

Inutrition

Alejandro Cifuentes Montoya y Alejandro Gallo Murillo

UNIVERSIDAD CATÓLICA DE ORIENTE

FACULTAD DE INGENIERÍAS

PROGRAMA DE INGENIERÍA DE SISTEMAS

RIONEGRO

2021



Inutrition

Alejandro Cifuentes Montoya y Alejandro Gallo Murillo

Trabajo para obtener el título de ingeniero de Sistemas

**Asesores**

Jaime de Jesús Guzmán Gómez

Universidad Católica de Oriente

Facultad de Ingenierías

Programa de Ingeniería de Sistemas

Rionegro

2021

**Nota de aceptación:**

---

---

---

---

---

**Firma del presidente del jurado**

---

**Firma del jurado**

---

**Firma del jurado**

**Dedicatoria:**

**Agradecimientos:**

Agradecemos a la universidad por todo el conocimiento impartido a nosotros, a profesores que nos enseñaron todas las bases para poder entender y solucionar problemas como ingenieros, tanto en las bases matemáticas como de gestión de desarrollo de software. Agradecemos también a la profesora Luz Mery Rios por estar siempre pendiente de nosotros como docente y como coordinadora de la facultad, siempre apoyándonos en los procesos formativos y profesionales; y por último pero no menos importante, agradecemos al profesor Jaime de Jesús Guzman por asesorarnos en todo el proceso de realización de este proyecto, por instruirnos en los sprint planning, por estar pendiente del avance del proyecto en cada reunión y por compartir con nosotros su conocimiento acerca de la gestión de proyectos.

## Contenido

<b>Antecedentes:</b>	<b>8</b>
<b>Planteamiento del problema:</b>	<b>11</b>
<b>Justificación:</b>	<b>12</b>
<b>Objetivos:</b>	<b>12</b>
General:	12
Específicos:	12
<b>Marco teórico</b>	<b>13</b>
Viabilidad	13
Deficiencias en el conocimiento	13
Definiciones - Sistema	13
Definiciones - Nutrición	15
<b>Diseño metodológico</b>	<b>17</b>
<b>Resultados</b>	<b>20</b>
Requisitos funcionales	20
Requisitos no funcionales	25
Historias de Usuario	26
Diseño	40
Desarrollo	44
<b>Conclusiones</b>	<b>45</b>
<b>Referencias bibliográficas</b>	<b>45</b>

## 1. Antecedentes:

Existen diferentes programas o software que tienen como finalidad la optimización del tiempo en la valoración nutricional, teniendo en cuenta la evaluación nutricional (cálculos antropométricos), elaboración de un plan de alimentación y seguimiento del objetivo del paciente, también funciones como la adición de un ejemplo de menú, posibles recetas a incluir en el plan de alimentación y adición de alimentos con su composición. En la actualidad, existen diferentes programas de nutrición que cuentan con características similares en función de facilitar la consulta nutricional de manera que la planificación de un plan de alimentación, valoración nutricional, educación nutricional, seguimiento y comunicación paciente-profesional sea más práctico en un mínimo tiempo. Dentro de los aplicativos de software que cumplen con las características mencionadas anteriormente son: (Nutrimind), (Dietopro, 2019), (Aznutrition), (diet-creator), (dietfarma), (enPeso).

- *Nutrimind*: Nutrimind es el software más usado por Nutriólogos y estudiantes Desde el año de 2007, se comercializa con gran éxito en todo América latina y España. Todos los resultados que genera el software, son basados en bibliografías de gran reconocimiento en Nutrición. Sus funciones de diseño de dietas se basan en el sistema mexicano de equivalentes y en la USDA.

Nutrimind es usado por miles de nutriólogos y estudiantes en México y todo el mundo de habla hispana, además de universidades e instituciones de investigación, como el Instituto Nacional de Ciencias Médicas y Nutrición Salvador Zubirán.(Información Obtenida de la página oficial de la aplicación)

- *DietoPro*: Es una aplicación móvil que está enfocado en los clientes, esta permite a los pacientes consultar sus planes dietéticos, acceder a las recetas e ingredientes, intercambiar sus platos, ver sus recomendaciones dietéticas, gestionar lista de compra, seguir la evolución de su peso y grasa corporal y consultar su próxima cita. Es gratuita para los pacientes y los nutricionistas pueden estar en contacto con sus pacientes. (Información Obtenida de la página oficial de la aplicación)

- *Az nutrition*: Az nutrition es la herramienta ideal para manejar de forma profesional los 5 elementos básicos de una consulta de nutrición: antropométrico, bioquímicos, clínicos, dietéticos y expediente. Permite elaboración de menús: permite realizar menús por medio de porciones o equivalentes, además lo acompañan herramientas que permiten ir seleccionando los ingredientes o escribirlos, permitiendo que las indicaciones del menú queden a gusto del usuario. Además, permite el coste de platillos y menús: maneja una lista de 2300 alimentos, los cuales puedes ingresar el precio de compra y al realizar menús te indica el costo de preparación. También tiene elaboración de platillos: Genera una cantidad infinita de platillos, y conoce su valor nutricional y su distribución de equivalentes, te da la opción de agregar su fotografía y modo de preparación. Otra de las funcionalidades es proyectos de investigación. La base de datos de alimentos, es completamente editable y también permite la adición de nuevos alimentos. Permite el respaldo de información para que esta no se pierda.
- *Diet\_creator*: Software de análisis nutricional: Cálculo predictivo. Reduce radicalmente el tiempo de creación de un plato o dieta gracias al motor de cálculo nutricional predictivo que te adelanta el resultado para que puedas ajustar lo más posible. El programa toma en cuenta el porcentaje del valor diario (%VD) y los macros y micro nutrientes que has establecido como objetivos.
- *Compatibilidad alimentaria*: El programa aprende qué es bueno para tus clientes. Ve todas las alertas que se crean cuando la interacción de tu cliente con los alimentos o medicamentos no es la apropiada o es peligrosa. Otras alertas te garantizan la idoneidad de lo que le estás recomendando, y otras: sus preferencias. (Información Obtenida de la página oficial de la aplicación)
- *Dietfarma*: Es una aplicación móvil y web que sirve para poder conocer su estado nutricional actual, permite realizar un seguimiento a sus objetivos, lleva un control de las

calorías que se ingieren y que se queman, visualizar el % porcentaje de micronutrientes de la dieta

- *EnPeso*: Es una APP móvil para pacientes con las siguientes características:

Registro diario

Planes alimentarios

Chat en tiempo real: Envía consultas al profesional y obtiene respuestas rápidas sobre tu tratamiento.

Sigue tu evolución: Gráficos con información sobre tu tratamiento. Peso, IMC, Masa grasa y libre de grasa. (Información Obtenida de la página oficial de la aplicación)

Sin embargo, nuestro proyecto sería una de las primeras aplicaciones en Colombia de este tipo, enfocada a mejorar la atención de las citas entre paciente-dietista, ya que permite generar más rápido las dietas, de acuerdo a los datos ingresados de la persona atendida, teniendo en cuenta posibles enfermedades o alergias a ciertos productos. Además, las aplicaciones mencionadas anteriormente trabajan con estándares alimenticios respectivos a su país, o a un país en específico, mientras nuestra aplicación contendrá los estándares alimenticios colombianos que se usan hasta la fecha, teniendo en cuenta los alimentos que se pueden encontrar en nuestro país de manera cotidiana.

La aplicación será web tanto para nutricionistas como pacientes y contará con una versión móvil a la que los pacientes podrán acceder con facilidad y podrán ver la información de sus dietas y procesos actualizados mediante el registro de este.

## **2. Planteamiento del problema:**

Desde hace varios años los nutriólogos vienen realizando los cálculos de las valoraciones y las creaciones de dietas utilizando tablas físicas y realizando los cálculos a mano, pasando también por tablas de cálculo en Excel, volviéndose un archivo demasiado extenso y poco

productivo, ya que para realizar consultas de las referencias alimenticias se gasta mucho tiempo para realizar las valoraciones lo que afecta la atención a los pacientes.

Según lo que nos dicen nuestros compañeros nutriólogos es que se vuelve muy complicado el proceso por la cantidad de cálculos que se necesitan para llegar a finalizarlo. En el proceso de la elaboración de un plan de alimentación es necesario ser específicos con los antecedentes familiares, personales y clínicos, así como la valoración actual realizada en cada consulta para indicar una alimentación suficiente. Además de los cálculos antropométricos, métodos de evaluación de ingesta por medio del recordatorio de 24 horas, junto con la frecuencia de consumo, así como el cálculo de los macronutrientes, micronutrientes, vitaminas y minerales, es también necesario ajustar la alimentación del paciente, este “lo determinan de manera fundamental los hábitos o tradiciones alimentarias” los cuales son definidos por la disponibilidad a los alimentos que tengan según la ubicación geográfica.

Lo que llevaremos a cabo con nuestro proyecto, será proporcionar un software el cual ayudará a optimizar el tiempo abordado en cada consulta, ayudando en la calidad de la atención ya que la aplicación permitirá mejorar el tiempo de consulta y realización de cálculos necesarios, además proporciona organización en el almacenamiento y fácil acceso de la información, tanto de los estándares alimenticios como de los pacientes a los que se está atendiendo, las valoraciones y diagnósticos que se les hace, además de las recetas que el nutricionista le puede recomendar a sus pacientes, esta información le será facilitada al paciente por medio de la aplicación ya que podrá visualizar sus diagnósticos y fórmulas dietéticas de manera que las pueda tener al alcance en el momento que desee.

### **3. Justificación:**

El software que se plantea a desarrollar será de ayuda para optimizar el tiempo abordado en cada consulta, mejorando la calidad de la atención realizando los cálculos necesarios,

además proporciona organización en el almacenamiento y fácil acceso de la información, teniendo en cuenta las morbilidades y ambiente de cada paciente.

A nivel económico, teniendo en cuenta la optimización del tiempo en consulta, que permite dar un servicio más eficiente, la cantidad de pacientes que se pueden atender en una jornada podrá aumentar significativamente y los seguimientos de estos serán de manera más efectiva ya que se lleva toda la historia con avances en la aplicación.

A nivel social se dará una mejor comunicación entre el nutricionista y el paciente además de aportar al ambiente cultural con las recetas que se montaran en la aplicación de acuerdo a las necesidades y condiciones de salud de los pacientes a los que se les hace valoraciones.

#### **4. Objetivos:**

##### **4.1. General:**

- Desarrollar un Software de tipo web para el apoyo al profesional de salud, en el análisis nutricional de pacientes y la creación del plan nutricional personalizado.

##### **4.2. Específicos:**

- Identificar la mejor arquitectura para la realización del proyecto.
- Realizar el diseño detallado de la estructura de la aplicación.
- Implementar el diseño con la arquitectura escogida de manera que cumpla con los requerimientos del proyecto.
- Implementar el diseño de manera que la aplicación sea cómoda y entendible para los profesionales de salud de manera que puedan realizar sus funciones de manera eficiente

## **5. Marco teórico**

### **Viabilidad**

Es bastante viable ya que no está enfocado a un solo proceso, sino que puede ser implementado para varios de forma que mejora la productividad y calidad en el servicio que brinda el nutricionista. Por ende, el software tendrá una gran disposición y brindará datos actualizados para que el trabajo del nutricionista sea más eficaz.

### **Deficiencias en el conocimiento**

Desconocimiento de los cálculos que se realizan para llevar a cabo las dietas y los análisis nutricionales, por esto se lleva a cabo el proyecto con los compañeros estudiantes de nutrición.

### **Definiciones - Sistema**

1. Software: Es un conjunto de aplicaciones que permite al computador realizar tareas para cumplir una o varias funciones.
2. APP: Aplicación
3. Dato: Información que puede ser valiosa e importante para realizar transacciones, almacenar, actualizar e incluso eliminar.
4. Aplicación Móvil: Aplicación que está construida para ser ejecutada en dispositivos móviles (Celulares, tabletas)

5. Aplicación WEB: Aplicación que está diseñada y construida para ser desplegada en un navegador y que puede hallarse en una búsqueda de internet.
6. Base de datos: Aplicación capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características establecidas.
7. Clase: Es un molde que representa objetos de la vida real, con los atributos y con las funciones que realiza y que son importantes para la aplicación.
8. Diagrama de Clases: Es un diagrama que representa las clases con las que se trabajará en la aplicación de forma que ayude al correcto funcionamiento de esta.
9. Diagrama de Comunicaciones: Es el diagrama que se encarga de mostrar cómo se comunican los componentes de la aplicación.
10. Diagrama de secuencias: Es un tipo de diagrama usado para modelar la interacción entre objetos en un sistema.
11. Diagrama de Componentes: Es el diagrama que muestra la estructura de la aplicación.
12. Diagrama de paquetes: Muestra las relaciones entre los paquetes que componen la aplicación.
13. Paquete: Es un contenedor que brinda orden a los archivos que estructuran la aplicación.
14. Diagramas de despliegue: Se utiliza para modelar la disposición física de los artefactos software en nodos. Muestra la arquitectura del sistema así como el despliegue de los artefactos de software a los objetivos de despliegue.
15. Atributos de calidad: son características no funcionales que se consideran deseables en un sistema de software.
16. Historias de usuario: Son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente del sistema.

## **Definiciones - Nutrición**

1. Caloría: “Una kcal es la cantidad de energía necesaria para incrementar la temperatura de 1 litro de agua destilada de 14,5 °C a 15,5 °C; utilizada de manera habitual a nivel nacional e internacional, en energía humana y para expresar el contenido energético en alimentos” (resolución 3803).
2. Tasa metabólica basal (TMB) Se define como:” la cantidad de energía que el organismo utiliza para mantener, la actividad metabólica de las células y tejidos, la circulación sanguínea, la respiración, el tono muscular, la temperatura corporal y la función renal” (Correa, 2006).
3. Gasto Energético Total (GET) Es la cantidad de energía que una persona gasta cada día (24h) y el GET es el resultado de tres componentes: el gasto de la tasa metabólica basal, el gasto por actividad física y la termogénesis de los alimentos. ( (Castillo, 2006).
4. Nivel de Actividad Física (PAL) “gasto de energía de 24 horas expresado como un múltiplo de la TMB y se calcula como Gasto Energético Total para 24 horas según la intensidad” (resolución).
5. Índice de Masa Corporal (IMC) Significa índice de masa corporal, se define como “la relación entre el peso y la talla de una persona” (Alimentación y nutrición: manual teórico práctico, 2005).
6. Efecto Calorígeno de los alimentos “aumento del gasto energético causado por la ingesta de alimentos e incluye la energía necesaria para la digestión, absorción, transporte, metabolismo y almacenamiento de los macronutrientes” (resolución).
7. Evaluación Dietética revisión de la ingesta usual que tiene un paciente de los diferentes nutrientes, comparándolos con los valores de referencia (EAR, RDA, AI, UL), para determinar si la ingesta del individuo es adecuada, deficiente o excesiva, expresada como la probabilidad de inadecuación por déficit o riesgo de efectos adversos por una ingesta excesiva. (resolución).
8. Recomendaciones Diarias Adecuadas (RDA) “es una estimación del nivel de ingesta diaria promedio de un nutriente, que es suficiente para cubrir las necesidades de casi todos los individuos saludables de un grupo particular (97 - 98%) por período de vida, condición fisiológica y género” (resolución).

9. **Requerimiento Promedio Estimado (EAR)** “nivel de ingesta promedio diario estimado que cubre el requerimiento de la mitad de los individuos saludables (50%) de un grupo de edad, por periodo de la vida, estado fisiológico y género específicos” (resolución).
10. **Ingesta Adecuada (AI):** Término establecido por la FAO - OMS, representa la mejor estimación de la necesidad realizada con base a la ingesta aceptable. (Resolución 3803).
11. **Ingesta Máxima Tolerable (UL):** nivel de ingesta promedio más alto de la ingesta diaria de un nutriente que no genera riesgo de efectos adversos para la salud de la mayoría de las personas de la población en general; no se debe ingerir niveles superiores a los rangos establecidos.
12. **Rango Aceptable de Distribución de Macronutrientes (AMDR):** Es el rango de ingesta de una fuente de energía que se asocia con la reducción de enfermedad crónica mientras aporta cantidades adecuadas de nutrientes esenciales. Se expresa como porcentaje de ingesta total de energía. (resolución).
13. **Proteínas:** Las proteínas son macromoléculas complejas desde el punto de vista físico y funcional, que desempeñan funciones de transporte, estructura, transporte, inmunológico, catalizador, entre otras funciones. Las proteínas están compuestas por aminoácidos; los cuales son estructuras pequeñas que forman proteínas. Los seres humanos forman 11 de los 20 L - alfa - aminoácidos, los otros 9 se obtienen de la dieta. (Citar; Bioquímica de Harper).
14. **Lípidos:** Son un grupo de compuestos heterogéneos, incluyen grasas, aceites, esteroides, ceras y compuestos relacionados más por sus propiedades físicas que por sus propiedades químicas. Contienen propiedades que los vuelven hidrofóbicos e insolubles en agua, pero solubles en solventes no polares como el éter y cloroformo. (Bioquímica de Harper).
15. **Grasas:** Son un componente importante de la dieta, siendo una fuente esencial de energía metabólica, síntesis de compuestos con actividad metabólica y un regulador de la expresión de genes, además de que sirve como portador de nutrientes como las vitaminas liposolubles (A, D, E y K) y precursores de vitaminas en el tubo digestivo y plasma (Nutrición y Dieta, Libro). Se encuentran en el cuerpo principalmente como estrés de

grasas y aceites naturales, pero existen en una forma no esterificada como ácidos grasos libres (Bioquímica de Harper).

16. **Ácidos grasos:** Los ácidos grasos se encuentran en el cuerpo como ésteres en grasas y aceites naturales, pero existen en una forma no esterificada, denominada ácidos grasos libres (Bioquímica de Harper). Están compuestas por una cadena de hidrocarburo (acilo) con un grupo metilo y uno carboxilo en cada extremo, en forma de cadena recta. La mayor parte de los ácidos grasos de la dieta contienen entre 4 a 22 carbonos de largo. (Nutrición y Dieta, libro).
17. **Ácidos Grasos Saturados (AGS):** Los Ácidos Grasos Saturados son cadenas hidrocarbonadas con un solo enlace entre un átomo de carbono y el otro. Esta característica le permite tener cadenas rectas que se agrupan estrechamente, formando grumos sólidos a temperatura ambiente.
18. **Ácidos Grasos Insaturados:** Son las cadenas hidrocarbonadas que, dentro de su estructura en forma de cadena, tienen 1 o más dobles enlaces entre los átomos de carbono, provocando flexiones (curvas) en las cadenas de ácidos grasos. Los aceites son líquidos en temperatura ambiente (Biología de la tierra). Los ácidos grasos insaturados se dividen en ácidos grasos monoinsaturados (tienen un solo doble enlace) y ácidos grasos poliinsaturados.

## **6. Diseño metodológico**

1. Planear la estructura de la aplicación.
  - 1.1. Modelo de secuencia de la arquitectura de la aplicación.
  - 1.2. Modelar los procesos teóricos en la aplicación.
    - 1.2.1. Identificar los requerimientos del sistema.
      - 1.2.1.1. Funcionales.
      - 1.2.1.2. No funcionales.
    - 1.2.2. Generar diagrama de flujo del sistema.
      - 1.2.2.1. Funcionamiento perfecto sin fallos.

