



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PROYECTO DE GRADUACIÓN

**PLATAFORMA WEB PARA LA VENTA EN LÍNEA DE PLANES DE ENTRENAMIENTO Y
DE ALIMENTACIÓN**

PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTADO POR:

11641402

KAREN MICHELLE LÓPEZ MELÉNDEZ

ASESOR: LICDA. TANIA LUCILA MEZA AMADOR

CAMPUS TEGUCIGALPA; ABRIL, 2021

RESUMEN EJECUTIVO

El presente informe se entrega como requisito para optar al título universitario de Ingeniero en Sistemas Computacionales, el cual fue desarrollado en el periodo de octubre de 2020 a marzo de 2021.

El proyecto se realizó para la clínica nutricionista Lucía Escobar, la cual se encarga de tratar a personas con enfermedades en las que la alimentación adecuada y el ejercicio son esenciales, personas que desean prevenir enfermedades o que quieran mantener un estilo de vida saludable. La clínica provee servicios de consultas, creación de menús alimenticios y planes de entrenamiento físico.

El proyecto permitirá a la clínica, vender planes de alimentación y de entrenamiento a sus pacientes, llevar un control de estos, vender libros y demás productos digitales que ofrecen, únicamente a través de la plataforma web sin interacción física del paciente con el nutricionista. Aplicando la experiencia de usuario para brindar a los pacientes una interfaz agradable visualmente y de uso fácil e intuitivo.

Para el desarrollo del sistema se hizo uso de tecnologías como Python, Django, SQLite y APIs para el backend de la aplicación, así como React, NextJS y Redux para el frontend de la misma.

EXECUTIVE SUMMARY

The objective of this research is to be delivered as a requirement to qualify for the university degree in Computer Systems Engineering, which was developed in the period from October 2020 to March 2021.

The project was carried out for the Lucía Escobar nutritionist clinic, which treats people with diseases in which proper nutrition and exercise are essential, people who want to prevent diseases or who want to maintain a healthy lifestyle. The clinic provides consultation services, creation of food menus and physical training plans.

The project allowed the clinic to sell food and training plans to their patients, keep track of them, sell books and other digital products they offer, only through the web platform without physical interaction between the patient and the nutritionist. Applying the user experience to provide patients with a visually pleasing, user-friendly and intuitive interface.

For the development of the system, technologies such as Python, Django, SQLite and APIs were used for the application's backend, as well as React, NextJS and Redux for its frontend.

ÍNDICE DE CONTENIDO

I.	Introducción	1
II.	Planteamiento del Problema	2
	2.1. Precedentes del Problema	2
	2.2. Definición del Problema	2
	2.3. Justificación	2
	2.4. Preguntas de Investigación	3
	2.5. Objetivos	3
	2.5.1. Objetivo General	3
	2.5.2. Objetivos Específicos	3
III.	Marco Teórico	4
IV.	Metodología	8
	4.1. Enfoque	8
	4.2. Variables de Investigación	8
	4.3. Técnicas e Instrumentos Aplicados	8
	4.3.1. Casos de Uso	8
	4.3.2. Mockups	10
	4.3.3. Diseño de Base de Datos	14
	4.3.4. React	15
	4.3.5. Django y Python	15
	4.3.6. GitHub	16
	4.3.7. Digital Ocean	16
	4.3.8. Arquitectura Web	16

4.3.9. Interfaz de Programación de Aplicaciones (API) de Transferencia de Estado Representacional (REST).....	16
4.3.10 Modelo de Base de Datos Implementado	16
4.3.11. Administrador Django.....	18
4.4 Metodología de Estudio.....	18
4.4.1. Modelo Ágil.....	18
4.5 Cronograma de Trabajo	18
V. Resultados y Análisis.....	20
5.1 Propuesta de Interfaz de Usuario.....	20
5.2 API REST	21
5.3 Capturas de la Aplicación.....	24
VI. Conclusiones.....	33
VII. Recomendaciones.....	34
Bibliografía.....	35

ÍNDICE DE ILUSTRACIONES

Ilustración 1 – Diagrama de caso de uso para usuario paciente	9
Ilustración 2 – Diagrama de caso de uso para usuario nutricionista	10
Ilustración 3 – Mockup de landing page	11
Ilustración 4 – Mockup de planes	12
Ilustración 5 – Mockup de productos	13
Ilustración 6 – Mockup de inicio de sesión	13
Ilustración 7 – Mockup de registro de pacientes	14
Ilustración 8 – Diagrama de clases	15
Ilustración 9 – Diagrama de arquitectura Web	16
Ilustración 10 – Modelo final de la base de datos	17
Ilustración 11 – Consola de Administración de Django	18
Ilustración 12 – Landing page de plantilla	20
Ilustración 13 – Plantilla de tienda en línea para exhibición y compra de productos	21
Ilustración 14 – API Planes	22
Ilustración 15 – Login de la aplicación	25
Ilustración 16 – Lista de planes y productos que tiene asignado un usuario	25
Ilustración 17 – Landing Page	26
Ilustración 18 – Gestión de productos	26
Ilustración 19 – Listado de productos	27
Ilustración 20 – Listado de Planes	28
Ilustración 21 – Detalle de producto	29
Ilustración 22 – Carrito de compras	29
Ilustración 23 – Detalles de la compra	30
Ilustración 24 – Visualización y control de planes	30
Ilustración 25 – Blog	31
Ilustración 26 – Gráfica de Vercel	32

ÍNDICE DE TABLAS

Tabla 1 – Cronograma de Trabajo Proyecto Fase I	17
Tabla 2 – Cronograma de Trabajo Proyecto Fase II	18
Tabla 3 – Descripción de cada API utilizada en la aplicación	24

LISTA DE SIGLAS Y GLOSARIO

API	Application Programming Interface
CSS	Cascading Style Sheets
ECNT	Enfermedades Crónicas No Transmisibles
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
OWASP	Open Web Application Security Project
PaaS	Platform as a Service
REST	Transferencia de estado representacional
SQL	Structured Query Language
WSTG	Web Security Testing Guide

REST: o RESTful es un conjunto de principios que sirven para transferir recursos entre cliente y servidor en distintos formatos a través de solicitudes HTTP. (RedHat, 2020).

Droplet: es una máquina virtual escalable que provee flexibilidad para alojar, probar y hacer crecer una aplicación Web. (Digital Ocean, 2020).

I. INTRODUCCIÓN

“La única forma de mantener nuestra salud es comer lo que no queremos comer, beber lo que no nos gusta y hacer lo que preferiríamos no hacer” (Twain, 1998). La buena alimentación y el ejercicio físico son factores fundamentales para mantener la salud y el bienestar físico y mental de las personas, y apoyan a la disminución de riesgo de padecer de enfermedades crónicas no transmisibles (ECNT) como diabetes, hipertensión, obesidad, entre otras.

Según el perfil nutricional de Honduras “el 26.9% de las mujeres y el 15.6% de los hombres padecen de obesidad” (Global Nutrition, 2019) porcentajes elevados y preocupantes para la población hondureña. Problemas que aumentan cuando las personas tratan de someterse a dietas o tratamientos no supervisados.

La clínica de la nutricionista Lucía Escobar brinda apoyo profesional a sus pacientes, creando planes de alimentación nutritivos y planes de entrenamiento físico, según las necesidades de cada persona, para mejorar su calidad de vida. Sin embargo, la situación actual del mundo debido a la pandemia de COVID-19, ha presentado un desafío para la clínica y la ha obligado a buscar oportunidades de mejora y llevar a cabo una transformación digital, aplicando la tecnología para mejorar sus procesos y evitar en la medida de lo posible el contacto físico.

Por estas razones se desarrollará un sistema Web, para que los nutricionistas de la clínica estén más cerca de sus pacientes a través de internet. Los nutricionistas llevarán un mejor control de la alimentación y entrenamiento de sus pacientes sin necesidad de tener citas presenciales ya que los pacientes tendrán a su disposición la compra de los planes de entrenamiento y de alimentación, recetarios, libros electrónicos y demás productos provistos por los nutricionistas. De esta manera, el sistema Web, podrá apoyar en el hecho de que, a pesar de los inconvenientes causados por la pandemia, las personas no descuiden su salud.

II. PLANTEAMIENTO DEL PROBLEMA

2.1. PRECEDENTES DEL PROBLEMA

La salud es un factor importante en la vida de una persona, problemas debido a la mala alimentación y el sedentarismo afectan a una gran parte de la población, es por esto por lo que las personas se abocan a nutricionistas para mejorar su estilo de vida y salud, esto requiere un seguimiento constante por parte de los nutricionistas que incluyen una numerosa cantidad de citas presenciales, las cuales hasta inicios del año 2020 se daban con normalidad.

Sin embargo, un problema no esperado surgió, la pandemia de COVID-19 es una situación sin precedentes para la humanidad, debido a esto la mayoría de las empresas se han visto obligadas a buscar formas de llegar a una transformación digital.

Actualmente la Clínica Nutricionista realiza seguimientos de control de alimentación y ejercicio físico de forma presencial, desafortunadamente, estas se han vuelto menos recurrentes debido al cumplimiento de las medidas impuestas por el gobierno de Honduras ante la situación por COVID-19.

2.2. DEFINICIÓN DEL PROBLEMA

Para la Clínica de la Nutricionista Lucía Escobar, la creación y venta de planes nutritivos y de entrenamiento físico y el seguimiento de estos de forma presencial, ya no son tan beneficiosos como lo eran antes de la pandemia, debido a toques de queda, restricciones de circulación diarios y la renuencia de algunas personas a salir a la calle por miedo al contagio.

2.3. JUSTIFICACIÓN

A pesar de todos los inconvenientes provocados por la pandemia de COVID-19, las personas no deben descuidar su salud, por lo que un sistema Web que ayude a mantener una mejor relación entre nutricionista y paciente sería ideal para que las personas no desatiendan su salud durante la situación actual, y representará una mejora del servicio que brinda la clínica incluso cuando las medidas de bioseguridad por la pandemia finalicen ya que incorporará también la venta de recetas, libros electrónicos y otros productos.

2.4. PREGUNTAS DE INVESTIGACIÓN

- ¿Cómo diseñar y desarrollar una plataforma Web que cumpla con las expectativas del cliente?
- ¿Cómo desarrollar una tienda en línea de los planes y productos que ofrece el cliente?
- ¿Cómo dar acceso a los pacientes a los productos comprados?
- ¿Cómo llevar el control de los planes adquiridos por los usuarios?

