

Especificación de requisitos de software

Proyecto:

“Connect Salud”



**Sitio Web de Cuidado de la Salud
Monitoreo de Peso, Información sobre hábitos
saludables, recetas y venta de planes
alimentarios.**

Creadores:

**Montenegro, Jonathan
Coronel, Ediberto
Sanchez, Samira
Mingorance Pico, Desirée**

2023

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. Calidad.
2023			

Contenido

FICHA DEL DOCUMENTO	2
CONTENIDO	
1 INTRODUCCIÓN	4
1.1 Propósito	4
1.2 Alcance	4
1.3 Personal involucrado	5
1.4 Definiciones, acrónimos y abreviaturas	6
1.5 Referencias	6
1.6 Resumen	6
2 DESCRIPCIÓN GENERAL	7
2.1 Perspectiva del producto	7
2.3 Características de los usuarios	7
2.4 Restricciones	8
3 REQUISITOS ESPECÍFICOS	8
3.2 Product Backlog	9
3.2.1 Product Backlog	9
3.3 Sprints	9
3.3.1 Sprint 0	9
3.3.2 Sprint 1	10
3.3.3 Sprint 2	12
3.3.4 Sprint 3	14
3.3.5 Sprint 4	16

Introducción

A continuación presentamos el Proyecto denominado “**Connect Salud**” como entrega del módulo Programador Web de la Tecnicatura Superior en Diseño Web y Aplicaciones Digitales.

1.1 Propósito

El proyecto consiste en el desarrollo de un sitio web para el cuidado de la Salud de las personas. El objetivo del mismo es brindar información sobre hábitos, cuidados y receta saludables, acceso a compra de planes alimentarios adecuados para cada necesidad y además proporciona un control de monitoreo de peso para uso del usuario.

1.2 Alcance

El sitio se denominará “**Connect Salud**”. Su plataforma es una web accesible desde cualquier navegador por Internet y tiene información referente al cuidado de la salud de las personas. Se brindará información completa para que el usuario encuentre todo lo necesario para llevar una vida saludable y de buenos hábitos. Encontramos dos tipologías de usuarios, están los **usuarios no registrados**, que únicamente pueden acceder a información general sobre el sitio y navegar en el mismo. Por otro lado, están los **usuarios registrados**, que accederán a una mayor funcionalidad dentro de la web, pueden realizar acciones tales como solicitar turnos a nutricionistas, hacer compras de planes alimentarios, controlar su peso, consultar recetas, etc., mientras que el usuario administrador gestionará la base de datos y el sistema.

1.3 Personal involucrado

Nombre	Sanchez, Samira
Rol	Desarrollador y Administrador
Categoría Profesional	Informático
Responsabilidad	Documentar, desarrollar el Frontend y Backend
Información de contacto	samsanchez99@icloud.com

Nombre	Montenegro, Jonathan
Rol	Desarrollador y Administrador
Categoría Profesional	Informático
Responsabilidad	Documentar, desarrollar el Frontend y Backend
Información de contacto	mrmontenegro@gmail.com

Nombre	Coronel, Ediberto
Rol	Desarrollador y Administrador
Categoría Profesional	Informático
Responsabilidad	Documentar, desarrollar el Frontend y Backend
Información de contacto	edibertocoronel@gmail.com

Nombre	Mingorance Pico, Desirée
Rol	Desarrollador y Administrador
Categoría Profesional	Informático
Responsabilidad	Documentar, desarrollar el Frontend y Backend
Información de contacto	desimingo15@gmail.com

1.4 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema a través de un sitio web.
Administrador	Personas encargadas de desarrollar el Frontend y Backend de la web.
Gestor de Contenido	Encargado de Gestionar contenido e información al sitio web creado.
SIS-I	Sistema de Información Web para la Gestión de Procesos Administrativos y Académicos
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
FTP	Protocolo de Transferencia de Archivos
Moodle	Aula Virtual

1.5 Referencias

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

1.6 Resumen

Se describen a continuación las secciones de este trabajo que conforman el proceso de desarrollo del proyecto.

En la **Introducción** se determinan los distintos alcances y propósitos del mismo.

En **Descripción General** se detalla la funcionalidad del sitio con los requisitos necesarios para acceder. El cliente vuelca todas sus necesidades al desarrollador en cuanto a la finalidad del sitio web.

En **Requisitos Específicos**: Se realizó el modelado conceptual de la futura solución mediante el uso de **Sprint Backlog**. Los Sprint ayudan a visualizar cómo es el sistema, proporcionando tareas que sirven de guía en la construcción del sitio web **Connect Salud**.

Se especifica aquí qué debe hacer cada desarrollador en el sitio web pero no cómo debe hacerlo.

