



Código:
NA

Versión:
1.0

Fecha:
10/03/24

Documento de Especificación de Arquitectura

Realizado por:

Santiago Monsalve
Daniela Silva
Sebastián Zambrano



Código:
NA

Versión:
1.0

Fecha:
10/03/24

HISTORIAL DE REVISIONES

Fecha	Versión	Autor	Descripción	Revisado Por
28/02/24	1.0	Equipo	Inicio de realización	Santiago Monsalve



Código: NA	Versión: 1.0	Fecha: 10/03/24
-----------------------------	-------------------------------	----------------------------------

Contenido

1.	Documento de Arquitectura de Software.....	4
1.1.	Introducción	4
1.2.	Propósito	4
1.3.	Alcance	4
1.4.	Referencias	4
1.5.	Definiciones acrónimos y abreviaciones	5
2.	Generalidades del Proyecto	6
2.1.	Problema a Resolver	6
2.2.	Descripción General del Sistema a Desarrollar	6
2.3.	Identificación de los Stakeholders y sus responsabilidades	6
3.	Vistas de la arquitectura	8
3.1.	Vista de Casos de Uso	8
3.2.	Vista de Procesos	9
3.3.	Vista Lógica	13
3.4.	Vista de Implementación	14
3.5.	Vista de Despliegue	14
4.	Arquitectura en capas.....	15
5.	Vista de Datos	17
5.1.	Modelo Relacional	17
6.	Definición de Interfaces de Usuario.....	18
7.	Características Generales de Calidad	21
8.	Stack Tecnológico	22
9.	Referencias	23



Código:
NA

Versión:
1.0

Fecha:
10/03/24

1. Documento de Arquitectura de Software

1.1. Introducción

En el mercado de aplicaciones que guían a los consumidores de alimentación y nutrición, no se encuentra actualmente una aplicación dedicada al consumo de suplementos deportivos. Por ello, se desarrollará una aplicación la cual podrá guiar y asistir al consumidor de suplementos deportivos. Para ello, se implementará un catálogo para la exploración y la búsqueda específica de los diferentes productos disponibles en el mercado, así como otras características funcionales que más adelante se detallarán.

1.2. Propósito

Desarrollar una aplicación móvil que ejerza como una guía para el consumidor de suplementos deportivos.

1.3. Alcance

El proyecto pretende desarrollar una aplicación móvil que guíe al consumidor de suplementos deportivos para que decida mejor al adquirir un producto específico. Esto en un plazo máximo de 3 años.

1.4. Referencias

1. Documento de Especificación de Requerimientos no funcionales.

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Velocidad del sistema
Descripción del requerimiento:	La velocidad de respuesta del sistema será óptima al consultar los suplementos disponibles en el aplicativo.
Prioridad del requerimiento:	Alta

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Actualizaciones
Descripción del requerimiento:	El sistema se mantendrá actualizado conforme a los avances tecnológicos que presentan los sistemas operativos (Android – iOS).
Prioridad del requerimiento:	Alta



Código: NA	Versión: 1.0	Fecha: 10/03/24
----------------------	------------------------	---------------------------

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Veracidad de la información
Descripción del requerimiento:	La información del sistema debe ser verídica, igual a la reportada en la etiqueta impresa en el producto.
Prioridad del requerimiento:	Alta

Identificación del requerimiento:	RNF04
Nombre del Requerimiento:	Actualizaciones del catálogo
Descripción del requerimiento:	El sistema incluirá actualizaciones al catálogo conforme salgan suplementos nuevos en el mercado.
Prioridad del requerimiento:	Alta

1.5. Definiciones acrónimos y abreviaciones

ARQUITECTURA DE SOFTWARE: conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente, este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

DESCRIPCION DE ARQUITECTURA: colección de productos de documentación.

VISTAS: es una representación de un área de interés o perspectiva del sistema en alto nivel.

TIPOS DE VISTAS: especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

STAKEHOLDER: Individuo, equipo u organización con intereses relativos en el sistema.

Código:
NAVersión:
1.0Fecha:
10/03/24

2. Generalidades del Proyecto

2.1. Problema a Resolver

Uno de los aspectos más importantes para llevar a cabo un estilo de vida saludable es la práctica regular de actividad física. Esta se constituye en un factor de protección contra diversas enfermedades y aporta numerosos beneficios para la salud (Organización Mundial de la Salud [OMS], 2020).

Entre las personas que profundizan en este estilo de vida, ha venido en aumento formas de complementar el desarrollo de la actividad física por medio de la suplementación deportiva. La suplementación deportiva es la actividad de consumir productos que ayuden a cubrir los requerimientos nutricionales adicionales de las personas que practican actividades físicas con frecuencia. Estos son utilizados tanto para mantener un buen estado de salud, como para mejorar el rendimiento durante la práctica del ejercicio (Palacios et al., 2019).

A pesar de que la tendencia de consumir suplementos deportivos ha venido en aumento en los últimos años a una tasa de crecimiento anual compuesto de 7.9% entre 2004 y 2018 (Oster, 2019), las aplicaciones que existen en el mercado actual para llevar a cabo estilos de vida saludable están enfocadas en la alimentación: como contadores de calorías y seguimiento de la ingesta diaria de cada persona; en el ejercicio: como guías de entrenamiento y seguimiento de objetivos; o en la descripción teórica de las funciones de los diferentes tipos de suplementos deportivos. Sin embargo, no se encuentra una aplicación dedicada especialmente a la descripción de cada suplemento deportivo disponible en el mercado.

2.2. Descripción General del Sistema a Desarrollar

La Guía de Suplementos Deportivos será una aplicación móvil diseñada para ayudar al consumidor de suplementos deportivos con un catálogo que le permitirá consultar varios productos disponibles en el mercado.

2.3. Identificación de los Stakeholders y sus responsabilidades

STAKEHOLDER	DESCRIPCIÓN	ESCENARIO	Caso de Uso
Usuario	Es el usuario final, persona natural, interesado en la información	<ul style="list-style-type: none"> Escenario de servicios 	<ul style="list-style-type: none"> C.U.1 (Registrar Usuario)



Código: NA	Versión: 1.0	Fecha: 10/03/24
	específica (productos) de la suplementación deportiva.	<ul style="list-style-type: none"> • C.U.2 (Iniciar Sesión) • C.U.3 (Gestionar Perfil) • C.U.4 (Consultar Catálogo)