- 1.2.2.2. Funcionamiento con condiciones.
        - 1.2.2.3. Funcionamiento en caso de fallos.
      - 1.2.3. Generar modelado de dominio del sistema.
        - 1.2.3.1. Identificar los usuarios involucrados en el sistema.
        - 1.2.3.2. Identificar los procesos que realizan los usuarios en el sistema.
    - 1.3. Identificar los componentes necesarios para la aplicación.
      - 1.3.1. Identificar el tipo de base de datos a utilizar.
        - 1.3.1.1. Analizar si es base de datos relacional o no relacional.
        - 1.3.1.2. Diseñar modelo entidad relación (De ser necesario).
      - 1.3.2. Diseñar la arquitectura de la aplicación.
        - 1.3.2.1. Realizar modelado de despliegue.
    - 1.4. Implementar el modelado de dominio en programación.
      - 1.4.1. Implementar los componentes del dominio.
        - 1.4.1.1. Clases con atributos y métodos.
  - 2. Diseñar
    - 2.1. Mockups de la aplicación.
    - 2.2. Vista lógica
      - 2.2.1. Diagrama de clases.
      - 2.2.2. Diagrama de secuencias.
    - 2.3. Vista de despliegue
      - 2.3.1. Diagrama de componentes.
      - 2.3.2. Diagrama de paquetes.
    - 2.4. Vista física
      - 2.4.1. Diagrama de despliegues.
      - 2.4.2. Atributos de calidad.
    - 2.5. Vista de Escenarios
      - 2.5.1. Casos de uso.
  - 3. Desarrollo
    - 3.1. Frontend.

- 3.1.1. Implementar componentes básicos para el funcionamiento de la aplicación.
- 3.1.2. Implementar los servicios para la comunicación con el API REST del backend.
- 3.1.3. Implementación de los módulos para tener la aplicación mejor estructurada.
- 3.1.4. Implementar routing para la aplicación.
- 3.1.5. Implementar el login de la aplicación que controle los roles del sistema.
- 3.1.6. Hacer uso de los componentes de la librería fullcalendar.
- 3.1.7. Implementar formularios reactivos para la aplicación.
- 3.1.8. Implementar el uso de observables para el manejo de datos provenientes del backend.
- 3.1.9. Uso de las librerías para la emulación de programas tipo excel.
- 3.1.10. Estilizar componentes de la aplicación.
- 3.1.11. Hacer uso de la librería Sweet Alert para informar al usuario acerca de los procesos que realice el usuario en la aplicación.

## 3.2. Backend

- 3.2.1. Implementación de distribución de carpeta correspondientes a la arquitectura.
- 3.2.2. Implementación de entidades necesarias para el correcto funcionamiento de la aplicación.
- 3.2.3. Implementación de JPA para conexión con base de datos y manejo de información.
- 3.2.4. Implementación de seguridad por medio de la tecnología JWT y Spring Security.
- 3.2.5. Planteamiento e implementación de motor de base de datos para el almacenamiento de la información.
- 3.2.6. Implementación de patrones de diseño Command y Factory para estructurar los datos recibidos desde los servicios REST expuestos.

- 3.2.7. Implementación de control de excepciones para posibles errores durante el uso de la aplicación.
      - 3.2.8. Implementación de envío de correos por medio de Spring-Email y SMTP de gmail.
      - 3.2.9. Implementación de encriptado de contraseñas generadas aleatoriamente por el sistema.
- 4. Testeo
  - 4.1. Testear las peticiones HTTP por medio de aplicativos REST.
  - 4.2. Hacer análisis estático de código por medio de SonarLint, SonarQube y SonarCloud para la parte del backend. Y TSLint para la parte del frontend.
- 5. Despliegue
  - 5.1. Planteamiento y despliegue de ambiente linux por medio de un EC2 de AWS
  - 5.2. Implementación de docker para la generación de imágenes del proyecto
  - 5.3. Implementación de NGINX para despliegue de frontend y como balanceador de cargas.
  - 5.4. Implementación de motor de base de datos MySql por medio de un RDS de AWS
- 6. Revisión
  - 6.1. Cerciorarse de que los componentes de la aplicación se rendericen de manera correcta y tengan el funcionamiento adecuado.
  - 6.2. Verificar que la aplicación se encuentre realizando las peticiones HTTP de manera correcta.
- 7. **Drivers Arquitectónicos**
  - 7.1. **Atributos de calidad**
    - 7.1.1. **Disponibilidad:** El aplicativo debe estar disponible las 24 horas ya que los usuarios que quieran conocer su información y la de sus pacientes en caso de ser nutricionistas, estos podrán verla en cualquier momento, siempre y cuando tengan conexión a internet.
    - 7.1.2. **Seguridad:** El aplicativo debe ser seguro, permitir el ingreso solo a los usuarios registrados en la aplicación y manejar inicio de sesión, además de

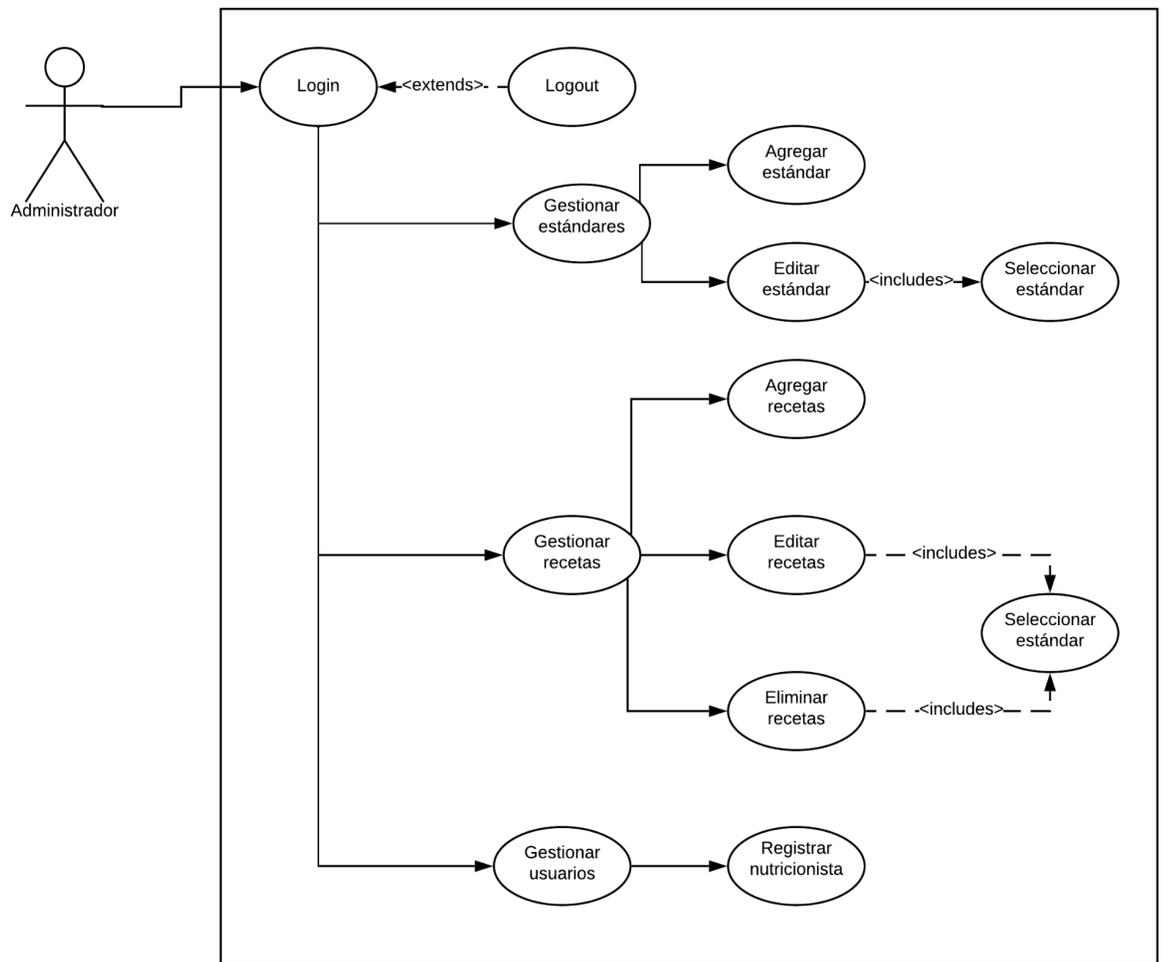
controlar que tipo de rol puede acceder a cierto tipo de información que puede ser sensible para los demás usuarios.

- 7.1.3. **Usabilidad:** El aplicativo debe ser usable por los diferentes usuarios de manera que sea entendible y fácil de usar de acuerdo a las capacidades de los diferentes tipos de usuarios

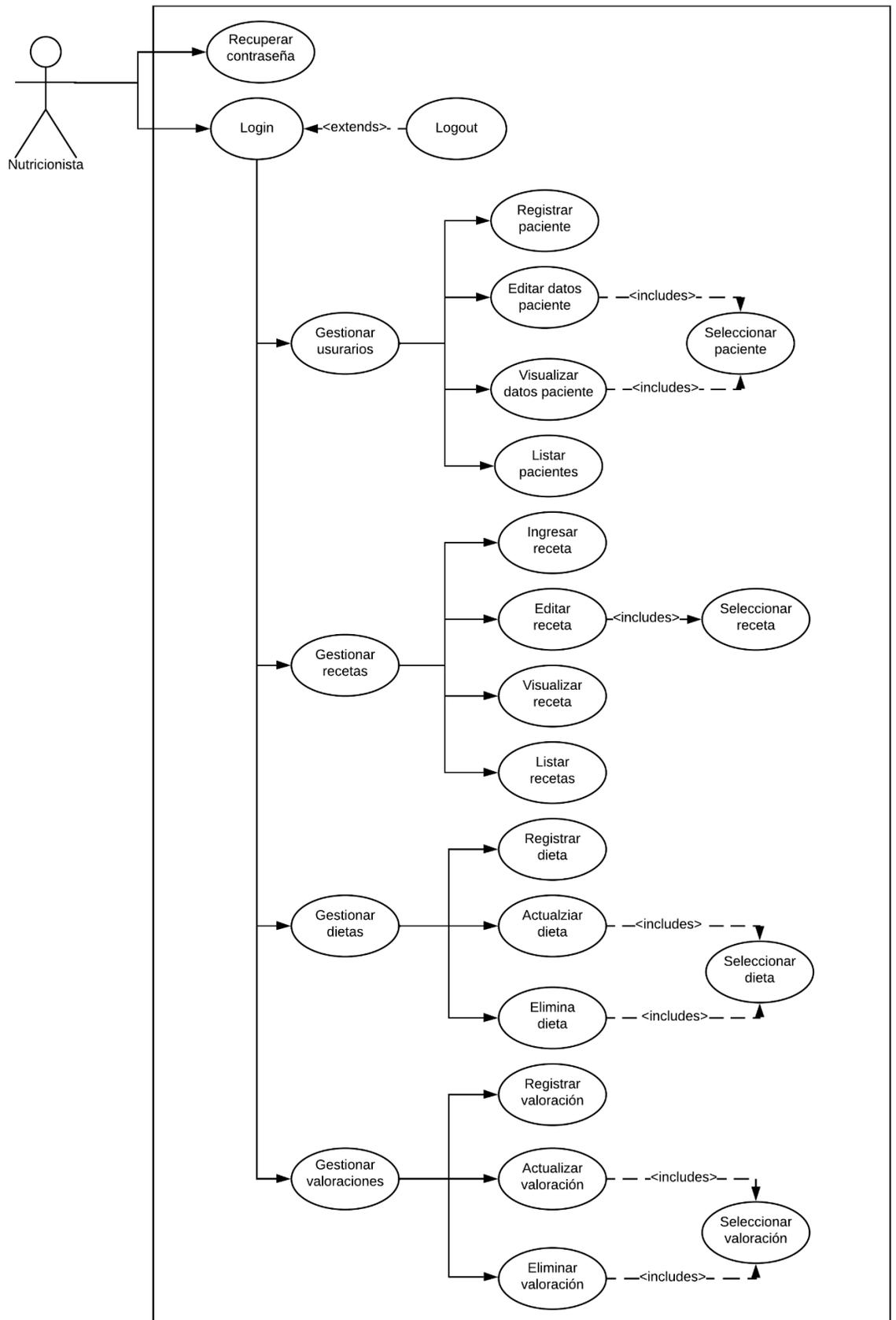
## 7.2. Casos de uso y restricciones

- 7.2.1. Si un usuario ingresa a la aplicación, esta debe identificar el tipo de usuario y redirigirlo a su menú en específico.
- 7.2.2. Si un usuario intenta ingresar a una ruta en específico a la cual no tiene permisos, el sistema debe redirigirlo a la página de inicio de la aplicación.
- 7.2.3. Si el usuario mantiene una sesión inactiva durante 30 minutos el sistema debe hacer logout del usuario.
- 7.2.4. Un usuario de tipo nutricionista solo puede ver y manipular la información y el registro de sus propios pacientes
- 7.2.5. Se debe llevar un historial de antropometrías de los pacientes las cuales se guarden automáticamente con la fecha y hora actual de acuerdo a la región en la que se encuentra
- 7.2.6. Se debe mostrar un spinner siempre que se hagan peticiones a servidores para evitar el envío de varias peticiones simultáneas innecesarias.
- 7.2.7. En el momento de que un usuario sea registrado en el aplicativo, se debe validar la existencia del correo, en caso de que no exista se debe retornar un error al usuario

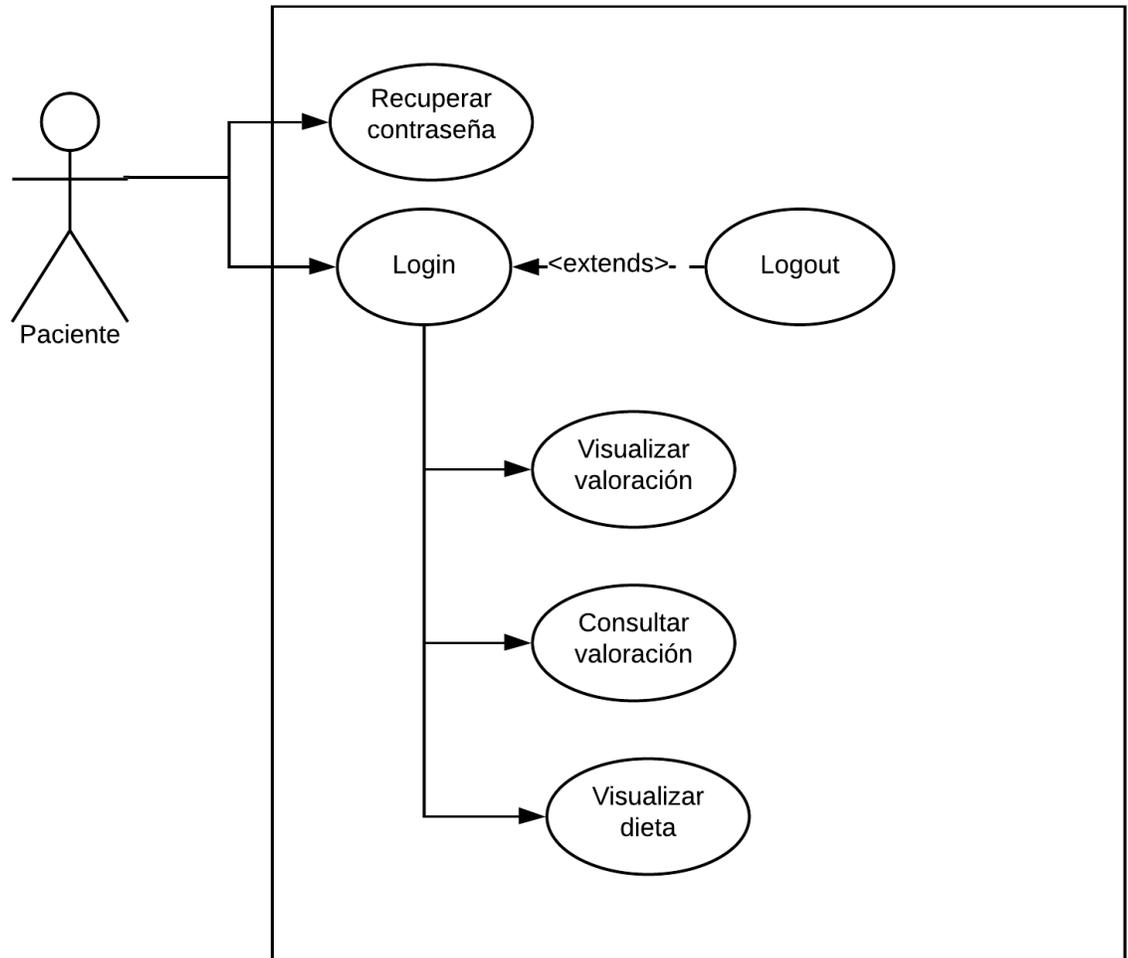
- 7.2.8.** En el momento de que un usuario sea registrado en el aplicativo y se valide la existencia del correo se debe enviar una contraseña generada automáticamente para que el usuario pueda realizar el primer ingreso
- 7.2.9.** Si un usuario intenta ingresar por primera vez al aplicativo con la contraseña enviada a su correo en el momento de registro, este deberá cambiar la contraseña para poder utilizar el aplicativo; en caso de que no cambie de contraseña no podrá realizar ninguna otra opción además de cerrar sesión.
- 7.2.10.** Si un usuario que no ha cambiado la contraseña intenta ingresar a otra página por medio de una ruta específica, el sistema deberá direccionar al usuario al menú de configuración de la cuenta para que realice el cambio de la contraseña.



**Figura: Casos de uso para el rol administrador**



**Figura: Casos de uso para el rol nutricionista**



**Figura: Casos de uso para el rol Paciente**

## 7. Resultados

### 7.1. Requisitos funcionales

7.1.1. Almacenar información de credenciales de usuarios y los permisos de cada rol.

- 7.1.2.** Almacenar las recetas con nombre, número de porciones, calorías, proteínas, grasas, carbohidratos, comida del día, ingredientes, preparación y foto de la receta.
- 7.1.3.** Almacenar el registro de los pacientes con los parámetros de nombre, apellidos, género, identificación, fecha de nacimiento, correo electrónico, teléfono, escolaridad, ocupación, estado civil, alergias, medicamentos que esté consumiendo, enfermedades de las que padezca, preferencias alimentarias, disgustos alimentarios, deporte que practica, frecuencia práctica, información sobre familiares enfermos, información sobre si ha sido hospitalizado, fecha de la hospitalización, tipo de paciente, identificación del paciente e identificación del nutricionista que realiza el registro.
- 7.1.4.** Se debe almacenar la información del registro de nutricionista con los campos nombre, apellido, identificación, fecha de nacimiento, correo electrónico, teléfono, contraseña, tarjeta profesional e información sobre posgrados realizados.
- 7.1.5.** Se debe almacenar la información del registro de administrador con los campos de nombre, apellido, identificación, especialidad, correo electrónico, celular y fecha de nacimiento.
- 7.1.6.** Se debe almacenar las fotos de seguimiento de los pacientes con información de la identificación del paciente, foto, fecha de la foto y la descripción que el nutricionista desee agregar.
- 7.1.7.** Se debe almacenar información sobre las consultas teniendo en cuenta la fecha de la consulta, la hora, el lugar, motivo de la consulta, identificación del paciente e identificación del nutricionista que atiende la consulta.
- 7.1.8.** Se debe almacenar información de frecuencia de consumo con grupo de alimento, la frecuencia de consumo, la medida casera, la medida real y la identificación del paciente.
- 7.1.9.** Se debe guardar información del complemento como nombre del complemento, la marca, dosis, cantidad, método de consumo y la identificación del paciente.