2 Descripción general

2.1 Perspectiva del producto

El sitio web Connect Salud permite el acceso a información de una manera práctica y rápida desde cualquier lugar a través de un navegador.

El objetivo es facilitar el acceso a información sobre hábitos saludables y de cuidado de la salud para todas las personas. El sitio Web permite además monitorear el peso del usuario, mediante tablas de IMC, acceder a un turnero para atención médica de nutricionistas, cuenta con un espacio de recetas saludables y novedades en el cuidado de la salud. Además, permite comprar planes alimentarios adecuados a la necesidad del consumidor.

2.2 Características de los usuarios

Tipo de usuario	Administrador
Formación	Manejo de herramientas informáticas
Actividades	Control y manejo del sistema en general

Tipo de usuario	Visitante
Formación	Manejo de entornos web
Actividades	Observa e indaga información del sitio web

Tipo de usuario	Usuario Registrado
Formación	Manejo de entornos web
Actividades	Observa e indaga información del sitio web, accede mediante el registro a la función de monitoreo de peso, recetas saludables y puede realizar compras.

Tipo de usuario	Gestor de Contenido
Formación	Manejo de herramientas web
Actividades	Sube y crea información para el contenido del sitio. Controla y gestiona el contenido creado (imágenes, textos, etc)

2.3 Restricciones

- El sitio web Connect Salud requiere de internet para ser utilizado.
 - Lenguajes y tecnologías en uso: HTML5, CSS3, SQL y JavaScript.
 - Se considerarán las políticas de una empresa estándar en el rubro Cuidado de la Salud.
 - Limitaciones de hardware: Los dispositivos que utilicen la página web van a necesitar si o si de un navegador web.
 - Consideraciones de Seguridad:
 - * Los usuarios que realicen pagos por las compras serán redirigidos a una web especializada como Mercado Pago.
 - *En el registro de usuarios poner como requisito contraseñas fuertes.
- Al dar el alta el usuario tendrá que introducir un código de autenticación que llegará al mail.

3 Requisitos específicos E-commerce

Product Backlog

#US01 Como usuario quiero ingresar al carrito para poder comprar Planes de Alimentación.

#US02 Como usuario quiero que tenga distintos medios de pago.

#US03 Como usuario quiero que el carrito de compra quede guardado en caso de no realizarla en el momento a la compra.

#US04 Como usuario quiero quede el historial de compra realizada.

#US05 Como usuario quiero poder cancelar la compra realizada.

#US06 Como usuario quiero que al comprar se abra una pestaña de compra.

#US07 Como usuario quiero que al realizar la compra envíe un mail con el detalle de compra al comprador logueado"

"#US08 Como usuario quiero que al realizar la compra tenga el icono de pdf y al hacer click descargue comprobante de compra con todo el detalle de la misma.

Sprints.

N° de sprint	00
Sprint Backlog	<ul style="list-style-type: none"> • Registrar la Especificación de Requerimientos mediante la documentación IEEE830 • Git/GitHub : continuar con el mismo repositorio • Crear Project estilo Kanban con incorporación de Historias de Usuarios, tareas, e incidencias. • Llevar ANÁLISIS • Definir requerimientos para el nuevo módulo a desarrollar e-commerce (colocarlos en el Product Backlog del Project), a su vez revisar si han cumplimentado todos los requerimientos previos, realizando mejoras del mismo • Plantear Historias de Usuarios y Tareas dependientes de las US para incorporarlas en el repositorio remoto GitHub. (Issues y Milestones) - Tener en cuenta la redacción adecuada para las US y nomenclatura, ej "#US01 Como usuario quiero ingresar al carrito para poder comprar" • Definir tareas dentro de las Historias de Usuario (GITHUB) ej dentro de las ISSUES #TK01 revisar IEEE830..
Responsabilidades	<p>Todas las ramas Plantean Historias de Usuarios y Tareas dependientes de las US para incorporarlas en el repositorio remoto GitHub. (Issues y Milestones) –</p> <p>Se designan Issues y Milestones.</p> <p>Se definen tareas dentro de las Historias de Usuario .</p> <p>#TK07 Crear base de datos de compra realizada y enlace para el envío por mail al usuario.</p> <p>#TK08 Crear icono de PDF y base de datos para descargar comprobante de compra.</p> <p>#TK01 AÑADIR CARRITO DE COMPRA EN PÁGINA DE PLANES ALIMENTARIOS</p>
Calendario	Fecha Inicio = 10/04/2023 - Fecha de Fin = 21/04/2023
Inconvenientes:	