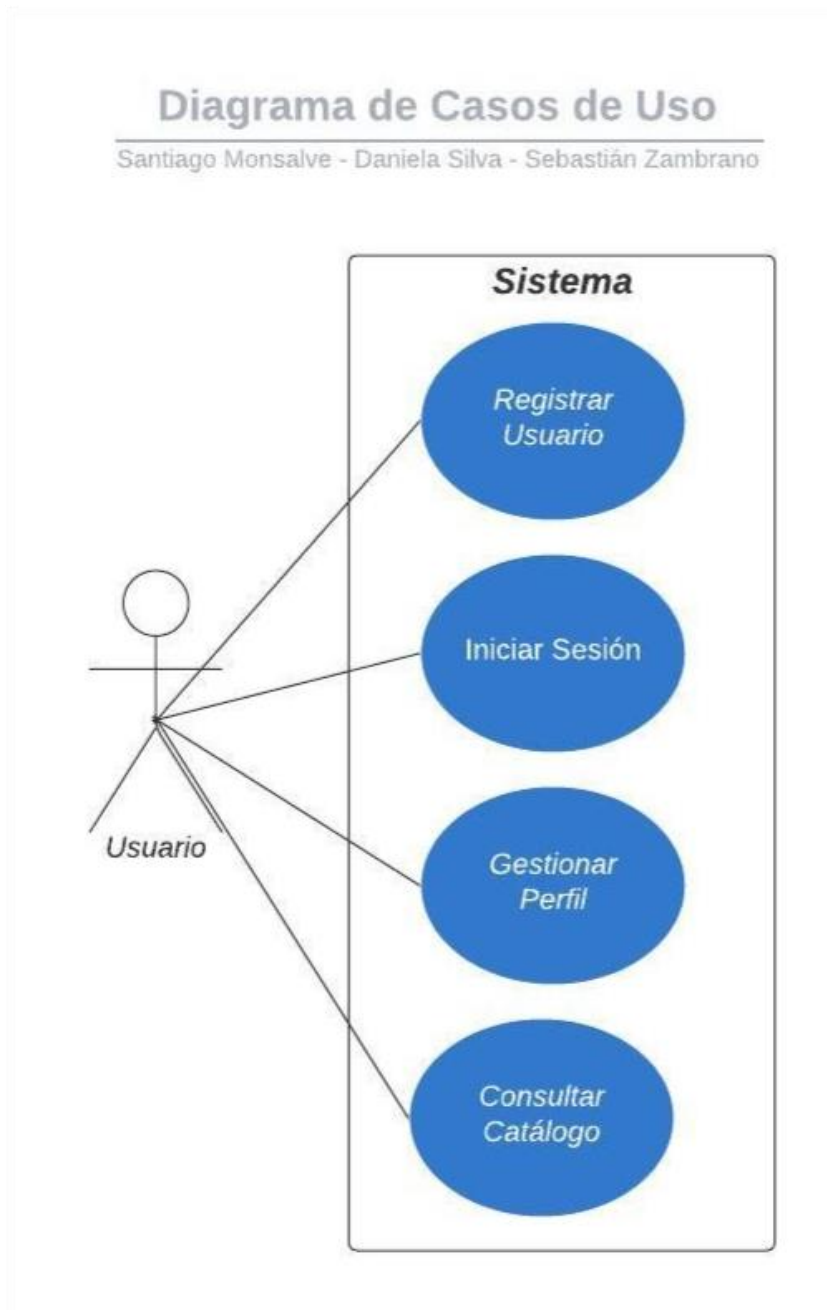
Código:
NA

Versión:
1.0

Fecha:
10/03/24

3. Vistas de la arquitectura

3.1. Vista de Casos de Uso





Código:
NA

Versión:
1.0

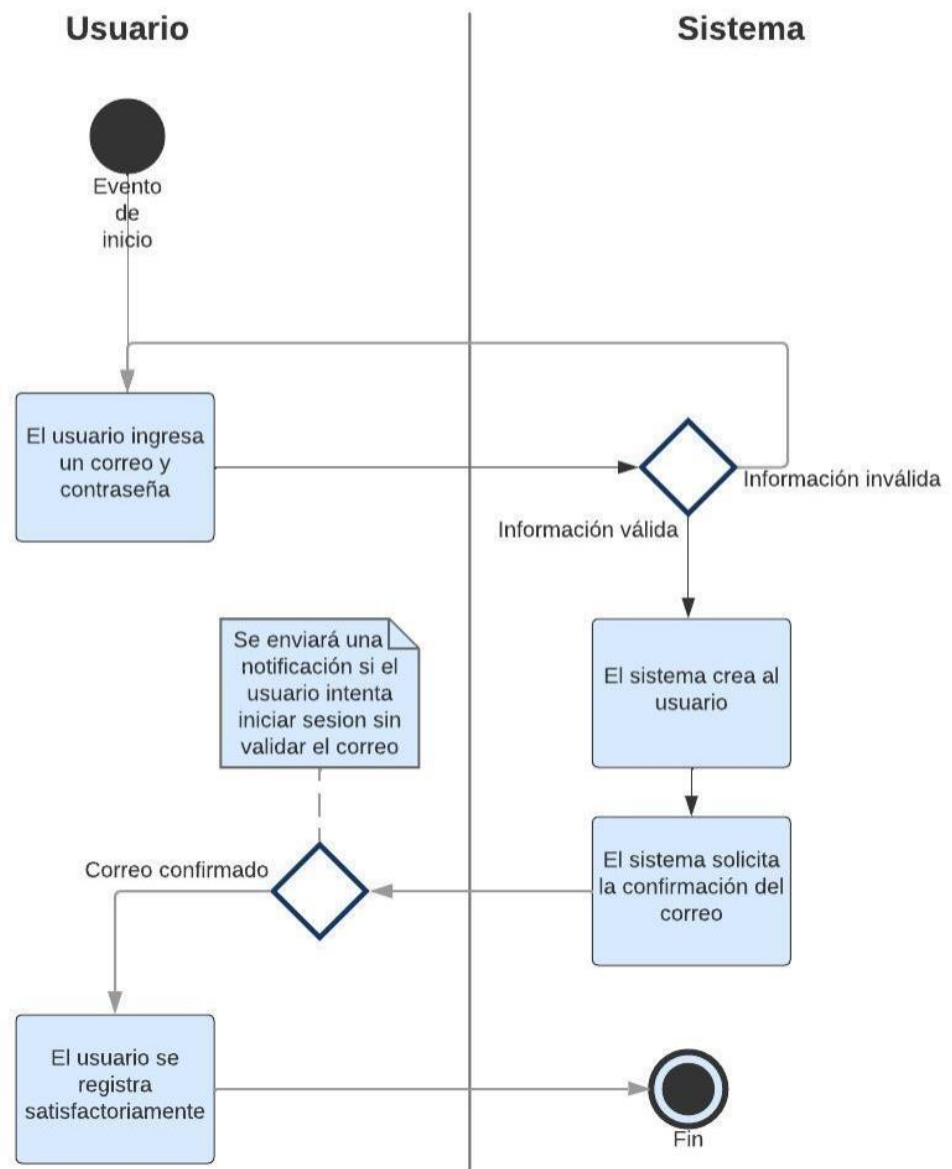
Fecha:
10/03/24

3.2. Vista de Procesos

3.2.1. Diagrama de Actividades

Diagrama Actividad - Registrar Usuario

Santiago Monsalve - Daniela Silva - Sebastián Zambrano