2.5. OBJETIVOS

2.5.1. OBJETIVO GENERAL

Desarrollar un sistema Web que permitirá a los nutricionistas, vender planes de alimentación y de entrenamiento físico a sus pacientes, llevar un control de estos y vender los demás productos digitales que ofrecen, con el fin de evitar en la medida de lo posible las citas presenciales.

2.5.2. OBJETIVOS ESPECÍFICOS

- Diseñar un sistema amigable e intuitivo para el usuario.
- Desarrollar una tienda en línea de planes alimenticios, planes de entrenamiento físico, libros electrónicos y recetarios para hacer llegar el contenido nutricional a más personas de forma más fácil e inmediata.
- Aplicar controles de seguridad para garantizar que cada usuario tendrá acceso únicamente a los productos por los que pagó.
- Aplicar prácticas de desarrollo seguro según la guía Web Security Testing Guide (WSTG) de Open Web Application Security Project (OWASP), para reducir el riesgo de un ataque de ciberseguridad al sistema.
- Realizar pruebas de seguridad posterior al desarrollo del sistema para validar que se cumpla con el nivel de seguridad adecuado.

III. MARCO TEÓRICO

El desarrollo web es la construcción y mantenimiento de software. Con el fin de lograr crear un sitio web funcional e intuitivo para el usuario, se deben integrar dos grandes aspectos, el frontend y el backend. (Kazandzhy, 2019) El desarrollo del frontend de una aplicación Web es también conocido como desarrollo del lado del cliente, es la habilidad de escribir código en HyperText Markup Language (HTML), Cascading Style Sheets (CSS) y JavaScript, para que un usuario pueda interactuar con un sitio Web. El objetivo del frontend es facilitar al usuario el uso del sistema de una forma que sea sencilla de comprender e intuitiva de usar. Debido a los distintos dispositivos que se usan en la actualidad, cada uno con distintos tamaños de pantalla y resoluciones, es importante desarrollar un sitio Web que se visualice correctamente en los distintos sistemas operativos y dispositivos. (Lindley, 2019)

Actualmente son tres las principales tecnologías que existen para desarrollar el frontend de un sitio web, la primera es HTML, un lenguaje de marcado que sirve para indicar cómo se van a ordenar todos los elementos de la página web. (Mestres, 2014). La segunda tecnología es CSS, es un lenguaje de hojas de estilo con el cual se puede mejorar el aspecto del sitio, modificando las propiedades de los elementos contenidos en él, como el color, el tipo de fuente y tamaño de los textos, la forma de los elementos, etc. (Andina, 2012). Por último, JavaScript, el lenguaje de programación interpretado que tiene como función manejar el comportamiento del sitio web y la interacción con los usuarios finales. Existen diversidad de frameworks para el uso de JavaScript, entre ellos React y Angular. (Luna, 2019)

Por otro lado, el backend de una aplicación Web representa la parte del sistema que se ejecuta en un servidor, puede ser en la nube o un servidor físico, el backend es el responsable de recibir y procesar las solicitudes de los clientes o del frontend de la aplicación. Algunos lenguajes como Java, Python y PHP son utilizados para desarrollar backend. (Filipova & Vilão, 2018)

El backend es también responsable del envío y recuperación de la información que se almacena en la base de datos. (Manzoor, 2012).

Cuando se utilizan bases de datos, es necesario encontrar una manera de comunicarse con ellas, Structured Query Language (SQL) es un lenguaje que permite ejecutar de forma eficiente tareas

sobre diferentes tablas para extraer información relacionada, en un sistema relacional los datos son representados como tablas y sobre ellas se pueden realizar operaciones. (Date, 2001). Las bases de datos NoSQL no tienen una estructura de tablas como la tienen las SQL, son no relacionales, son sistemas de almacenamiento de información que cumplen con algunos aspectos principales que les dan ventaja sobre las bases de datos relacionales. El primero es el tamaño y cantidad de la información, debido a que mientras más información se maneje, las consultas en bases de datos SQL se vuelven más complejas. La segunda ventaja es que es un paradigma de base de datos orientado a la velocidad de carga de información. (Castro Romero, González Sanabria, & Callejas Cuervo, 2012)

Para poner en funcionamiento el sitio web es necesaria la preparación de un ambiente y arquitectura adecuados, que usualmente requieren de espacio físico, mantenimiento y monitoreo constante, aspectos que aumentan la dificultad y el costo de mantener un sitio web. Debido a esto surgen nuevas alternativas, como las Platform as a Service (PaaS). Según la computación en la nube, la modalidad de plataforma como servicio, brinda a un desarrollador la arquitectura necesaria para ejecutar, alojar y gestionar las aplicaciones Web sin necesidad de diseñar e implementar una arquitectura de hardware y software. (RedHat, 2020). Microsoft Azure, Heroku y RedHat, son algunos proveedores de PaaS.

Para lograr la integración exitosa del frontend y el backend, y conseguir así poner en funcionamiento un sitio web, es necesario el uso de metodologías o marcos de trabajo que permitan planificar y optimizar el proceso de desarrollo del sistema, y de esta forma garantizar la calidad del producto final. (Alonzo, Martinez, & Segovia, 2005).

Se encuentran disponibles una gran cantidad de metodologías que influyen en el desarrollo de software, existen marcos de trabajo tradicionales, que se centran en llevar un riguroso control de los procesos, actividades y herramientas utilizadas para el desarrollo del sitio. Estos presentan algunas desventajas si el software a desarrollar requiere de flexibilidad, pues con este tipo de metodologías los cambios tienen un alto costo ya que todo está planificado y detallado desde antes de iniciar con el proceso de desarrollo del software. (Maida & Pacienza, 2015)

Y, por otro lado, se pueden encontrar las metodologías ágiles, estas toman en cuenta la alta dificultad de llevar a cabo el desarrollo de un proyecto de software sin realizar ningún cambio en el camino, ya sea porque el mismo proyecto lo requiere o porque es solicitado por el usuario final. El desarrollo ágil es una metodología incremental, que permite entregar software funcional en ciclos pequeños, y es capaz de adaptarse a cambios repentinos, también permite una mejor comunicación con el cliente. (Fuentes, 2015). Es necesario estudiar y evaluar el producto a desarrollar para determinar cuál tipo de metodología es la que mejor se adapta al software que se va a desarrollar.

Hoy en día son muchos los negocios, empresas, clínicas, personas naturales, entre otros, los que ven la necesidad de tener un sitio web para brindar información de los productos o servicios que ofrecen. Algunas veces no solo para dar información, sino que necesitan su página web para realizar ventas formales en línea de forma fácil, rápida, confiable y sin necesidad de que sus clientes se muevan de sus hogares. La forma en que se exhiben los productos en línea es clave para realizar una venta. La información que se le presenta al usuario debe ser específica, con imágenes representativas de cada producto y una descripción clara de los mismos. (K., 2013)

Para que los usuarios sean capaces de realizar compras en un sitio web, es requerido un carrito de compras, este es el nombre que recibe la plataforma, módulo o plugin que se encarga de recopilar los pedidos que hacen los usuarios en una página web que cuente con una tienda en línea. Su función es igual a la de un carrito de supermercado. Se recorren los pasillos mientras se cargan todos los artículos que se desean comprar, la diferencia es que se hace vía internet desde la comodidad de tu casa. (Luis, 2020)

Una vez el usuario haya seleccionado los productos que desea comprar, es imprescindible que el sitio web les permita pagar para obtener sus artículos. Para que ellos puedan realizar sus pagos, existen muchas opciones, una de ellas son las transferencias bancarias que son envíos de dinero realizados a la orden de un cliente desde su cuenta bancaria en una entidad (ordenante) a otra designada (beneficiario). (Comisiones, 2015)

Paypal es otra buena y segura opción, es un método de pago en línea, PayPal ofrece a los clientes de un negocio una variedad de métodos de pago que permiten que las ventas se lleven a cabo de una forma rápida y segura. (PayPal, s.f.)

IV. METODOLOGÍA

4.1 ENFOQUE

“El enfoque cuantitativo, que representa un conjunto de procesos, es secuencial y probatorio. Cada etapa precede a la siguiente y no podemos “brincar” o eludir pasos”. (Sanpieri, Fernandez Collado, & Baptista Lucio, 2014). Por esta razón se ha decidido usar un enfoque de metodología cuantitativo, ya que el desarrollo de un proyecto de software implica seguir una serie de pasos ordenados que van uno después de otro y no es posible pasar por alto ninguno. El desarrollo del proyecto debe seguir un modelo y una serie de pasos estructurados.

4.2 VARIABLES DE INVESTIGACIÓN

- Disponibilidad de la PaaS donde estará desplegado el proyecto.
- Nivel de satisfacción de la nutricionista con el sitio web desarrollado.

4.3 TÉCNICAS E INSTRUMENTOS APLICADOS

4.3.1. CASOS DE USO

Al iniciar con la fase de diseño del proyecto, se elaboran los diagramas de casos de uso, para plasmar la idea general de cómo funciona el sistema. En la Ilustración 1 se muestra el caso de uso del sistema para el usuario paciente, se observa que se tienen dos tipos de pacientes, el paciente invitado, es aquel que no está registrado en el sistema y puede explorar y comprar productos y planes. El paciente con plan es aquel paciente que ya tiene acceso al sistema porque ha realizado alguna compra.