N° de sprint	01
Sprint Backlog	<p>Convertir los archivos .html del proyecto previo, en una SPA en Angular con módulos (Layouts/Shared y Pages son un ejemplo) con sus components correspondientes. (HOME, REGISTRO, LOGIN, DASHBOARD)</p> <p>Crear módulos y componentes para la tienda virtual o ecommerce. (PRODUCTO o SERVICIOS, PRODUCTO INDIVIDUAL, COMPRA).</p> <p>Incorporar la navegabilidad de la aplicación mediante Routing con Angular.</p> <p>Diseño del BACKEND</p> <p>Subir en la carpeta documentación las tablas actualizadas de la base de datos:</p> <p>Diseño conceptual al que ya hemos visto como DER: Diagrama Entidad-Relación.</p> <p>Diseño Lógico que consta de la Normalización y el Modelo Relacional.</p> <p>Diseño Físico que es el script legible de la DB en MySQL.</p> <p>Crear un Diagrama de Clases y Casos de Uso para facilitar el modelado en POO.</p> <p>GESTION DE PROYECTO</p> <p>Subir al repo grupal en GitHub, mediante comandos GIT desde el local, pueden crearla en una branch por desarrollador para que cada uno tenga una copia, o directamente crear una branch FEATURE, para actualizar los cambios que no se encuentran aún en condiciones de incorporarse a la branch DEVELOP.</p> <p>En GitHub, mediante la sección WIKI deben reflejar los avances individuales y de equipo en relación a Scrum (Ceremonias, roles, historias de usuarios, etc). Actualizar IEEE830.</p> <p>"FEATURE" son las branches en las que se trabaja la mayor parte del tiempo. Cuando no se utilizan más se borran.</p> <p>"DEVELOP" es la branch donde se mantiene la versión de desarrollo más estable y reciente. La que es candidata para pasar a ser la principal (MAIN).</p> <p>La branch "MAIN" o "MASTER" es la que suele usarse para mantener las versiones estables (productivas) de nuestros proyectos. Listas para salir a producción. Se supone que en cada cierre de Sprint, queda lo solicitado en esa condición, por lo que: Para el cierre de cada SPRINT si deberá realizarse un MERGE a la</p>

	rama MAIN/MASTER como constancia que está completo dicho Sprint para la Review (en producción)
Responsabilidades	<p>Se asignan tareas a las ramas:</p> <p>Jhonatan Montenegro:</p> <p>Samira sanchez:</p> <p>Ediberto:</p> <p>Desimingo:</p> <p>Micaela Galeano</p> <p>Adan Reynoso 702</p> <p>#TK01-1: Incorporar la navegabilidad de la aplicación mediante Routing con Angular.</p> <p>Diseño del BACKEND: Crear un Diagrama de Clases y Casos de Uso para facilitar el modelado en POO.</p> <p>Diseño del BACKEND: realizar Diseño Físico que es el script legible de la DB en MySQL.</p> <p>Diseño del BACKEND: realizar Diseño Lógico que consta de la Normalización y el Modelo Relacional.</p> <p>Diseño del BACKEND: realizar Diseño conceptual como DER: Diagrama Entidad-Relación.</p> <p>Implementación FRONTEND: Crear módulos y componentes para la tienda virtual o ecommerce.</p>
Calendario	Fecha Inicio = 24/04/2023 - Fecha de Fin = 07/05/2023
Inconvenientes:	

N° de sprint	02
Sprint Backlog	<p>Frontend</p> <p>El goal de este sprint es construir un frontend dinámico. Por ello, se requiere crear servicios que provean los datos a los componentes. Para lograrlo, los servicios deben consumir una API Rest de prueba</p> <p>Se requiere:</p> <ul style="list-style-type: none"> • Crear servicios que consuman un API Rest de prueba. Para realizar las peticiones http, utilizar el servicio de HttpClient. • Inyectar a los componentes los correspondientes servicios a fin de que los componentes puedan acceder a los datos y mostrarlos en pantalla. Incluye enlace de datos (Data binding). • Incorporar la programación reactiva (asincrónica) mediante observables para la comunicación entre servicios y componentes. <p>Backend</p> <ul style="list-style-type: none"> • Establecer el modelo de base de datos del primer sprint en Django usando clases. • Establecer el sistema de usuarios en Django-Admin (registro, login y roles) • Crear un ABM de productos/servicios utilizando el panel de control de Django.
Responsabilidades	<p>Las Ramas:</p> <p>Jonathan Montenegro Samira Sanchez Ediberto_nueva Desimingo</p> <p>Se encargarán de trabajar en equipo:</p> <p>#TK04-1: Crear un ABM de productos/servicios utilizando el panel de control de Django.</p> <p>#TK01-5: Establecer el sistema de usuarios en Django-Admin (registro, login y roles)</p> <p>#TK03-1: Establecer el modelo de base de datos del primer sprint en Django usando clases.</p> <p>#TK01-4: Incorporar la programación reactiva (asincrónica)</p>