- 7.1.10.** Se debe guardar la información de la programación dietética con fórmula, calorías, restricción calórica, porcentaje de restricción, peso ajustado, peso saludable y la identificación del paciente.
- 7.1.11.** Se debe registrar la información del alimento propósito con APM, marca, dosis, cantidad, método de consumo y la identificación del paciente
- 7.1.12.** Se debe guardar información del suplemento como nombre del suplemento, marca, dosis, cantidad, método de consumo y la identificación del paciente.
- 7.1.13.** Se debe guardar la fórmula sintética de los pacientes con la información de nutrientes, porcentajes, calorías, gramos y la identificación del paciente.
- 7.1.14.** Se debe almacenar la información de la programación sintética con los datos de proteínas, grasas, carbohidratos e identificación del paciente.
- 7.1.15.** Se debe almacenar la información del dato antropométrico de cada paciente con los datos de peso, talla, subescapular, cresta iliaca, muslo anterior, tricípital, supraespinal, pierna media, bicípital, abdominal, diámetro de fémur, diámetro de humero, perímetro brazo corregido, perímetro pierna corregida e identificación del paciente.
- 7.1.16.** Se debe almacenar la información del tipo de examen con los datos de hemoglobina glicada, glicerina, glucosa en ayunas, creatinina suero, triglicéridos, colesterol total, colesterol LDL, colesterol HDL, calcio, potasio, sodio, albúmina y hemograma.
- 7.1.17.** El software debe permitir realizar una valoración dietética completa de manera dinámica y eficiente.
- 7.1.18.** La aplicación debe permitir el cálculo calórico, de macronutrientes y micronutrientes.
- 7.1.19.** La aplicación debe permitir la elaboración de cálculo antropométrico restringido (Pliegues, diámetros y perímetros para el cálculo de somatocarta).
- 7.1.20.** Proporcionar recetas saludables.
- 7.1.21.** Gestionar alimentos de acuerdo a la información en el banco de alimentos que se está actualizando de manera periódica.

- 7.1.22.** Debe permitir registrar administradores con nombres, apellidos, especialidad, correo electrónico y contraseña.
- 7.1.23.** La aplicación debe permitir que los administradores puedan editar el contenido de la aplicación, como las recetas para agregar más contenido para los usuarios.
- 7.1.24.** La aplicación debe permitir crear nutricionistas con nombres, apellidos, correo electrónico, celular, tarjeta profesional y especialidad.
- 7.1.25.** La aplicación debe manejar roles para que se puedan activar las diferentes funciones de acuerdo al rol del usuario y a las funciones que tendría.
- 7.1.26.** Gestionar pacientes
- 7.1.27.** La aplicación debe guardar las fotografías del avance en el estado de los pacientes.
- 7.1.28.** Gestionar calendario de programación de citas de pacientes.
- 7.1.29.** Debe proporcionar un menú de administrador en el que se le permita realizar las operaciones de creación, eliminación, actualización de datos y consulta de los diferentes usuarios del sistema. como del sistema de recetas.
- 7.1.30.** Debe proporcionar un registro para cada usuario en el que solo el usuario Nutricionista puede registrarse de manera autónoma mientras que los administradores solo pueden ser registrados por otro administrador y los pacientes solo pueden ser registrados por un usuario Nutricionista, los administradores pueden registrar Nutricionistas y Pacientes solo en casos especiales o por proporcionar ambientes de prueba de la aplicación.
- 7.1.31.** La aplicación debe permitir que cada usuario pueda cambiar sus datos personales básicos como el correo, celular, foto y contraseña.
- 7.1.32.** La aplicación debe listar las recetas registradas en el sistema categorizándolas por su tipo, de acuerdo a la comida del día.
- 7.1.33.** Se debe dar la opción al nutricionista de poder agregar más recetas al recetario de forma que pueda recomendarles a sus pacientes.
- 7.1.34.** La aplicación debe contar con el apartado de antropometría del paciente seleccionado donde de acuerdo a datos ingresados como peso, talla y subescapular

pueda mostrar las gráficas correspondientes de acuerdo al tipo de paciente seleccionado.

- 7.1.35.** La aplicación debe permitir visualizar en detalle la receta seleccionada para su correspondiente seguimiento por parte de los usuarios de la aplicación. Si se ingresa al sistema como usuario Nutricionista se puede ver en el apartado de citas, la programación de consultas que cada paciente seleccionado ha tenido hasta el momento.
- 7.1.36.** Si se ingresa al sistema como usuario Nutricionista se puede ver en el apartado de calendario las citas que tiene programadas con sus pacientes.
- 7.1.37.** Si el usuario Nutricionista ha ingresado al sistema, en este se debe poder visualizar los datos personales de los pacientes que fueron registrados en el sistema como lo son nombre, apellidos, género, identificación, fecha de nacimiento, correo electrónico, teléfono, escolaridad, ocupación, estado civil, alergias, medicamentos que esté consumiendo, enfermedades de las que padezca, preferencias alimentarias, disgustos alimentarios, deporte que practica, frecuencia práctica, información sobre familiares enfermos, información sobre si ha sido hospitalizado, fecha de la hospitalización, tipo de paciente, identificación del paciente e identificación del nutricionista que realiza el registro.
- 7.1.38.** El nutricionista debe contar con la opción de agregar fotos de seguimiento del paciente con la fecha y una descripción.
- 7.1.39.** El usuario de tipo nutricionista debe tener la opción de registrar la lista de opciones de alimentos para el paciente con información del tipo de comida, la hora de alimentación, las medidas caseras y reales y la preparación de acuerdo al recetario.
- 7.1.40.** El usuario de tipo nutricionista debe poder rellenar los campos de la parte de evaluación dietética en la que se tienen los apartados de recordatorio 24 horas, los complementos, los suplementos, la frecuencia de consumo y la fórmula sintética de evaluación.

- 7.1.41.** El usuario Nutricionista debe tener la opción de visualizar todos sus pacientes registrados de acuerdo al tipo de paciente o puede buscar por nombre o identificación.
- 7.1.42.** El nutricionista debe tener la opción de agregar la minuta patrón en la que puede agregar los alimentos que el paciente puede consumir además del tipo de dieta y las recomendaciones nutricionales.
- 7.1.43.** El nutricionista podrá agregar y eliminar opciones de alimentos para su paciente, obteniendo la información de los alimentos de la base de datos externa.
- 7.1.44.** El nutricionista podrá contar con el apartado de programación dietética en el que de acuerdo a los datos ingresados se realicen los cálculos en la fórmula desarrollada de programación, este proceso se realiza en tiempo de manera que la fórmula desarrollada de programación no debe guardarse en la base de datos.
- 7.1.45.** El usuario de tipo nutricionista debe poder realizar el registro de sus pacientes con los datos básicos de forma que el paciente al ser registrado pueda ingresar y ver los componentes que estén habilitados de acuerdo a su rol.
- 7.1.46.** Para el nutricionista en el apartado de seguimiento se debe mostrar un resumen de las pestañas que ha rellenado con la información de su paciente y las recomendaciones alimentarias que le ha dado.
- 7.1.47.** El usuario de tipo paciente debe poder imprimir su plan nutricional.
- 7.1.48.** El usuario de tipo paciente debe poder ver el apartado de programación dietética que el nutricionista ha creado para él o ella con la información de la fórmula sintética de programación y la fórmula desarrollada de programación.
- 7.1.49.** El usuario de tipo paciente debe poder tener acceso a su seguimiento de manera que pueda visualizar pero no editar lo que el nutricionista ha registrado.
- 7.1.50.** El usuario de tipo paciente debe poder visualizar el calendario con las fechas de sus próximas consultas programadas con el nutricionista.
- 7.1.51.** El usuario de tipo paciente debe tener acceso al historial de consultas que ha tenido con el nutricionista hasta la fecha actual.

- 7.1.52. El usuario de tipo paciente debe tener acceso a las fotografías del avance que ha tenido desde que inició el proceso para así poder visualizar los cambios.
- 7.1.53. El usuario de tipo paciente al ingresar al sistema debe tener la opción de visualizar las recetas que se encuentren en el sistema o las que su nutricionista le haya recomendado de acuerdo a sus condiciones.
- 7.1.54. El usuario de tipo paciente debe poder visualizar las opciones de alimentos que el nutricionista le ha recomendado.
- 7.1.55. El usuario de tipo paciente debe tener la opción de restablecer la contraseña, esto con una contraseña genérica enviada al correo que se encuentre registrado en su usuario.

## 7.2. Requisitos no funcionales

- 7.2.1. **Usabilidad:** El software debe ser entendible para el nutricionista y fácil de usar para el paciente.
- 7.2.2. **Rendimiento:** La página debe cargar de manera rápida todos sus componentes.
- 7.2.3. **Mantenibilidad:** La aplicación debe permitir el soporte por parte de los desarrolladores a lo largo del tiempo.
- 7.2.4. **Disponibilidad:** La aplicación debe estar corriendo 24/7 para que los usuarios puedan acceder a ella en cualquier momento.
- 7.2.5. **Hardware:** Se debe tener una impresora instalada en el equipo si se requiere imprimir algún informe de la aplicación.
- 7.2.6. **Apariencia:** Se debe tener en cuenta en las vistas que deben contar con la estructura especificada en los mockups.

## 7.3. Historias de Usuario

<b>HU01</b>	
<b>Nombre Historia:</b> Registro paciente	<b>Usuario:</b> Usuario

**Descripción:** Yo como usuario con permisos dados por el rol en la aplicación quiero poder registrar pacientes para tener una base de datos organizada en el sistema

**Criterios de aceptación:**

- El registro debe tener
  - Nombres
  - Apellidos
  - Género
  - Identificación
  - Fecha de nacimiento
  - Correo electrónico
  - Teléfono
  - Escolaridad
  - Ocupación
  - Estado civil
  - Alergias
  - Medicamentos que esté consumiendo
  - Enfermedades de las que padezca
  - Preferencias alimentarias
  - Disgustos alimentarios
  - Deporte que práctica
  - Frecuencia práctica
  - Información sobre familiares enfermos
  - Información sobre si ha sido hospitalizado
  - Fecha de la hospitalización
  - Tipo de paciente
  - Identificación del paciente
  - Identificación del nutricionista que realiza el registro.
  
- La página de registro debe contar además del formulario con un botón de guardar y otro de cancelar.
- Si se da clic en el botón guardar, se debe mostrar un popup con el resultado de la operación, ya sea exitosa o fallida por algún cambio en específico
- Al dar clic en el botón cancelar se debe dirigir a la ventana del menú de usuario

**HU02**

**Nombre Historia:** Registro Nutricionista

**Usuario:** Administrador

**Descripción:** Yo como Administrador quiero poder registrar nutricionistas para tener una base de datos organizada en el sistema

**Criterios de aceptación:**

- EL registro debe tener obligatoriamente:
  - Nombre
  - Apellido
  - Identificación
  - Fecha nacimiento
  - Correo electrónico
  - Teléfono
  - Contraseña
  - Tarjeta Profesional
  - Posgrados
  
- La aplicación debe tener la posibilidad de no realizar la operación solicitada, si hace falta datos necesarios
- El formulario debe estar en la capacidad de adaptarse al tamaño de la pantalla del dispositivo de donde se esté realizando la transacción

**HU03****Nombre Historia:** Gestión de pacientes**Usuario:** Administrador**Descripción:** Yo como Administrador quiero realizar una gestión de pacientes, teniendo la opción de ver todos los pacientes registrados el registro de pacientes y la eliminación de pacientes para poder tener control sobre la base de datos de la aplicación**Criterios de aceptación:**

- La página debe poseer las siguientes opciones:
  - Listado de pacientes
  - Registro de pacientes
  - Eliminación de pacientes
- La página debe contar con un campo de tipo de usuario para que al dar clic en particular o deportista, este muestre solo los pacientes del tipo seleccionado
- La página debe contar con un campo de número de identificación para que se pueda agregar un número de identificación y así filtre los usuarios que contengan los números ingresados en su identificación.

**HU04****Nombre Historia:** Menú principal paciente**Usuario:** Paciente

**Descripción:** Yo como Paciente quiere tener un menú de opciones para tener una mejor experiencia y control sobre lo que deseo realizar

**Criterios de aceptación:**

- La página debe tener las opciones de:
  - Seguimientos
  - Recetas
  - Banco de alimentos
  - Consultas
  - Imprimir
  - Calendario
  - Fotografías

**HU05**

**Nombre Historia:** Implementar registros de antropometrías

**Usuario:** Nutricionista

**Descripción:** Yo como Nutricionista quiero ver el historial de antropometrías que he registrado de mis pacientes para llevar el control histórico y pueda definir las dietas a recomendarles

**Criterios de aceptación:** Al dar clic en antropometría en el menú lateral de debe mostrar la lista de las antropometrías del paciente seleccionado con los datos:

- Fecha
- Nombre del paciente
- Identificación

**HU06**

**Nombre Historia:** Programación dietética

**Usuario:** Paciente

**Descripción:** Yo como Paciente quiero ver la programación dietética que me ha desarrollado el nutricionista para ver la cantidad en porcentaje de calorías y grasa, además de las raciones, que puedo consumir durante el día

**Criterios de aceptación:**

- Se debe mostrar un excel para realizar los cálculos dietéticos del paciente cuando el nutricionista entre a esta opción

<b>HU07</b>	
<b>Nombre Historia:</b> Banco de recetas	<b>Usuario:</b> Nutricionista
<b>Descripción:</b> Yo como Nutricionista quiero visualizar en el apartado de banco de recetas para saber las recetas que puedo recomendar a mis pacientes y agregar nuevas recetas	
<b>Criterios de aceptación:</b> La receta debe contener los siguientes datos: <ul style="list-style-type: none"> <li>● Foto</li> <li>● Nombre de la receta</li> <li>● Calorías</li> <li>● Proteínas</li> <li>● Grasas</li> <li>● Carbohidratos</li> <li>- Se deben mostrar en la sección de recetas, cada una de estas con su correspondiente imagen</li> <li>- Se debe poder realizar una edición de la receta, por si llega a tener cambios en el transcurso del tiempo</li> </ul>	

<b>HU08</b>	
<b>Nombre Historia:</b> Ejemplo menú	<b>Usuario:</b> Nutricionista
<b>Descripción:</b> Yo como Nutricionista quiero tener la posibilidad de ver el menú que hasta el momento le estoy recetando a mi paciente para llevar un control de los alimentos que he escogido y así poder agregar o quitar alimentos de acuerdo a mi paciente	
<b>Criterios de aceptación:</b> Los datos de la tabla son los mismos para los pacientes de tipo: <ul style="list-style-type: none"> <li>● Deportista</li> <li>● Particular</li> </ul>	

<b>HU09</b>	
<b>Nombre Historia:</b> Antropometría particular	<b>Usuario:</b> Nutricionista
<b>Descripción:</b> Yo como Nutricionista quiero visualizar en el apartado de antropometría de los pacientes particulares para así poder tener la información del paciente y como la alimentación recomendada puede ayudar a sus necesidades	

**Criterios de aceptación:**

- Se requiere mostrar los datos antropométricos del paciente teniendo información como:
  - **Básicos**
    - Peso kg
    - Talla m
  - **Pliegues Cutáneos**
    - Subescapular mm
    - Cresta iliaca mm
    - Muslo Anterior mm
    - Pierna Medial mm
    - Tricipital mm
    - Supraespinal mm
    - Pierna medial mm
    - Bicipital mm Abdominal mm
  - **Diámetros**
    - Supraescapular mm
    - Subescapular mm
    - Perimetro de brazo corregido cm
    - Perimetro de pierna Corregido cm
  
- La página de registro debe contar además del formulario con un botón de guardar.
- Si se da clic en el botón guardar, se debe mostrar un popup con el resultado de la operación, ya sea exitosa o fallida por algún cambio en específico
- Si no se da clic en guardar la información ingresada se perderá cuando la página se recargue

**HU10**

**Nombre Historia:** Consultas

**Usuario:** Nutricionista

**Descripción:** Yo como Nutricionista quiero visualizar en el apartado de consultas de los pacientes para saber qué consultas han tenido conmigo anteriormente y que consultas hay próximas a agendar

**Criterios de aceptación:** Al dar clic en las consultas se debe mostrar como calendario o como una tabla en la que se muestre:

- Número de consulta
- Nombre
- Apellidos
- ID

- Tipo de Paciente
- Fecha
- Estado
- Se debe mostrar esta información en el calendario

### HU11

**Nombre Historia:** Evaluación dietética

**Usuario:** Nutricionista

**Descripción:** Yo como Nutricionista quiero tener la posibilidad de generar y ver la minuta patrón de mis pacientes para poder indicarle las opciones de comidas que tendrá durante la semana para cumplir su objetivo de contextura corporal deseado y recomendado

**Criterios de aceptación:**

- Se debe implementar en tablas teniendo en cuenta
  - Grupo de alimentos
  - Porciones
  - Desayuno
  - Media Mañana
  - Almuerzo
  - Cena
  - Alimentos post entrenamiento
  - Suplementos
  - Complemento
  - Alimento con propósitos médico
- Cada nuevo elemento se debe agregar de forma dinámica, dado que si el usuario da clic en el botón agregar , se muestre una nueva fila para agregar los otros campos de ese nuevo elemento, esto de manera dinámica
- Si el usuario da clic en el botón eliminar, se borrará la celda
- solo se guardará la información en el momento que el usuario de clic en el botón guardar, sino esta información se perderá en el momento en el que la página se recargue.

### HU12

**Nombre Historia:** Fotografías

**Usuario:** Paciente

**Descripción:** Yo como Paciente quiero ver el historial de fotografías que me han tomado en el proceso para ver si se han presentado cambios durante el proceso

**Criterios de aceptación:**

- Las imágenes que se suben deben tener un tamaño máximo de 20MB y un formato JPG o JPEG.
- Cada imagen debe estar registrada con la fecha de subida.

**HU13**

**Nombre Historia:** Ver administradores

**Usuario:** Administrador

**Descripción:** Yo como Administrador quiero tener la posibilidad de ver todos los administradores registrados en la aplicación para poder tener control de los administradores que se encuentran registrados y que especialidades tienen que ayuden al desarrollo y al soporte de la aplicación para los usuarios.

**Criterios de aceptación:**

- El sistema debe mostrar la lista de administradores
- La página debe contar con un campo de número de identificación para que se pueda agregar un número de identificación y así filtre los usuarios que contengan los números ingresados en su identificación.

**HU14**

**Nombre Historia:** Gestionar administrador

**Usuario:** Administrador

**Descripción:** Yo como Administrador quiero tener la posibilidad de gestionar administradores para poder ver los administradores registrados eliminar los administradores que dejen de trabajar para la aplicación y poder registrar a los nuevos administradores que se puedan ocupar de dar mantenimiento y soporte a la aplicación en situaciones futuras

**Criterios de aceptación:**

- En la ventana debe haber 3 opciones:
  - Ver todos los administradores
  - Registrar Administrador
  - Eliminar administrador

- Al dar clic en una de las opciones se debe redirigir a cada uno de los componentes de acuerdo al botón seleccionado

### HU15

**Nombre Historia:** Eliminar administrador

**Usuario:** Administrador

**Descripción:** Yo como Administrador quiero tener la posibilidad de eliminar los usuarios administradores que ya no se estén utilizando para así mantener un control de los usuarios registrados en el sistema y poder eliminar los usuarios que ya no se utilizarán más

**Criterios de aceptación:**

- Un administrador no puede eliminarse a sí mismo, debe haber otro usuario administrador que lo haga, esto para controlar la mantenibilidad del control en la aplicación

### HU16

**Nombre Historia:** Registrar administrador

**Usuario:** Administrador

**Descripción:** Yo como Administrador quiero tener la posibilidad de registrar nuevos usuarios administradores para así tener más usuarios que ayuden a soportar la aplicación

**Criterios de aceptación:**

- Se debe tener en cuenta los datos:
  - Nombre
  - Apellidos
  - Identificación
  - Especialidad
  - Correo Electrónico
  - Celular
  - Fecha de Nacimiento
- Si se da clic en el botón guardar, se debe mostrar un popup con el resultado de la operación, ya sea exitosa o fallida por algún cambio en específico
- La página de registro debe contar además del formulario con un botón de guardar y otro de cancelar.
- Al dar clic en el botón cancelar se debe dirigir a la ventana del menú de usuario

<b>HU17</b>	
<b>Nombre Historia:</b> Gestionar nutricionista	<b>Usuario:</b> Administrador
<b>Descripción:</b> Yo como Administrador quiero tener la posibilidad de gestionar nutricionistas para poder ver los nutricionistas registrados en la aplicación que se encuentran pagando	
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"> <li>- En la ventana debe haber 3 opciones: <ul style="list-style-type: none"> <li>● Ver todos los nutricionistas</li> <li>● Registrar nutricionista</li> <li>● Eliminar nutricionista</li> </ul> </li> <li>- Al dar clic en una de las opciones se debe redirigir a cada uno de los componentes de acuerdo al botón seleccionado</li> </ul>	

<b>HU18</b>	
<b>Nombre Historia:</b> Ver nutricionista	<b>Usuario:</b> Administrador
<b>Descripción:</b> Yo como Administrador quiero tener la posibilidad de ver todos los nutricionistas registrados en la aplicación para poder tener control de los nutricionistas que se encuentran registrados y las especialidades que tienen, para dar reporte del crecimiento de la aplicación	
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"> <li>- El sistema debe mostrar la lista de nutricionistas, teniendo en cuenta la siguiente información <ul style="list-style-type: none"> <li>● Nombre</li> <li>● Apellido</li> <li>● Identificación</li> <li>● Especialidad</li> </ul> </li> <li>- La página debe contar con un campo de número de identificación para que se pueda agregar un número de identificación y así filtre los usuarios que contengan los números ingresados en su identificación.</li> </ul>	

<b>HU19</b>	
<b>Nombre Historia:</b> Deshabilitar nutricionista	<b>Usuario:</b> Administrador

**Descripción:** Yo como Administrador quiero tener la posibilidad de deshabilitar los usuarios nutricionistas que ya no estén pagando membresía para así mantener control de los usuarios registrados en el sistema

**Criterios de aceptación:**

- No se puede deshabilitar un usuario que haya pagado membresía mientras la fecha esté cubierta por esta.