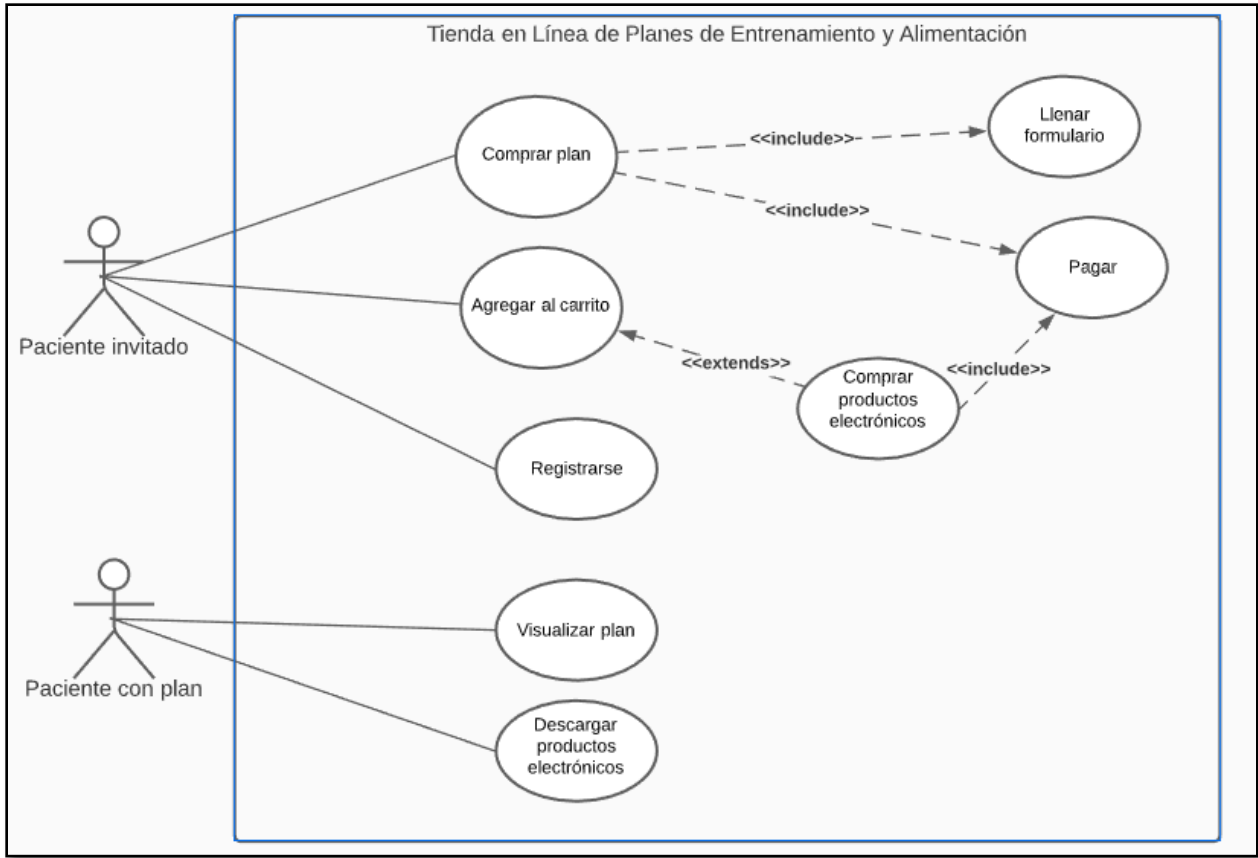


Ilustración 1 – Diagrama de caso de uso para usuario paciente.

Fuente: Elaboración propia

La Ilustración 2 muestra el caso de uso para el usuario nutricionista, este puede gestionar planes y productos, así como gestionar las transferencias bancarias que se realicen.

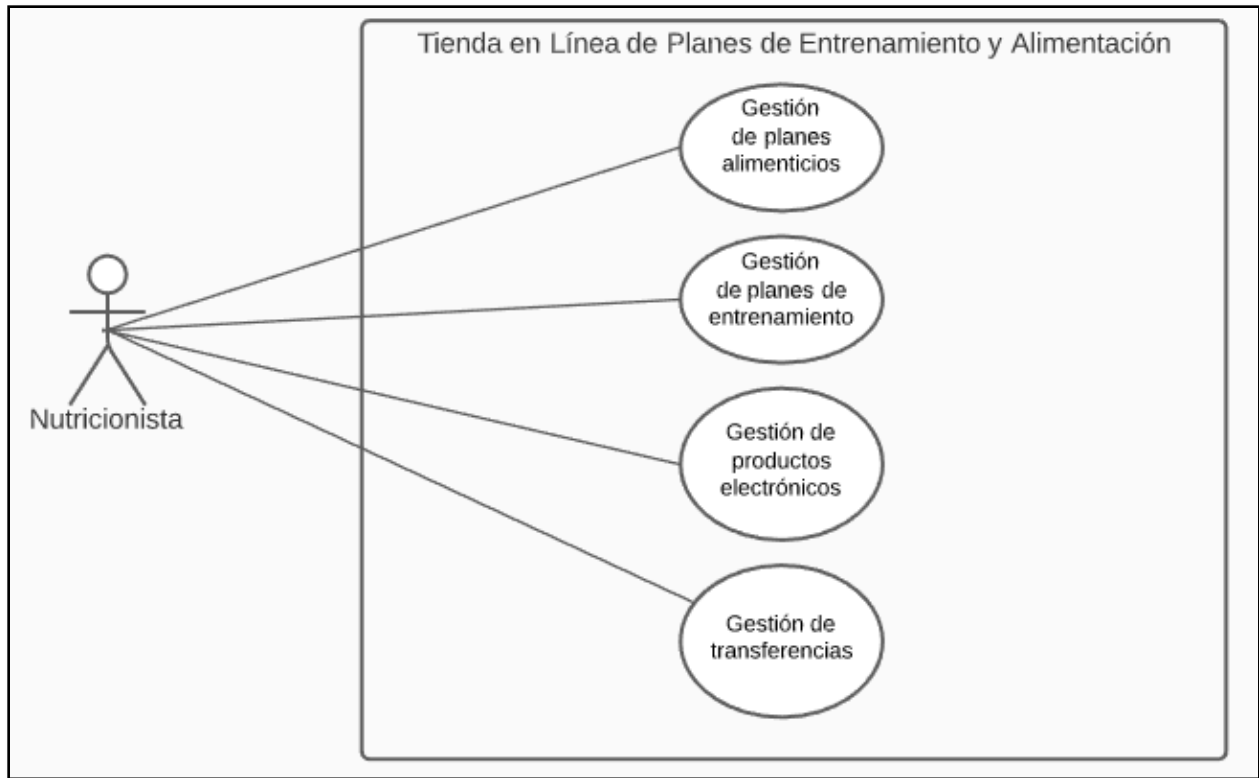


Ilustración 2 – Diagrama de caso de uso para usuario nutricionista

Fuente: Elaboración propia.

4.3.2. MOCKUPS

Los mockups son una herramienta muy útil para diseñar un sistema, sirven de base a la hora de diseñar la interfaz gráfica del sitio web y son ideales para mostrarle al cliente una idea de cómo se verá su producto final.

En la Ilustración 3 se puede observar el diseño general de la landing page del sitio, se pidió que se mostrara una presentación de imágenes de transformaciones de pacientes de la clínica.

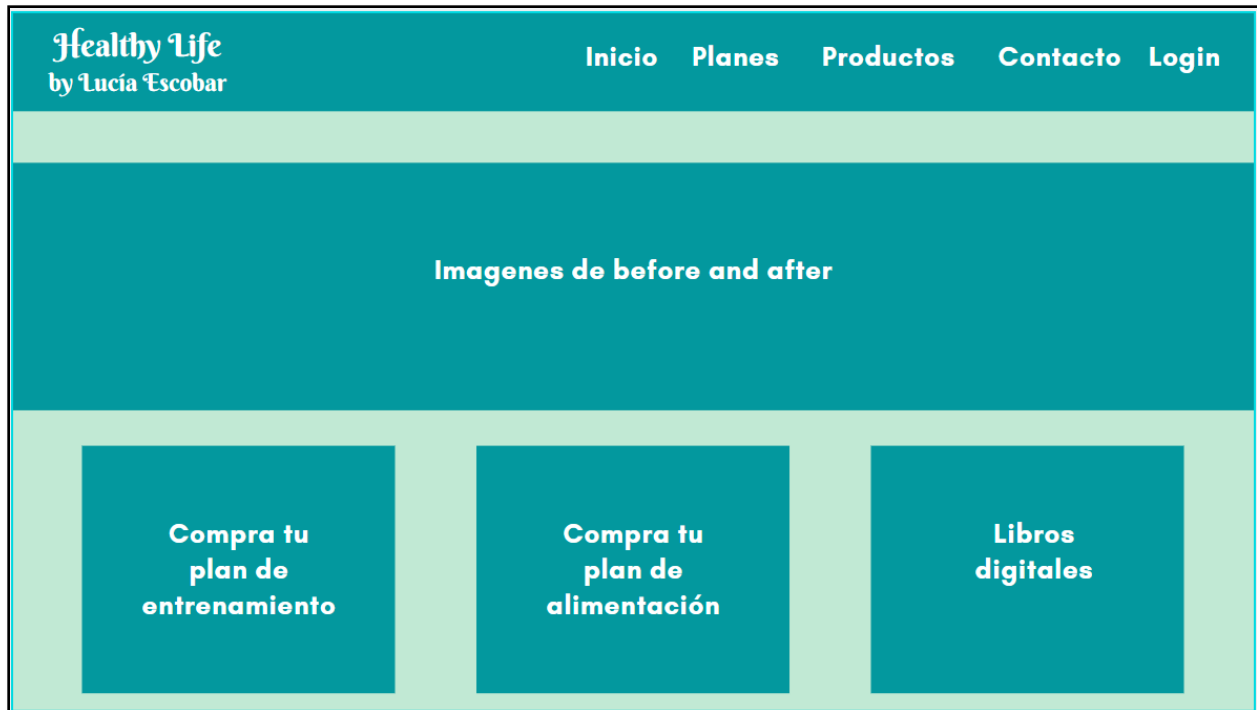


Ilustración 3 – Mockup de landing page

Fuente: Elaboración propia.

En la Ilustración 4 se muestra el mockup del módulo de planes, donde el usuario podrá seleccionar el plan que mejor se adapte a sus objetivos y presupuesto.

