

Universidad ORT Uruguay

Facultad de Ingeniería

# **Ingeniería de Software 2**

## **Obligatorio 2**

Joselen Cecilia (233552)

Belén Martinez (223694)

Camila Burgueño (221951)

Entregado como requisito de la materia Ingeniería de  
Software 2

7 de diciembre de 2020

<https://www.overleaf.com/read/mnjzrcmgdgcj>

# Declaraciones de autoría

Nosotras, Camila Burgueño, Joselen Cecilia y Belén Martínez, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos Ingeniería de Software 2 ;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

## **Resumen**

Este documento forma parte del proyecto Alimentación Saludable. El mismo tiene como objetivo detallar el estado de calidad del mismo, a la vez que justificarlo. Contiene los estándares y métricas de calidad usados por el equipo. También se habla acerca del desarrollo del producto. Además, el documento cuenta con una descripción del producto en lo que respecta a funcionalidades, alcance, capacidad. Este documento es un componente crucial del producto, y un complemento de la documentación entregada como documentación del obligatorio

# Índice general

<b>1. Introducción</b>	<b>2</b>
1.1. Objetivos del documento . . . . .	2
1.2. Descripción del sistema . . . . .	2
<b>2. Versionado</b>	<b>4</b>
2.1. Repositorio . . . . .	4
2.1.1. Criterios de versionado . . . . .	4
<b>3. Funcionalidades del Sistema</b>	<b>5</b>
<b>4. Requerimientos</b>	<b>6</b>
4.1. Introducción . . . . .	6
4.1.1. Requerimientos Funcionales (RF) . . . . .	6
<b>5. Modelado de usuarios</b>	<b>10</b>
5.1. Interesados . . . . .	10
5.1.1. Registro de interesados . . . . .	10
5.2. Descripción de usuarios . . . . .	11
5.2.1. Perfil de usuarios . . . . .	11
5.2.2. Contexto de uso . . . . .	12
<b>6. Calidad del código</b>	<b>13</b>
6.1. Estándares de codificación . . . . .	13
6.2. Cobertura de pruebas unitarias . . . . .	16
6.3. Análisis de código . . . . .	18
<b>7. Evaluación de usabilidad</b>	<b>26</b>
<b>8. Pruebas funcionales</b>	<b>31</b>
8.1. Técnicas de pruebas aplicadas . . . . .	31
8.1.1. Casos de uso . . . . .	31
8.2. Casos de prueba . . . . .	40
8.2.1. Casos de prueba . . . . .	55
<b>9. Reporte de defectos</b>	<b>71</b>
9.1. Definición de categoría de defectos . . . . .	71
9.2. Defectos encontrados por iteración . . . . .	72

# 1. Introducción

## 1.1. Objetivos del documento

Este documento es una nueva versión de la documentación original del proyecto Alimentación Saludable la cual nos fue entregada en el obligatorio 1. En esta nueva versión se mejora la documentación anterior y además se agregan nuevas secciones.

Secciones que ya estaban y serán mejoradas:

- Estándar de codificación
- Pruebas unitarias
- Análisis de código
- Interfaz de usuario y usabilidad
- Pruebas funcionales
- Reporte de defectos
- Reflexión
- Bibliografía

Secciones nuevas:

- Estándar de codificación
- Versionado
- Funcionalidades del sistema
- Requerimientos Funcionales y No-Funcionales
- Modelado de usuarios

El siguiente documento tiene como objetivo presentar la documentación necesaria para el desarrollo de un sistema de software. Este documento es muy importante para que el proyecto sea de alta calidad y por lo tanto mantenible. Es útil si a futuro se cambia el equipo de desarrollo y hay que mantener o agregarle nuevas funcionalidades al proyecto.

## 1.2. Descripción del sistema

Al abrir la aplicación se muestra una pantalla principal con tres usuarios (Usuario(Cliente), Profesional, Administrador) donde elegís el tipo de usuario. En caso de elegir cliente o profesional se pide el nombre de usuario para ingresar al sistema. Al

elegir cliente se muestran los botones del cliente los cuales son:

- Home
- Perfil
- Consulta directa
- Pedir plan
- Agregar comida a dieta
- Cambiar usuario

Al elegir profesional se muestran los botones del profesional, estos son:

- Home
- Realizar plan
- Consulta directa
- Cambiar usuario

Por último en caso de elegir administrador se pedirá el nombre de usuario y la contraseña del mismo para ingresar al sistema. En caso de no estar registrado, se solicita que se registre con un nombre de usuario y una contraseña. Luego al iniciar sesión se muestran los botones: -Registrar Usuario

- Registrar Profesional
- Registrar Alimento
- Cambiar Usuario.

## 2. Versionado

### 2.1. Repositorio

Como establece la ingeniería de software, a la hora de asegurar la calidad de un producto en su desarrollo, es necesario un sistema que coordine los cambios a lo largo de toda la vida del producto. En este caso la herramienta elegida por el equipo para mantener el versionado es Git.

Git es un sistema de control de versiones distribuido, cuyo propósito es mantener un registro de los cambios en las versiones de un software y ayudar en la coordinación de desarrollos entre varias personas que realizan cambios sobre archivos ubicados en repositorios locales.

Para este proyecto creamos un repositorio Git en la nube, de nombre obligatorio-2 y lo clonamos en un repositorio local en la computadora de cada una de las participantes de esta tarea, donde estaremos aplicando cambios y subiendo mediante los comandos commit y push al repositorio remoto.

#### 2.1.1. Criterios de versionado

Para el versionado decidimos utilizar la estructura GitFlow, la cual consta de una rama Master donde se encuentra la versión final y estable, y una rama Develop donde se crean las features branch correspondientes a cada uno de los cambios.

En cada una de las ramas, realizamos una serie de commits los cuales agregaban los cambios realizados en el código.

Luego de completado cada uno de los cambios, hicimos una revisión, realizamos correcciones que nos parecían pertinentes y fuimos mergeando la versión final de cada una a la rama Develop, para luego hacer un merge final con la rama Master y así dejar la última versión entregable de este proyecto en la rama principal del repositorio.

A continuación agregamos el link de nuestro repositorio: <https://github.com/JoselenC/Obligatorio-2>

### 3. Funcionalidades del Sistema

1. El sistema permite al usuario loguearse como usuario(cliente), profesional o administrador.
2. El sistema permite que el administrador registre usuarios nuevos como profesionales y como usuarios(clientes), llenando todos los campos que solicitan información necesaria de cada uno. Dentro de estos campos está la opción de incluir una foto en formato “.png”, “.jpg” o “.jpeg”. para identificar al usuario.
3. El sistema permite que el administrados pueda registrar un alimento indicando nombre, tipo de alimento y nutrientes principales.
4. El usuario puede solicitar una dieta a alguno de los profesionales registrados en el sistema.
5. El profesional puede generar una dieta para un usuario que haya solicitado una.
6. El usuario puede enviar una consulta directa a un profesional y recibir respuesta del mismo.
7. El profesional puede atender consutlas directas.
8. El usuario tiene la posibilidad de agregar las comidas ingeridas diariamente
9. Tanto el usuario como el profesional pueden editar los datos de sus perfiles



## 4. Requerimientos

### 4.1. Introducción

En esta sección vamos a especificar los Requerimientos Funcionales y los No-Funcionales. Los primeros especifican servicios o funciones del sistema. Brindan una descripción del comportamiento del sistema en condiciones específicas. Los segundos, pueden ser restricciones a los servicios provistos por el sistema o atributos de calidad.