**HU21**

**Nombre Historia:** Generación de contraseñas | **Usuario:** Usuario

**Descripción:** Yo como usuario de la aplicación quiero que al realizar el registro se genera una contraseña aleatoria para tener seguridad al ingresar a mi correo

**Criterios de aceptación:**

- La contraseña debe tener una longitud de 8 caracteres y debe ser generada de manera aleatoria.

**HU22**

**Nombre Historia:** Envío de contraseña | **Usuario:** Usuario

**Descripción:** Yo como usuario de la aplicación quiero que la aplicación me envíe un correo con mi contraseña para poder hacer ingreso a la aplicación

**Criterios de aceptación:**

- El sistema debe enviar un mensaje con la contraseña aleatoria creada al correo registrado por el usuario en el momento del registro o el que se encuentre en el campo correo, en los datos personales del usuario.

**HU23**

**Nombre Historia:** Login de usuario | **Usuario:** Usuario

**Descripción:** Yo como usuario de la aplicación quiero tener la posibilidad de ingresar al sistema para poder cumplir con las funciones de acuerdo a mi usuario

**Criterios de aceptación:**

- Solo se puede ingresar si el usuario se encuentra registrado en el sistema, en caso de que no se debe enviar una alerta de que el usuario o la contraseña es inválida
- Si el usuario pone el usuario(identificación) o la contraseña errónea, debe aparecer un mensaje de que se ha ingresado un usuario o contraseña incorrecta.
- Si el usuario olvida la contraseña puede seleccionar la opción de olvidé contraseña y se le hará envío de la contraseña al correo registrado de acuerdo al usuario.

**HU24**

**Nombre Historia:** Ventana inicio nutricionista

**Usuario:** Nutricionista

**Descripción:** Yo como Nutricionista quiero que al momento de iniciar sesión me lleve a la ventana de opciones disponibles para mí para poder seleccionar la opción requerida de acuerdo al paciente que esté atendiendo

**Criterios de aceptación:**

En la ventana se debe mostrar la foto del nutricionista, su nombre completo y las especialidades que tiene. Además, debe mostrar 3 opciones principales las cuales son:

- Pacientes
- Banco de recetas
- Banco de alimentos

Al dar clic en la opción de pacientes se mostrarán los tipos de pacientes, en el momento de seleccionar un tipo se puede indicar la opción de listar los pacientes o registrar pacientes.

Al dar clic en Banco de recetas, se mostrará la opción para escoger el tipo de recetas a ver, las cuales son:

- Desayunos
- Snacks
- Cenas
- Almuerzos

Al dar clic en banco de alimentos debe mostrar la lista de intercambio de alimentos, la cual se debe consultar desde la base de datos de los alimentos registrados

<b>HU25</b>	
<b>Nombre Historia:</b> Lista particulares/deportista	<b>Usuario:</b> Nutricionista
<b>Descripción:</b> Yo como Nutricionista quiero tener la opción de ver la lista de los paciente de tipo particular/deportista para llevar el control de los pacientes que se encuentran a mi cuidado que son particulares y poder desarrollar los planes de alimentación adecuados para sus necesidades	
<b>Criterios de aceptación:</b>	
<ul style="list-style-type: none"> <li>- Al ingresar el número de identificación del paciente debe mostrar solo al paciente con el mismo número de identificación o a pacientes que tienen el mismo número inicial pero luego cambia; si no se agrega una identificación debe mostrar todos los pacientes de tipo particular.</li> </ul>	

<b>HU26</b>	
<b>Nombre Historia:</b> Deshabilitar pacientes	<b>Usuario:</b> Nutricionista
<b>Descripción:</b> Yo como Nutricionista quiero tener la opción de deshabilitar los pacientes que ya no están realizando consultar conmigo para tener solo registro de los pacientes que sí están en consulta	
<b>Criterios de aceptación:</b> Se deben implementar los componentes para cada tipo de usuario, aunque se tiene en cuenta que todos consumen el mismo servicio pero la parte visual es diferente.	

<b>HU27</b>	
<b>Nombre Historia:</b> Actualizar pacientes	<b>Usuario:</b> Nutricionista
<b>Descripción:</b> Yo como Nutricionista quiero tener la opción de actualizar información de los pacientes para corregir errores en la información del usuario que se pueda presentar	
<b>Criterios de aceptación:</b>	
<ul style="list-style-type: none"> <li>- Se deben implementar los componentes para cada tipo de usuario, aunque se tiene en cuenta que todos consumen el mismo servicio pero la parte visual es diferente.</li> </ul>	

<b>HU28</b>	
<b>Nombre Historia:</b> Banco de alimentos	<b>Usuario:</b> Nutricionista
<b>Descripción:</b> Yo como Nutricionista quiero tener acceso a la información del banco de alimentos para tener los alimentos con la información nutricional de acuerdo a los estándares de alimentación colombianos	
<b>Criterios de aceptación:</b>	
<ul style="list-style-type: none"> <li>- Se debe poder realizar una alimentación específica por paciente y poder obtenerla y realizar cambios si se requiere</li> </ul>	

<b>HU29</b>	
<b>Nombre Historia:</b> Seguimiento paciente	<b>Usuario:</b> Paciente
<b>Descripción:</b> Yo como Paciente quiero visualizar en el apartado de seguimiento todos mis datos antropométricos, familiares, hospitalarios y dietéticos para así poder visualizar las recomendaciones que el nutricionista brinda para el apartado de los datos dietéticos	
<b>Criterios de aceptación:</b>	
<ul style="list-style-type: none"> <li>- En la página se deben visualizar: <ul style="list-style-type: none"> <li>● Datos personales</li> <li>● Datos familiares</li> <li>● Datos hospitalarios</li> <li>● Datos Antropométricos con su correspondiente gráfica</li> <li>● Dietas con los apartados de: <ul style="list-style-type: none"> <li>● Suplementos</li> <li>● Complementos</li> <li>● Alimentos con propósitos médico</li> <li>● Método de consumo</li> <li>● Tipo de dieta</li> <li>● Fórmula de programación</li> </ul> </li> </ul> </li> </ul>	

<b>HU30</b>	
<b>Nombre Historia:</b> Imprimir plan nutricional	<b>Usuario:</b> Paciente
<b>Descripción:</b> Yo como Paciente quiero tener la posibilidad de imprimir el plan nutricional para tenerlo físico y realizar las indicaciones diarias con mayor comodidad	
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"> <li>- El sistema debe permitir la impresión del plan nutricional</li> <li>- Al dar clic en el botón imprimir se debe abrir una nueva ventana con la pre visualización de como se va a ver la impresión.</li> </ul>	

<b>HU31</b>	
<b>Nombre Historia:</b> Opciones de Alimentos	<b>Usuario:</b> Paciente
<b>Descripción:</b> Yo como Paciente quiero visualizar una tabla con todos los alimentos recomendados por el nutricionista en la valoración realizada para así tener un control más exacto de los alimentos que se deben tener en cuenta para la alimentación recomendada	
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"> <li>- La página debe tener todos los alimentos recomendados por el nutricionista con: <ul style="list-style-type: none"> <li>● Kcal</li> <li>● Alimentos</li> <li>● Medida Casera</li> <li>● Medida Real</li> </ul> </li> <li>- Cada nuevo elemento se debe agregar de forma dinámica, dado que si el usuario da clic en el botón agregar , se muestre una nueva fila para agregar los otros campos de ese nuevo elemento, esto de manera dinámica</li> </ul>	

<b>HU32</b>	
<b>Nombre Historia:</b> Recetas	<b>Usuario:</b> Paciente
<b>Descripción:</b> Yo como Paciente quiero visualizar una página única donde pueda ver distintas recetas recomendadas para cada comida para tener diferentes opciones de alimentación	
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"> <li>- La página debe tener: <ul style="list-style-type: none"> <li>● Apartado de recetas para Desayunos</li> </ul> </li> </ul>	

- Apartado de recetas para Snacks
- Apartado de recetas para Almuerzos
- Apartado de recetas para Cena
- La página debe contar con la opción de seleccionar cada uno de los tipos de comida
- Cada receta se debe visualizar en forma de tarjeta
- Cada tarjeta tiene dos botones, uno de ver y uno de eliminar
- Al dar clic en el botón ver se debe mostrar un modal con la información de la receta
- Al dar clic en el botón eliminar se debe mostrar una alerta que pregunte si está seguro de realizar esa acción.

<b>HU33</b>	
<b>Nombre Historia:</b> Configuración información de la cuenta	<b>Usuario:</b> Usuario
<b>Descripción:</b> Yo como usuario de la aplicación quiero poder gestionar los cambios de contraseña, imagen de reconocimiento, correo y número de contacto para llevar una gestión actualizada de mi cuenta	
<p><b>Criterios de aceptación:</b> Se debe tener:          La posibilidad de realizar cambio de:</p> <ul style="list-style-type: none"> <li>● Correo Electrónico</li> <li>● Numero de Celular</li> <li>● Contraseña cumpliendo los criterios de aceptación de esta</li> <li>● Imagen de reconocimiento</li> </ul> <p>cos campos no editables en este menú, por motivo de seguridad serán:</p> <ul style="list-style-type: none"> <li>● Nombre</li> <li>● Apellidos</li> <li>● Identificación</li> <li>● Especialidad</li> </ul> <p>En caso de que el usuario desee cambiar de contraseña, este debe confirmarla en otro campo del formulario          En caso de que el usuario no cambie la contraseña, no se hace la validación del campo de nueva contraseña</p>	

<b>HU35</b>	
<b>Nombre Historia:</b> Recuperación de contraseña	<b>Usuario:</b> Usuario

**Descripción:** Yo como usuario de la aplicación quiero recuperar la contraseña de mi cuenta para tener control sobre mi cuenta

**Criterios de aceptación:**

- El sistema debe permitir la opción de recuperar contraseña
- Al ingresar el correo, el sistema debe validar este mismo y si se encuentra registrado, enviar una nueva contraseña para el ingreso a la aplicación

**HU34**

**Nombre Historia:** Datos personales

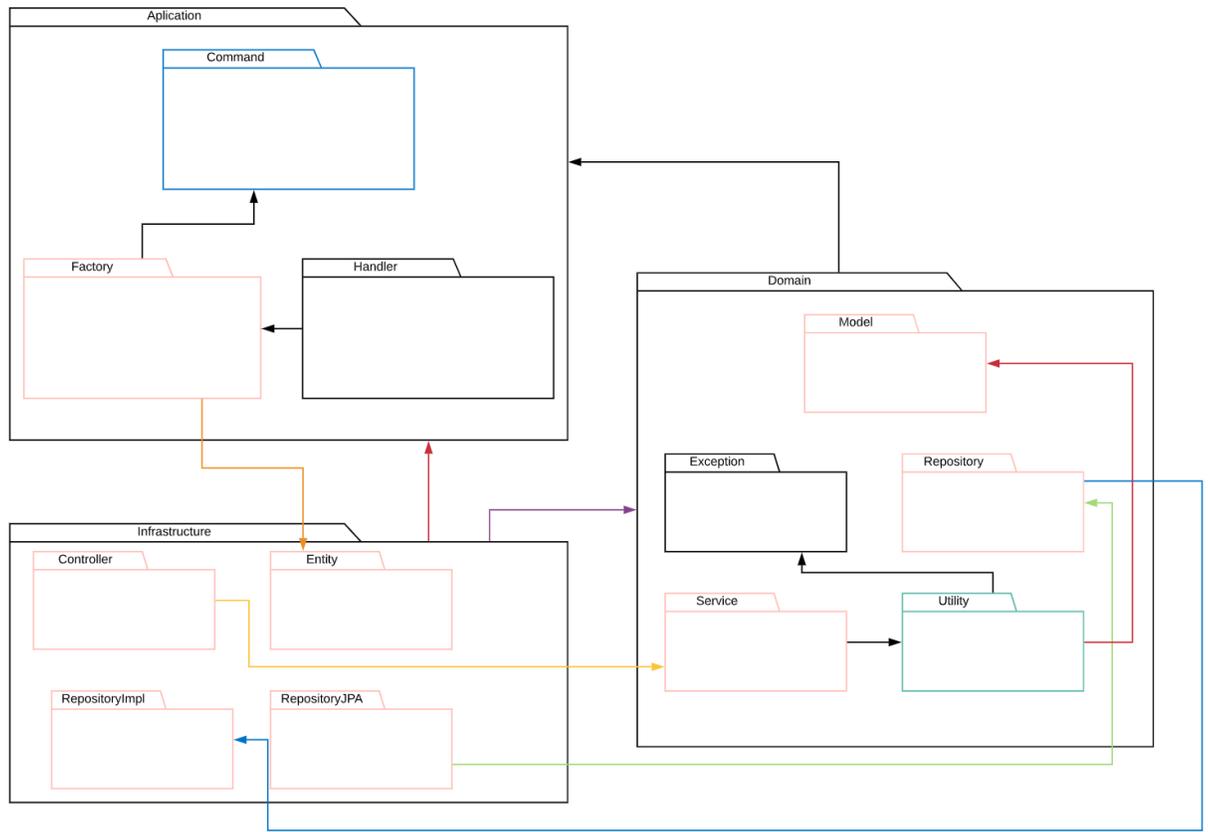
**Usuario:** Nutricionista

**Descripción:** Yo como Nutricionista quiero tener la posibilidad de ver los datos personales de mis pacientes que fueron registrados por mí en la primera consulta para llevar un control de su estado de salud, sus preferencias alimenticias, sus expedientes y demás datos que me permitan dar una correcta asignación de comidas para cumplir con sus necesidades

**Criterios de aceptación:**

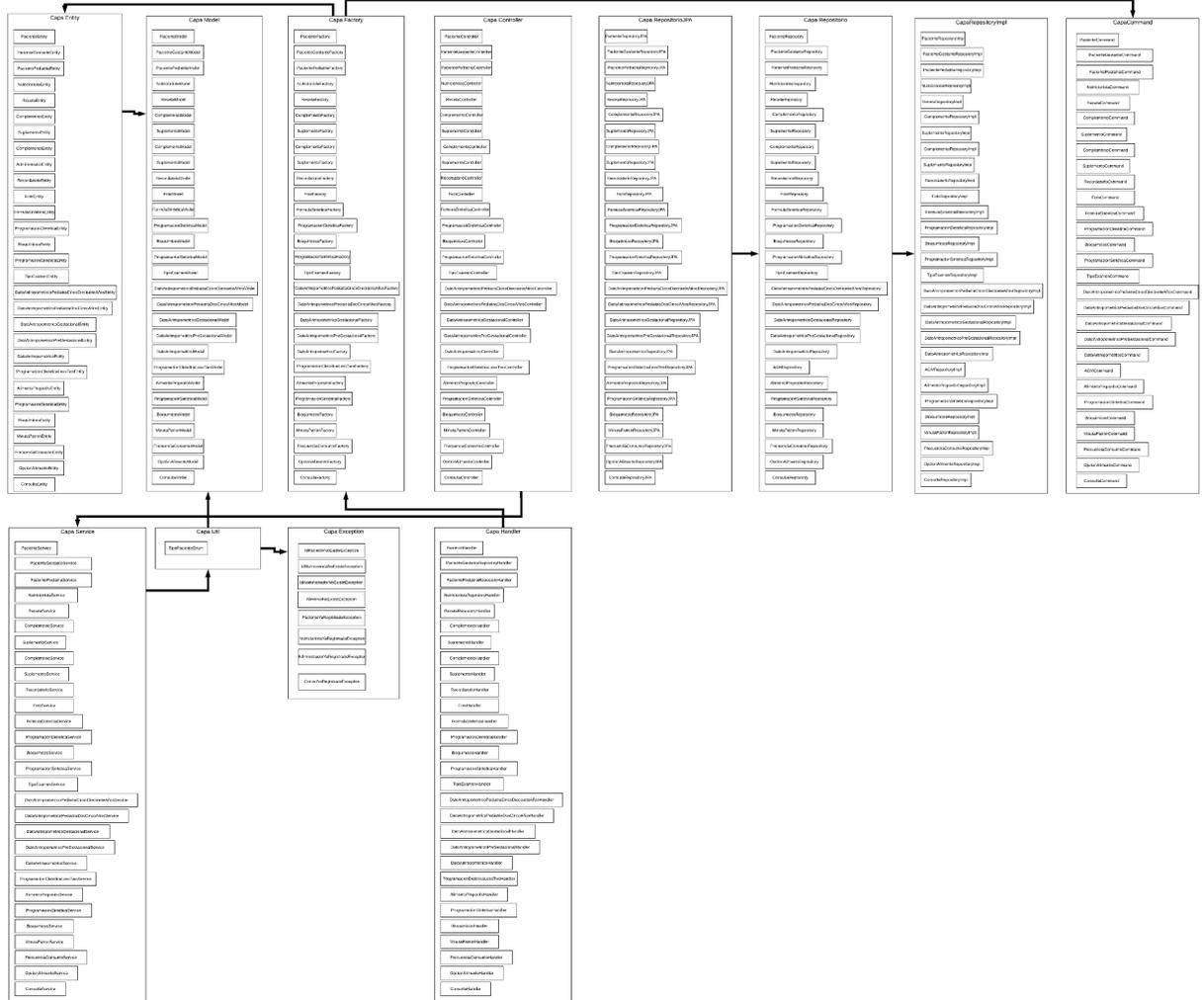
- Se debe tener en cuenta
  - Datos personales
  - Datos familiares
  - Datos hospitalarios
  - Bioquímicos
- El nutricionista podrá editar los datos del paciente que no estén relacionados con su usuario.

## 7.4. Diseño



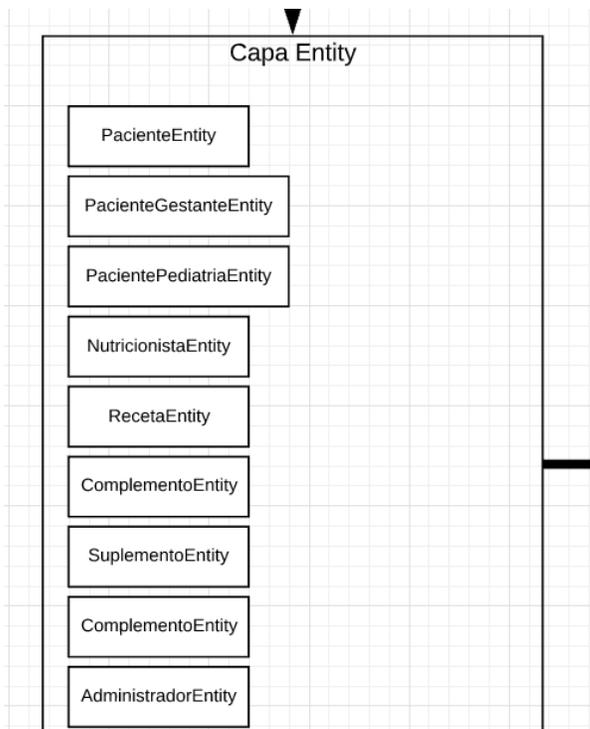
**Figura 1: Diagrama de paquetes**

Este diagrama está compuesto por 3 grandes secciones las cuales son estructuradas como los paquetes principales de la aplicación a nivel de backend, contiene los diferentes paquetes que controlan los modelos, los servicios y los controladores que se ocupan de funcionar como API REST y de acuerdo a las peticiones dar la respuesta o realizar el procedimiento esperado.