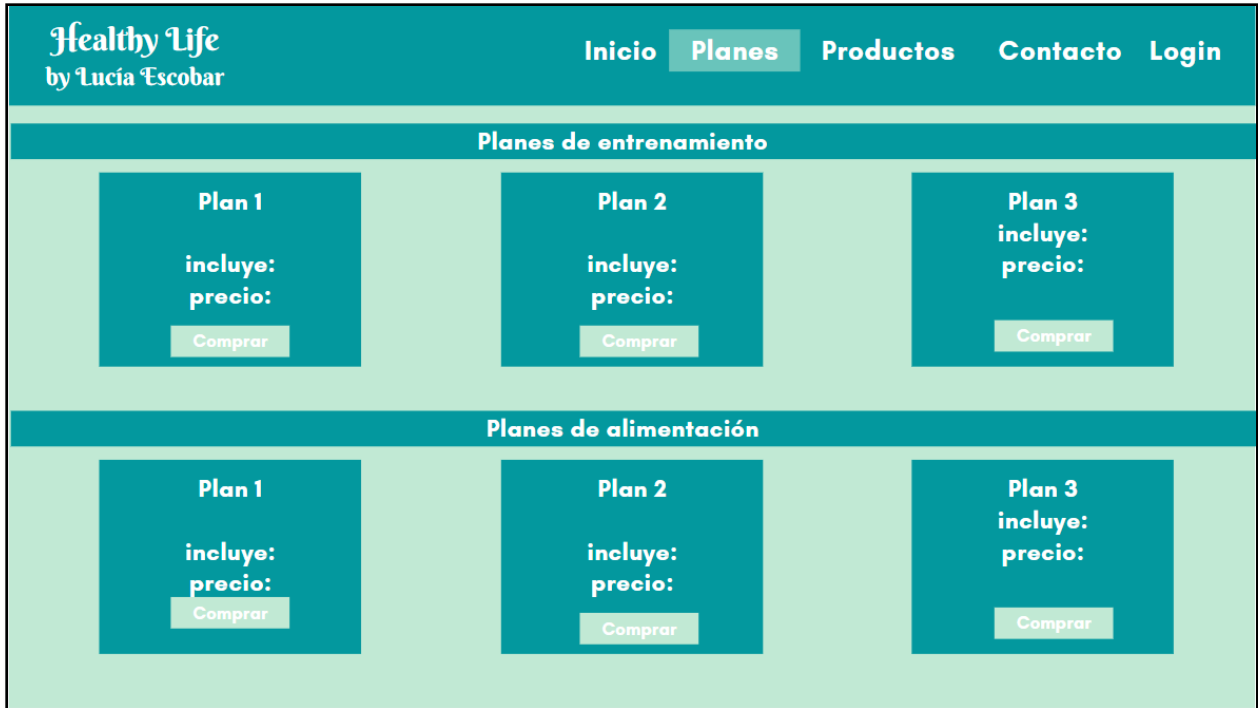


Ilustración 4 – Mockup de planes

Fuente: Elaboración propia.

En la Ilustración 5, se muestra la tienda de productos, que puede ser explorada por el usuario, puede realizar búsqueda de productos y puede agregar productos al carrito de compras.



Ilustración 5 – Mockup de productos

Fuente: Elaboración propia.

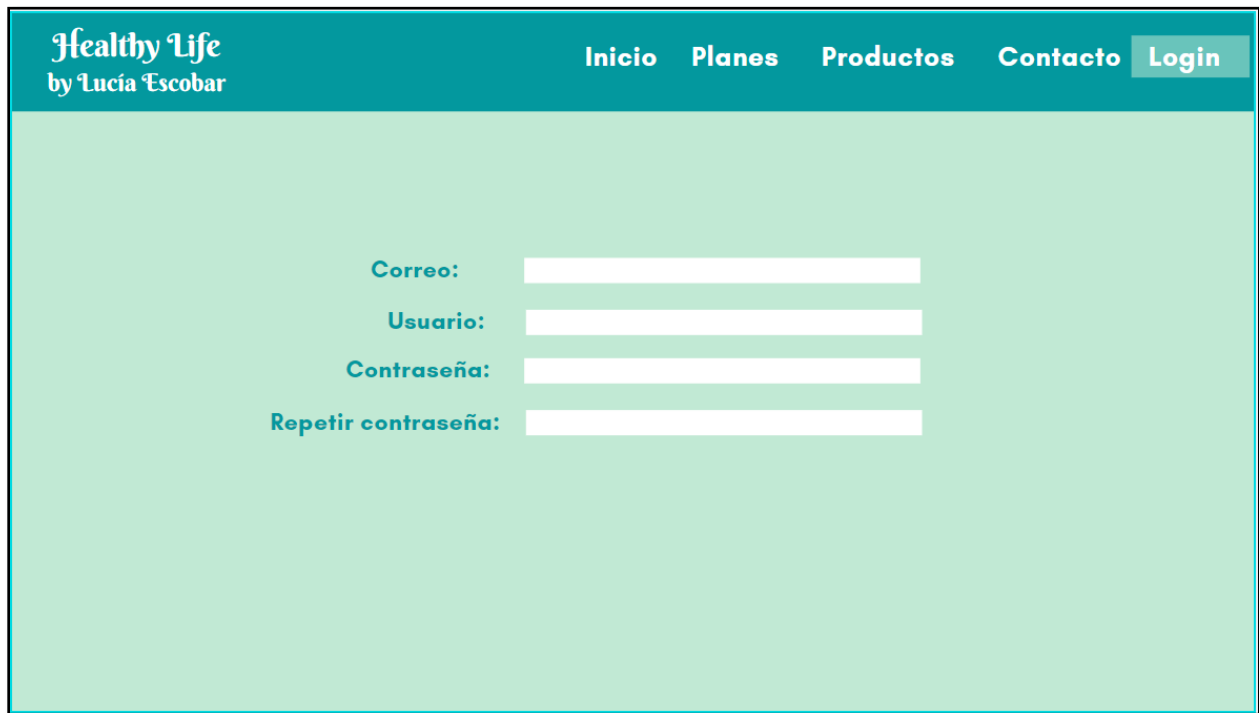
El usuario registrado podrá iniciar sesión, como se muestra en la Ilustración 6.



Ilustración 6 – Mockup de inicio de sesión

Fuente: Elaboración propia.

La Ilustración 7 presenta el mockup del registro de usuarios.



The image shows a web page mockup for 'Healthy Life by Lucía Escobar'. The header is teal with the logo on the left and navigation links 'Inicio', 'Planes', 'Productos', 'Contacto', and 'Login' on the right. The 'Login' link is highlighted in a lighter teal box. The main content area is light green and contains a registration form with four white input fields. The labels for the fields are 'Correo:', 'Usuario:', 'Contraseña:', and 'Repetir contraseña:'.

Ilustración 7 – Mockup de registro de pacientes

Fuente: Elaboración propia.

4.3.3. DISEÑO DE BASE DE DATOS

Se procedió a realizar un diagrama de clases, como se muestra en la Ilustración 8, que sirvió como modelo para la creación de la base de datos, el tipo de base de datos elegido fue el relacional y se usó SQLite.

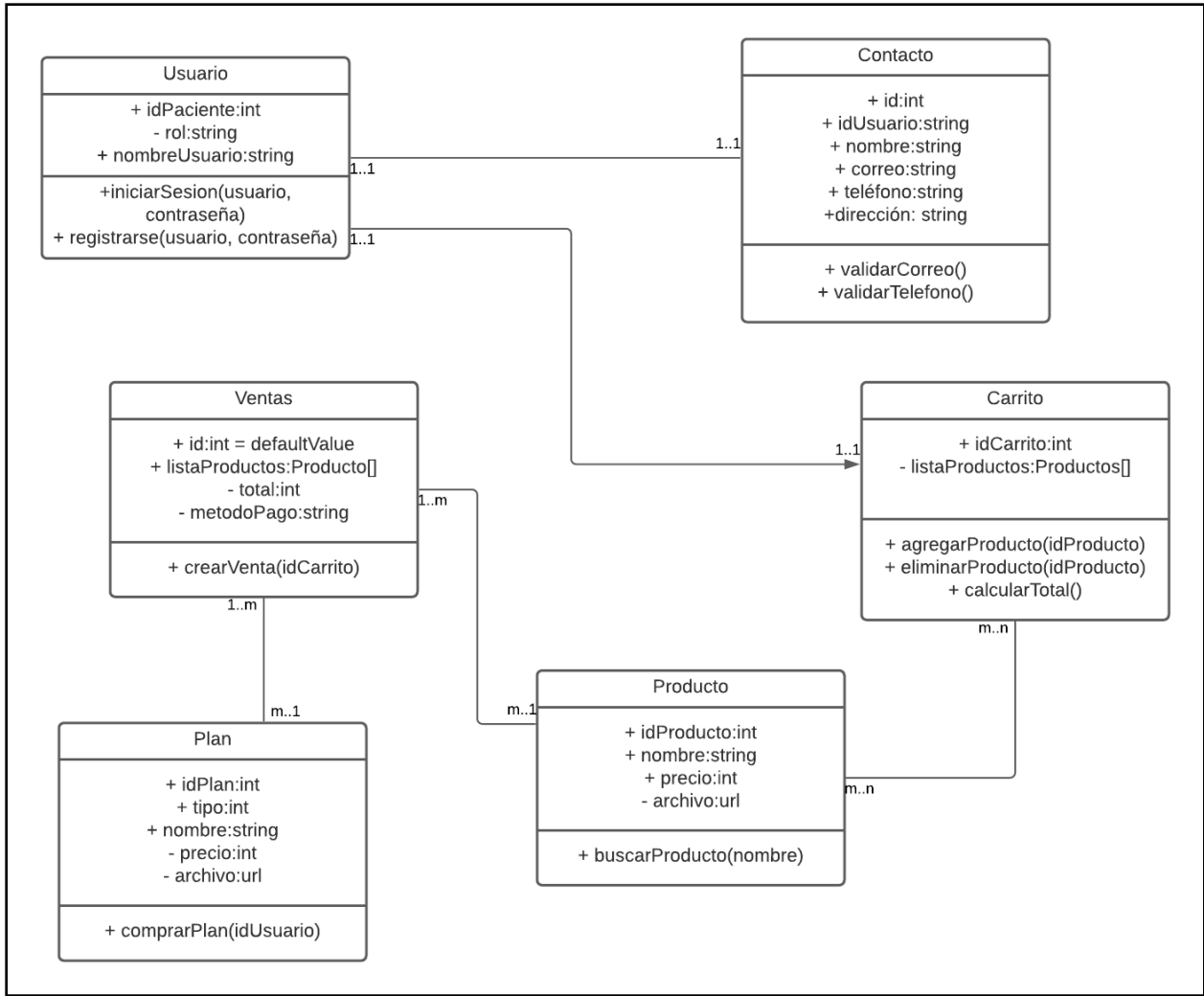


Ilustración 8 – Diagrama de clases

Fuente: Elaboración Propia.

4.3.4. REACT

Como framework para el frontend del proyecto, se escogió React debido a la gran cantidad de información y documentación pública de la librería que facilita su aprendizaje.

4.3.5. DJANGO Y PYTHON

El lenguaje de programación de backend que se decidió utilizar es Python y su framework Django, debido a que Django brinda una gran cantidad de beneficios y facilidades al desarrollador, la más sobresaliente es su plataforma de administración integrada.

4.3.6. GITHUB

Es un sistema de control de versión y será utilizado para almacenar el repositorio del proyecto y gestionar todos los cambios realizados en él.

4.3.7. DIGITAL OCEAN

Es una empresa proveedora de servicios de computación en la nube, que ofrece una plataforma de infraestructura como servicio (IaaS) para desarrolladores de software (Rouse, 2019), por sus múltiples ventajas y precio se utilizará un droplet como servidor para lanzar a producción el frontend y backend del sistema.