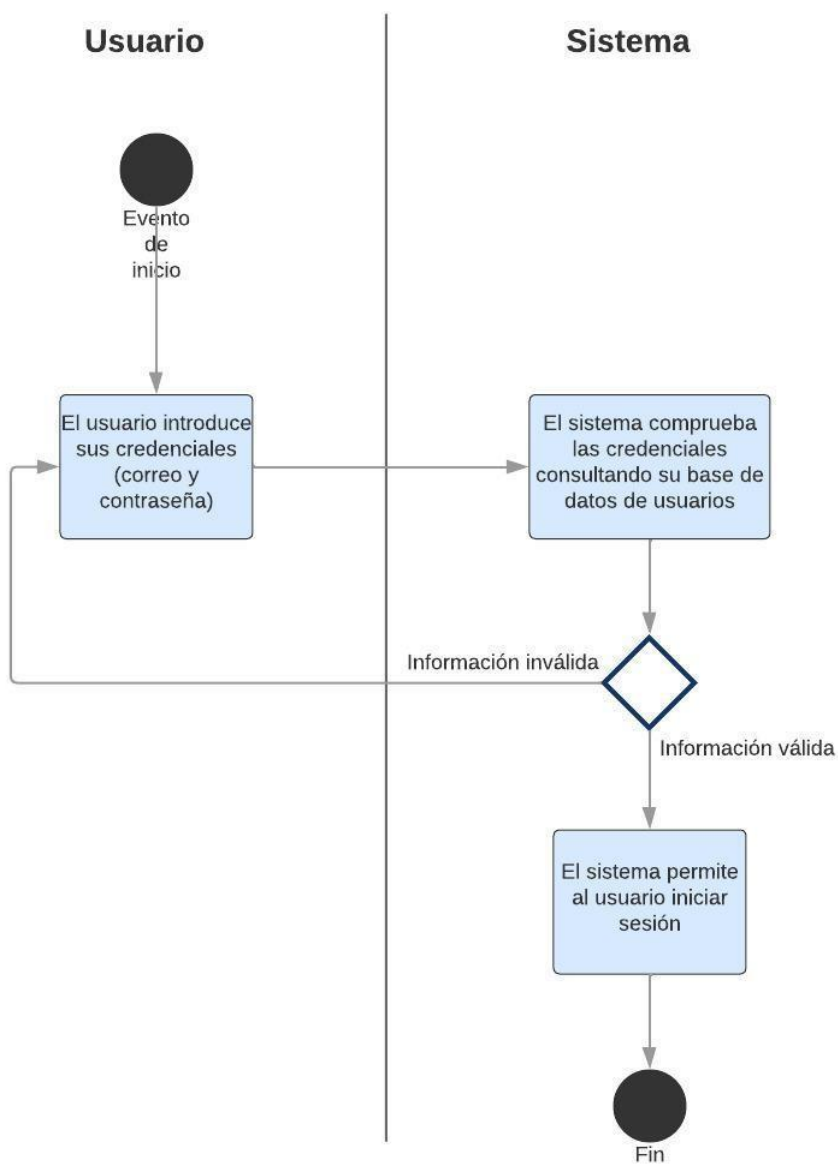
Código:
NA

Versión:
1.0

Fecha:
10/03/24

Diagrama Actividad - Iniciar Sesión

Santiago Monsalve - Daniela Silva - Sebastián Zambrano





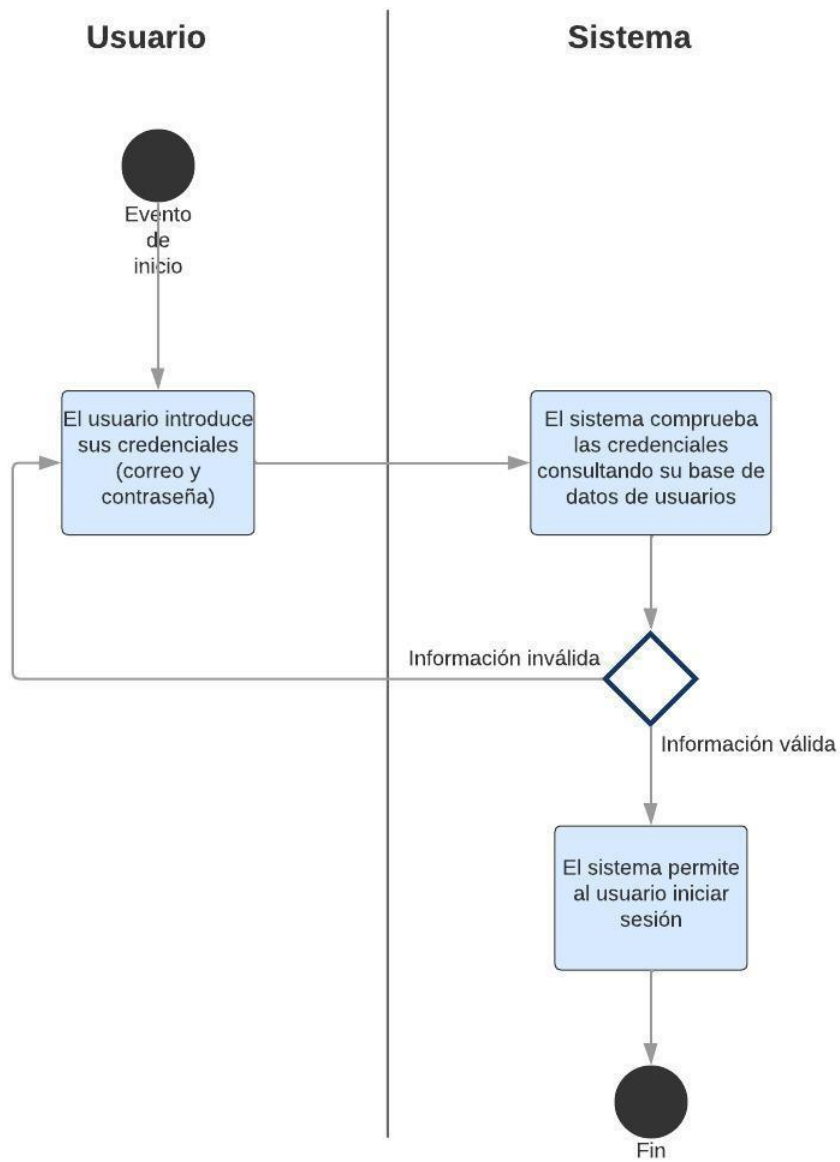
Código:
NA

Versión:
1.0

Fecha:
10/03/24

Diagrama Actividad - Iniciar Sesión

Santiago Monsalve - Daniela Silva - Sebastián Zambrano





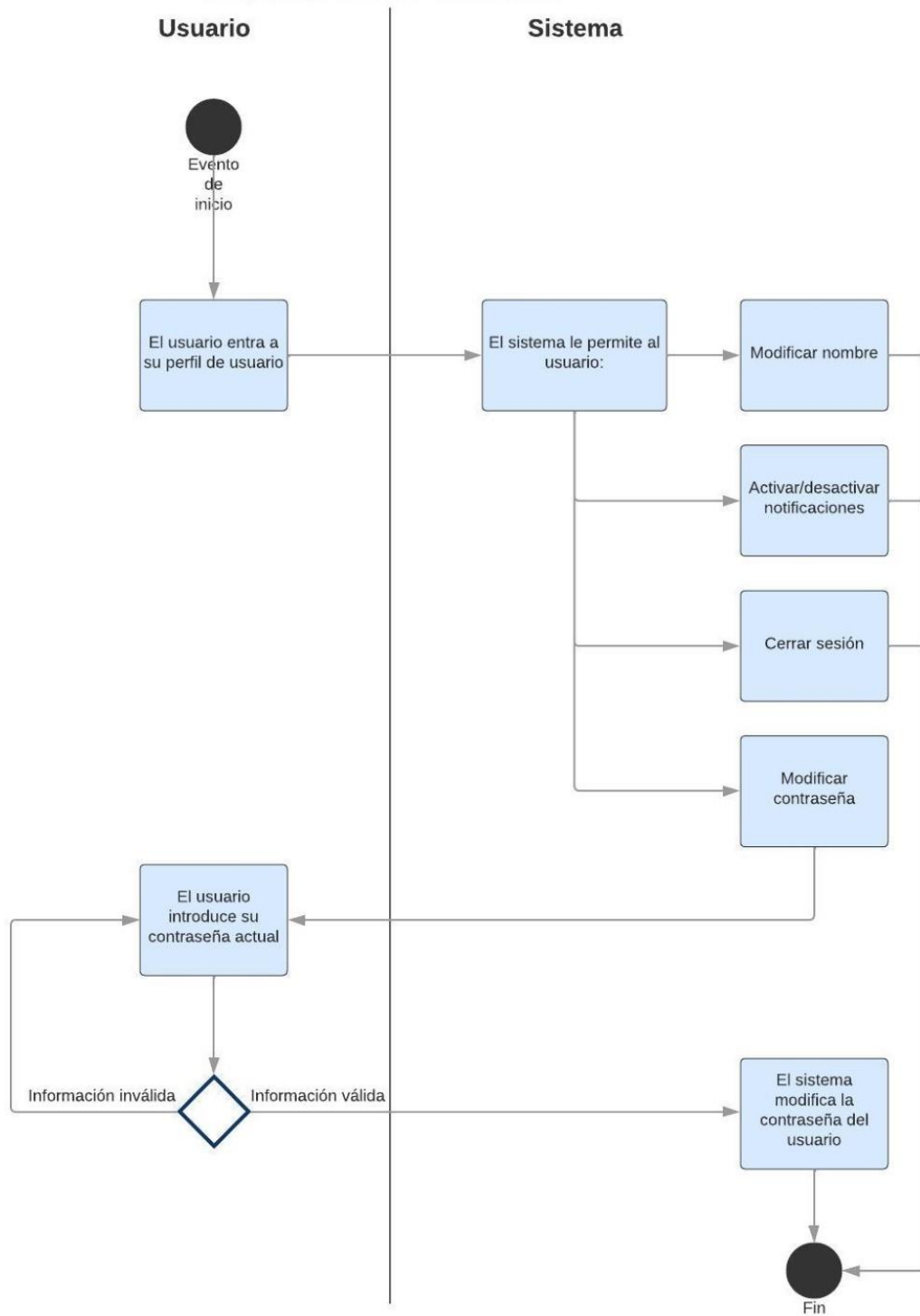
