

2023



Escuela Superior Politécnica de Chimborazo

Aplicaciones informáticas II

Octavo Semestre

Integrantes:

Luis López (6880)

Fernanda Moreno (6884)

Roberto Pilco (6893)

Erika Villavicencio (6915)

Periodo:

Octubre 2023 – Marzo 2024

Entregado 22 de Noviembre

Producto Backlog

Desarrollo ágil: Pila de Producto (Product Backlog)

Elaborado por: Luis López, Fernanda Moreno, Roberto Pilco, Erika Villavicencio

Identificador (ID) de la Historia	Enunciado de la Historia	Alias	Estado	Dimensión / Esfuerzo	Iteración (Sprint)	Prioridad	Comentarios
HT-01	Configuración del repositorio para proyecto	Configuración del repositorio para proyecto	planificado	5 horas	sprint 1	Alta	Se creará un proyecto compartido en GitHub con los colaboradores desarrolladores del sistema para tener acceso a las versiones del proyecto a través de ramas individuales, además, se podrá tener una facilidad para generar una documentación.
HT-02	Modelamiento de la base de datos	Modelamiento de la base de datos	planificado	130 horas	sprint 1	Alta	El objetivo de esta tarea es llevar a cabo el modelamiento de la base de datos del sistema. El modelamiento de la base de datos es crucial para garantizar una estructura eficiente y coherente que respalde las funcionalidades del sistema.
HT-03	Configuración de un servidor de base de datos	Configuración de un servidor de base de datos	planificado	5 horas	sprint 1	Alta	Crear y configurar un servidor de base de datos mediante las herramientas de Docker o Railway, con el fin de agilizar el proceso de desarrollo y mantener actualizado los datos del proyecto.
HT-04	Definición de la Arquitectura del Sistema	Definición de la Arquitectura del Sistema	planificado	160 horas	sprint 2	Alta	La tarea de Definición de la Arquitectura del Sistema tiene como objetivo establecer la estructura y los componentes principales del sistema. Esto implica tomar decisiones clave sobre la organización de los módulos, la gestión de datos, la comunicación entre componentes y la selección de tecnologías fundamentales.
HT-05	Implementación de Seguridad JWT	Implementación de Seguridad JWT	planificado	140 horas	sprint 3	Alta	La tarea de Implementación de Seguridad JWT tiene como objetivo integrar un mecanismo de seguridad basado en JSON Web Tokens (JWT) en el sistema.
HT-06	Implementación de la Capa de Presentación (Interfaz de Usuario)	Implementación de la Capa de Presentación (Interfaz de Usuario)	planificado	160 horas	sprint 4	Alta	La tarea de Implementación de la Capa de Presentación tiene como objetivo llevar a cabo el desarrollo de la interfaz de usuario del sistema.
HT-07	Implementación de la capa Lógica de Negocio	Implementación de la capa Lógica de Negocio	planificado	160 horas	sprint 5	Alta	Configurar e implementar la capa Lógica de Negocio, que es fundamental para el funcionamiento de software, ya que se encarga de procesar y gestionar la lógica específica del dominio o sector al que pertenece la aplicación.
HT-08	Capa de Acceso a Datos (Persistencia de Datos)	Capa de Acceso a Datos (Persistencia de Datos)	planificado	160 horas	sprint 6	Alta	La tarea de Implementación de la Capa de Acceso a Datos tiene como objetivo desarrollar la lógica necesaria para interactuar con la base de datos del sistema.
HU-01	Como paciente, quiero tener un diario de actividades que me permita registrar, editar, eliminar y ver mis actividades diarias. Esto me ayudará a mantener un seguimiento organizado de mis rutinas y logros, facilitándome la gestión y revisión de mi progreso a lo largo del tiempo.	Diario de actividades	planificado	160 horas	sprint 7	Por determinar	El Diario de Actividades permite a los usuarios registrar, editar y visualizar actividades diarias, facilitando la gestión y revisión del progreso de los pacientes a lo largo del tiempo.