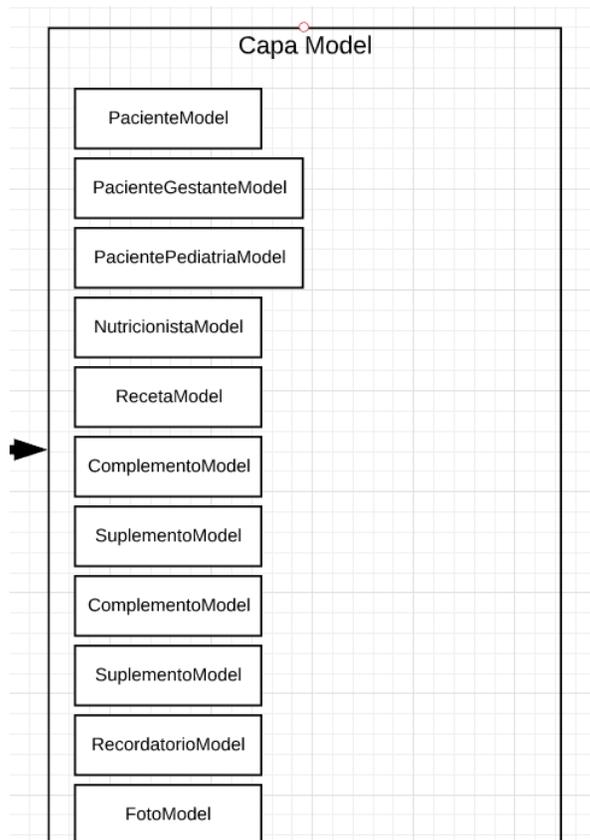


**Figura 2: Diagrama de componentes BE**

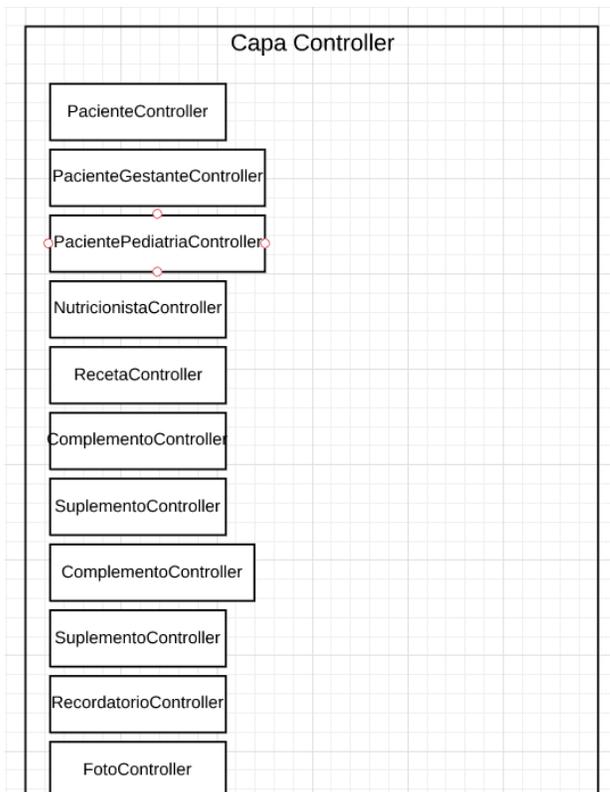
Este diagrama tiene la estructura de los componentes que componen el proyecto de backend el cual fue desarrollado en el IDE IntelliJ, cada componente se encarga de realizar la acción específica, también contiene los modelos y los controladores implementados.



En la capa Entity nos observaremos las declaraciones y las estructuras de las tablas que nos encontraremos en la base de datos, con sus correspondientes atributos y tipos para cada campo, así mismo, los campos obligatorios y no que no son obligatorios, contiene las etiquetas **@Entity** y **@Table**

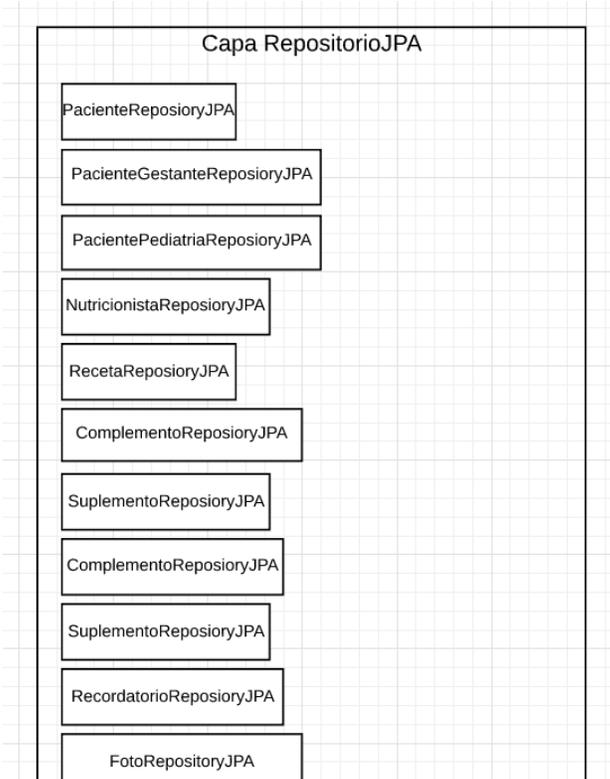


En la capa Model se encuentran los modelos que representan el mapeo de los objetos provenientes de la base de datos y con los que se opera para que no se tenga el acceso directo a las entidades, estos modelos se utilizan para realizar las operaciones CRUD en la base de datos al ser utilizados en los otros componentes del backend.

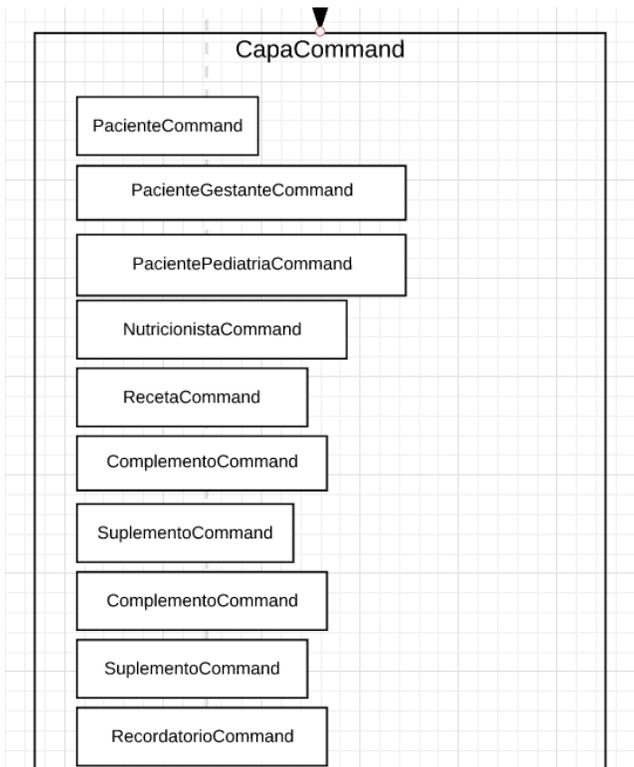


La capa Controller se encarga de almacenar todos los controladores de la aplicación, los cuales sirven para establecer las rutas por las que se puede realizar peticiones HTTP desde otra aplicación. En estos se encuentra también el control de accesos a estas peticiones y el control de los roles que pueden acceder a ciertas peticiones a las que tengan permiso de acuerdo a la lógica de negocio.

La capa controller se identifica con la etiqueta **@RestController**

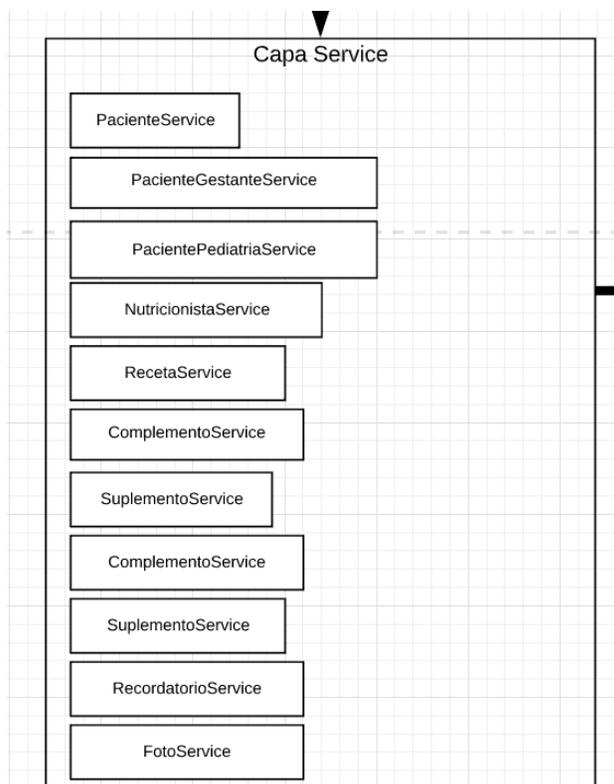


En la capa JPA se encuentran las clases JPA que contienen la etiqueta **@Repository**, estas se encargan de conectar la base de datos y al API REST para así poder realizar las operaciones de lectura y escritura en la base de datos.

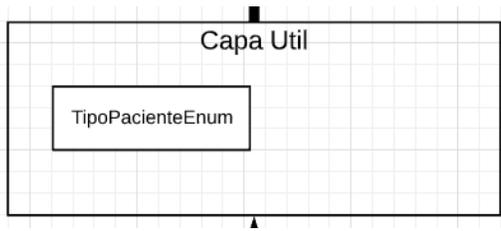


En la capa command se encuentran los objetos que son mapeados por los controller al momento de acceder a una ruta por medio de peticiones HTTP, esto con el fin de realizar operaciones y de poder realizar operaciones CRUD (Create, Read, Update and Delete).

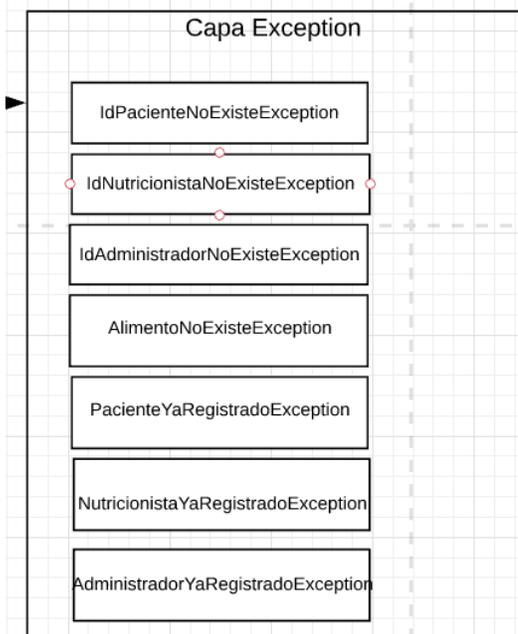
Al ser un componente de la aplicación, se debe utilizar la etiqueta **@Component**, para evitar errores a la hora de la compilación, si se pueda inyectar esta clase en otras



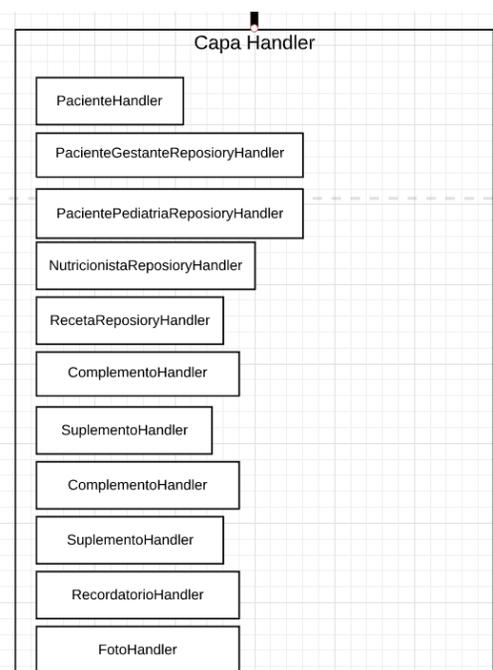
En la capa service se encuentran todas las operaciones que se requieran realizar llamando directamente los servicios implementados en el desarrollo de la aplicación (JPA), se identifica y es necesario que siempre contengan las etiqueta **@Service**.



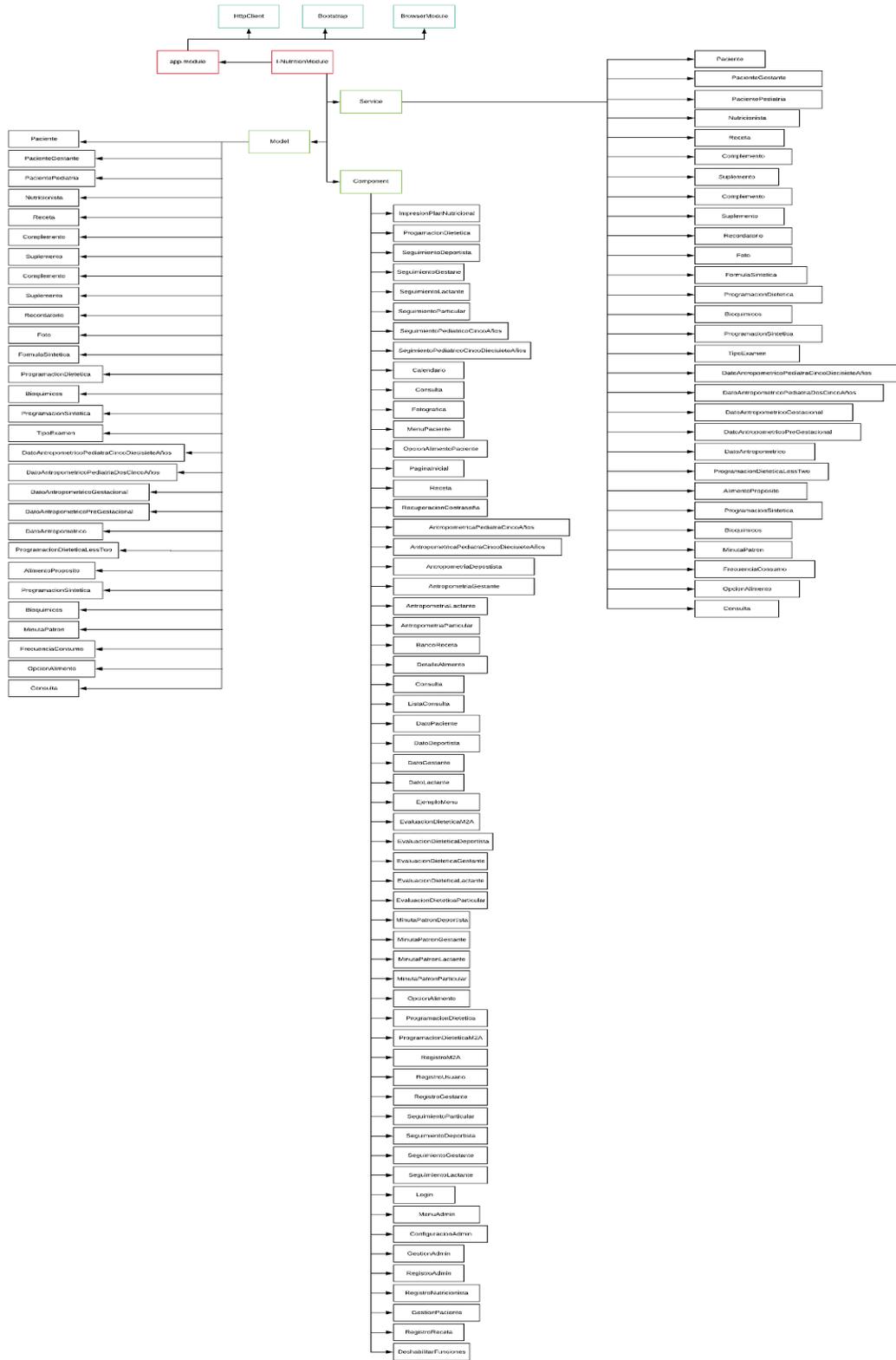
En la capa Útil se encuentran todas las utilidades que tengan a aplicación pueden ser enumeradores, métodos u operaciones que requieran utilizar en varios lugares, cada una de estas clases se identifican con la etiqueta **@Component** para poder realizar inyección de estos servicios en otras clases



En la capa Exception se encuentran todos los Handler que se encargan de arrojar las excepciones customizadas de la aplicación.

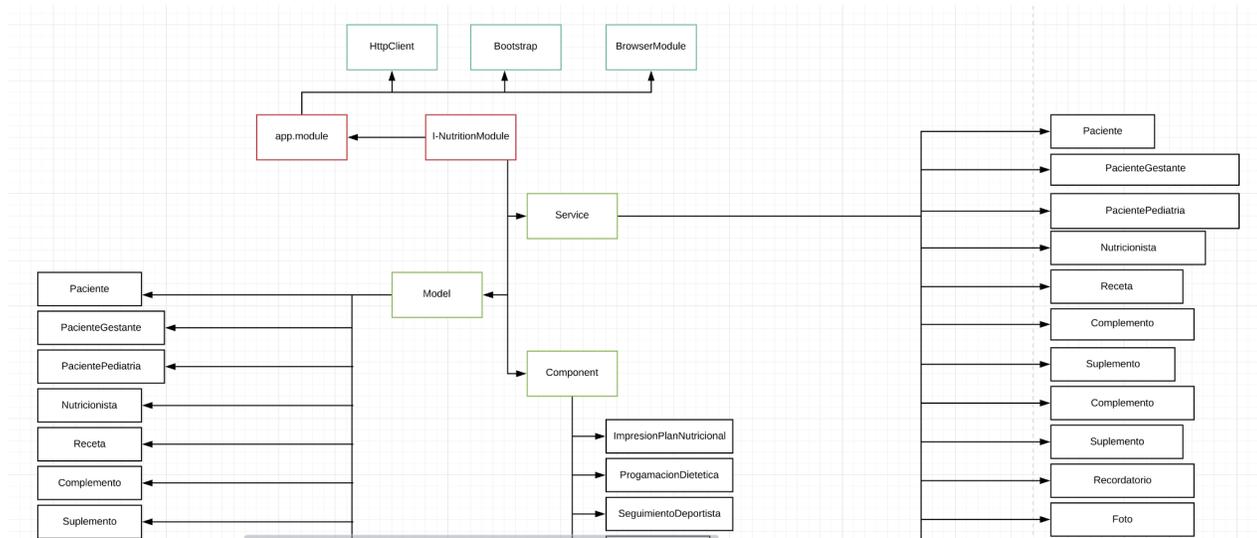


En la capa se encarga de manejar la capa factory para utilizar los datos obtenidos de este y enviarlos a la capa service para realizar todo el flujo necesario, esta capa debe poseer la etiqueta **@Component**, para realizar la inyección de esta clase



**Figura 3: Diagrama de componentes FE**

Este diagrama tiene la estructura de los componentes que componen el proyecto de frontend el cual fue desarrollado en el IDE WebStorm, cada componente se encarga de maquetar y funcionar como vista de la aplicación, que se encarga de ser la conexión con el API REST del backend y que realiza toda la interacción con el usuario final.



**app.module:** Es el módulo principal de la aplicación, el cual contiene las librerías a nivel global y todo lo relacionado con rutas de componentes en toda la aplicación.