Es importante seguir las formalidades establecidas a la hora de generar el ESRE (documento de especificación de requerimientos). Las formalidades elegidas consisten en identificar cada requerimiento con un ID (RFXX o RNFX, siendo XX un número natural de 2 cifras), un nombre o título (descriptivo del mismo), una descripción, y una prioridad asignada al mismo, definimos un rango de prioridad que varía entre:

Prioridad alta: consideramos que estos requerimientos son los que aportan funcionalidades o restricciones que se deben implementar necesariamente para lograr el objetivo del sistema.

Prioridad media: consideramos que estos requerimientos son importantes, pero no son fundamentales para lograr el objetivo del sistema. Son requerimientos que aportan calidad a la solución, pero no son imprescindibles.

Prioridad baja: consideramos que estos requerimientos agregarían valor a la experiencia de usuario, es una buena implementación como mejora del sistema; pero no aportan al cumplimiento del objetivo.

#### 4.1.1. Requerimientos Funcionales (RF)

##### **Requerimientos Funcionales (RF)**

RF01: Registro administrador.

Descripción: El sistema debe permitir registrarse como administrador. Se debe ingresar un nombre de usuario y una contraseña que contenga menos de 8 caracteres.

Prioridad: Alta

RF02: Registro usuario/profesional

Descripción: El sistema debe permitir al administrador registrar nuevos usuarios(clientes) y profesionales.

Prioridad: Alta

RF03: Ingresar al sistema

Descripción: El sistema debe permitir ingresar al sistema al usuario(cliente) y profesional pidiéndole el número de usuario y al administrador pidiéndole nombre de usuario y contraseña.

Prioridad: Alta

RF04: Imagen de usuario/profesional

Descripción: El sistema permite subir imágenes al momento de registrarse como usuario o profesional.

Prioridad: Baja

RF05: Vista de los Menús

Descripción: El sistema debe visualizar un menú con distintas funcionalidades para usuario, profesional y administrador.

Prioridad: Alta

RF06: Solicitar plan de alimentación

Descripción: El sistema permitirá a los usuarios solicitar un plan de alimentación al profesional que desee.

Prioridad: Alta

RF07: Historial de comidas

Descripción: El sistema debe permitirle mostrarle al usuario su historial de comidas por fecha.

Prioridad: Baja

RF08: Comidas Ingeridas

Descripción: El sistema debe permitirle a los usuarios agregar y/o quitar las comidas que el ingiere diariamente.

Prioridad: Baja

RF09: Consultas a profesionales

Descripción: El sistema debe permitir a los usuarios enviar consultas privadas mediante mensajes a sus profesionales.

Prioridad: Media

RF10: Realizar plan de alimentación

Descripción: El sistema debe permitir a los profesionales realizar un plan de alimentación para los clientes que se lo soliciten

Prioridad: Alta

RF11: Agregar o quitar alimentos

Descripción: El sistema debe permitir a los profesionales registrar y/o quitar alimentos que los clientes pueden consumir.

Prioridad:Alta

RF12: Contestar consultas

Descripción: El sistema debe permitir a los profesionales contestar los mensajes de sus clientes.

Prioridad:Alta

RF13: Registrar alimento

Descripción: El sistema debe permitirle al administrador registrar un alimento. Para ello se le pedirá que complete los siguientes campos: Nombre, tipo de alimento y Principales nutrientes.

Prioridad: Alta

RF14: Informar sobre el usuario autenticado

Descripción: Informar sobre el usuario autenticado en la aplicación en cada ventana.

Prioridad: Alta

RF15: Cambiar de usuario

Descripción: El sistema debe permitir al administrador cambiar de usuario cuando desee, esto hace que se vuelva a la pantalla principal.

Prioridad: Alta

RF16: Redactar consulta directa profesional

Descripción: Panel redactar consulta directa profesional desde Profesional al presionar el botón enviar, notificar al usuario que se envió el mensaje. Prioridad: Alta

## **Requerimientos No Funcionales (RNF)**

RNF01: Lenguaje de desarrollo.

Descripción: El sistema debe ser desarrollado en el lenguaje Java en la versión 8.

Prioridad: Media.

RNF02: Idioma.

Descripción: El sistema debe tener como idioma el español.

Prioridad: Alta.

RNF03: Extensión imagenes

Descripción: Los archivos multimedia subidos al sistema deberán ser de alguno de los siguientes formatos: png, jpg, jpeg.

Prioridad: Media.

RNF04: Tamaño imagenes.

Descripción: Los archivos multimedia subidos al sistema del tipo imagen deben ser de hasta 10 MB.

Prioridad: Alta.

RNF05: Serialización

Descripción: El sistema debe poder persistir en memoria los datos de usuarios(usuario(cliente), profesional, administrador), planes alimenticios, comidas y conversaciones.

Prioridad: Alta

## **5. Modelado de usuarios**

### **5.1. Interesados**

Según PMBOK los interesados son aquellos individuos y/o organizaciones que están inmersos activamente en el proyecto, o cuyos intereses pueden ser afectados positivamente o negativamente como resultado de la ejecución del proyecto, o de su culminación exitosa. Pueden tener intereses contrapuestos entre sí.

#### **5.1.1. Registro de interesados**

El registro de interesados recoge toda la información acerca de los individuos y grupos que tienen interés en el trabajo que se realiza.

Los requerimientos indicados en la columna de requerimientos se encuentran especificados detalladamente en la sección 4.

Nombre	Posición	Roles	Expectativas	Influencia(alta/baja)	Clasificación (amigo/enemigo/neutral)	Requerimientos
Grupo (Camila, Joselen y Belén)	Área de desarrollo	Líderes	Lograr los objetivos planteados, mejorar la calidad	Alta	Amigo	Todos los requerimientos especificados
Nutricionistas/Dietistas	Área de profesionales	Profesionales, brindan información	Brindar un plan de alimentación más fácilmente.	Alta	Amigo	RF01-RF02-RF04-RF09-RF10-RF13-RF11-RF12-RF14-RF15-RF17-RF19-RF20
Lugares de comida rápida/Restaurantes/Hoteles	Area de alimentacion	Cliente	Dar una buena alimentación a sus clientes	Media	Neutral	RF01-RF02-RF03-RF04-RF05-RF06-RF07-RF08-RF14-RF18-RF20
Comunidad	Usuarios	Usuarios	Mejorar su alimentación	Alta	Amigo	RF01-RF02-RF03-RF04-RF05-RF06-RF07-RF08-RF14-RF18-RF20
Ministerio de la salud/OMS/A DU/ACELU	Área de profesionales	Sponsor	Fomentar una buena alimentación y brindar el servicio de la aplicación	Alta	Amigo	RF01-RF02-RF04-RF09-RF10-RF13-RF11-RF12-RF14-RF15-RF17-RF19-RF20
Nooddle	Área de competencia	Referente y competencia	Que no funcione	Media	Enemigo	RF01-RF02-RF03-RF04-RF05-RF06-RF07-RF08-RF14-RF18-RF20
Hospitales/Gimnasios	Área de profesionales	Cliente	Brindar un plan de alimentación a cada persona	Media	Neutral	RF01-RF02-RF04-RF09-RF10-RF13-RF11-RF12-RF14-RF15-RF17-RF19-RF20

## 5.2. Descripción de usuarios

### 5.2.1. Perfil de usuarios

Son las características detalladas de las personas para las cuales está realizado el sistema, incluyendo profesión, educación, edad.