HU-02	Como paciente con un plan nutricional asignado, necesito poder gestionar mis citas con la nutrióloga para agendar nuevas consultas, cancelar citas existentes si es necesario y revisar las fechas y horas de mis próximas citas. Esto garantizará un seguimiento adecuado de mi progreso y una comunicación continua con la nutrióloga para mejorar mi salud y bienestar.	Gestión de Citas con Nutrióloga	planificado	160 horas	sprint 7	Por determinar	Esta funcionalidad permitirá a los pacientes con planes nutricionales gestionar citas fácilmente, agendar nuevas consultas, cancelar citas existentes y revisar fechas/horas, facilitando un seguimiento proactivo de su progreso y una comunicación continua con la nutrióloga para mejorar su salud.
HU-03	Como administrador, necesito poder registrar el historial clínico de mis pacientes, visualizar la información detallada cuando sea necesario y editarla según sea requerido. Esto permitirá ofrecer un tratamiento nutricional personalizado y mantener un seguimiento efectivo del progreso de los pacientes.	Registro de Historial Clínico	planificado	80 horas	sprint 8	Por determinar	Los administradores podrán eficientemente ingresar, visualizar y editar los historiales clínicos de pacientes, facilitando tratamientos personalizados y un seguimiento efectivo del progreso.
HU-04	Como administrador, necesito ver el historial de citas agendadas y visualizar las próximas citas pendientes. Esto permitirá un control eficiente sobre la gestión de citas, facilitando una atención organizada y oportuna a los pacientes.	Gestión de Citas Asignadas	planificado	80 horas	sprint 8	Por determinar	Los administradores podrán eficientemente acceder al historial de citas, filtrarlas por paciente y visualizar las próximas citas pendientes, mejorando la organización y la puntualidad en la atención a los pacientes
HU-05	Como administrador, necesito tener acceso a un seguimiento detallado de mis pacientes, revisando su consumo alimenticio según el plan nutricional asignado y visualizando las actividades físicas registradas. Esto permitirá evaluar su progreso y realizar ajustes pertinentes en sus planes de salud y nutrición.	Seguimiento de Pacientes	planificado	160 horas	sprint 9	Por determinar	Los administradores podrán eficientemente revisar el consumo alimenticio y las actividades físicas registradas de los pacientes, facilitando la evaluación del progreso y la realización de ajustes personalizados en los planes de salud y nutrición.
HU-06	Como administrador y paciente del sistema, necesito poder actualizar datos, guardar nueva información, eliminar registros innecesarios y visualizar la información existente de los usuarios. Esto facilitará la gestión eficiente de la base de datos, asegurando que la información sea precisa y relevante para la operación del sistema.	Gestión de Usuarios	planificado	160 horas	sprint 10	Por determinar	Proporciona a administradores y pacientes herramientas eficientes para actualizar, guardar, eliminar y visualizar información de usuarios, asegurando la integridad y relevancia de la base de datos del sistema.
HU-07	Como administrador de la aplicación nutricional, necesito la capacidad de gestionar los planes nutricionales de los pacientes, permitiéndome visualizar, editar, asignar, guardar y eliminar planes según sea necesario. Esto facilitará una gestión efectiva de la información nutricional de cada paciente.	Gestión de Planes Nutricionales	planificado	160 horas	sprint 11	Por determinar	Ofrece a administradores la capacidad completa de gestionar planes nutricionales, y permite a pacientes visualizar y marcar recetas realizadas para una gestión personalizada de la información nutricional.
HT-09	Diseño responsivo	Diseño responsivo	planificado	140 horas	sprint 12	Alta	Implementar el diseño responsivo para garantizar una experiencia de usuario óptima en una variedad de dispositivos y tamaños de pantalla.
HT-10	Implementación de mejoras en la documentación del código fuente	Implementación de mejoras en la documentación del código fuente	planificado	20 horas	sprint 12	Alta	Tiene el objetivo de mejorar la calidad y la accesibilidad del código fuente de la aplicación, para que los desarrolladores comprendan rápidamente la lógica y el propósito de cada componente.