**HttpClient:** Es la librería encargada de las peticiones HTTP al API REST en el backend.

**Bootstrap:** Es la librería encargada de los estilos en la aplicación.

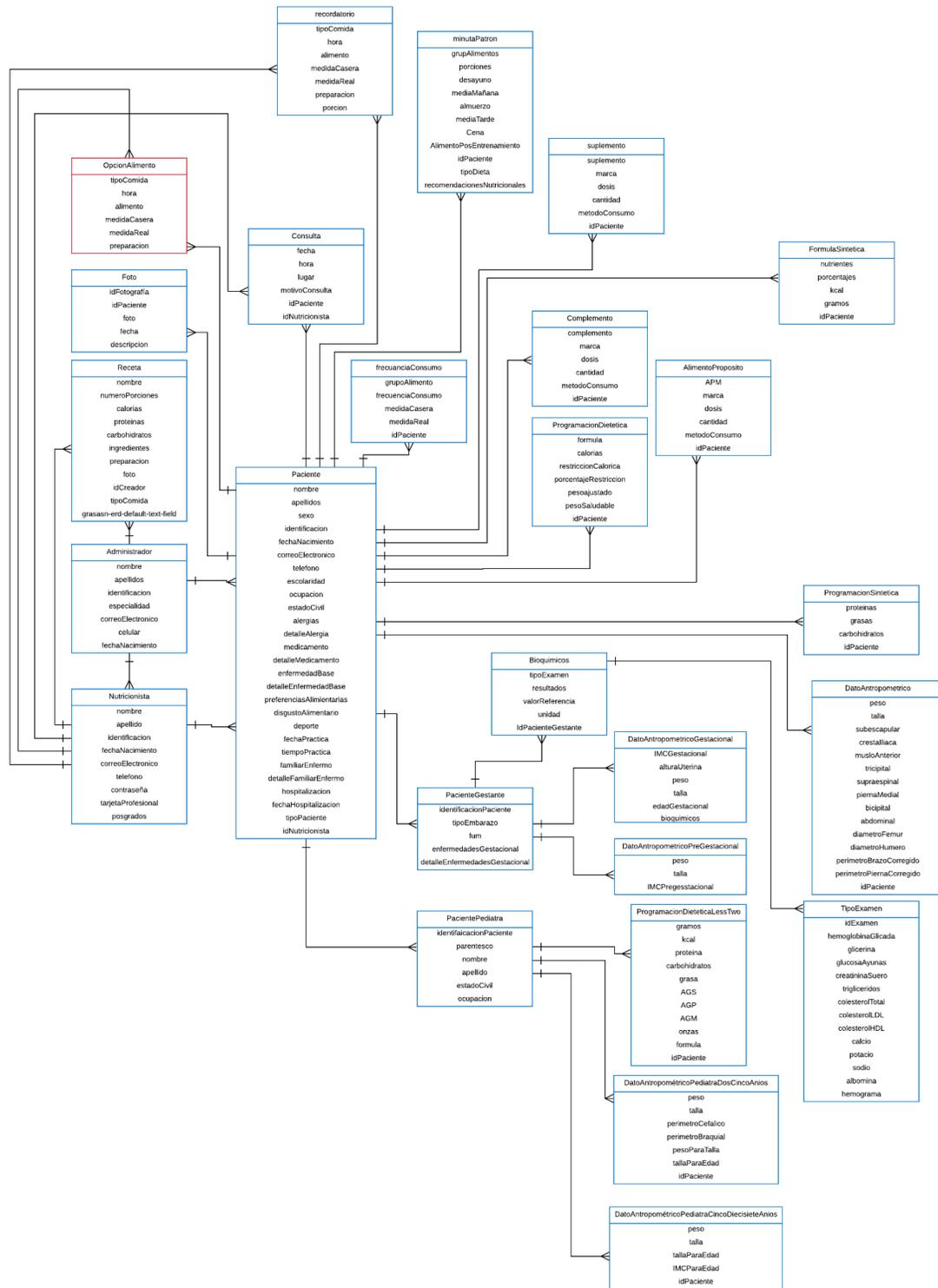
**BrowserModule:** Es el módulo encargado de importar todas las librerías requeridas por el navegador.

**I-NutritionModule:** Es el módulo encargado de los componentes, servicios y guardias de la aplicación.

**Service:** Es el directorio donde se encuentran todos los servicios para comunicarse con el API REST del backend mediante los métodos HTTP, en estos servicios se encuentra todo el proceso de peticiones de manera asíncrona para así proporcionarle la información necesaria a los componentes para trabajar de manera adecuada.

**Model:** En este directorio se encuentran todos los modelos que cumplen el papel de mapear los objetos que se encuentran almacenados en la base de datos, estos se encuentran como unas interfaces para así poder instanciarse en los componentes.

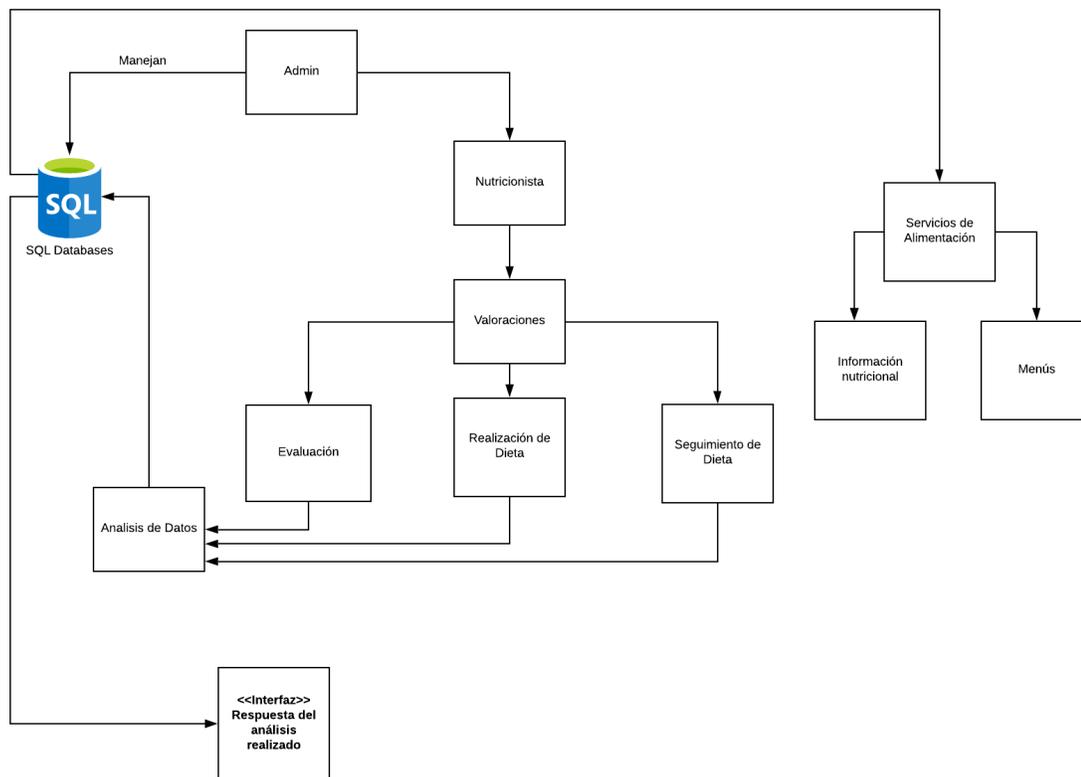
**Component:** En este directorio se encuentran los componentes de la aplicación, como header, footer, carruseles, tablas de contenido, cuadros de búsqueda, contenedores de imágenes, listas de tarjetas y otros componentes más avanzados como calendarios y vistas de excel.



**Figura 4: Diagrama de entidades**

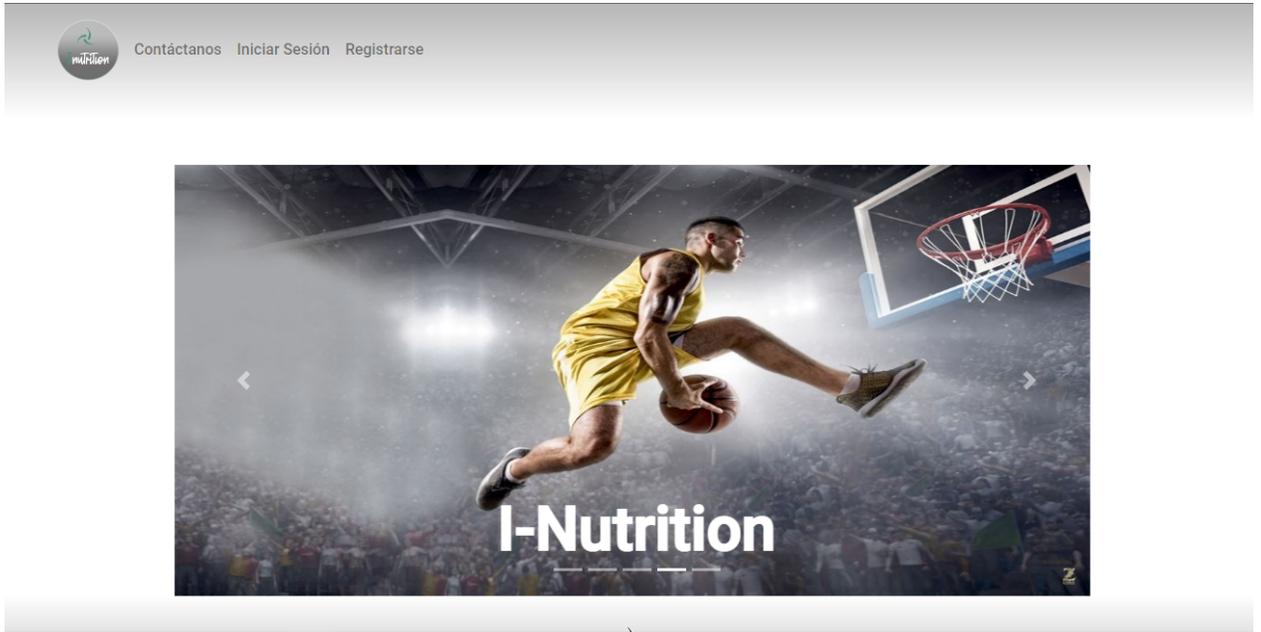
En este diagrama se muestran las entidades involucradas en el desarrollo de la aplicación, los usuarios y otras entidades relacionadas con ellos, que permiten el correcto

funcionamiento de la aplicación según los requisitos de los stakeholders. En este diagrama se encuentran Los pacientes, las consultas, las opciones de alimento, las fotos de avance de los pacientes, las recetas, las frecuencias de consumo, los usuarios administradores, los usuario nutricionistas, los complementos, suplementos, programación dietética de los pacientes y los datos antropométricos.

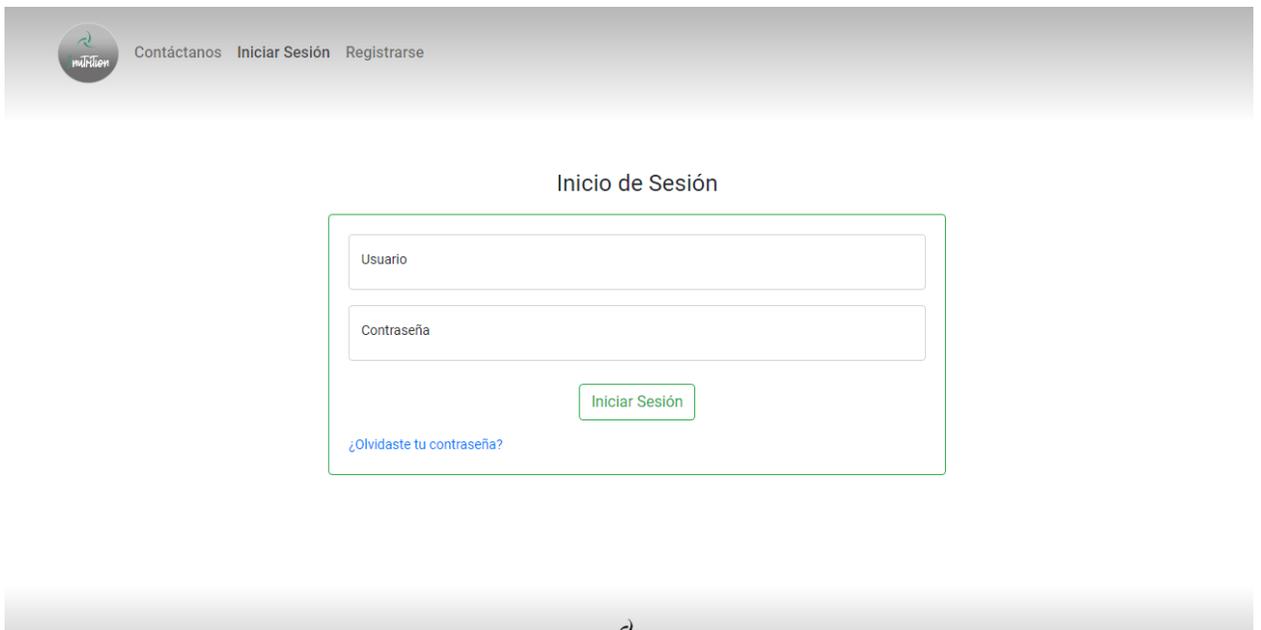


**Figura 5: Diagrama de despliegue**

Los administradores manejan la base de datos y los registros de los nutricionistas, los cuales están encargados de manejar las valoraciones de los pacientes que se encuentran registrados en la base de datos y relacionados a ellos. La información de menús, recetas y alimentos se encuentran almacenados en la base de datos y todas esta información se consume/obtiene por medio de peticiones realizadas desde la interfaz del usuario.



**Figura 6: Inicio de la aplicación**



**Figura 7: Página de inicio de sesión**

### Registro de nutricionista

Nombre	Apellido
Identificación	Fecha de nacimiento dd/mm/aaaa 
Especialidad	No. Tarjeta Profesional
Correo Electrónico	Número de Celular
<a href="#">Guardar</a>	<a href="#">Cancelar</a>

**Figura 8: Formulario de registro de nutricionista**

*Alejandro Montoya*  
Nutricionista con énfasis en nutrición deportiva

Pacientes	Banco de recetas	Configuración de la cuenta
-----------	------------------	----------------------------



**Figura 9: Menú nutricionista**

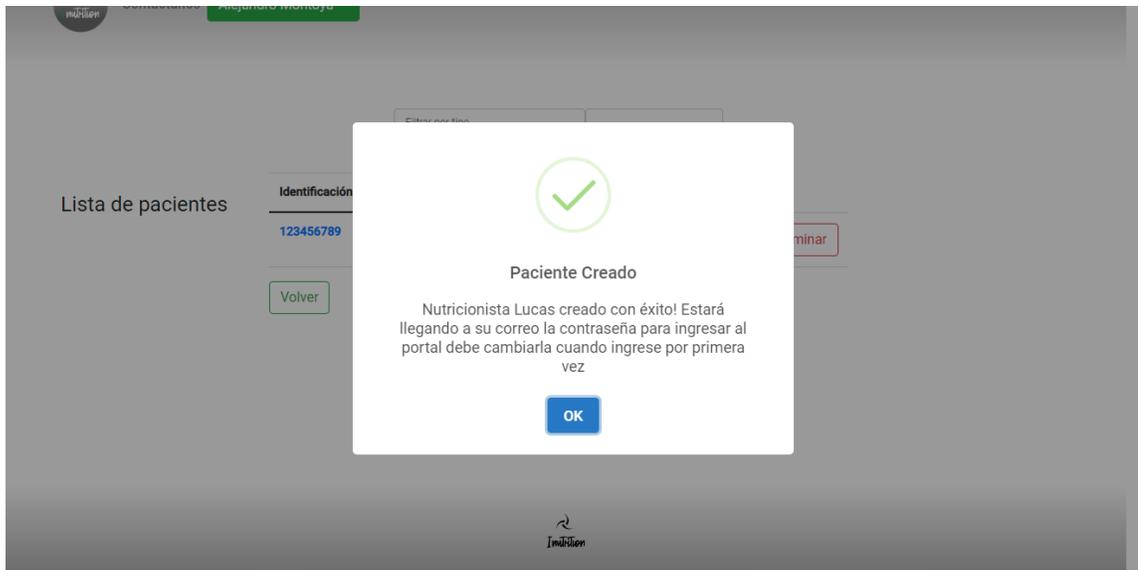
Ver Paciente	Registrar Paciente	Pacientes Deshabilitados	Eliminar Paciente
			Identificación

**Figura 10: Opciones gestión pacientes**

### Registro del paciente

Nombre
Apellido
Género ▾
Identificación
Correo Electrónico
Número de Celular
Fecha de nacimiento dd/mm/aaaa 
Escolaridad

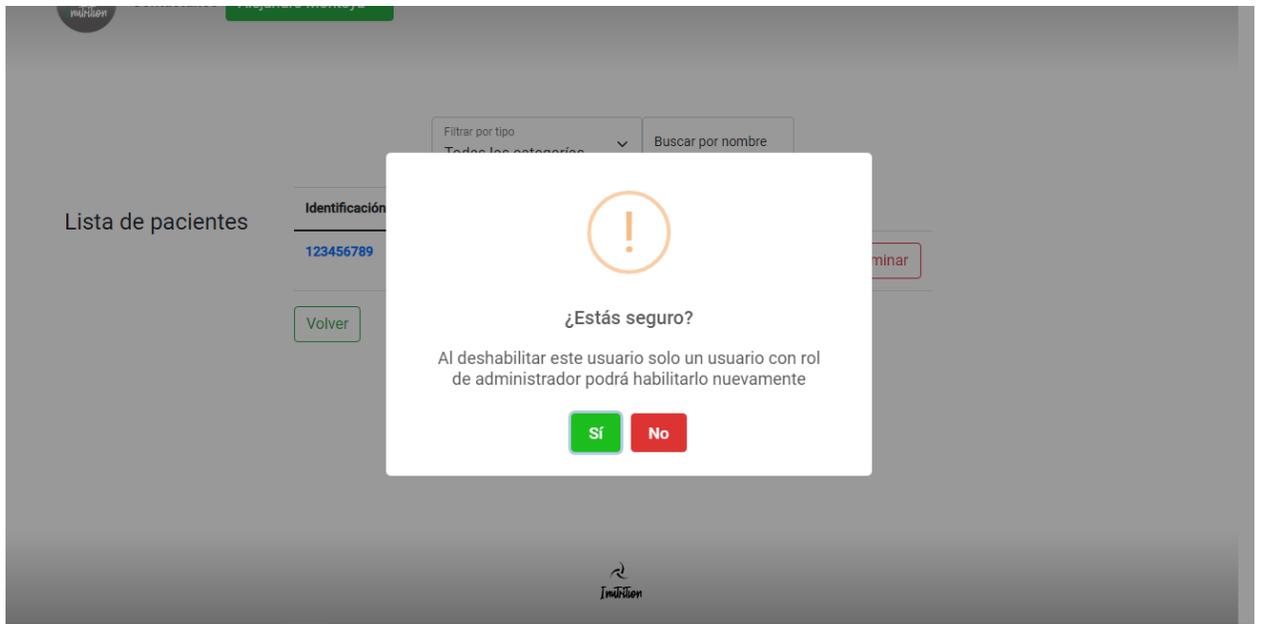
**Figura 11: Formulario registro paciente**