En este caso tenemos por un lado profesionales de la salud especialmente los

especializados en nutrición y por el otro lado cualquier persona con la intención de bajar de peso, o mantener un buen estado físico.

### **5.2.2. Contexto de uso**

Son situaciones particulares en las que los usuarios utilizarán el sistema, prestando atención a factores influyentes en su uso y los criterios de éxito de los usuarios principales.

1. Fernando, adulto de 45 años posee obesidad y quiere bajar de peso. Este en vez de sacar cualquier dieta de internet lo cual no es conveniente, desea hacerlo a través de la aplicación para conseguir una dieta escrita por un profesional capacitado, el cual tiene un título que lo respalda. Fernando decide solicitarle un plan de alimentación a un nutricionista de la aplicación el cual le recomendaron.

2. Florencia es una entrenadora física de 22 años, ella sabe mucho sobre alimentación y le gusta ayudar a otros, al no tener una larga experiencia laboral tiene muy pocos clientes. Esta aplicación le da la posibilidad de tener más clientes y le permite brindar su servicio.

3. Nicolás es un joven de 21 años el cual desea tener una dieta saludable y balanceada a pesar de estar bien físicamente. Solicita un plan de alimentación para tener una dieta saludable y mantener bien su estado físico.

## 6. Calidad del código

### 6.1. Estándares de codificación

Un código de calidad es aquel que se rige por determinados estándares de codificación previamente definidos, ya sea por una entidad, un equipo de desarrollo o estándares ya definidos para un lenguaje o plataforma.

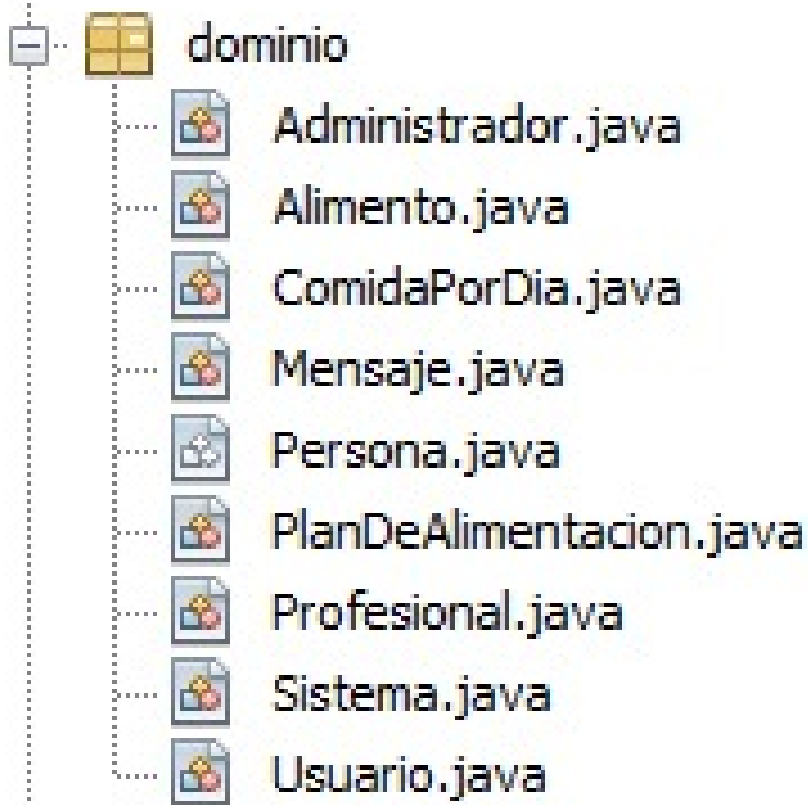
Por ejemplo, Google tiene establecidos distintos estándares de codificación. Estos estándares son la base de la calidad de un código, permitiendo además un fácil trabajo en equipo, ya que todos los integrantes se basan en un mismo estilo de código, lo que simplifica a la hora de entender desarrollados hechos por otro integrante del equipo.

Utilizamos el estándar Java Code Conventions, este consta con:

- Las clases deben comenzar con mayúscula.

Se adjunta imagen como muestra del cumplimiento de este punto.





- Paquetes y métodos comienzan con minúscula.  
Se adjunta imagen como muestra del cumplimiento de este punto.



```

public boolean existeNombreProfesional(String nombre) {
    for (int i = 0; i < profesionales.size(); i++) {
        Profesional profesional = profesionales.get(i);
        if (profesional.getNombreUsuario().equals(nombre)) {
            return true;
        }
    }
    return false;
}

```

- Uso de camelCase para diferenciar palabras.

Camel case es un estilo de escritura que se aplica a frases o palabras compuestas. Existen dos tipos, UpperCamelCase, cuando la primer letra de cada una de las palabras es mayúscula; y lowerCamelCase, es igual que el anterior con la excepción de que la primer letra es minúscula.

Como ejemplo del uso de UpperCamelCase, se tienen los nombres de las clases; aquellos que son compuestos por más de una palabra. Ejemplo: PlanDeAlimentacion, ComidaPorDia.

Por otro lado, como ejemplo del uso de lowerCamelCase, se tienen los nombres de métodos compuestos por más de una palabra. Ejemplo existeNombreProfesional entre otros.

- Llaves estilo Kernigham.

Este estilo plantea que cada función tiene sus llaves de apertura en la misma línea que sus respectivas declaraciones de control; las llaves de cierre permanecen en una línea propia, a menos que le siga una palabra clave else o while. Se adjunta imagen como muestra del cumplimiento de este punto.

```

public boolean existeNombreProfesional(String nombre) {
    for (int i = 0; i < profesionales.size(); i++) {
        Profesional profesional = profesionales.get(i);
        if (profesional.getNombreUsuario().equals(nombre)) {
            return true;
        }
    }
    return false;
}

```

Además de la implementación de los puntos anteriores, se tomaron en cuenta otros elementos y sus buenas prácticas.

Se mantiene una indentación adecuada importante para navegar en estructuras de

controles. Todo código que se incluya dentro de llaves (código interno de otro método o estructura de control) se encuentra un nivel hacia la derecha de su método padre. Facilitando la lectura del código y la visualización de las jerarquías.

Se utilizan líneas en blanco únicamente en los casos en que es necesario, por ejemplo, en la separación de métodos, atributos, importaciones, entre otros. También se evitaron los comentarios innecesarios.

## 6.2. Cobertura de pruebas unitarias

Para analizar la cobertura de las pruebas unitarias se utiliza la herramienta JaCoCo.

A la hora de testear el código nace un factor importante, la cobertura o coverage, el coverage nos ayuda a saber qué porcentaje del código hemos probado satisfactoriamente.