Pila de Producto (Product Backlog): Instructivo

Elaborado por: pmoinformatica.com

Columna	Instrucciones
Identificador (ID) de la Historia	Código que identifica a la historia de forma unívoca, una vez asignado, no debe ser re-usado en otra historia, ni siquiera si la historia es descartada. El código identifica la historia en otros documentos, como por ejemplo la plantilla de historias de usuario.
Enunciado de la Historia	Nombre de la historia, el cual debe ser el mismo que se utiliza en otros documentos. Se puede utilizar el formato siguiente: Como un [Rol], Necesito [Descripción de la Funcionalidad], con la finalidad de [Razón o Resultado]
Alias	Título de la historia alternativo a la descripción, que servirá para identificar más fácilmente la historia sin tener que repetir todo su enunciado. Se puede utilizar por ejemplo el nombre de la funcionalidad o requerimiento que se pretende desarrollar.
Estado	Identifica los posibles estados de la historia durante su ciclo de vida: - Vacío: La historia fue identificada pero aún no ha sido asignada a una iteración. - Planificada: La historia fue asignada a una iteración y aún no ha comenzado su ejecución. Puede tener este estado incluyendo en la iteración donde está planificado ejecutarla (pero que aún no ha comenzado). - En Proceso: La historia fue seleccionada por el equipo y está en proceso de desarrollo (en ejecución). - Hecho (Donde): La historia fue desarrollada. Es importante clarificar la definición de "Hecho" con el equipo de trabajo. "Hecho" no sólo incluye el desarrollo sino la integración y pruebas integrales del Software. Una historia hecha puede presentarse al dueño de producto para sus pruebas de aceptación. - Descartada: Se determinó que la historia ya no es relevante, su contenido se incluyó en otro grupo de historias o fue cancelada.
Dimensión / Esfuerzo	Medida del esfuerzo (tamaño) que implica desarrollar la historia, existen distintos métodos para medirlo, un ejemplo es los "puntos de historia" una medida de complejidad no necesariamente relacionado con jornadas o días. Otra forma de medirlo es con días o jornadas ideales.
Iteración (Sprint)	Iteración o Sprint al que se asigna la historia. Esta asignación puede cambiar en cada iteración donde se haga la revisión de la pila de producto (ProductBacklogReview), según las prioridades indicadas por el dueño de producto. Por medio de este campo se puede crear un "Plan de Salidas a Productivo" (Release Plan).
Prioridad	Seguendo el marco de trabajo ágil y Scrum, se le deben asignar prioridades a las historias, según las instrucciones del dueño de producto (ProductOwner). De esta forma pueden ordenarse. Las historias de mayor prioridad deben ser las que agregan más valor al negocio, y deben ser originadas en sus necesidades.
Comentarios	Comentarios o detalles relacionadas que expliquen la historia. Para definiciones de mayor longitud deben usarse documentos externos, por ejemplo la plantilla de historias de usuario.

Historias técnicas:

HT-01	Configuración del repositorio para proyecto
<p>Descripción: Se creará un proyecto compartido en GitHub con los colaboradores desarrolladores del sistema para tener acceso a las versiones del proyecto a través de ramas individuales, además, se podrá tener una facilidad para generar una documentación. Esta configuración nos permitirá desplegar el proyecto tanto en desarrollo como en producción</p>	
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none">• Cuando se aloje el código y se pueda comprobar su existencia• Cuando se pueda desplegar el código en los servidores	
<p>Tareas:</p> <ul style="list-style-type: none">• Configurar los servidores con ramas específicas para generar una integración entre ellos	

Dependencias: <ul style="list-style-type: none"> • Servidores • Extensiones para generación de documentación 	
Estimación de Esfuerzo: 3 horas	Prioridad: Alta