Código:
NA

Versión:
1.0

Fecha:
10/03/24

Diagrama Actividad - Gestionar Perfil

Santiago Monsalve - Daniela Silva - Sebastián Zambrano





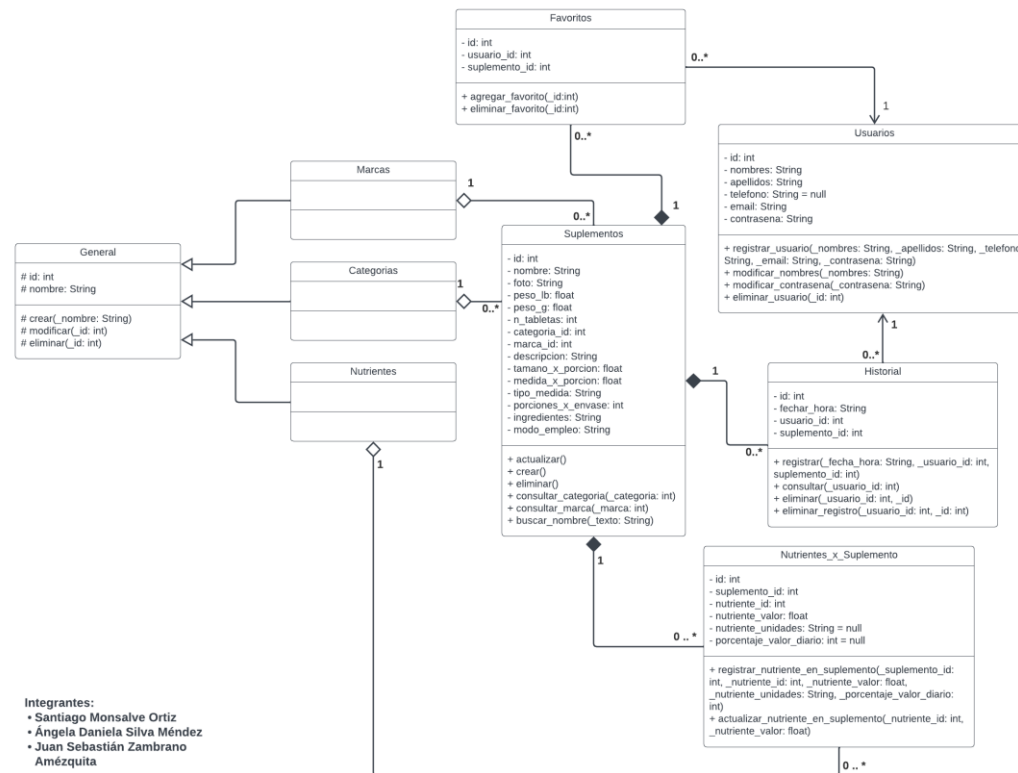
Código:
NA

Versión:
1.0

Fecha:
10/03/24

3.3. Vista Lógica

3.3.1. Diagramas – Clases



Nota: favor hacer zoom para apreciar correctamente la información, la imagen cuenta con buena resolución.