4.3.8. ARQUITECTURA WEB

En la Ilustración 9 se muestra el diagrama de la arquitectura web del sistema.

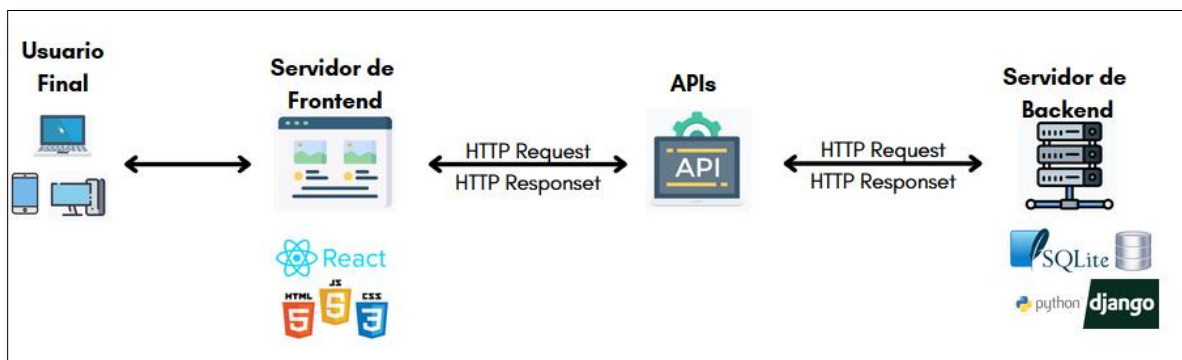


Ilustración 9 – Diagrama de arquitectura Web

Fuente: Elaboración Propia.

4.3.9. INTERFAZ DE PROGRAMACIÓN DE APLICACIONES (API) DE TRANSFERENCIA DE ESTADO REPRESENTACIONAL (REST)

Para que el frontend y el backend puedan comunicarse y ejecutar funciones se necesita un intermediario, por esta razón se hacen uso de las API REST, estas permiten interactuar con la base de datos y gestionar toda la información almacenada en ella.

4.3.10 MODELO DE BASE DE DATOS IMPLEMENTADO

La Ilustración 10 muestra el modelo final de la base de datos que fue implementado en Django incluyendo las tablas de grupos y permisos y las tablas generadas por defecto en Django.

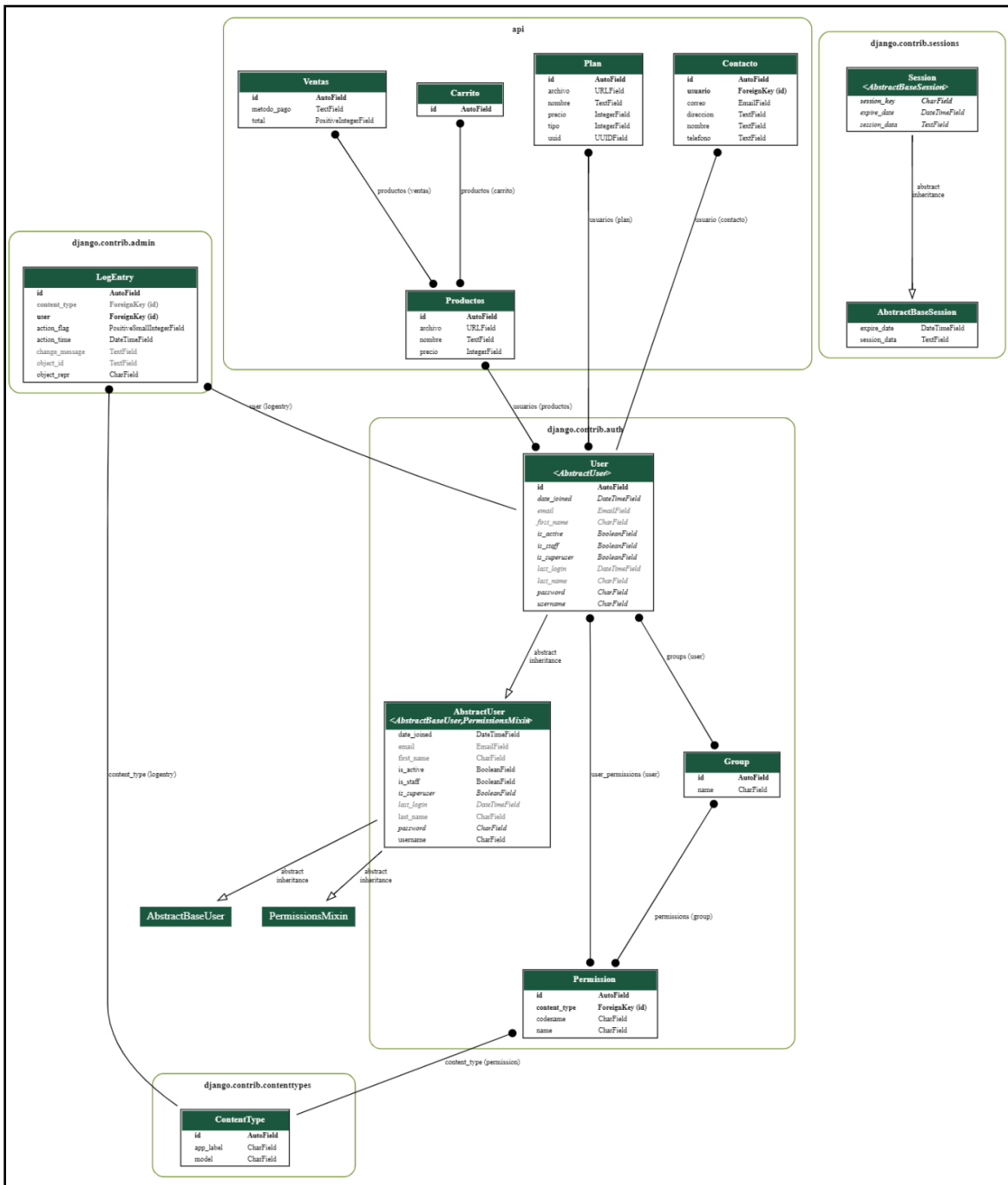


Ilustración 10 – Modelo final de la base de datos

Fuente: Elaboración propia

4.3.11. ADMINISTRADOR DJANGO

Una de las principales ventajas que posee Django es su consola de administración, desde ella el administrador del sitio puede gestionar toda la información del sistema, después de definir los modelos, el administrador puede agregar y eliminar usuarios, gestionar planes y productos, entre otros. La Ilustración 11 muestra un ejemplo de la consola.

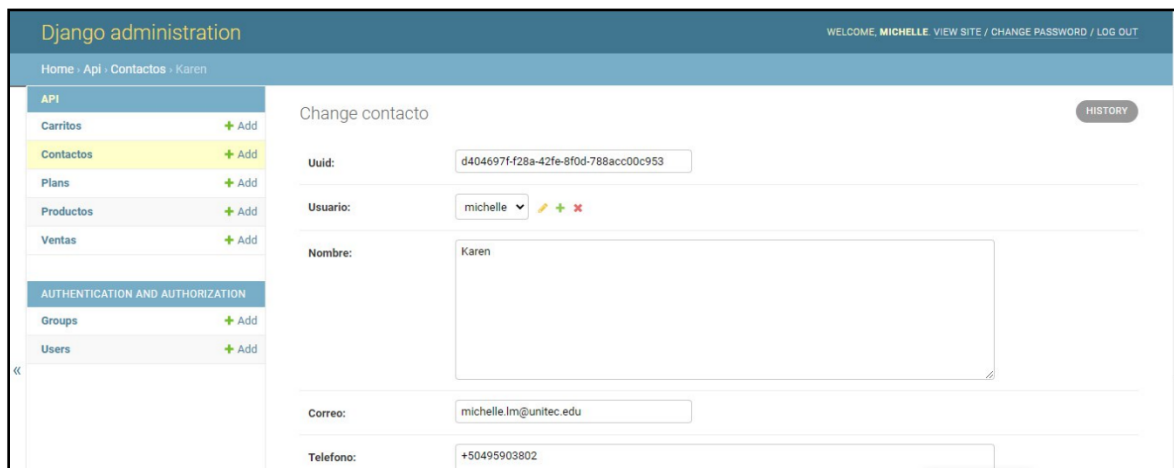


Ilustración 11 – Consola de Administración de Django.

Fuente: Elaboración propia.

4.4 METODOLOGÍA DE ESTUDIO

4.4.1. MODELO ÁGIL

Mientras que las metodologías tradicionales como el modelo de cascada son ideales para proyectos en los que se tiene bien definido el alcance y se sabe que no existirán modificaciones, las metodologías ágiles son flexibles, adaptables e iterativas. Para el desarrollo del proyecto se optó por un modelo ágil, ya que es sabido desde el inicio, que el cliente aún tenía bastantes ideas que no estaban bien definidas lo que implicarían cambios ligeros durante el desarrollo del proyecto. Adicionalmente, este modelo permite hacer entregables durante tiempos cortos lo que permite una mejor comunicación con el cliente dueño del proyecto.

4.5 CRONOGRAMA DE TRABAJO

En la Tabla 1 se muestra el cronograma utilizado para el desarrollo del proyecto en su fase 1.

Tabla 1 – Cronograma de Trabajo Proyecto Fase I

Actividad	Octubre			Noviembre				Diciembre		
	12	19	26	2	9	16	23	30	7	14
Recopilación de información										
Investigar sobre Plantillas Web ideales para páginas de nutrición										
Investigar sobre Plantillas Web de venta de productos										
Investigar sobre Frameworks modernos de desarrollo de aplicaciones Web										
Reunion con cliente para definir plantilla										
Análisis y diseño										
Diagramas de casos de uso para el sistema										
Diagrama de clases para el sistema										
Diseño de Mockups										
Diseño de Base de datos										
Desarrollo										
Creación de roles y usuarios										
Login para administradores y pacientes sin registro										
Implementar el framework de frontend										
Registro de datos del paciente										
Paciente puede descargar su rutina de ejercicio										
Paciente puede visualizar su rutina de ejercicio										
Paciente puede descargar su rutina de Alimentación										
Paciente puede visualizar su rutina de Alimentación										

Fuente: Elaboración propia

En la Tabla 2 se muestra el cronograma utilizado para el desarrollo del proyecto en su fase 2.