HT-02	Modelamiento de la base de datos
<p>Descripción: El objetivo de esta tarea es llevar a cabo el modelamiento de la base de datos del sistema. El modelamiento de la base de datos es crucial para garantizar una estructura eficiente y coherente que respalde las funcionalidades del sistema. Se espera que el modelo de datos refleje con precisión los requisitos del negocio y facilite la manipulación de la información de manera eficiente.</p>	
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> • Se ha creado un modelo de datos que refleja con precisión la estructura y las relaciones entre las entidades del sistema. • El modelo de datos ha sido revisado y validado por el equipo de desarrollo. • Se han documentado las decisiones de diseño y cualquier consideración importante relacionada con el modelo de datos. 	
<p>Tareas:</p> <ul style="list-style-type: none"> • Análisis de Requisitos: Revisar detalladamente los requisitos del sistema para identificar las entidades y relaciones clave que deben estar presentes en el modelo de datos. • Diseño del Modelo: Utilizar herramientas de modelado de bases de datos para diseñar un modelo que cumpla con los requisitos identificados. • Revisión y Validación: Colaborar con el equipo de desarrollo para revisar y validar el modelo de datos propuesto. • Documentación: Documentar de manera completa y clara el modelo de datos, incluyendo decisiones de diseño y cualquier consideración relevante. 	
<p>Dependencias:</p> <ul style="list-style-type: none"> • Disponibilidad de los requisitos del sistema. • Software de modelamiento. • Colaboración activa con el equipo de desarrollo para validar el modelo propuesto. 	
Estimación de Esfuerzo: 10 horas	Prioridad: Alto

HT-03	Configuración de un servidor de base de datos
<p>Descripción: Crear y configurar un servidor de base de datos mediante las herramientas de Docker o Railway, con el fin de agilizar el proceso de desarrollo y mantener actualizado los datos del proyecto.</p>	
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> Realizar la conexión la base de datos mediante credenciales de para autorizar el acceso a cada desarrollador del equipo Realizar una consulta de la versión de la base de datos 	
<p>Tareas:</p> <ul style="list-style-type: none"> Configurar el servidor Realizar una conexión con la base de datos 	
<p>Dependencias:</p> <ul style="list-style-type: none"> Servidores Gestor de base de datos 	
Estimación de Esfuerzo: 2 horas	Prioridad: Alta

HT-04	Definición de la Arquitectura del Sistema
<p>Descripción: La tarea de Definición de la Arquitectura del Sistema tiene como objetivo establecer la estructura y los componentes principales del sistema. Esto implica tomar decisiones clave sobre la organización de los módulos, la gestión de datos, la comunicación entre componentes y la selección de tecnologías fundamentales. La arquitectura del sistema sienta las bases para un desarrollo eficiente y escalable, asegurando que el sistema sea robusto y cumpla con los requisitos funcionales y no funcionales establecidos.</p>	
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> Se ha definido una arquitectura del sistema que cumple con los requisitos funcionales y no funcionales del proyecto. Se han identificado y justificado los patrones de diseño y las tecnologías utilizadas en la arquitectura. La arquitectura del sistema ha sido revisada y aprobada por el equipo de desarrollo del proyecto. 	
<p>Tareas:</p> <ul style="list-style-type: none"> Análisis de Requisitos: Revisar detalladamente los requisitos del sistema para comprender los aspectos clave que deben ser considerados en la arquitectura. Definición de Componentes: Identificar los principales componentes del sistema y definir sus responsabilidades y funciones. 	

<ul style="list-style-type: none"> • Selección de Tecnologías: Evaluar y seleccionar las tecnologías adecuadas para implementar los componentes identificados, considerando aspectos de rendimiento, seguridad y mantenimiento. • Diseño de la Interfaz: Especificar cómo los componentes del sistema interactuarán entre sí, incluyendo la definición de interfaces y protocolos de comunicación. • Modularidad y Reutilización: Diseñar la arquitectura de manera modular para facilitar la mantenibilidad y fomentar la reutilización de componentes. 	
Dependencias: <ul style="list-style-type: none"> • Disponibilidad de requisitos funcionales y no funcionales. • Colaboración activa con el equipo de desarrollo y los stakeholders para validar la arquitectura propuesta. 	
Estimación de Esfuerzo: 3 horas	Prioridad: Alta