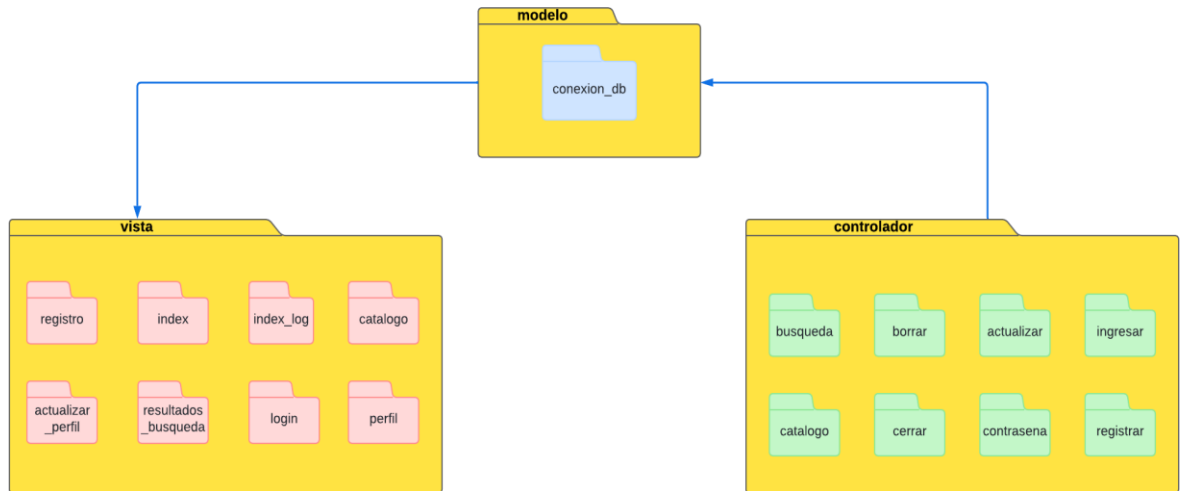
Código:
NA

Versión:
1.0

Fecha:
10/03/24

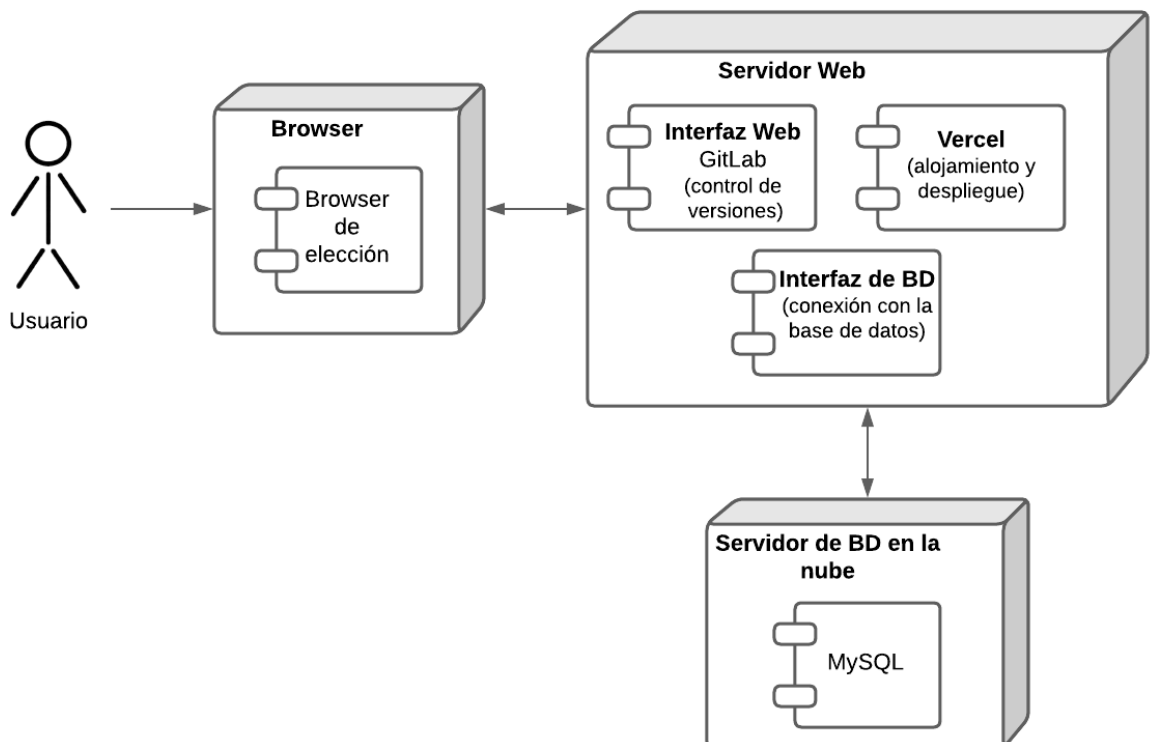
3.4. Vista de Implementación

3.4.1. Diagrama de Paquetes



3.5. Vista de Despliegue

3.5.1. Diagrama de despliegue





Código: NA	Versión: 1.0	Fecha: 10/03/24
----------------------	------------------------	---------------------------

4. Arquitectura en capas ("Modelo Vista-Controlador")

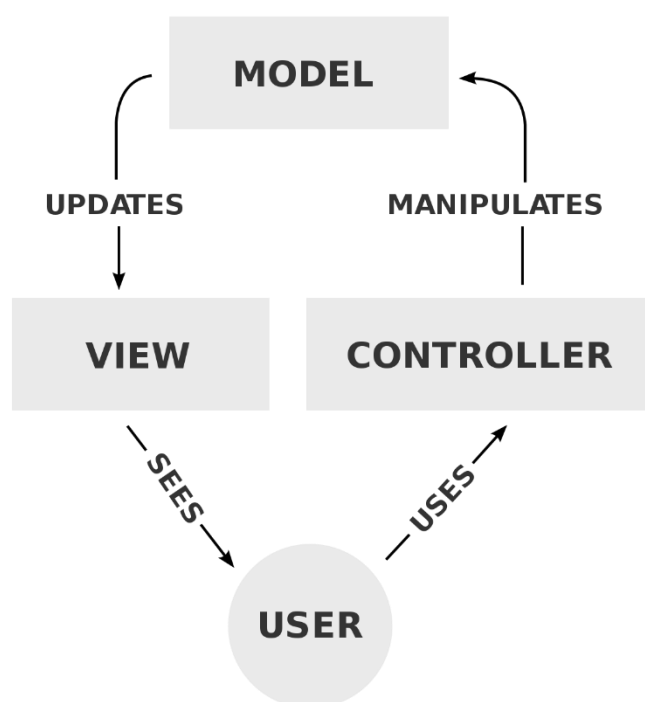
(capas, patrones, plataforma)

Respecto al apartado del patrón de arquitectura/diseño seleccionado, se eligió el modelo MVC (Modelo-Vista-Controlador). Microsoft, en su apartado de patrón MVC para ASP.NET, lo define como un patrón de diseño utilizado para "desacoplar" -es decir, separar- las funciones de la interfaz de usuario (vista), la información o data (modelo) y la lógica informática (controlador) de una misma entidad como podría ser una aplicación web; esto para así facilitar su diseño, construcción y mantenimiento (Microsoft).

Siguiendo el paradigma de programación orientada a objetos, cada uno de los elementos de este patrón se definen en clases diferentes:

- Modelo: clase encargada de la interacción con la base de datos.
- Vista: clase encargada de proveer "salida" al usuario para ser mostrada.
- Controlador: clase encargada de recibir "entrada" del usuario y procesarla.

Y la naturaleza de su relación se puede apreciar en la siguiente imagen:



Nota: diagrama de interacciones de una forma de implementar el patrón MVC.

<https://commons.wikimedia.org/wiki/File:MVC-Process.svg>

La adopción del Modelo-Vista-Controlador (MVC) en nuestra aplicación web ofrece beneficios significativos en términos de estructura, mantenimiento y escalabilidad. Este enfoque proporciona una clara separación de



Código: NA	Versión: 1.0	Fecha: 10/03/24
----------------------	------------------------	---------------------------

responsabilidades, mejorando la modularidad y facilitando la colaboración eficiente entre desarrolladores.