	<p>mediante observables para la comunicación entre servicios y componentes.</p> <p>#TK01-3: Inyectar a los componentes los correspondientes servicios a fin de que los componentes puedan acceder a los datos y mostrarlos en pantalla. Incluye enlace de datos (Data binding).</p> <p>#TK01-2: Crear servicios que consuman un API Rest de prueba. Para realizar las peticiones http, utilizar el servicio de HttpClient.</p>
Calendario	<p>Fecha Inicio = 08/05/2023 - Fecha de Fin = 21/05/2023</p>
Inconvenientes:	

N° de sprint	03
Sprint Backlog	<p>Frontend:</p> <ul style="list-style-type: none"> • Incorporar formularios reactivos de Registro y Login para la autenticación y formularios para la manipulación (Alta) de servicios/productos/suscripción, según el caso con sus respectivas validaciones. • Conexión con el backend para el registro e inicio de sesión. • Conexión con backend para almacenar los datos de los productos/servicios y detalles de producto/servicios en la DB a través del modelo en Django. • Opcional: Agregar rutas protegidas a la navegación en base al usuario autenticado/autorizado. Ej. para acceder al tablero de control o bien, para acceder al registro de productos/servicios. <p>Backend</p> <p>Definición de la Api Rest y los endpoints necesarios para las funcionalidades de:</p> <ul style="list-style-type: none"> • Registro de usuarios. • Login de usuarios. • Registro / Alta de al menos 3 productos/servicios/suscripción (solo desde el rol Administrador, mediante su panel de control). • Tablero de control (Gestión) o Dashboard con módulos planteados en la maquetación del FrontEnd según proyecto: <ul style="list-style-type: none"> ◦ El rol ADMIN pueda ver la lista de usuarios registrados y el listado de los productos/servicios/suscripción según corresponda. ◦ El rol USUARIO registrado solo que pueda ver su Dashboard para editar su perfil. • Los datos serán consumidos desde la Base de datos MySQL. <p>OPCIONAL: Token validación del login con JWT.</p>
Responsabilidades	<p>Ramas:</p> <p>Jonathan Montenegro:</p>

	<p>#TK01-6: Formulario reactivo de Registro y Login para autenticación.</p> <p>TK06-1: conexión de backend para registro e inicio de sesión.</p> <p>Samira Sanchez:</p> <p>#TK01-7: Formularios para manipulación (Alta) de productos</p> <p>#TK04-2: Componente Detalle de Planes de alimentación.</p> <p>Ediberto_nueva:</p> <p>#TK01-8: Registro y Login de usuarios (funcionalidad con la definición de la Api Rest y endpoints)</p> <p>#TK01-9: Crear Dashboard rol ADMIN.</p> <p>#TK01-10: Dashboard rol Usuario</p> <p>Desimingo:</p> <p>#TK06-2: Detalle de producto y Formulario de Productos.</p> <p>#TK03-1: Creación de Formulario para agrega y eliminar productos.</p>
Calendario	<p>Fecha Inicio = 22/05/2023 - Fecha de Fin = 04/06/2023</p>
Inconvenientes:	

N° de sprint	04
Sprint Backlog	<p>Requerimientos Incremento del producto: Funcionalidades completas (front + back integrado)</p> <ul style="list-style-type: none"> • <i>Tablero de control para usuarios:</i> Mostrar catálogo (mínimo 3 productos o servicios o suscripción). Seleccionar para comprar. Simular Pago con API Mercado Pago. • <i>Tablero de control administrador CRUD</i> básico funcional de producto y cliente. • Actualizar documentación IEEE830 - tablero Kanban (project) y Wiki. • Actualizado todo el proyecto en GitHub como respaldo • <i>Demo final</i> en video con participación de todo el equipo describiendo todas las funcionalidades. (quien no explique en el mismo alguna de las funcionalidades/código/gestión del proyecto, no será tenido en cuenta para calificación) • OPCIONAL: Sitio publicado en Servidor
Responsabilidades	<p>Ramas: Jonathan Montenegro Samira Sanchez Ediberto_nueva Desimingo Se encargarán en equipo de trabajar: Tablero de control administrador CRUD básico funcional de producto y cliente. Tablero de control para usuarios: Mostrar catálogo (mínimo 3 productos o servicios o suscripción). Seleccionar para comprar. Simular Pago con API Mercado Pago</p>
Calendario	Fecha Inicio = 05/06/2023 - Fecha de Fin = 18/06/2023
Inconvenientes:	