HT-05	Implementación de Seguridad JWT
Descripción: La tarea de Implementación de Seguridad JWT tiene como objetivo integrar un mecanismo de seguridad basado en JSON Web Tokens (JWT) en el sistema. La utilización de JWT proporcionará una capa adicional de seguridad para la autenticación y autorización de usuarios, asegurando la integridad de los datos transmitidos y validando la identidad de los usuarios que acceden al sistema.	
Criterios de Aceptación: <ul style="list-style-type: none"> • Se ha implementado correctamente el proceso de generación y firma de JWT para la autenticación de usuarios. • La verificación y decodificación de JWT se ha incorporado de manera efectiva en el sistema para garantizar la integridad de la información. • Se han establecido mecanismos adecuados para la gestión de tokens, incluyendo la expiración y la renovación cuando sea necesario. 	
Tareas: <ul style="list-style-type: none"> • Integración de la Biblioteca JWT: Seleccionar y integrar una biblioteca de JWT en el sistema para facilitar la generación, firma y verificación de tokens. • Configuración de Parámetros de Seguridad: Configurar los parámetros de seguridad adecuados, como algoritmos de cifrado y configuración de tiempo de expiración de los tokens. • Validación de JWT en Solicitudes: Integrar la verificación y decodificación de JWT en las solicitudes entrantes para garantizar la autenticidad del usuario. • Gestión de Tokens: Implementar mecanismos para gestionar la expiración de los tokens y renovarlos según sea necesario. 	

Dependencias: <ul style="list-style-type: none"> Definición clara de los requisitos de seguridad. Integración con el sistema de autenticación existente. 	
Estimación de Esfuerzo: 5 horas	Prioridad: Alta

HT-06	Implementación de la Capa de Presentación (Interfaz de Usuario)
Descripción: La tarea de Implementación de la Capa de Presentación tiene como objetivo llevar a cabo el desarrollo de la interfaz de usuario del sistema. Esta capa es fundamental para proporcionar una experiencia de usuario intuitiva y eficiente, cumpliendo con los requisitos de diseño y brindando una interfaz atractiva y funcional.	
Criterios de Aceptación: <ul style="list-style-type: none"> La interfaz de usuario se ha implementado de acuerdo con las especificaciones de diseño y los requisitos funcionales establecidos. La navegación es clara y la interacción con los elementos de la interfaz es intuitiva. La capa de presentación se integra adecuadamente con las capas subyacentes del sistema. 	
Tareas: <ul style="list-style-type: none"> Desarrollo de Vistas: Implementar las diferentes vistas de la interfaz de usuario según las especificaciones de diseño, asegurando la consistencia visual. Interactividad: Incorporar interactividad a la interfaz mediante la implementación de eventos y animaciones que mejoren la experiencia del usuario. Integración con Backend: Conectar la capa de presentación con las funcionalidades del backend, asegurando una comunicación eficiente y segura. Documentación: Documentar la estructura y el funcionamiento de la capa de presentación para facilitar futuras actualizaciones y mantenimiento. 	
Dependencias: <ul style="list-style-type: none"> Especificaciones detalladas de diseño de interfaz de usuario. Disponibilidad de las funcionalidades del backend. 	
Estimación de Esfuerzo: 10 horas	Prioridad: Alta

HT-07	Implementación de la capa Lógica de Negocio
<p>Descripción: Configurar e implementar la capa Lógica de Negocio, que es fundamental para el funcionamiento de software, ya que se encarga de procesar y gestionar la lógica específica del dominio o sector al que pertenece la aplicación.</p>	
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> Definición de Reglas de Negocio: Analizar y definir las reglas de negocio que guiarán el comportamiento del sistema. Implementación de Funcionalidades: Desarrollar las funciones y operaciones principales del sistema de acuerdo con las reglas de negocio establecidas. Validación de Datos: Implementar mecanismos de validación de datos para garantizar la integridad y consistencia de la información. 	
<p>Tareas:</p> <ul style="list-style-type: none"> Análisis detallado de los requisitos y especificaciones de la lógica del negocio. Diseño y planificación de la estructura y funcionalidad de la capa de lógica del negocio. Implementación de algoritmos y reglas del negocio específicas del dominio Realización de pruebas de integración para garantizar la cohesión entre las distintas capas 	
<p>Dependencias:</p> <ul style="list-style-type: none"> Implementación de capa de datos Documentación de la arquitectura 	
Estimación de Esfuerzo: 10 horas	Prioridad: Alta