Tabla 2 – Cronograma de Trabajo Proyecto Fase II

Actividad	Enero			Febrero				Marzo			
	2	3	4	1	2	3	4	1	2	3	4
Creación de usuario automáticamente con una contraseña OTP											
Desarrollo y configuración de envío de correos a usuario con notificaciones											
Gestión de productos											
Administrador puede crear productos y gestionarlos											
Categorizar productos											
Filtro y búsqueda de productos para la vista del cliente											
Mostrar detalles de productos											
Proceso de compra											
Agregar carrito											
Solicitar datos											
Solicitar dirección											
Solicitar método de envío											
Solicitar método de pago											
Pago por transferencia											
Administrados confirma transferencia efectuada											
Acceso de usuario a producto comprado											
Pago por PayPal											
Pruebas de seguridad											

Fuente: Elaboración propia.

V. RESULTADOS Y ANÁLISIS

5.1 PROPUESTA DE INTERFAZ DE USUARIO

La interfaz gráfica de usuario es un componente importante del proyecto, es el encargado de la comunicación del sistema con el usuario final. Para la dueña del proyecto es muy importante que su sitio web tenga una interfaz gráfica intuitiva, moderna, llamativa y bonita. Debido a estos requerimientos se decidió usar una plantilla web para adaptarlo a las necesidades de la tienda de planes y productos. Se le mostró al cliente una diversidad de plantillas que pudieran adaptarse al proyecto, entre ellas gratis y pagadas para que ella decidiera. Ella optó por una plantilla pagada de ReactJS, llamada Zonata – React Next.js Organic Food Template. En la Ilustración 12 se muestra la landing page de la plantilla que será adaptada a los requisitos del sistema.

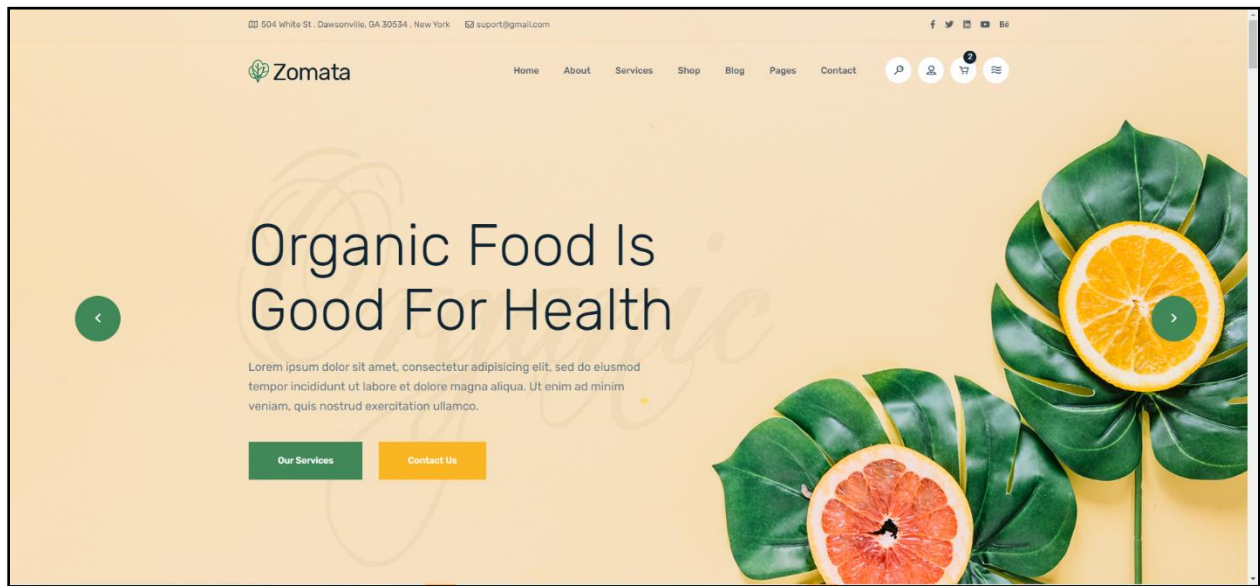


Ilustración 12 – Landing page de plantilla.

Fuente: (BDevs, 2020)

En la Ilustración 13 se puede observar que la plantilla incluye un carrito de compras para listar y agregar productos para su compra.

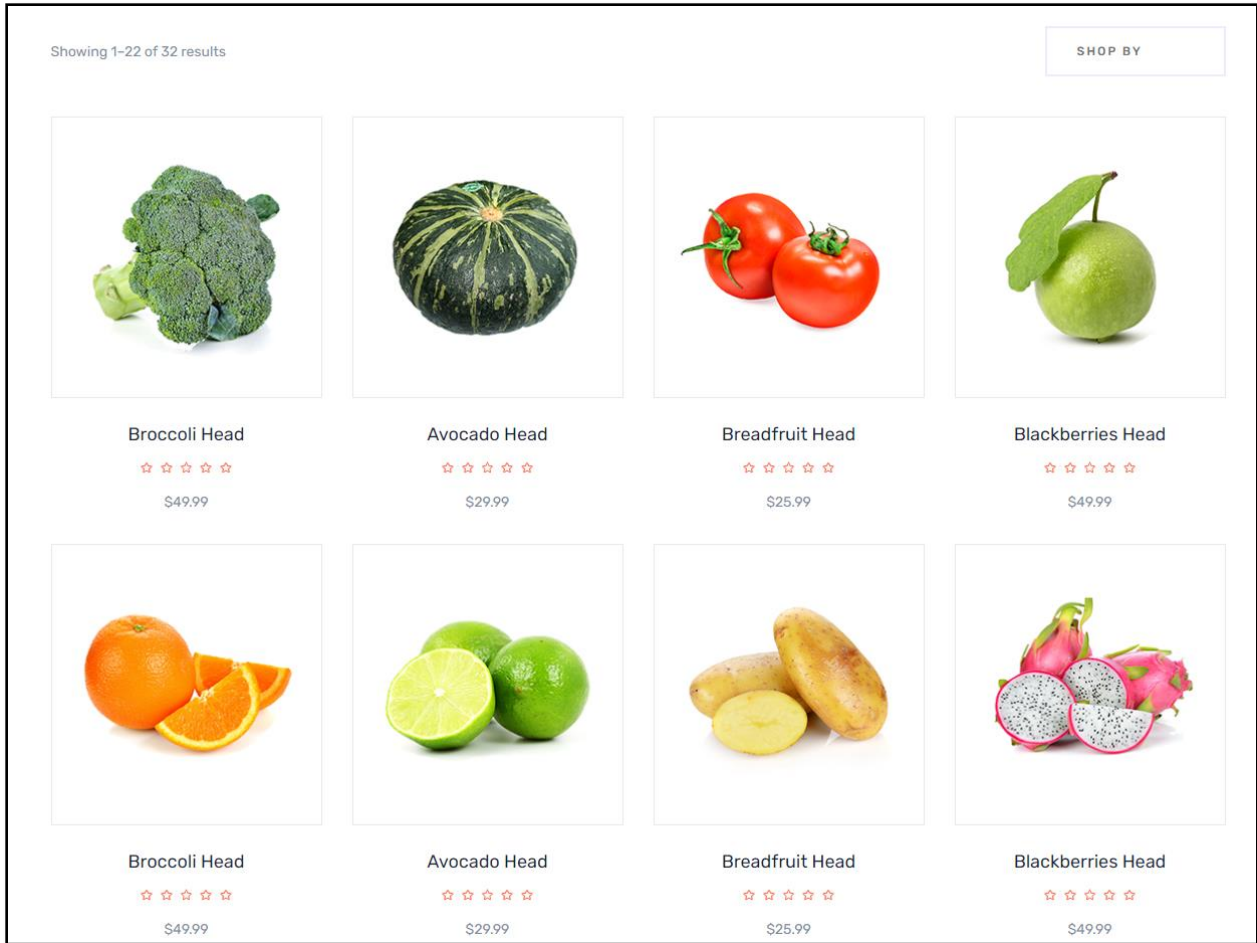


Ilustración 13 – Plantilla de tienda en línea para exhibición y compra de productos.

Fuente: (BDevs, 2020)

5.2 API REST

Para que el usuario final pueda agregar, modificar, editar o eliminar información de la base de datos del sistema, es necesario crear un intermediario que procese la información entre el servidor de frontend y el de backend a través de peticiones HTTP, esta información viaja usualmente en forma de Notación de Objeto JavaScript (JSON) (RedHat, 2020).

En la Ilustración 14 se observa la forma en la que se visualiza una de las APIs del sistema.

```
localhost:8000/api/contact/7ffbee43-40fc-4cf1-b772-4cbbce6a4c8e/plans/

Django REST framework

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 1,
    "uuid": "2a4ac712-62fa-41c8-83e2-040a9221c213",
    "tipo": 1,
    "nombre": "Plan 1",
    "precio": 200,
    "archivo": "/files/files/Plan1.png",
    "usuarios": [
      3
    ]
  },
  {
    "id": 2,
    "uuid": "1a085d9b-14de-4657-8941-70c8bd37eba6",
    "tipo": 1,
    "nombre": "Plan ejemplo 2",
    "precio": 1000,
    "archivo": "/files/files/plan.pdf",
    "usuarios": [
      3
    ]
  }
]
```

Ilustración 14 – API Planes

Fuente: Elaboración propia

Fue necesario crear cinco APIs en Django de Python para el funcionamiento de la aplicación Web. La Tabla 3 muestra los detalles de las API utilizadas por el sistema.

Tabla 3 – Descripción de cada API utilizada en la aplicación

Parámetro	Descripción
Nombre	Planes
Ruta	/api/uuid/planes
Función	Listar los planes alimenticios y de entrenamiento de cada usuario para su visualización y descarga.
Permisos	Usuario Autenticado
Campos	uuid = UUIDField tipo = IntegerField nombre = TextField() precio = IntegerField() archivo = FileField() usuarios = ManyToManyField(User)
Nombre	Contacto
Ruta	/api/uuid/contacto
Función	CRUD Usuario
Permisos	Administrador
Campos	uuid = UUIDField usuario = OneToOneField(User) nombre = TextField correo = EmailField telefono = TextField direccion = TextField
Nombre	Producto
Ruta	/api/uuid/producto
Función	CRUD Producto

Parámetro	Descripción
Permisos	Usuario autenticado, administrador
Campos	nombre = TextField precio = IntegerField archivo = FileField usuarios = models.ManyToManyField(User)
Nombre	Venta
Ruta	/api/venta
Función	Gestionar la venta de planes y productos
Permisos	Usuario autenticado
Campos	productos = ManyToManyField(Producto) total = PositiveIntegerField() método_pago = TextField
Nombre	Carrito
Ruta	/api/uuid/carrito
Función	Gestionar el carrito de compra
Permisos	Usuario autenticado
Campos	productos = ManyToManyField(Producto) usuario = OneToOneField(User)

Fuente: Elaboración propia.