Razones para Implementar el Modelo MVC:*

- **Separación de Responsabilidades:** la arquitectura MVC divide la aplicación en tres componentes principalmente, facilitando la gestión de la lógica de negocio (Modelo), la presentación (Vista) y el control del flujo (Controlador). Esto garantiza una clara separación de responsabilidades, lo que resulta en un código más organizado y mantenible.
- **Escalabilidad:** la modularidad inherente del MVC permite escalar cada componente de manera independiente. Esto es esencial para adaptarse a cambios en los requisitos y facilita la expansión gradual de la aplicación sin afectar otras partes del sistema.
- **Reutilización de Código:** al dividir la aplicación en capas claramente definidas, se fomenta la reutilización de código. Los componentes del Modelo, la Vista y el Controlador pueden utilizarse en diferentes contextos, reduciendo la duplicación y aumentando la eficiencia en el desarrollo.
- **Mantenimiento Simplificado:** la estructura modular del MVC facilita la identificación y corrección de errores. Los desarrolladores pueden centrarse en áreas específicas de la aplicación sin interferir con otras partes, lo que agiliza el proceso de mantenimiento y actualización.
- **Facilita la Colaboración:** la clara definición de roles en el MVC facilita la colaboración entre equipos de desarrollo. Los diseñadores pueden trabajar en la interfaz de usuario (Vista) sin afectar la lógica de negocio (Modelo), y los desarrolladores pueden implementar nuevas características sin alterar la presentación.