HT-08	Capa de Acceso a Datos (Persistencia de Datos)
<p>Descripción: La tarea de Implementación de la Capa de Acceso a Datos tiene como objetivo desarrollar la lógica necesaria para interactuar con la base de datos del sistema. Esta capa es esencial para garantizar la persistencia y recuperación eficiente de la información, así como para mantener la integridad de los datos almacenados.</p>	
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> Se ha implementado la capa de acceso a datos de acuerdo con el modelo de datos y los requisitos de persistencia del sistema. Las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) se han implementado de manera eficiente y segura. La capa de acceso a datos se integra adecuadamente con las capas de lógica de negocio y presentación. 	

Tareas: <ul style="list-style-type: none"> • Diseño de la Capa de Acceso a Datos: Definir la estructura y las operaciones necesarias para interactuar con la base de datos. • Implementación de Operaciones CRUD: Desarrollar las funciones necesarias para crear, leer, actualizar y eliminar datos en la base de datos. • Gestión de Conexiones: Implementar mecanismos eficientes de gestión de conexiones a la base de datos para optimizar el rendimiento. • Seguridad de Datos: Aplicar medidas de seguridad, como la validación de datos y la prevención de ataques de inyección, para proteger la integridad de la base de datos. 	
Dependencias: <ul style="list-style-type: none"> • Modelo de datos definido y validado. • Disponibilidad y configuración de la base de datos. 	
Estimación de Esfuerzo: 10 horas	Prioridad: Alta

HT-09	Diseño responsivo
Descripción: Implementar el diseño responsivo para garantizar una experiencia de usuario optima en una variedad de dispositivos y tamaños de pantalla. Esencial para mejorar accesibilidad, retención de usuarios y satisfacción general del usuario.	
Criterios de Aceptación: <ul style="list-style-type: none"> • Utilizar emuladores de dispositivos en navegadores web o herramientas paraprevisualizar la adaptabilidad en diferentes dispositivos • Adaptarse diferentes resoluciones de pantalla • Consistencia 	
Tareas: <ul style="list-style-type: none"> • Maquetación Responsiva: Adaptar o diseñar las vistas y componentes de la interfaz para que respondan de manera adecuada a diferentes tamaños de pantalla. • Optimización de Imágenes: Optimizar el manejo de imágenes y recursos multimedia para mejorar el rendimiento en dispositivos móviles. • 	
Dependencias: <ul style="list-style-type: none"> • Integración del frontend 	

Estimación de Esfuerzo: 12 horas	Prioridad: Alta
---	------------------------

HT - 10	Implementación de mejoras en la documentación del código fuente
<p>Descripción: Tiene el objetivo de mejorar la calidad y la accesibilidad del código fuente de la aplicación, para que los desarrolladores comprendan rápidamente la lógica y el propósito de cada componente.</p>	
<p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> • Completa cobertura: la documentación debe cubrir todas las clases y métodos esenciales del código fuente. • Formato consistente: se debe mantener un formato consistente para la documentación, utilizando convenciones claras y fácilmente comprensibles. 	
<p>Tareas:</p> <ul style="list-style-type: none"> • Identificación de áreas críticas: identificar partes del código que requieran una documentación más detallada • Mejora de comentarios en el código: Agregar o mejorar los comentarios en el código fuente para facilitar la comprensión • Revisión de la documentación actual: evaluar la calidad y cobertura de la documentación existente en el código fuente. 	
<p>Dependencias:</p> <ul style="list-style-type: none"> • Acceso directo al código fuente: se debe tener acceso al código fuente de la aplicación. • Colaboración con el equipo de desarrolladores: coordinación con el equipo de desarrolladores para comprender mejor las áreas críticas que requieren una documentación más detallada 	
Estimación de Esfuerzo: 7 horas	Prioridad: Alta