interfaces públicas como los métodos internos importantes.

Criterios de aceptación:

- Cada módulo, clase, y función debe tener su respectiva documentación en el estándar del lenguaje.
- La documentación debe estar actualizada con cada cambio en el código.
- La documentación debe incluir ejemplos de uso y explicaciones claras de parámetros y valores de retorno.
- La documentación externa (manual o guía) debe describir el propósito general del sistema, cómo funciona, y las interacciones entre módulos.

Área de interés: Comprobabilidad

Requisito 20: Se deben realizar pruebas exhaustivas para asegurar el buen funcionamiento de la plataforma.

Número de historia: 21

Nombre de la historia: Pruebas exhaustivas para asegurar el funcionamiento de la plataforma

Usuario: Equipo de Desarrollo

Puntos estimados de esfuerzo: 5

Descripción de la historia de usuario: Quiero realizar pruebas exhaustivas para asegurar que la plataforma funcione correctamente en términos de funcionalidad, rendimiento, seguridad, y usabilidad.

Observaciones: Las pruebas deben incluir no solo las pruebas unitarias e integradas, sino también pruebas como funcionalidad y seguridad.

Criterios de aceptación:

- Se deben realizar pruebas unitarias, de integración, funcionales y seguridad.
- Todos los módulos y funcionalidades deben ser probados exhaustivamente antes de la entrega del producto.
- Los resultados de las pruebas deben documentarse adecuadamente.
- Los errores encontrados deben ser corregidos antes de la entrega final.