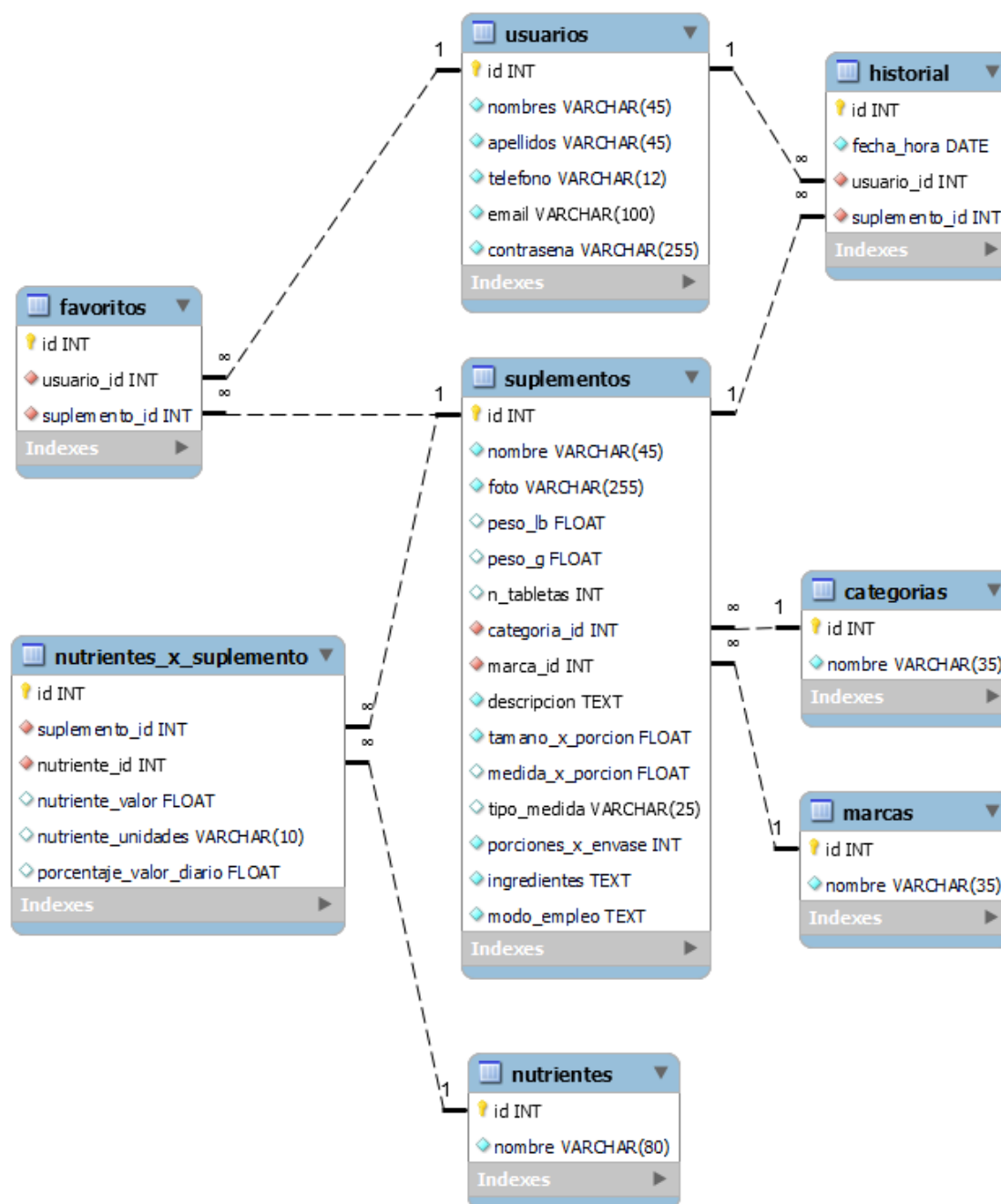
Código:
NA

Versión:
1.0

Fecha:
10/03/24

5. Vista de Datos

5.1. Modelo Relacional





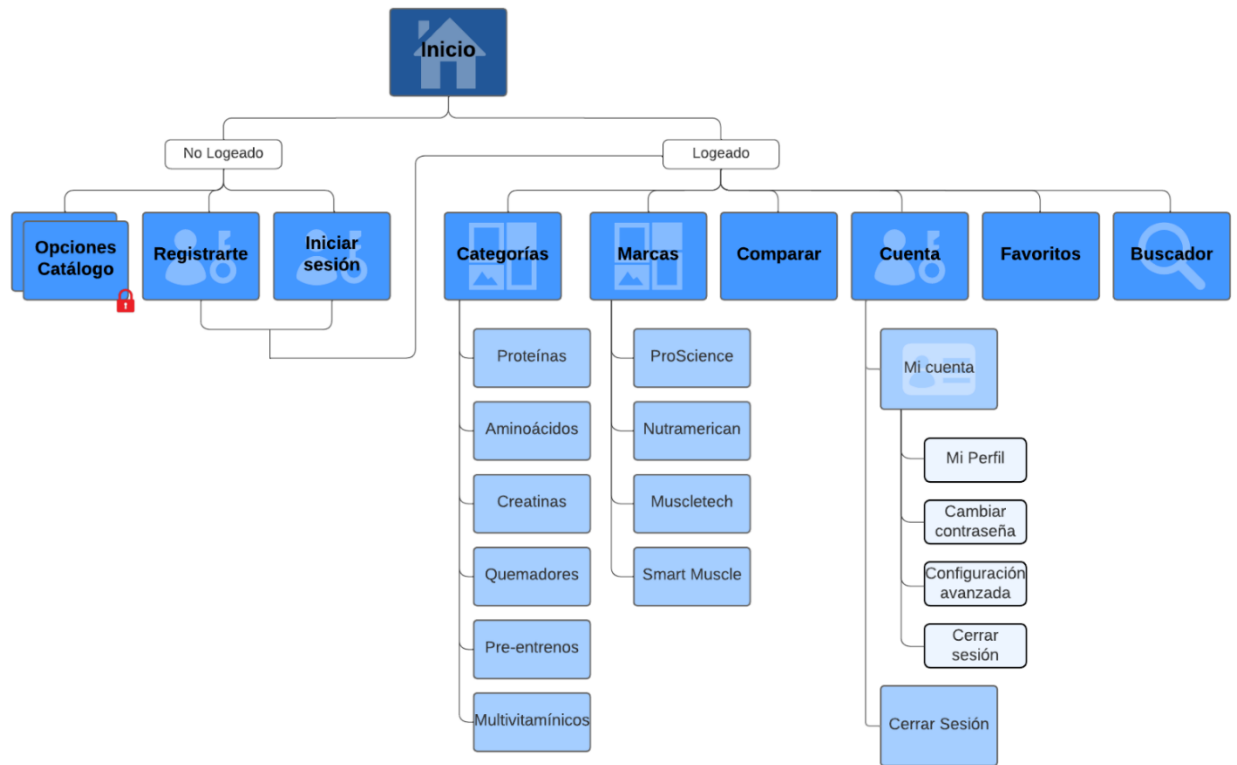
Código:
NA

Versión:
1.0

Fecha:
10/03/24

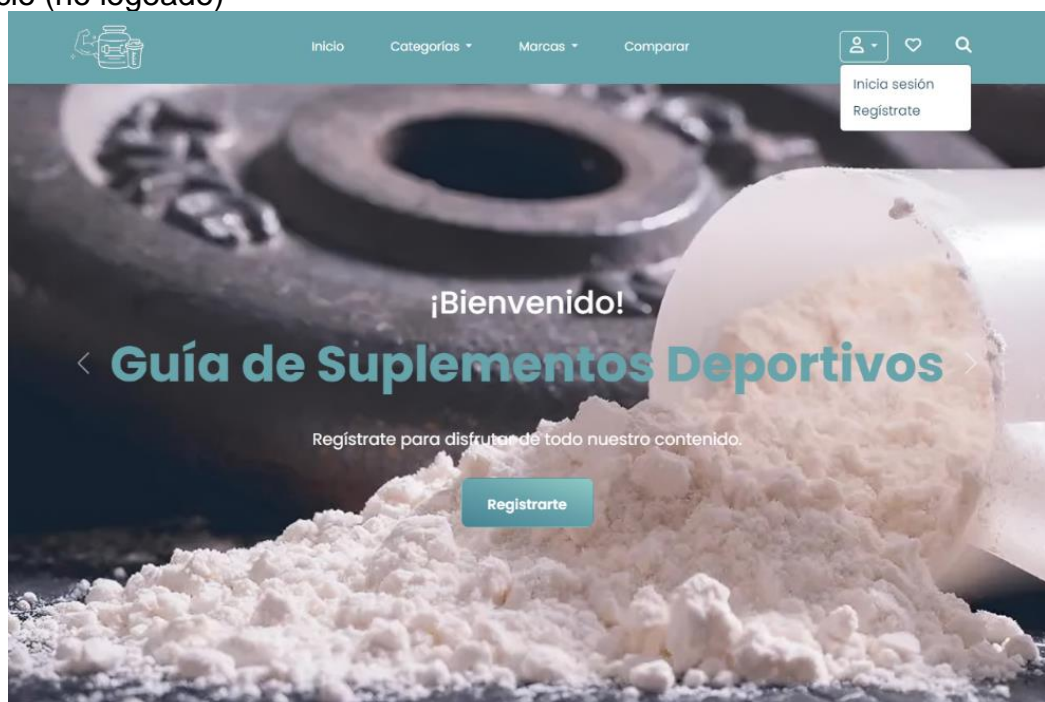
6. Definición de Interfaces de Usuario

- Mapa de navegación



- Demostración de las interfaces

Inicio (no logeado)





Código:
NA

Versión:
1.0

Fecha:
10/03/24

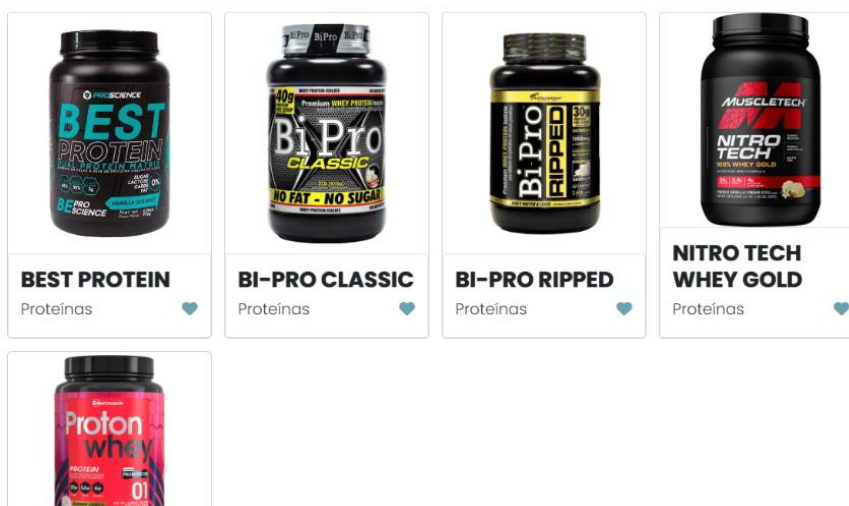
Inicio (logueado)