5.3 CAPTURAS DE LA APLICACIÓN

En esta sección se muestran algunas capturas de las funcionalidades del sitio Web. En la Ilustración 15 se observa el login funcional de la aplicación con un HTML genérico previo a la compra de la plantilla de React a utilizar.



Ilustración 15 – Login de la aplicación

Fuente: Elaboración propia

La Ilustración 16 muestra otra funcionalidad completa de la aplicación previo a la compra de la plantilla de React, el usuario al iniciar sesión puede ver la lista de los planes que tiene asignados para su visualización o descarga.

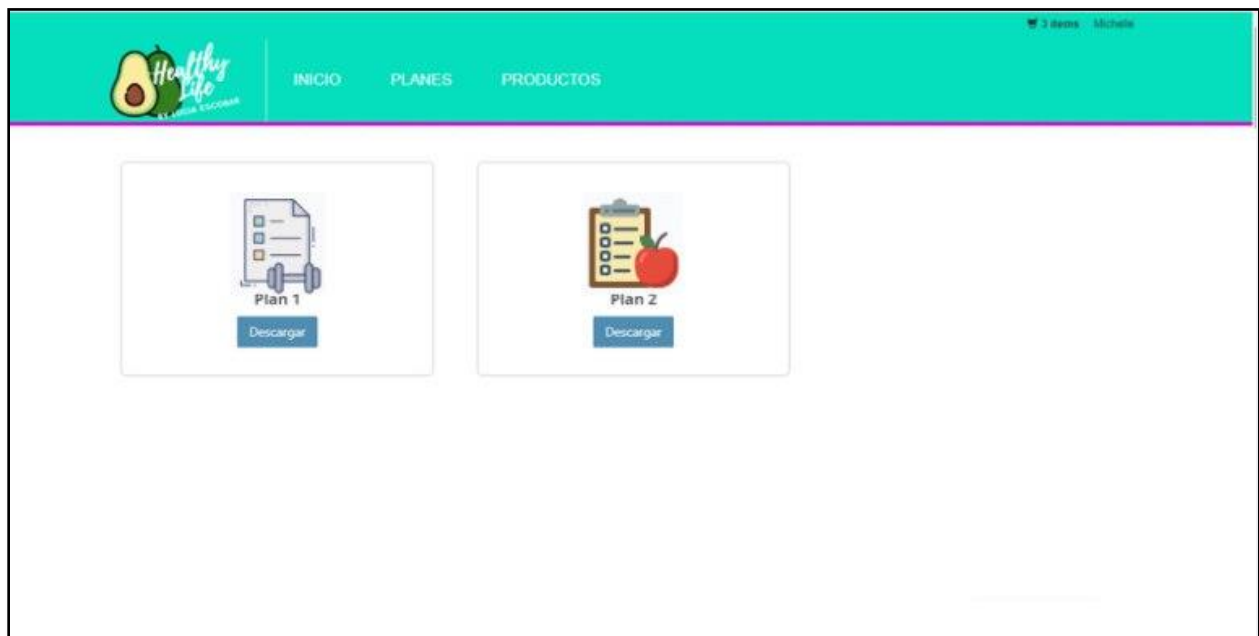


Ilustración 16 – Lista de planes y productos que tiene asignado un usuario.

Fuente: Elaboración propia.

La Ilustración 17 muestra la versión final de la landing page del sitio Web.



Ilustración 17 – Landing Page.

Fuente: Elaboración propia.

La Ilustración 18 muestra la administración del nutricionista, en donde puede gestionar los productos y planes.

The image shows a web form titled 'Add producto'. It has several fields: 'Nombre:' with a text input containing 'Desayunos Saludables'; 'Precio:' with a numeric input set to '200'; 'Archivo:' with a 'Choose File' button and the filename 'Desayunos d...ludables.pdf'; and 'Usuarios:' with a dropdown menu showing 'michelle' and a green plus sign to add more. At the bottom right are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'. A small note at the bottom of the form says 'Hold down "Control", or "Command" on a Mac, to select more than one.'

Ilustración 18 – Gestión de productos.

Fuente: Elaboración propia.

En la Ilustración 19 se puede ver cómo se listan los productos creados por el nutricionista para que el usuario pueda adquirirlos, estos pueden ser filtrados y al colocar el mouse sobre el producto aparece la opción de agregar al carrito.



Ilustración 19 – Listado de productos.

Fuente: Elaboración propia.

En la Ilustración 20 se muestra el listado de los planes para que el usuario pueda adquirirlos, estos pueden ser filtrados y al colocar el mouse sobre el plan aparece la opción de agregar al carrito.

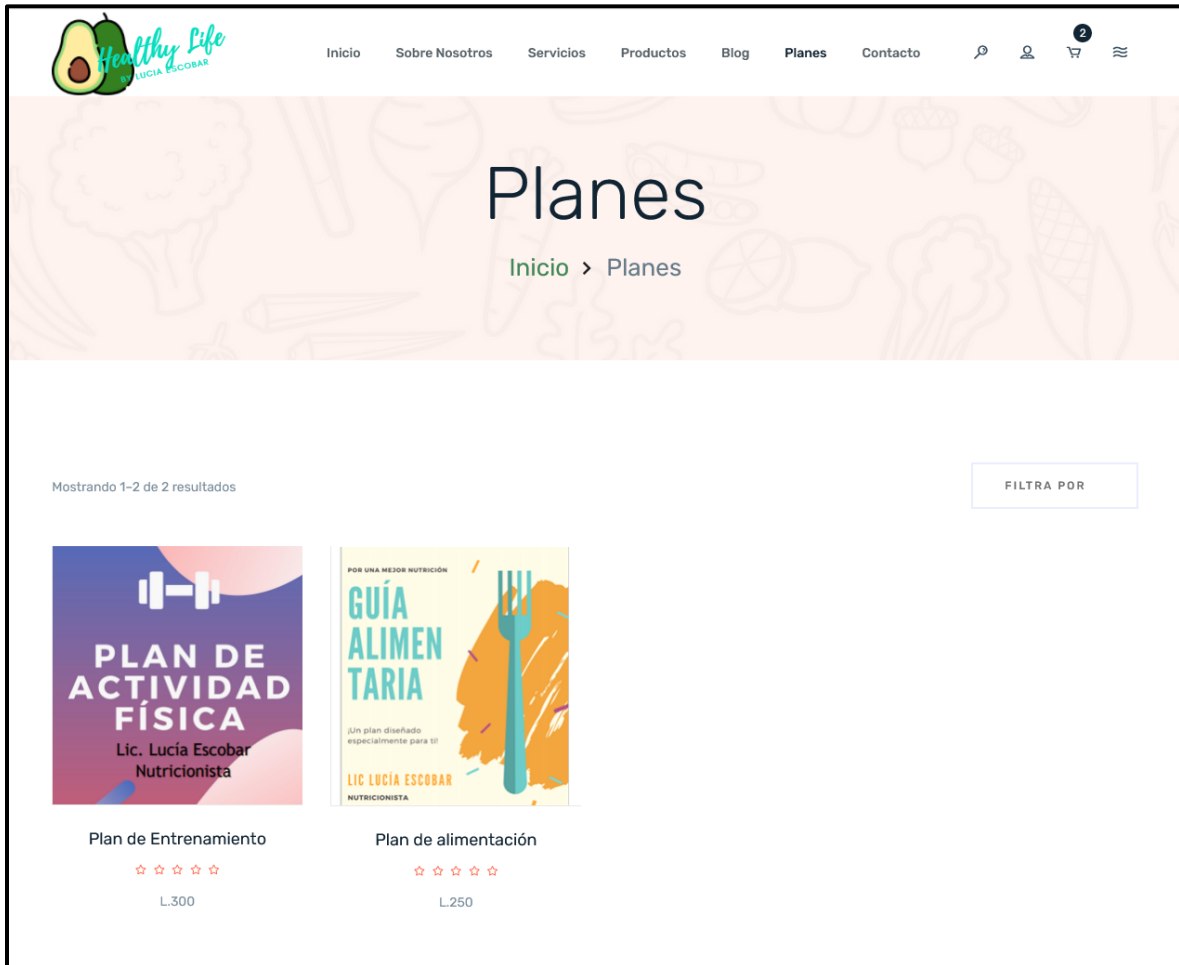


Ilustración 20 – Listado de Planes

Fuente: Elaboración propia.

Se pueden ver los detalles de cada producto y puede ser agregado al carrito como se muestra en la Ilustración 21.



Ilustración 21 – Detalle de producto

Fuente: Elaboración propia.

Se pueden visualizar los artículos agregados al carrito como se muestra en la Ilustración 22.

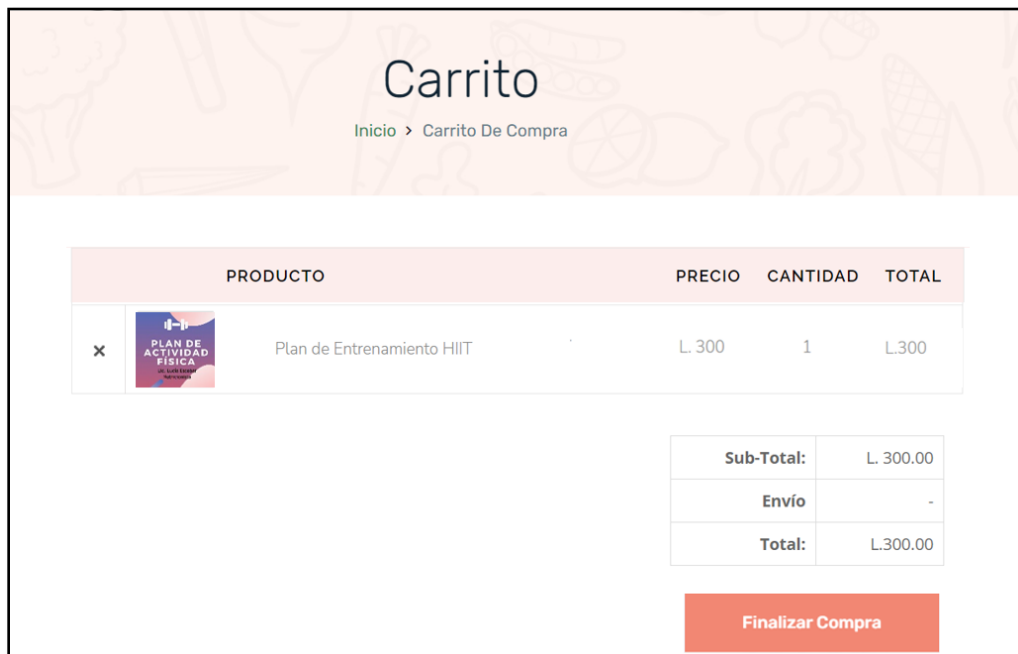


Ilustración 22 – Carrito de compras

Fuente: Elaboración propia.