La cobertura de líneas de código consiste en medir en porcentaje la cantidad de líneas de código que fueron probadas en las pruebas unitarias. Mientras que la cobertura de ramas es la medición de los puntos de decisión y lógica condicional mostrando que porcentaje de los resultados de decisión (ramas) se probaron.

Hacer este seguimiento de manera manual es imposible, especialmente cuando hay muchas líneas de código. Es por esto por lo que se creó la herramienta JaCoCo, una herramienta que analiza la cobertura en pruebas que se tienen y genera un reporte en formato HTML.

En el primer obligatorio planteamos como uno de los objetivos mejorar la cobertura de las pruebas unitarias aumentándola a un porcentaje del 95 % en el paquete dominio del proyecto.

Ya que como habíamos visto el porcentaje era de 79 % ,mucho de estos test no pasaban y algunas clases estaban con cobertura 0 % es decir no habían sido probadas

### JaCoCoverage analysis of project "AlimentacionSaludable" (powered by JaCoCo from EcEmma)

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Cla
interfaz		0%		0%	653	653	3.555	3.555	480	480	157	
dominio		79%		68%	91	186	151	424	77	164	2	
alimentacionsaludable		0%		n/a	2	2	10	10	2	2	1	
Total	20.183 of 22.342	10%	360 of 390	8%	746	841	3.716	3.989	559	646	160	

Created with JaCoCo 0.7.6.201602180812

## dominio

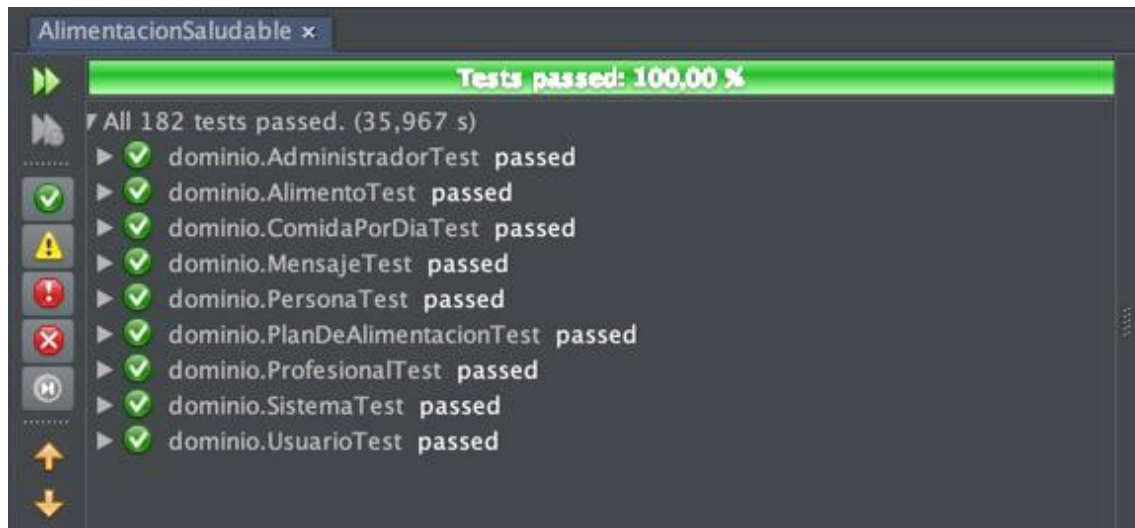
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	C
Usuario		70%		67%	16	41	18	96	12	35	0	
Sistema		63%		80%	11	25	48	108	9	20	0	
Mensaje		0%		n/a	10	10	20	20	10	10	1	
PlanDeAlimentacion		42%		n/a	16	17	24	34	16	17	0	
Administrador		0%		n/a	6	6	11	11	6	6	1	
Alimento		81%		50%	7	16	15	39	6	15	0	
ComidaPorDia		62%		50%	9	13	11	27	4	8	0	
Profesional.Pais		97%		n/a	2	4	0	5	2	4	0	
Alimento.TipoAlimento		94%		n/a	2	4	0	4	2	4	0	
Alimento.Nutrientes		91%		n/a	2	4	0	3	2	4	0	
Persona		89%		100%	3	15	3	32	3	12	0	
Usuario.Nacionalidades		98%		n/a	1	4	0	6	1	4	0	
Usuario.Restricciones		93%		n/a	1	4	0	2	1	4	0	
Usuario.Preferencias		93%		n/a	1	4	0	2	1	4	0	
Sistema.tipoUsuario		90%		n/a	1	4	0	2	1	4	0	
Profesional		99%		50%	3	15	1	33	1	13	0	
Total	567 of 2.726	79%	14 of 44	68%	91	186	151	424	77	164	2	

Ahora con los nuevos test y arreglando aquellos que estaban implementados pero no pasaban , logramos tener una cobertura de 95 , con todos los test funcionando Esto implica que todas las clases menos sistema están por encima del 90 % de cobertura, incluso tenemos algunas con cobertura del 100 % con todos los test funcionando y con sus principales funcionalidades probadas.

## JaCoCoverage analysis of project "AlimentacionSaludable" (powered by JaCoCo from EcEmma)

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Cla
interfaz		0%		0%	656	656	3.674	3.674	476	476	156	
dominio		95%		77%	30	210	35	469	14	175	0	
alimentacionsaludable		0%		n/a	2	2	10	10	2	2	1	
Total	20.457 of 23.214	12%	376 of 430	13%	688	868	3.719	4.153	492	653	157	





JaCoCoverage analysis of project "AlimentacionSaludable" (powered by JaCoCo from EcEmma) > dominio [Source Files](#) [Sessions](#)

## dominio

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed
Sistema	<div><div></div></div>	79%	<div><div></div></div>	81%	7	41	30	137	2	28	0
Profesional.Pais	<div><div></div></div>	97%		n/a	2	4	0	5	2	4	0
Alimento.TipoAlimento	<div><div></div></div>	94%		n/a	2	4	0	4	2	4	0
Alimento.Nutrientes	<div><div></div></div>	91%		n/a	2	4	0	3	2	4	0
Alimento	<div><div></div></div>	96%	<div><div></div></div>	100%	2	16	3	39	2	15	0
Usuario.Nacionalidades	<div><div></div></div>	98%		n/a	1	4	0	6	1	4	0
ComidaPorDia	<div><div></div></div>	94%	<div><div></div></div>	50%	5	13	2	27	0	8	0
Usuario.Restricciones	<div><div></div></div>	93%		n/a	1	4	0	2	1	4	0
Usuario.Preferencias	<div><div></div></div>	93%		n/a	1	4	0	2	1	4	0
Sistema.tipoUsuario	<div><div></div></div>	90%		n/a	1	4	0	2	1	4	0
Usuario	<div><div></div></div>	100%	<div><div></div></div>	67%	4	41	0	96	0	35	0
Profesional	<div><div></div></div>	100%	<div><div></div></div>	50%	2	15	0	33	0	13	0
PlanDeAlimentacion	<div><div></div></div>	100%		n/a	0	18	0	36	0	18	0
Persona	<div><div></div></div>	100%	<div><div></div></div>	100%	0	15	0	32	0	12	0
Administrador	<div><div></div></div>	100%	<div><div></div></div>	100%	0	13	0	25	0	8	0
Mensaje	<div><div></div></div>	100%		n/a	0	10	0	20	0	10	0
Total	160 of 2.917	95%	16 of 70	77%	30	210	35	469	14	175	0

Created with JaCoCo 0.7.6.201602180812

## 6.3. Análisis de código

En esta sección se analiza el estado del código del producto luego del refactor y de los cambios implementados.