Categorías



Proteínas





Código:
NA

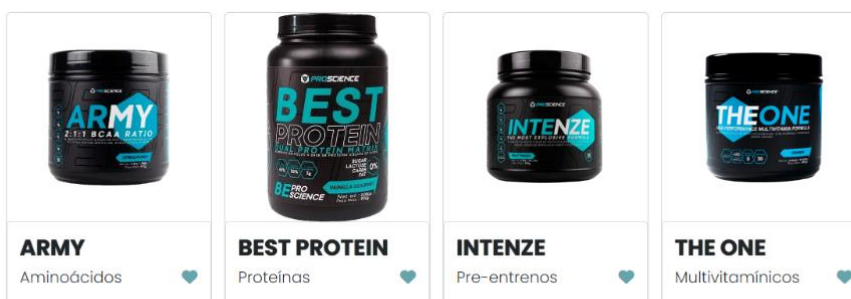
Versión:
1.0

Fecha:
10/03/24

Marcas



ProScience



Mi cuenta





Código:
NA

Versión:
1.0

Fecha:
10/03/24

7. Características Generales de Calidad

- **Compatibility (compatibilidad):** Definida como la capacidad de adaptarse a diferentes entornos, nuestra aplicación será capaz de ser utilizada en diferentes navegadores, asegurando una experiencia consistente y eficaz para los usuarios, independientemente del sistema que utilicen.
- **Correctness (exactitud):** Definida como la capacidad de ser preciso, nuestra aplicación aportará información correcta al usuario, garantizando que los datos proporcionados sean exactos y confiables, lo que contribuirá a una toma de decisiones informada y precisa.
- **Customizability (personalización):** Definida como la capacidad de cambiar de acuerdo con las preferencias, nuestra aplicación permitirá iniciar sesión al usuario, lo cual le permitirá recibir un servicio más personalizado al incluir funcionalidades como guardar favoritos. Esto asegurará que la experiencia del usuario se adapte a sus necesidades individuales y preferencias.
- **Integrity (integridad):** Definida como la calidad de mantener la coherencia y la exactitud de los datos. En el contexto de nuestra aplicación, nos aseguraremos de que la información y los procesos se mantengan íntegros, evitando alteraciones no autorizadas.
- **Responsiveness (sensibilidad):** Definida como la capacidad de responder rápidamente a las interacciones del usuario. En nuestra aplicación, la



Código: NA	Versión: 1.0	Fecha: 10/03/24
-----------------------------	-------------------------------	----------------------------------

sensibilidad se refiere a la velocidad con la que la interfaz responde a las acciones del usuario, garantizando una experiencia fluida y sin demoras perceptibles.

- **Simplicity (sencillez):** Definida como la cualidad de ser fácil de entender y usar. Nuestra aplicación se diseñará con una interfaz sencilla y sin complicaciones innecesarias, facilitando así la experiencia del usuario, incluso para aquellos con poca experiencia.
- **Interoperability (interoperabilidad):** Definida como la capacidad de nuestro sistema para funcionar de manera eficiente con otros sistemas o aplicaciones. Garantizaremos que nuestra aplicación pueda colaborar sin problemas con diferentes plataformas, mejorando la integración y la comunicación con otras herramientas.
- **Upgradability (actualizabilidad):** Definida como la capacidad de nuestro software para recibir y aplicar actualizaciones de manera fácil y eficiente. Nuestra aplicación permitirá realizar actualizaciones sin problemas, garantizando que los usuarios siempre tengan acceso a las últimas características y mejoras.
- **Flexibility (flexibilidad):** Definida como la capacidad de adaptarse y ajustarse a diferentes situaciones o requisitos. Nuestra aplicación ofrecerá flexibilidad al usuario, permitiéndole personalizar su experiencia y adaptar ciertos aspectos según sus preferencias individuales.
- **Scalability (escalabilidad):** Definida como la capacidad de nuestro sistema para crecer y manejar un aumento en la carga de trabajo. Nos aseguraremos de que nuestra aplicación sea escalable, permitiendo un rendimiento eficiente incluso cuando se incrementa el número de usuarios o la complejidad de las operaciones.

8. Stack Tecnológico

Describe en este apartado detalles sobre los lenguajes de programación, frameworks, bases de datos y otras herramientas tecnológicas relevantes para el desarrollo del software, servicios para servidor en la nube (pagina, base de datos).

Para el desarrollo de la aplicación, se va a utilizar PHP como lenguaje de programación debido a su versatilidad y amplia adopción en el desarrollo web. Este lenguaje en el aplicativo proporciona una gran flexibilidad y potencia para implementar diversas funcionalidades, así como también permite aprovechar las ventajas de la programación orientada a objetos, lo que facilita la modularidad y reutilización del código. Además, PHP cuenta con una amplia comunidad de



Código: NA	Versión: 1.0	Fecha: 10/03/24
-----------------------------	-------------------------------	----------------------------------

desarrolladores y una gran cantidad de recursos disponibles, lo que brinda un sólido respaldo para el desarrollo del aplicativo.

En cuanto al framework, se ha optado por utilizar Laravel como marco de desarrollo backend. Este framework cuenta con una amplia documentación, además permite desarrollar la aplicación utilizando una sintaxis elegante y una estructura organizada, entre otras ventajas. Esto es de alta importancia para el progreso y estructura del software que se está desarrollando.

Para el frontend se va a utilizar Bootstrap ya que facilita el diseño web responsivo y la interacción de interfaces de usuario atractivas y funcionales, que para el desarrollo de la aplicación es muy útil su implementación.

Con respecto a la base de datos se ha venido utilizando MySQL Workbench, esta base de datos ha permitido poder desarrollar de manera local código abierto y en donde su interfaz es sencilla y fácil de utilizar, además ha permitido gestionar de manera eficaz los datos del proyecto. No obstante, una vez se tenga totalmente sólida y funcional la base de datos, esta migrara a un servidor de datos en la nube con el fin de poder tener de manera remota un entorno de nube híbrida a fin de ayudar a organizar, almacenar, y administrar los datos de la aplicación de manera más asequible y donde la seguridad prioricé en la base de datos.

También se va a implementar toda la estructura de desarrollo de la aplicación en Gitlab, esto con el objetivo de proporcionar un conjunto de herramientas completas para la gestión de vida del desarrollo del aplicativo, lo que permite tener un control total sobre la infraestructura de desarrollo. Con GitLab se logrará colaborar, gestionar y desplegar el proyecto de manera más efectiva.

Cabe agregar que junto a GitLab se va a implementar el uso de la herramienta llamada Vercel que es una plataforma de alojamiento y despliegue para aplicaciones web. Esta se integra estrechamente con GitLab lo cual facilita el proceso de desarrollo, que para la aplicación ayuda a simplificar y automatizar el proceso de mejora y despliegue del aplicativo móvil.

9. Referencias

- Organización Mundial de la Salud. (2020). Directrices de la oms sobre actividad física y hábitos sedentarios. <https://apps.who.int/iris/bitstream/handle/10665/337004/9789240014817-spa.pdf>
- Oster, M. (2019). The Expansion of Sports Nutrition. Euromonitor International. <https://www.euromonitor.com/article/the-expansion-of-sports-nutrition>



Código: NA	Versión: 1.0	Fecha: 10/03/24
-----------------------------	-------------------------------	----------------------------------

- Palacios, N., Manonelles, P., Blasco, R., Contreras, C., Franco, L., Gaztañaga, T., Manuz, B., Galván, C., Soto, M., García, A., Villegas, J. (2020). Suplementos nutricionales para el deportista. Ayudas ergogénicas en el deporte - 2019. Documento de consenso de la Sociedad Española de Medicina del Deporte. *Arch Med Deporte*, 36 (1), 7-83. <https://archivosdemedicinadeldeporte.com/articulos/upload/Doc-consenso-ayudas-2019.pdf>
- Microsoft. ASP.NET MVC Pattern. [https://dotnet.microsoft.com/en-us/apps/aspnet/mvc#:~:text=Model%20View%20Controller%20\(MVC\),to%20achieve%20separation%20of%20concerns.](https://dotnet.microsoft.com/en-us/apps/aspnet/mvc#:~:text=Model%20View%20Controller%20(MVC),to%20achieve%20separation%20of%20concerns.)