Para hacer la compra el usuario puede seleccionar entre dos métodos de pago, por transferencia bancaria o PayPal, esto se ve reflejado en la Ilustración 23.



Ilustración 23 – Detalles de la compra

Fuente: Elaboración propia.

Cuando el usuario compra un producto, tiene la opción de descargarlo o de visualizarlo para llevar un control de lo realizado en forma de calendario, como se puede ver en la Ilustración 24.



Ilustración 24 – Visualización y control de planes

Fuente: Elaboración propia.

La ilustración 25 muestra el Blog del sitio Web el cual puede ser editado por los nutricionistas desde la consola de administración de Django y las entradas pueden ser visualizadas de forma pública por usuarios autenticados y no autenticados.



Ilustración 25 – Blog

Fuente: Elaboración propia.

Para finalizar, se validó la disponibilidad de la aplicación. La Ilustración 26 muestra una gráfica de Vercel, la PaaS utilizada para desplegar en producción el front end de la aplicación, donde se puede observar que de las peticiones de los usuarios, han surgido 0% de errores de tiempo de espera.

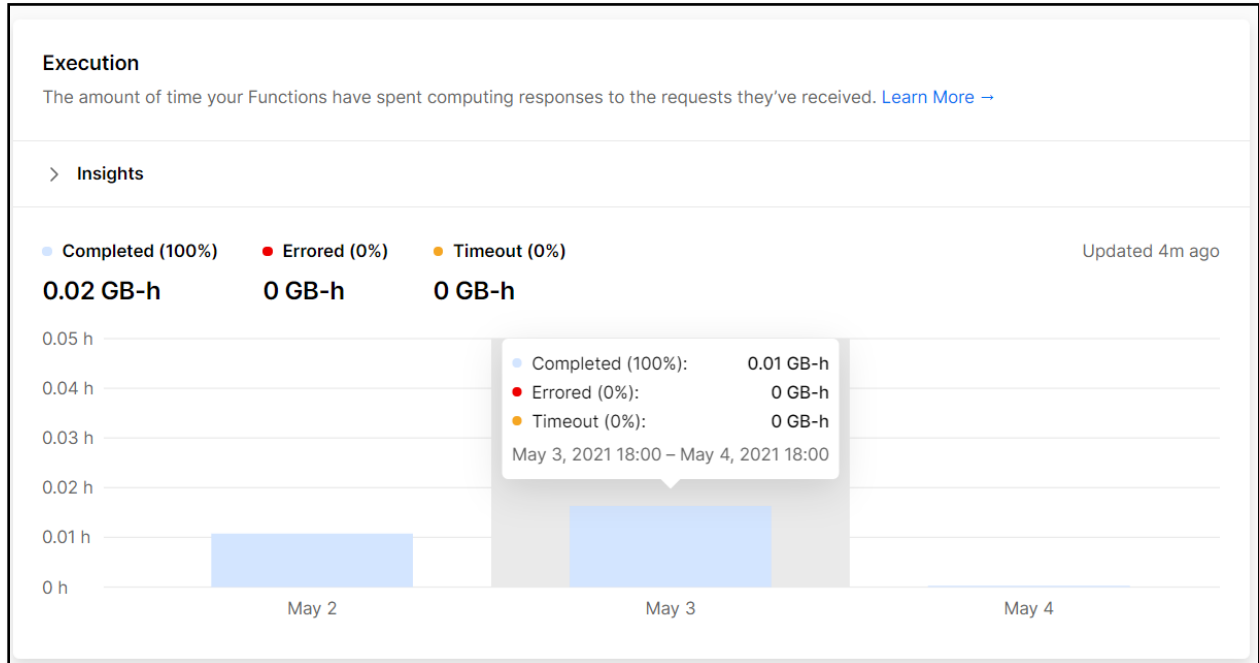


Ilustración 26 – Gráfica de Vercel

Fuente: Elaboración propia.

VI. CONCLUSIONES

Al finalizar el proyecto de investigación y con base en los objetivos planteados anteriormente en el presente documento, se detallan las siguientes conclusiones:

- Se desarrolló un sistema Web que permite a los usuarios comprar planes de entrenamiento y alimentación, y a los nutricionistas llevar un mejor control y seguimiento de sus pacientes.
- Se diseñó una plataforma con la experiencia de usuario necesaria para que el sistema sea de agrado y de fácil uso para las personas que lo utilicen.
- Se desarrolló una tienda en línea con pagos por medio de transferencia bancaria y de PayPal para que los usuarios puedan adquirir los productos de su interés sin necesidad de ir a la clínica.
- Se aplicaron controles de seguridad para proteger la información de la aplicación y validar que cada usuario pueda acceder únicamente a su información, los controles aplicados fueron autenticación de usuarios, controles de acceso y gestión de sesiones.
- Se contemplaron prácticas de desarrollo seguro basadas en OWASP desde la primera fase del proyecto y se aplicaron en todo el desarrollo de la plataforma web.
- Se realizaron pruebas de calidad y seguridad manuales y automatizadas a la aplicación, después del desarrollo de cada módulo y se corrigieron a tiempo y con éxito las fallas encontradas en la aplicación antes de lanzarla a producción.
- Se logró validar que la aplicación estará siempre disponible en internet a través de las PaaS seleccionadas para su despliegue.

VII. RECOMENDACIONES

A continuación, se presentan algunas recomendaciones para los encargados de dar mantenimiento a la aplicación y personas en general que quieran desarrollar un sistema Web:

- Se recomienda a las personas que deseen desarrollar un sistema Web, mantener una comunicación abierta y constante con el dueño del sistema, para definir detalladamente los objetivos y conocer las expectativas del cliente, garantizando que esté conforme con el resultado final.
- Se recomienda a los encargados del mantenimiento de la Plataforma Web para la Venta en Línea de Planes de Entrenamiento y de Alimentación, investigar sobre pagos con tarjeta de crédito/debito para implementar en el futuro este tipo de pago en el sitio Web.
- Se recomienda a los encargados del mantenimiento de la Plataforma Web para la Venta en Línea de Planes de Entrenamiento y de Alimentación, generar un nuevo certificado de seguridad cuando el implementado actualmente expire.
- Se recomienda a los encargados del mantenimiento de la Plataforma Web para la Venta en Línea de Planes de Entrenamiento y de Alimentación, realizar evaluaciones de seguridad cada seis meses para garantizar que la aplicación mantiene una postura de seguridad alta, pues al pasar el tiempo las versiones de librerías, frameworks y sistemas operativos se vuelven obsoletos y pueden introducir vulnerabilidades a la aplicación.
- Se recomienda a las personas que deseen desarrollar un sistema Web realizar una planificación adecuada, considerando tiempo suficiente para el diseño, la investigación de tecnologías, el desarrollo y las pruebas de calidad y seguridad a la plataforma.
- Se recomienda a las personas que quieran desarrollar un sistema desde cero, que estudien y comprendan a fondo el proyecto que se llevará a cabo, para poder seleccionar la mejor arquitectura y tecnologías basado en las necesidades de la aplicación.

BIBLIOGRAFÍA

- Filipova, O., & Vilão, R. (2018). *Software Development From A to Z*. Berlin: Apress, Berkeley, CA.
- Alonzo, F., Martínez, L., & Segovia, F. J. (2005). *¿Qué son las metodologías de desarrollo de software?* Madrid: Delta Publicaciones Universitarias. Obtenido de OBS Business School.
- Andina, F. (2012). *Diseño Web con HTML y CSS*. Sevagraf: Dalaga S.A.
- BDevs. (4 de noviembre de 2020). *The Theme Forest*. Obtenido de <https://themeforest.net/item/zomata-react-nextjs-organic-food-template/29266784>
- Castro Romero, A., González Sanabria, J. S., & Callejas Cuervo, M. (2012). Utilidad y funcionamiento de las bases de datos NoSQL. Tunja, Colombia.
- Comisiones. (25 de Noviembre de 2015). ¿Qué es una transferencia bancaria y cuál es su clasificación? España.
- Date, C. J. (2001). *Introducción a los sistemas de bases de datos*. Pearson Educación.
- Digital Ocean. (2020). Recuperado el 3 de febrero de 2020, de <https://www.digitalocean.com/products/droplets/>
- Fuentes, J. R. (2015). *Desarrollo de Software Ágil: Extreme Programming y Scrum. 2ª Edición*. IT Campus Academy.
- Global Nutrition, R. (2019). *Honduras Nutrition Profile*.
- K., A. (30 de octubre de 2013). Cómo vender un producto en Internet.
- Kazandzhy, L. (2019). *Freelance Web Developer 101: How to Start, Grow, and Succeed in Freelance Web Development from A to Z*. Howexpert.
- Lindley, C. (2019). *Front-end Developer Handbook*. Frontend Masters.
- Luis, V. (16 de Mayo de 2020). ¿Qué es un carrito de compras online? Lima, Perú: Pearson Prentice Hall.

- Luna, F. (2019). *JavaScript - Aprende a programar en el lenguaje de la red*. Buenos Aires: Six Ediciones.
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software. Tesis de Licenciatura en Sistemas y*. Buenos Aires.
- Manzoor, A. (2012). *Relational Databases: Design, Implementation, and Application Development*.
- Mestres, A. F. (2014). *Introducción al frontend y backend*. Cataluña: Creative Commons.
- PayPal. (s.f.). *PayPal*. Recuperado el 25 de Octubre de 2018, de PayPal: <https://www.paypal.com/hn/webapps/mpp/what-is-paypal>
- RedHat. (2020). Obtenido de <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- RedHat. (29 de octubre de 2020). *Cloud Computing*.. Obtenido de Red Hat: <https://www.redhat.com/es/topics/cloud-computing/what-is-paas>
- Rouse, M. (2019). *Tech Target*. Obtenido de <https://searchcloudcomputing.techtarget.com/definition/DigitalOcean>
- Sanpieri, R., Fernandez Collado, C., & Baptista Lucio, P. (2014). *Métodología de la Investigación*. México D.F.: McGRAW-HILL.
- Twain, M. (1998). *The Wit and Wisdom of Mark Twain*. Dover Thrift Editions.