Se analiza con el fin de conocer los problemas y aspectos que se mejoraron con los cambios implementados.

Uno de los aspectos que siempre se debe tener en cuenta a la hora de mantener y mejorar un sistema, es la calidad de este, por lo que intentamos apegarnos a un estándar que nos guíe. Para poder garantizar esta calidad y cumplimiento de las reglas del estándar es conveniente utilizar ciertos analizadores de código a la hora del desarrollo para poder evitar y encontrar ciertos errores.

Las herramientas utilizadas fueron SonarQube(para cuantificar bugs y code smells) y CheckStyle (para corregir errores de indentación,nombre de variables, largo de las lineas de código, etc).

## CheckStyle

Se arreglaron los errores de indentación,espacios y nombres de variables, además se agregaron los @override que faltaban.

Esto hizo que los errores marcados por CheckStyle se redujeran al 0% en los paquetes dominio, alimentacionSaludable y test y al 20% en el paquete interfaz.

Adjuntamos pruebas de las mejoras en algunas clases.

```
1  package dominio;
2
3  import java.util.Objects;
4  import java.io.Serializable;
5
6  public class Administrador implements Serializable {
7      private static final long serialVersionUID = 6106269076155338045L;
8      private String usuario;
9      private String contrasenia;
10
11     public Administrador(String miUsuario, String miContrasenia) {
12         usuario = miUsuario;
13         contrasenia = miContrasenia;
14     }
15
16     public Administrador() {
17     }
18
19     public String getUsuario() {
20         return usuario;
21     }
22
23     public void setUsuario(String usuario) {
24         this.usuario = usuario;
25     }
26
27     public String getContrasenia() {
28         return contrasenia;
29     }
30
31     public void setContrasenia(String contrasenia) {
```

```

1 package dominio;
2
3 import java.io.Serializable;
4 import java.util.Arrays;
5 import java.util.Objects;
6 import java.util.Optional;
7
8 public class Alimento implements Serializable {
9
10     private static final long serialVersionUID = 6106269076155338045L;
11     private String nombre;
12     private TipoAlimento tipo;
13     private TipoAlimento[] listaTipoAlimento = inicializoTipoAlimento();
14     private Nutrientes[] listaNutrientes = inicializoNutrientes();
15     private boolean[] listaNutrientesSeleccionados;
16
17
18
19     public Alimento() {
20         this.nombre = "sin nombre";
21         this.tipo = TipoAlimento.Otro;
22         this.listaNutrientesSeleccionados = new boolean[listaNutrientes
23             .length];
24     }
25
26     public String getNombre() {
27         return nombre;
28     }
29
30     public void setNombre(String nombre) {

```

```

1 package dominio;
2
3 import java.io.Serializable;
4 import javax.swing.ImageIcon;
5
6 public abstract class Persona implements Serializable {
7
8
9     private static final long serialVersionUID = 6106269076155338045L;
10     private String nombre;
11     private String apellido;
12     private String aliasUsuario;
13     private String fechaNacimiento;
14     private ImageIcon fotoPerfil;
15
16
17     public Persona(String nombre, String apellidos, String nombreUsuario,
18         String fechaNacimiento, ImageIcon fotoPerfil) {
19         this.nombre = nombre;
20         this.apellido = apellidos;
21         this.aliasUsuario = nombreUsuario;
22         this.fechaNacimiento = fechaNacimiento;
23         this.fotoPerfil = fotoPerfil;
24     }
25
26     public Persona(String nombre, String apellidos, String nombreUsuario,
27         String fechaNacimiento) {
28         this.nombre = nombre;
29         this.apellido = apellidos;
30         this.aliasUsuario = nombreUsuario;

```

```

5
6 public class Mensaje implements Serializable {
7
8     private static final long serialVersionUID = 6106269076155338045L;
9     private Persona origen;
10    private Persona destino;
11    private String asunto;
12    private String mensaje;
13
14
15    public Mensaje() {
16        Persona persona = (Persona) new Usuario();
17        this.origen = persona;
18        this.destino = persona;
19        this.asunto = "No se asigno un asunto";
20        this.mensaje = "No se ha escrito un texto de mensaje";
21    }
22
23
24    public Persona getOrigen() {
25        return origen;
26    }
27
28    public void setOrigen(Persona origen) {
29        this.origen = origen;
30    }
31
32    public Persona getDestino() {
33        return destino;
34    }
35

```

## SonarQube

Luego de realizados los cambios se realiza un análisis con sonarqube para medir el porcentaje de cobertura del código y la cantidad de bugs.

SonarQube es una herramienta que ayuda a la revisión de código, detección de errores y vulnerabilidades, permitiendo al equipo de desarrollo gestionar, hacer seguimiento y mejorar la calidad del código fuente.

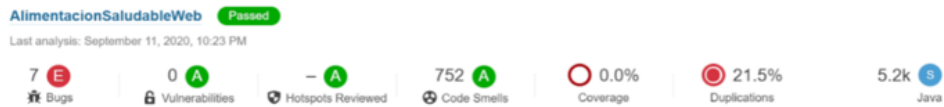
Entre sus funciones están la de dar métricas acerca de código duplicado, pruebas unitarias, cobertura del código, potenciales errores o estándares. Además, nos hace sugerencias sobre qué partes del código son mejorables.

Antes de ver el resultado de los análisis nos habíamos planteado que si la herramienta encuentra menos de un 10 por ciento de código duplicado diremos que es un resultado aceptable, para un porcentaje entre 10 y 17 por ciento se dirá que se puede mejorar y para un porcentaje mayor a 17 diremos que no cumple con una buena calidad.

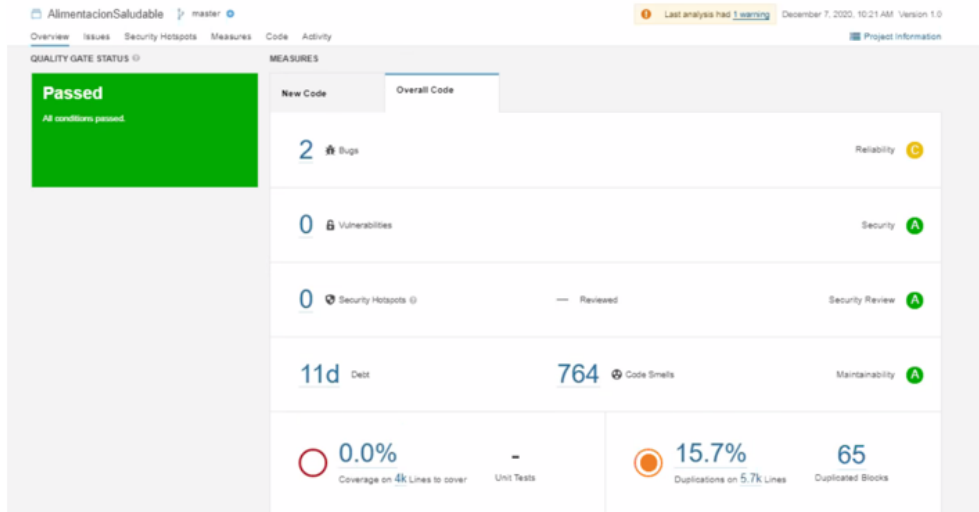
A continuación presentamos los resultados de la aplicación de la herramienta sobre el código corregido:



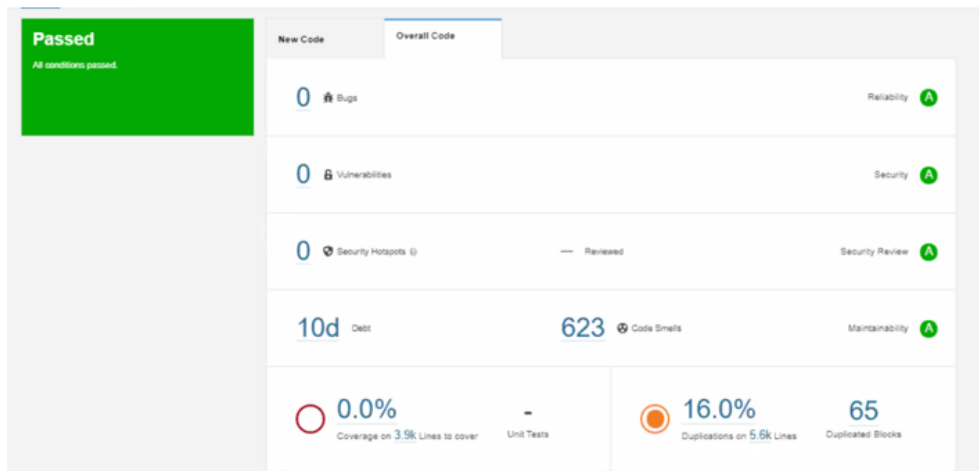
16/10/2020



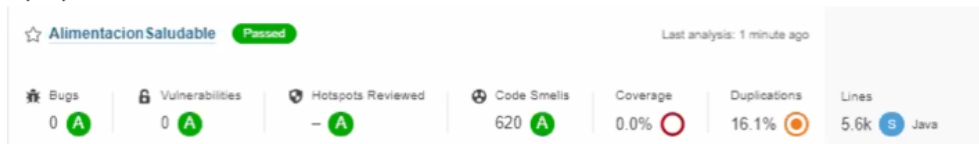
23/11/2020



29/11/2020



6/12/2020



Como se ve en la imagen logramos bajar los bugs que anteriormente eran 7 a 0 bugs, esto se logro corrigiendo los marcados por sonarQube.

Logramos disminuir los codeSmells de 752 a 620.

Los codeSmells que no se corrigen son de código autogenerated en la interfaz que no se puede modificar, y por lo tanto no se pueden corregir, adjuntamos algunos ejemplos:

The first screenshot shows a code snippet with a code smell message: "Defina una constante en lugar de duplicar este literal 'Tahoma' 5 veces." (Define a constant instead of duplicating this literal 'Tahoma' 5 times). The message includes a link to learn more and a "cert, sin usar" (cert, not used) status.

The second screenshot shows a code snippet with a code smell message: "Elimine este parámetro de método no utilizado 'evt'." (Remove this unused method parameter 'evt'). The message includes a link to learn more and a "cert, sin usar" (cert, not used) status.

The third screenshot shows a code snippet with three code smell messages, all related to variable declarations. The first message says: "Elimine el campo 'brnInicioSesion' y declárelo como una variable local en los métodos relevantes." (Remove the field 'brnInicioSesion' and declare it as a local variable in the relevant methods). The second message says: "Elimina el campo 'etiquetaTitulo' y declara como variable local en los métodos relevantes." (Remove the field 'etiquetaTitulo' and declare it as a local variable in the relevant methods). The third message says: "Eliminar el campo 'etiquetaTitulo1' y declararlo como variable local en los métodos relevantes." (Remove the field 'etiquetaTitulo1' and declare it as a local variable in the relevant methods). All three messages include a link to learn more and a "trampa" (trap) status.

Se muestra la clasificación de estos defectos



En los resultados también vemos que el código duplicado bajo a 16,1 % no es el resultado esperado pero todo este código duplicado se encuentra únicamente en el paquete interfaz y es de código autogenerado que no se puede modificar, cuando nos propusimos bajar a menos de 10 % el código duplicado no sabíamos que este se encontraba en código autogenerado.

AlimentacionSaludable - src		Visto como	Lista	para seleccionar archivos		para navegar		27 archivos
							Líneas duplicadas	Líneas duplicadas (%)
src / interfaz / PanelEditarPerfilProfesional.java							214	55,3%
src / interfaz / PanelRegistroProfesional.java							214	49,9%
src / interfaz / PanelEditarPerfilUsuario.java							205	29,2%
src / interfaz / PanelRegistroUsuario.java							205	28,6%
src / interfaz / PanelConsultaProfesional.java							47	22,6%
src / interfaz / PanelConsultaProfesionalDesdeProfesional.java							47	27,2%
src / interfaz / InterfazBotonesProfesional.java							36	21,4%
src / interfaz / InterfazBotonesUsuario.java							36	15,8%
src / interfaz / PanelAgregarComidasADietaDiariaUsuario.java							31	23,8%
src / interfaz / PanelHistorialDeComidasUsuario.java							31	22,8%
src / interfaz / PanelAgregarComidasAPlan.java							27	20,9%
src / interfaz / PanelHomeUsuario.java							20	12,7%
src / interfaz / PanelRealizarPlanAlimentacion.java							dieciséis	6,3%
src / interfaz / PanelRedactarConsulta.java							11	5,8%
src / interfaz / PanelRedactarConsultaDesdeProfesional.java							11	6,6%
Hay 12 componentes ocultos con una puntuación de 0. <a href="#">Mostrarles</a>								

Pudimos observar que la cantidad de líneas de código aumento a 5582 ya que se implementaron nuevas funcionalidades, también nos dimos cuenta con el análisis de sonarQube que la mayoría de estas líneas son también de código autogenerado, por lo que no pudimos reducir la cantidad de líneas en el paquete interfaz quedando 5067.

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
src	5,582	0	0	623	0	0.0%	16.0%
alimentacionSaludable	17	0	0	0	0	0.0%	0.0%
dominio	498	0	0	25	0	0.0%	0.0%
interfaz	5,067	0	0	598	0	0.0%	19.2%

## 7. Evaluación de usabilidad

### Heurísticas de Nilsen:

Uno de los puntos mas importantes era mejorar el estado de la interfaz de la aplicación, para esto modificamos el diseño de esta y agregamos nuevas validaciones y funcionalidades. Adjuntamos muestras de la nueva apariencia de la interfaz.

Al ingresar a la aplicación se muestra esta pantalla donde la persona decide con que tipo de usuario quiere usar la aplicación profesional, administrador o usuario



Cuando seleccionamos uno de estos tipos se pide al usuario que se loguee en el sistema



Como se ve se uno de los principales cambios en la apariencia fue el color ya que el anterior era muy intenso y molestaba a la vista, también se cambio el tipo de letra, y tamaño según que tan importante era este.