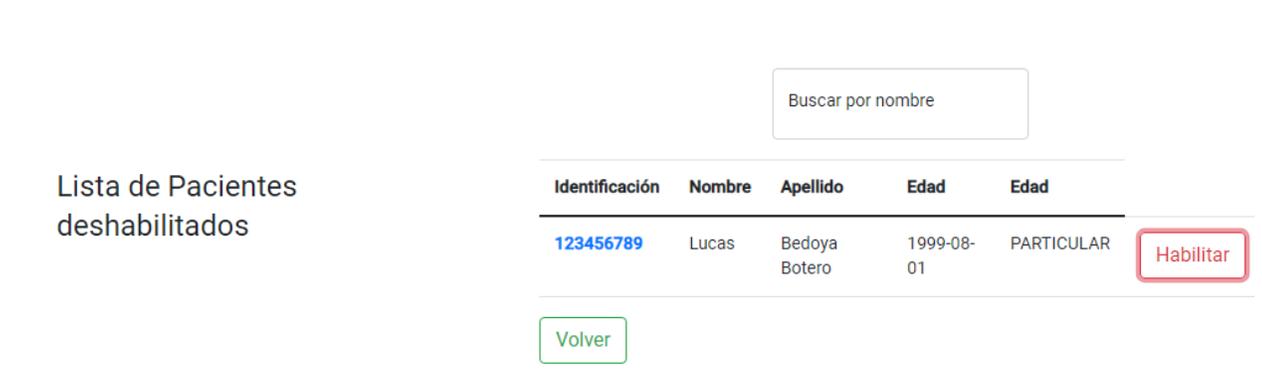
**Figura 12: Modal de registro de paciente**



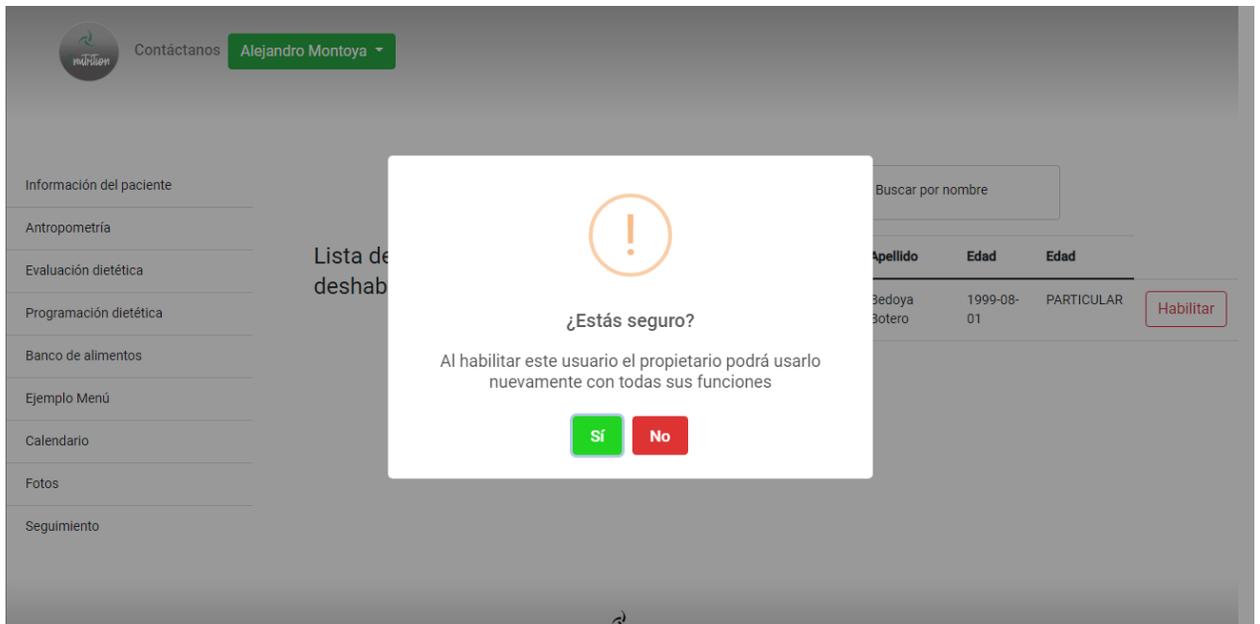
**Figura 13: Lista de pacientes**



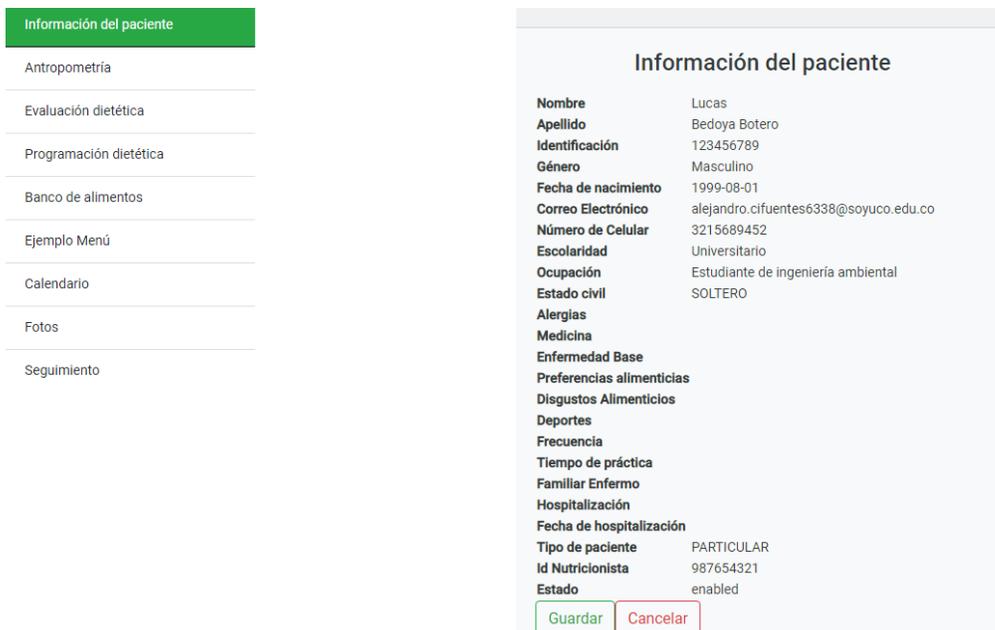
**Figura 14: Modal de eliminación de paciente**



**Figura 15: Lista de pacientes deshabilitados**



**Figura 16: Modal habilitar paciente**



**Figura 17: Información del paciente**

Información del paciente Historial antropométrico para paciente con id 123456789

**Antropometría**

Evaluación dietética

Programación dietética

Banco de alimentos

Ejemplo Menú

Calendario

Fotos

Seguimiento

Buscar por nombre

[Crear una nueva](#)

Thu Oct 07 2021 17:49:36 GMT-0500 (hora estándar de Colombia)

Thu Oct 07 2021 17:50:39 GMT-0500 (hora estándar de Colombia)

**Intuition**

**Figura 18: Historial antropométrico del paciente**

Información del paciente Datos Antropométricos

**Antropometría**

Evaluación dietética

Programación dietética

Banco de alimentos

Ejemplo Menú

Calendario

Fotos

Seguimiento

**Basicos**

Peso  kg      Talla  mt

**Pliegues Cutáneos**

Subescapular  mm      Cresta Iliaca  mm

Muslo Anterior  mm      Tricipital  mm

Supraescapular  mm      Pierna Medial  mm

Bicipital  mm      Abdominal  mm

**Diámetros**

Diámetro de femur  mm      Diámetro de Húmero  mm

Perímetro de brazo corregido  cm      Perímetro de pierna corregida  cm

**Figura 19: Formulario de registro antropometría paciente**

Lista de alimentos

buttonText.searchByFoodGroup

	Grupo de Alimentos	kcal	Alimento	Medida Casera	Medida Real
<b>Agregar</b>	Leches enteras frescas y fermentadas	153	Kumis de leche entera	1 vaso pequeño	200
<b>Agregar</b>	Leches enteras frescas y fermentadas	138	Leche de cabra cruda	1 vaso pequeño	200
<b>Agregar</b>	Leches enteras frescas y fermentadas	120	Leche de vaca entera cruda	1 vaso pequeño	200
<b>Agregar</b>	Leches enteras frescas y fermentadas	129	Leche de vaca entera en polvo	3 cucharadas soperas colmadas	26
<b>Agregar</b>	Leches enteras frescas y fermentadas	122	Leche de vaca entera pasteurizada	1 vaso pequeño	200
<b>Agregar</b>	Leches enteras frescas y fermentadas	140	Yogur regular de leche entera	1 vaso pequeño	200

**Figura 20: Modal lista de alimentos para evaluación dietética**

Información del paciente	Ejemplo menu						
Antropometría	Tipo de Comida	Hora	Alimento	Medida Casera	Medida Real	Porción	Preparación
<b>Evaluación dietética</b>	X Comida del <b>Desayuno</b>	9:00am	Leche de cabra cruda	1 vaso pequ	200	1	
Programación dietética	X Comida del <b>Almuerzo</b>	1:00pm	Salchichón de pollo	1 trozo medi	40	1	
Banco de alimentos	X Comida del <b>Cena</b>	6:00pm	Lengua de res	1/5 libra	95	1	
Ejemplo Menú	X Comida del <b>Desayuno</b>	9:00am	Mantequilla sin sal	1 cuchara de	5	1	
Calendario							
Fotos							
Seguimiento							

**Figura 21: Ejemplo menú**

Información del paciente **Ejemplo menú**

Tipo de Comida	Hora	Alimento	Medida Casera	Medida Real	Porción	Preparación
X Comida del Desayuno	9:00am	Leche de cabra cruda	1 vaso pequi	200	1	
X Comida Almuerzo					1	
X Comida Cena					1	
X Comida Desayuno					1	

Guardado exitosamente

El recordatorio se ha guardado exitosamente para el usuario con identificación 123456789

OK

Agregar más (General) | Agregar más (Niños) | Agregar más (Adultos) | Guardar

Frecuencia de consumo

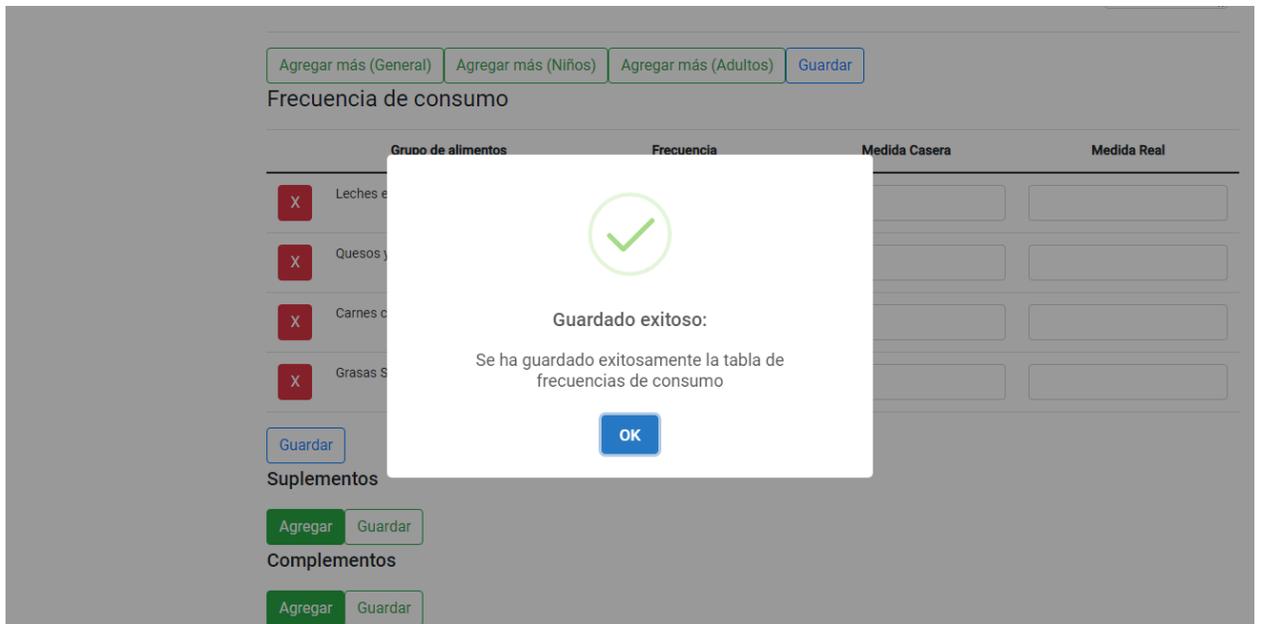
Grupo de alimentos	Frecuencia	Medida Casera	Medida Real
X Leches enteras frescas y fermentadas	Diariamente		
X Quesos y sustitutos	Cada dos días		
X Carnes crudas altas en lípidos	Diariamente		
X Grasas Saturadas	1 vez por semana		

**Figura 22: Modal ejemplo menú guardado**

Frecuencia de consumo

Grupo de alimentos	Frecuencia	Medida Casera	Medida Real
X Leches enteras frescas y fermentadas	Diariamente		
X Quesos y sustitutos	Cada dos días		
X Carnes crudas altas en lípidos	Diariamente		
X Grasas Saturadas	1 vez por semana		

**Figura 23: Frecuencia de consumo**



**Figura 24: Modal frecuencia de consumo guardada**

**Suplementos**

	Suplemento	Marca	Dosis	Cantidad	Método de consumo
X	1KVITC	1KVITC	3 mg	300ml	En jugo
X	Biotin	21st Century	3mg	300ml	En jugo

Agregar Guardar

**Complementos**

	Complemento	Marca	Dosis	Cantidad	Método de consumo
X	Amino Core	all max	5mg	300ml	En jugo

Agregar Guardar

**Figura 25: Complementos y suplementos**

## Banco de alimentos

Alimentos para regímenes especiales
Bebidas alcohólicas y no alcohólicas
Carnes altas en lípidos
Carnes y derivados
Cereales adultos
Cereales niños
Cereales y derivados
Frutas y derivados_
Grasas monoinsaturadas
Grasas saturadas
Grasas y aceites

Figura 26: Banco de alimentos

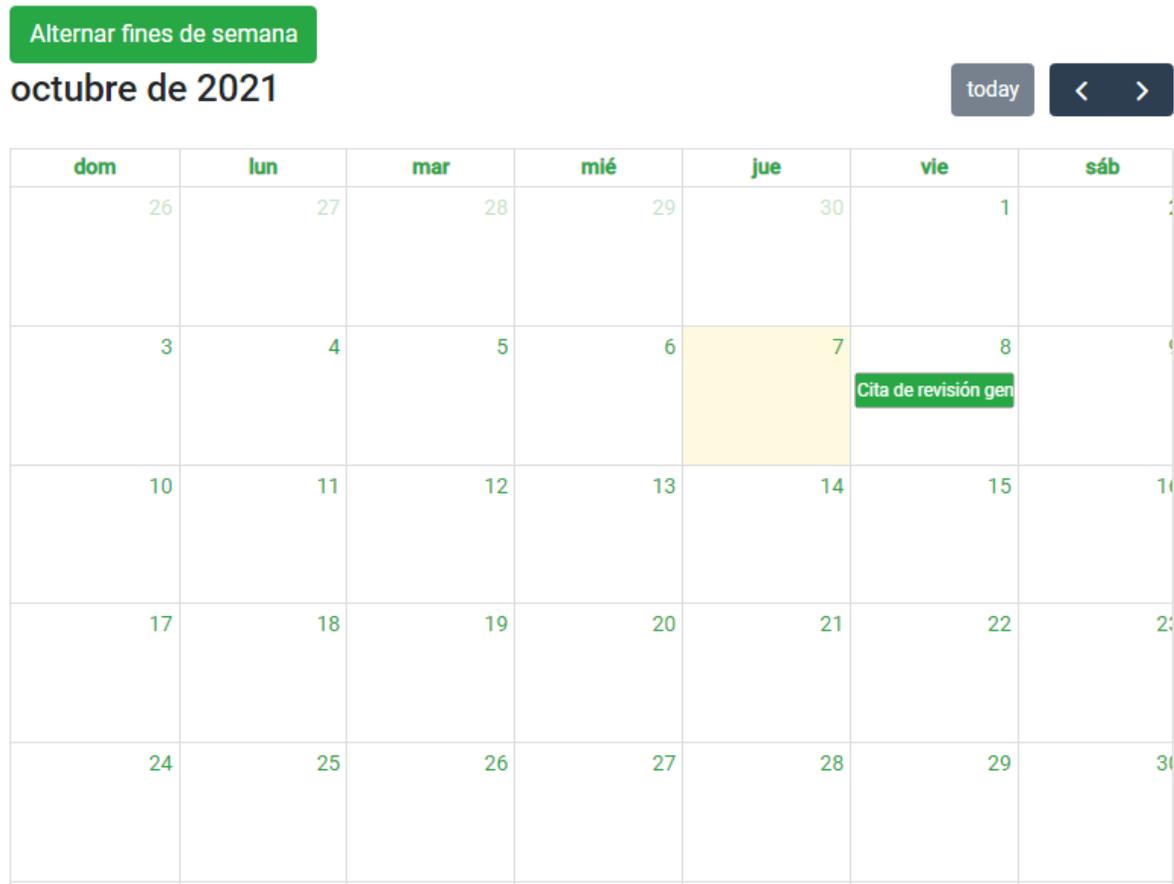
# Creación de consulta

Título Cita de revisión general	
Identificación del paciente 123456789	Identificación del nutricionista 987654321
Fecha de la consulta 2021-10-08	Hora de la consulta 2:00pm
Lugar de la consulta Centro comercial san francisco local 301	Estado de consulta Pendiente

**Guardar**

Cerrar

Figura 27: Formulario creación consulta



**Figura 28: Calendario de citas**

Registro fotografía

Fecha Creación  
07-10-2021

Identificación del paciente  
123456789

Descripción  
|

Elegir archivo No se ha seleccionado ningún archivo

Guardar

Cerrar

**Figura 29: Registro fotografía**

## Registro de receta

Nombre	Número de porciones
Calorías	Proteínas
Grasas	Carbohidratos
Comida del día <span>▼</span>	
Ingredientes	
Preparación	
<a href="#">Guardar</a>	<a href="#">Cancelar</a>

Figura 30: Registro de receta

### Lista de Recetas

Filtrar por tipo  
Todas las categorías ▼

[Crear una nueva](#)



**Arroz Con huevo**

Calorías	324
Proteínas	10.21
Grasas	8.6
Carbohidratos	1.1

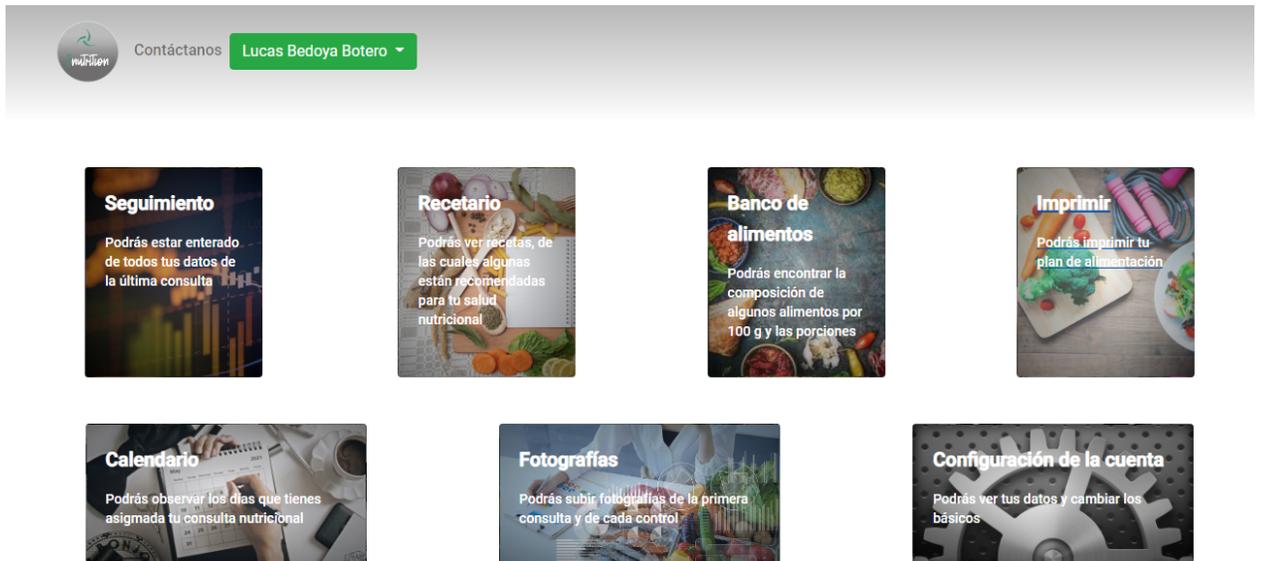
[Ver](#)