Además se eligieron distintas tonalidades a los botones, poniendo en tonos mas oscuros los botones principales

Para analizar la nueva interfaz, decidimos realizar un Análisis Heurístico de Usabilidad de Nielsen.

El análisis consiste en una serie de comprobaciones que velan por la usabilidad y la consecución de los objetivos de negocio de la aplicación, obteniendo conclusiones y propuestas de mejora, de qué se debería implementar.

Algunas de las heurísticas que evaluaremos son:

#### -1. Visibilidad del estatus del sistema:

En la nueva interfaz se agregan mensajes de error/éxito al ingresar datos y presionar botones, para que el usuario pueda mantenerse informado de qué es lo que se está haciendo al usar el programa.

Cada vez que el usuario se mueve a una nueva ventana se muestra resaltado el titulo de esa ventana, indicando que funcionalidades se encuentran en esta.

Además se muestra en cada ventana que usuario se encuentran logueado en ese momento.

## -2.Vínculo entre el sistema y el mundo real:

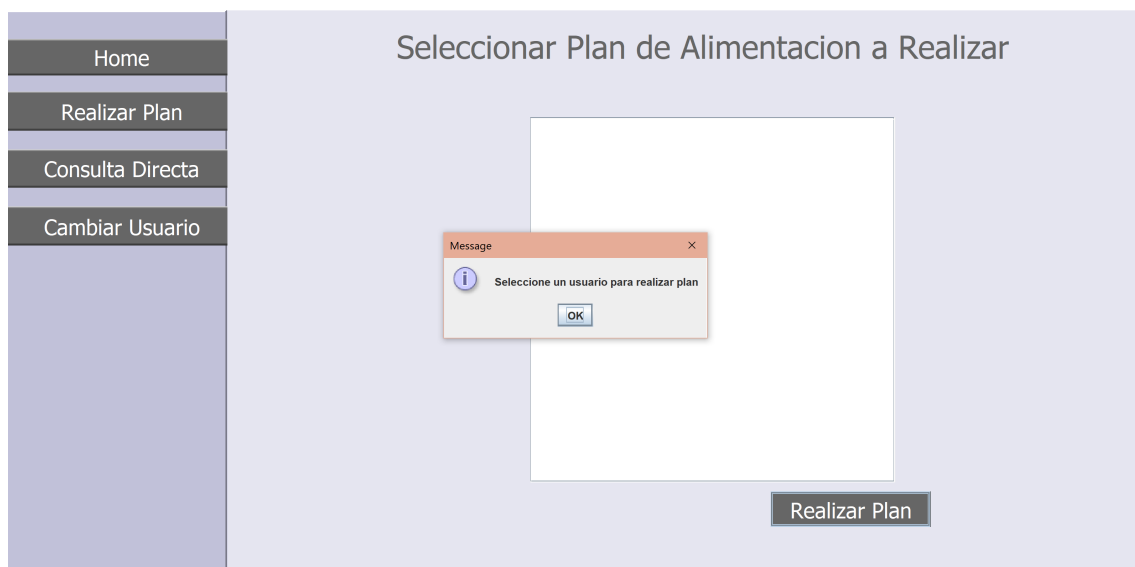
El sistema utiliza palabras del lenguaje español que son fáciles de interpretar para el usuario, no se usa lenguaje complejo, si no que se validan los errores en el código y se traducen estos errores a mensajes en lenguaje entendible. También los títulos de las ventanas y texto de los botones son simples.

## -3.Libertad y control del usuario:

El usuario puede moverse dentro de la página con los distintos botones, tiene la posibilidad de volver atrás y cerrar la aplicación en el momento que lo desee.

## -4.Prevencción de errores:

Se muestran mensajes de error cuando el usuario ingresa datos que no corresponden con los requeridos, o cuando clickea botones para notificar que acción se esta ejecutando .



**Registro Usuario**

Foto de Perfil:  **Cambiar Foto**

Nombre:

Apellidos:

Nombre de Usuario:

Nacionalidad:

Fecha de Nacimiento:

Sexo: ☒ Masculino ☐ Femenino

Restricciones: ☒ Celíaco ☐ Intolerante a la lactosa ☐ Diabético ☐ Hipertensión

Preferencias: ☐ Vegano ☐ Vegetariano ☐ Macrobiótico ☐ Orgánico

**Aceptar**

Message: Usuario registrado correctamente

**Historial de Comidas**

Home

Perfil

Consulta Directa

Pedir Plan

Agregar Comida a Di

Cambiar Usuario

**Ver Historial**

Message: Seleccione una comida

choche choche

##### -5.Diseño minimalista:

El sistema sólo muestra lo necesario para que el usuario pueda utilizar las funcionalidades en forma eficiente y no le genere dudas al usar el sistema. Con el cambio en la estética de la interfaz se intentó que esta fuera mas sobria y sencilla ya que anteriormente con todos los colores y dibujos abrumaba y confundía al usuario que la estaba usando

##### -6.Ayudar a los usuarios a reconocer, diagnosticar, y recuperarse de los errores:

Los errores mostrados por el programa están escritos en lenguaje común, para que el usuario pueda identificarlo y solucionarlo fácilmente, además estos errores se lanzan inmediatamente suceden para facilitar la recuperación del usuario, además se notifican estos con diálogos para llamar la atención del usuario y que ningún error pase desapercibido.



#### -7.Consistencia y estándares

El usuario no debe preguntarse qué hace un botón, se trata de seguir con un estándar, es decir si un botón dice enviar es porque envía, al igual que en todas las aplicaciones, o paginas web, no se reescriben estos estándares, sino que se siguen cumpliendo.

#### -8.Reconocer en lugar de recordar

En todas las ventanas se intenta ayudar al usuario auto completando campos, o dando sugerencias con mensajes de como realizar determinada acción.

## 8. Pruebas funcionales

### 8.1. Técnicas de pruebas aplicadas

Para probar el sistema utilizamos además de las pruebas unitarias:

Por un lado aplicamos casos de prueba, las cuales son un tipo de prueba donde se desconoce los métodos y código interno de la aplicación y solo se evalúa el funcionamiento de la misma. Únicamente se tienen en cuenta las entradas y salidas para verificar el correcto funcionamiento del sistema.

Por otro lado se realizaron también casos de uso. Los casos de uso son una descripción de las actividades que deberá realizar alguien para poder llevar a cabo un proceso.

#### 8.1.1. Casos de uso

Un caso de uso describe la secuencia de interacciones entre el sistema y uno o varios actores, obteniendo como resultado el logro de un resultado de valor para los involucrados. Es importante destacar que los casos de uso no describen ninguna funcionalidad interna del sistema, ni explican cómo se implementará, sino que muestran lo que el o los actores hacen o deben hacer para lograr cumplir una funcionalidad. Para este caso, decidimos aplicar estas herramientas para las que, a nuestro criterio, son las funcionalidades más importantes del sistema en el proceso de completar el objetivo principal. Consideramos que estas funcionalidades describen la sucesión de pasos que deben cumplirse para lograr el objetivo de la aplicación.