**Figura 31: Lista de recetas**

**Configuración de la cuenta**

Nombre Alejandro	Apellido Montoya
Identificación 987654321	Especialidad Nutricionista con énfasis en nutrición deportiva
Correo Electrónico alejox123321@gmail.com	Número de Celular 123456789
Fecha de nacimiento 01/08/1999	No. Tarjeta Profesional 123456789
Contraseña actual	Nueva Contraseña
Confirmar nueva contraseña	
<a href="#">Guardar Cambios</a>	<a href="#">Cancelar</a>

**Figura 32: Formulario de configuración de la cuenta para nutricionista**



**Figura 33: Menú paciente**

## 7.5. Desarrollo

### 7.5.1. Arquitectura (Backend):

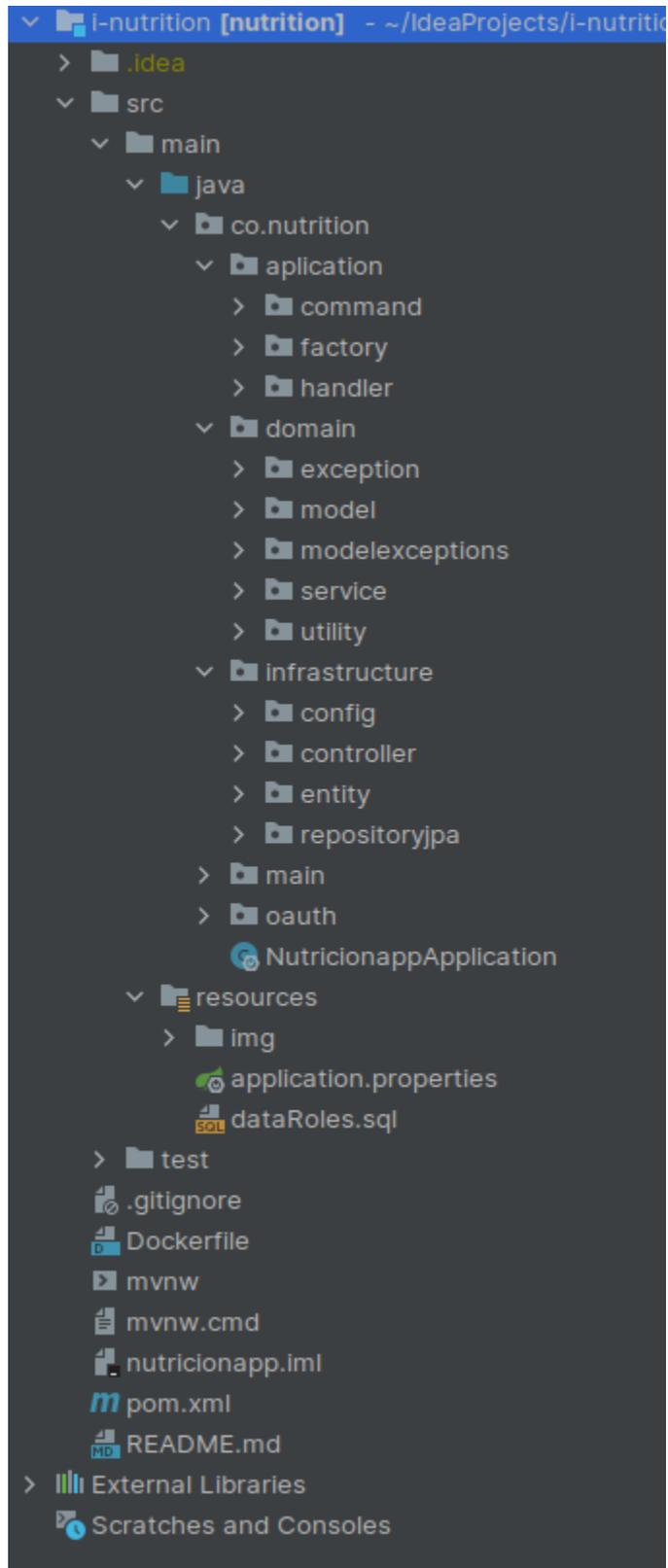
A nivel de Back-End se implementó un patrón de diseño hexagonal distribuido en carpetas los cuales se dividen en dominio, aplicación e infraestructura

La tecnología utilizada para el desarrollo es Spring-Boot acompañado de Maven como gestor de dependencia, realizando APIs expuestas las cuales son consumidas desde postman para metodología de pruebas, y ya unificado para con el Front-End para la finalización del proceso

Se realizó la implementación de seguridad para la aplicación con un Oauth2, complementado de una codificación en base 64 de RSA para la manutención segura de las contraseñas de los usuarios de la aplicación

Se implementaron propiedades y dependencias de Spring-Boot email, para hacer llegar a los usuarios registrados un correo el cuál tendrá la contraseña aleatoria que el sistema le brinda para realizar su ingreso desde el servidor que contiene el aplicativo INutition, está contraseña se guardará en Base de datos, pero su versión codificada para así mantener la transparencia y la seguridad de cada usuario

El motor de Base de Datos utilizado es MySQL, ya que es unas de las opciones más estables y con mejor manejo dentro de los servidores, pensando a la hora de realizar el despliegue de la aplicación.



**Figura 34. Estructura del proyecto backend**

### 7.5.1.1. Herramientas tecnológicas utilizadas en backend

**7.5.1.1.1. Java:** Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

**Porque se utiliza:** Java es el lenguaje de programación, con el cual nos sentimos cómodos trabajando, ya que dominamos y tenemos más conocimiento de este, comparados con otros lenguajes.

**7.5.1.1.2. JWT:** JWT (JSON Web Token) es un estándar que está dentro del documento RFC 7519. En el mismo se define un mecanismo para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario, además con una serie de claims o privilegios. Estos privilegios están codificados en objetos de tipo JSON, que se incrustan dentro de del payload o cuerpo de un mensaje que va firmado digitalmente.

**Porque se utiliza:** Ya que es una herramienta fácil de utilizar y es muy confiable a la hora de realizar la encriptación de los datos, ya que este lo

realiza con un RSA, lo hace difícil de descifrar por una persona externa al sistema.

**7.5.1.1.3. Spring Boot:** El uso de inyección de dependencias facilita la programación contra interfaz, permitiendo a los distintos componentes depender únicamente de interfaces y produciendo así un código menos acoplado. No solo eso, también permite implementar el patrón singleton de una forma extremadamente sencilla

**Porque se utiliza:** Spring Boot nos permite automatizar gran parte de la configuración necesaria, facilita la configuración del proyecto ya que se autoconfigura y también nos facilita el empaquetado.

**7.5.1.1.4. Maven:** Maven es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant, pero tiene un modelo de configuración de construcción más simple, basado en un formato XML.

**Porque se utiliza:** Ya que evitan tiempos de configurar e implementar el entorno de desarrollo para poder centrarnos en lo realmente interesante de

un proyecto Java, facilita el manejo de dependencias y la estructuración inicial de un proyecto

**7.5.1.1.5. MySql:** MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basado en código abierto. Desarrollado originalmente por MySQL AB, fue adquirida por Sun Microsystems en 2008 y esta su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL.

**Porque se utiliza:** Es un gestor de base de datos fácil de utilizar y seguro para almacenar data confidencial, y el uso dentro de contenedores de docker tiene un uso fácil.

**7.5.1.1.6. IntelliJ:** es un IDE inteligente y sensible al contexto para trabajar con Java y otros lenguajes JVM como Kotlin, Scala y Groovy en todo tipo de aplicaciones.

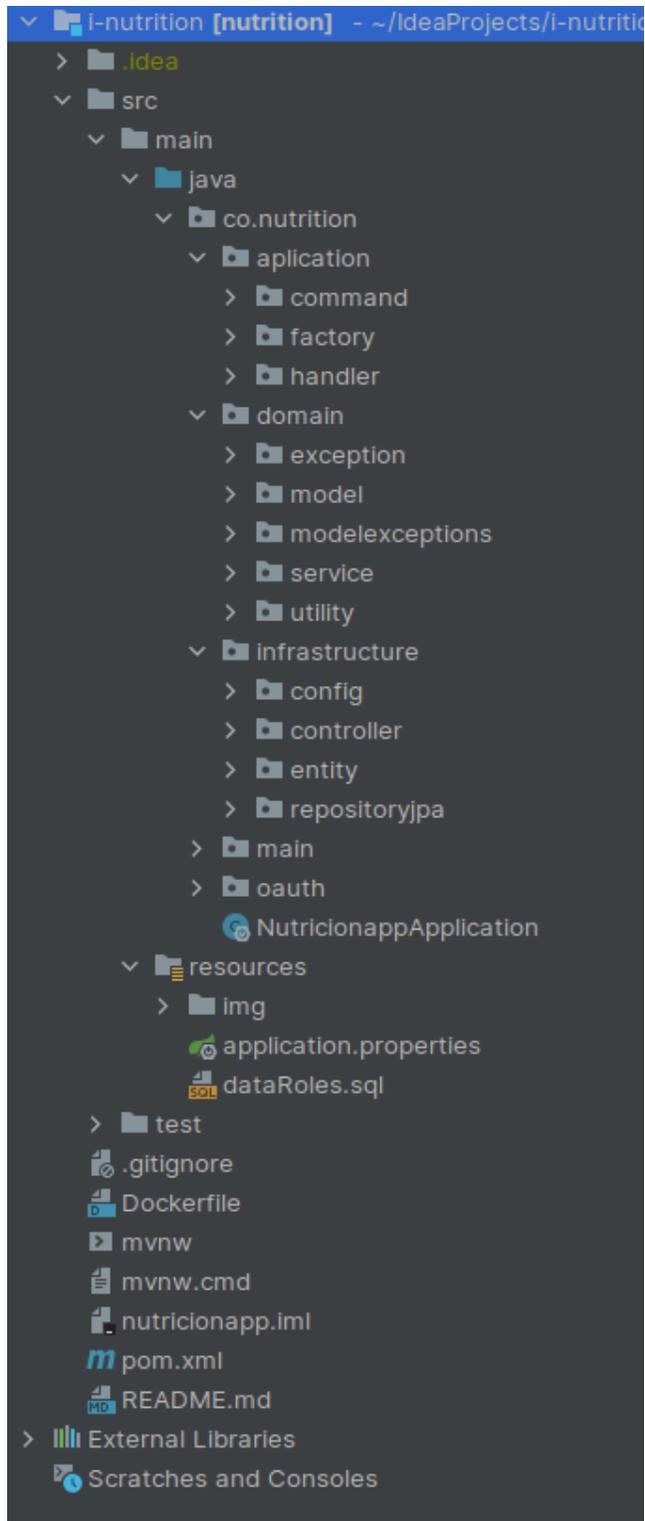
**Porque se utiliza:** Se opta por este IDE ya que proporciona un alto soporte para el framework Spring boot y para Java, además de que permite su integración con GitHub por lo que no se requiere utilizar herramientas externas para el control de versiones. Todo se controla desde el mismo entorno de desarrollo y esto permite que los recursos

consumidos por el equipo en el proceso de desarrollo sean menores, lo que optimiza la ejecución del proyecto para visualizar cambios.

### **7.5.2. Arquitectura (Frontend):**

Para el frontend se usaron librerías de angular para implementar toda la parte visual y funcional de la capa de presentación. Se usaron librerías como Router que es utilizada para direccionar entre los diferentes componentes de la aplicación; ReactiveForm para el uso de los formularios reactivos que cambien de acuerdo a las acciones de los usuarios, RxJs y sus operadores para el manejo de observables al hacer consultas al backend. Se usaron apis externas como lo son FullCalendar para el manejo de calendarios y un API de excel para poder visualizar los archivos de cálculo.

Para la seguridad a nivel de frontend se implementaron guards para proteger los componentes de personas que no tengan una cuenta en la aplicación activa y también para prevenir que un usuario no ingrese a los componentes a los que no debería tener acceso según indica la lógica de negocio del proyecto. Para el proceso de maquetado se utilizaron formularios, tablas, banners y tarjetas que ayudaron a mejorar la experiencia de usuarios en la aplicación, estos componentes se programaron para que funcionen en diferentes dispositivos como celulares, tablets, computadores o televisores de alta resolución.



**Figura 35. Estructura del proyecto frontend**

### 7.5.2.1. Herramientas tecnológicas utilizadas en frontend

**7.5.2.1.1. TypeScript:** Es un lenguaje de programación desarrollado por Microsoft basado en JavaScript. is a strongly-typed superset of JavaScript. TypeScript adds static types to JavaScript. TypeScript agrega la parte de fuertemente tipado que JavaScript no tiene pero utilizando sus estructuras declarativas. TypeScript usa la sintaxis de JavaScript y C# y compilado como JavaScript nativo, este lenguaje es utilizado para el desarrollo de aplicaciones open-source en proyectos como Angular.

**¿Por qué se utiliza?** Se usa este lenguaje ya que Angular lo utiliza por defecto para la programación de la lógica de componentes, servicios, módulos y guardias.

**7.5.2.1.2. Angular:** Es una plataforma de desarrollo construida en TypeScript que incluye un framework basado en componentes para construir aplicaciones web escalables; una colección de librerías bien integradas que abarcan una gran variedad de características, incluyendo enrutamiento, manejo de formularios, comunicación entre cliente y servidor además de una suite de herramientas de desarrollador que ayudan en el proceso de desarrollo. (angular.io, 2021).

**¿Por qué se utiliza?** Se prefirió este debido a que Angular separa los componentes de forma estructurada, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC

(Modelo-Vista-Controlador) asegurando los desarrollos con rapidez gracias a su sistema de línea de comandos.

**7.5.2.1.3. Angular CLI:** Es una herramienta de interfaz de línea de comandos que se utiliza para inicializar, desarrollar, escalar y mantener aplicaciones de angular directamente desde la línea de comandos. (angular.io, 2016).

**¿Por qué se utiliza?** Se optó por esta herramienta al estar utilizando angular, ya que este permite crear componentes, servicios, guardias y módulos que permiten estructurar de manera adecuada los componentes de la aplicación.

**7.5.2.1.4. CSS:** Siglas de Cascading Style Sheets, las cuales describen como los elementos HTML son desplegados en la pantalla, artículo u otro medio. CSS ahorra mucho trabajo ya que puede controlar la apariencia de múltiples páginas a la vez. (W3School).

**¿Por qué se utiliza?** Se usa esta tecnología ya que el alcance de la aplicación no era tan avanzado para implementar el uso de SCSS, además de que se evita el paso de compilar los estilos para luego pasarlos a CSS en el servidor.

**7.5.2.1.5. HTML:** Siglas de Hyper Text Markup Language, el cual es el lenguaje estándar de etiquetado para las páginas web. Sus elementos son contruidos en bloques los cuales son representados por <> como etiquetas. (W3School).

**¿Por qué se utiliza?** Se usa esta tecnología ya que es el lenguaje de etiquetas de la web y es con lo que todas las páginas se encuentran maquetadas en la actualidad.

**7.5.2.1.6. RxJs:** Es una extensión reactiva para JavaScript, la cual es una librería para programación reactiva usando observables que hacen más fácil componer de manera asíncrona el código. Proporciona una implementación del tipo Observable, el cual es necesario hasta que el tipo se vuelva parte del lenguaje y hasta que el navegador lo soporte. Cuenta con múltiples operadores que permiten realizar diferentes operaciones con estos observables. (Angular.io, 2010).

**¿Por qué se utiliza?** Se usa esta tecnología ya que al manejar observables se realizan procesos asíncronos permite que se ejecuten varias peticiones HTTP de manera simultánea.

**7.5.2.1.7. Bootstrap:** Es una colección gigante de bits de código reusable escrito en HTML, CSS y JavaScript, también es un entorno de programación frontend que permite a los desarrolladores y diseñadores construir rápidamente sitios web responsive. (Alexander Ouellete, 2021).

**¿Por qué se utiliza?** Se utilizan clases de bootstrap para no reinventar los estilos que ya este trae en sus librerías.

**7.5.2.1.8. SweetAlert:** Es un plugin de jQuery y con el cual podremos dar un aspecto profesional a los mensajes que lanzamos a los usuarios acorde a las tendencias actuales.

**¿Por qué se utiliza?** Se utiliza para ahorrar tiempo en el desarrollo de componentes de alertas.

**7.5.2.1.9. FullCalendar:** un plugin jQuery que proporciona un completo calendario que se puede alimentar por AJAX (o no, si definimos los eventos en el momento de su creación), y que nos permite incluso arrastrar y soltar eventos, o incluso sincronizarlo con Google Calendar.

**¿Por qué se utiliza?** Se optó por esta aplicación ya que permite mostrar de manera dinámica las fechas de las consultas.

**7.5.2.1.10. WebStorm:** WebStorm es un entorno de desarrollo integrado para JavaScript y las tecnologías relacionadas. (Webstorm Official Site).

**¿Por qué se utiliza?** Se opta por este IDE ya que proporciona un alto soporte a las diferentes tecnologías y lenguajes de programación utilizados en el proyecto, además de que permite su integración con github por lo que no se requiere utilizar herramientas externas para el control de versiones. Todo se controla desde el mismo entorno de desarrollo y esto permite que los recursos consumidos por el equipo en el proceso de desarrollo sean menores, lo que optimiza la renderización de componentes.

### **7.5.3. Despliegue de la aplicación**

El despliegue de la aplicación se realizó en AWS (Amazon Web Service), utilizando un sistema operativo Linux, para el manejo rápido, la base de datos sigue siendo un MySQL alojado en la nube de AWS.

Para realizar el despliegue de el Back-End y el Front-End, se realiza por medio de Docker, cada proyecto tiene su distintiva configuración para la complicación correspondiente y adecuada de cada ambiente

El Front-End posee un archivo adicional el cuál es el NGinxs el cuál funciona también como un balanceador de cargas hacia el Back-End

## **8. Conclusiones**

Con la realización de este proyecto se puede destacar como se implementa todo lo aprendido durante la carrera y durante nuestros procesos de prácticas, como se estructuró la arquitectura de la aplicación, como se desarrollaron los diseños iniciales, como se dividen las tareas en los diferentes sprint y como hubo buena comunicación en el equipo de desarrollo para la completación del proyecto.

## **9. Referencias bibliográficas**

- Nutrimind. (s.f.). Nutrimind. Jardines de la Asunción, Aguascalientes, México. Obtenido de <https://www.nutrimind.net/>
- Dietopro. (2019). Dietopro. Puerto de Sagunto, Valencia, España. Obtenido de Dietopro.com

- Aznutrition. (s.f.). Aznutrition. Lago Ginebra, Guadalajara, México. Obtenido de <https://www.aznutrition-oficial.com/>
- diet-creator. (s.f.). diet-creator. s.f, s.f, s.f. Obtenido de <https://www.diet-creator.com/es>
- dietfarma. (s.f.). dietfarma. Sevilla, España. Obtenido de <https://www.dietfarma.com/>
- enPeso. (s.f.). enPeso.com. s.f, s.f, s.f. Obtenido de [enPeso.com](https://www.enpeso.com)
- What is TypeScript? (2019) Tomado de: [What Is TypeScript? \(c-sharpcorner.com\)](https://c-sharpcorner.com/2019/07/what-is-typescript/)
- What is Angular? (2021) Tomado de: <https://angular.io/guide/what-is-angular>
- CLI Overview and Command Reference. (2016) Tomado de: <https://angular.io/cli>
- What is CSS? Tomado de: [https://www.w3schools.com/whatis/whatis\\_html.asp](https://www.w3schools.com/whatis/whatis_html.asp)
- What is HTML? Tomado de: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)
- The RxJS Library. (2010) Tomado de: <https://angular.io/guide/rx-library>
- What is Bootstrap? (2021) Tomado de: <https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/>
- What is WebStorm. Tomado de: <https://www.jetbrains.com/es-es/webstorm/>