**Estas funcionalidades son:**

- Registrar Administrador. RF01
- Registrar Profesional. RF02
- Registrar Usuario. RF02
- Solicitar plan de alimentación. RF06
- Consulta con profesional. RF09
- Realizar plan de alimentación. RF10
- Registrar alimento. RF13

#### 7.1 Caso de uso 1: Registrar Administrador

<b>Nombre</b>	<b>Registrar Administrador</b>
<b>ID</b>	CU_001
<b>Descripción</b>	En el menú principal del sistema está la opción de registrar un administrador
<b>Actores</b>	Administrador
<b>Beneficios</b>	Que se maneje correctamente el sistema y se puedan ingresar alimentos, profesionales y usuarios al mismo
<b>Disparador</b>	El administrador hace clic en el icono de administrador
<b>Precondiciones</b>	
<b>Postcondiciones</b>	El administrador ya puede registrar profesionales, usuarios y alimentos
<b>Curso Normal</b>	1.Ingresa un nombre de usuario 2.Ingresa una contraseña 3.Hace clic en el botón registrar
<b>Curso Alternativo</b>	<p>3.1 Entrada: No ingresa un nombre de usuario Respuesta del sistema: Mensaje de error "El usuario no puede ser vacío"</p> <p>3.2 Entrada: No ingresa una contraseña Respuesta del sistema: Mensaje de error "La contraseña no puede ser vacía"</p> <p>3.3 Entrada: Ingresa nombre y contraseña de un administrador ya registrado Respuesta del sistema: Mensaje de error "El administrador ya está registrado en el sistema, presiona iniciar sesión"</p> <p>3.4 Entrada: Ingresa una contraseña no válida Respuesta del sistema: Mensaje de error "La contraseña debe tener más de 8 caracteres y ser alfanumérica"</p>

## **7.2 Caso de uso 2: Registrar Profesional**

En el caso del registro del profesional no se valida el país en el que este se graduó ya que al momento del ingreso hay uno predeterminado, este es Uruguay. Con la foto pasa algo similar ya que en caso de que el usuario no cambie la foto le asigna la foto predeterminada

<b>Nombre</b>	<b>Registrar Profesional</b>
<b>ID</b>	CU_002
<b>Descripción</b>	Cuando un profesional quiere ser parte de la aplicación, un administrador lo registra completando todos los campos que se requieren para hacer el registro del mismo
<b>Actores</b>	Administrador
<b>Beneficios</b>	Que haya más profesionales en el sistema para ayudar a las personas a tener una alimentación saludable
<b>Disparador</b>	El administrador hace clic en el botón "Registrar Profesional"
<b>Precondiciones</b>	El administrador debe estar registrado en el sistema
<b>Postcondiciones</b>	El profesional ya puede usar el sistema
<b>Curso Normal</b>	1.Ingresa un nombre 2.Ingresa un apellido 3.Ingresa un nombre de usuario 4.Selecciona una fecha de nacimiento 5.Ingresa nombre del título profesional 6.Selecciona una fecha de graduación 7.Selecciona país de graduación 8.El sistema al presionar aceptar muestra un mensaje "Profesional registrado correctamente" y además se muestran todos los mensajes de error correspondientes
<b>Curso Alternativo</b>	1.1 Entrada: No ingresa un nombre Respuesta del sistema: Mensaje de error "El nombre no puede ser vacío"  2.1 Entrada: No ingresa un apellido Respuesta del sistema: Mensaje de error "El apellido no puede ser vacío"  3.1 Entrada: No ingresa un nombre de usuario Respuesta del sistema: Mensaje de error "El alias del usuario no puede estar vacío"  5.1 Entrada: No ingresa un título profesional Respuesta del sistema: Mensaje de error "El nombre del título no puede estar vacío"  8.1 No se ingresaron correctamente los campos. Mensaje de error "Error al registrar al profesional" y además se muestran todos los mensajes de error correspondientes

### 7.3 Caso de uso 3: Registrar Usuario

Nombre	Registrar Usuario
ID	CU_003
Descripción	Cuando un usuario quiere ser parte del sistema, un administrador lo registra completando todos los campos que se requieren para hacer el registro del mismo
Actores	Administrador
Beneficios	Tener más usuarios en el sistema
Disparador	El administrador hace clic en el botón "Registrar Usuario"
Precondiciones	El administrador debe estar registrado en el sistema
Postcondiciones	El usuario ya puede usar el sistema
Curso Normal	<ol style="list-style-type: none"><li>1. Ingresa un nombre</li><li>2. Ingresa un apellido</li><li>3. Ingresa un nombre de usuario</li><li>4. Selecciona una nacionalidad</li><li>5. Selecciona una fecha de nacimiento</li><li>6. Ingresa una altura</li><li>7. Ingresa un peso</li><li>8. Selecciona su sexo</li><li>9. Selecciona restricciones</li><li>10. Selecciona preferencias</li><li>11. El sistema al presionar aceptar muestra un mensaje "Usuario registrado correctamente"</li></ol>

<p><b>Curso Alternativo</b></p>	<p>1.1 Entrada: No ingresa un nombre Respuesta del sistema: Mensaje de error "El nombre no puede ser vacío"</p> <p>2.1 Entrada: No ingresa un apellido Respuesta del sistema: Mensaje de error "El apellido no puede ser vacío"</p> <p>3.1 Entrada: No ingresa un nombre de usuario Respuesta del sistema: Mensaje de error "El nombre de usuario no puede estar vacío"</p> <p>6.1 Entrada: No ingresa una altura Respuesta del sistema: Mensaje de error "La altura no puede estar vacía"</p> <p>6.2 Entrada: Se ingresa una altura invalida Respuesta del sistema: Mensaje de error "Debe ingresar un valor numérico"</p> <p>6.3 Entrada: Se ingresa una altura muy grande Respuesta del sistema: Mensaje de error "El dato debe estar entre:0 y 265"</p> <p>7.1 Entrada: No ingresa un peso Respuesta del sistema: Mensaje de error "El peso no puede estar vacío"</p>
	<p>7.2 Entrada: Se ingresa un peso invalido Respuesta del sistema: Mensaje de error "Debe ingresar un valor numérico"</p> <p>7.3 Entrada: Se ingresa un peso muy grande Respuesta del sistema: Mensaje de error "El dato debe estar entre:0 y 265"</p> <p>11.1 No se ingresaron correctamente los campos. Mensaje de error "Error al registrar el usuario" y además se muestran todos los mensajes de error correspondientes</p>

En el caso de registro del usuario no se valida la nacionalidad ni el sexo ya que al momento del ingreso hay uno predeterminado. Tampoco se validan las restricciones ni las preferencias alimenticias ya que se asume que si no se selecciona ninguna el usuario no tiene preferencias ni restricciones. Con la foto pasa algo similar ya que en caso de que el usuario no cambie la foto le asigna la foto predeterminada.

#### 7.4 Caso de uso 4: Solicitar Plan de Alimentación

<b>Nombre</b>	<b>Solicitar Plan de Alimentación</b>
<b>ID</b>	CU_004
<b>Descripción</b>	Dentro de las opciones del menú de usuarios se encuentra Plan de alimentación donde el usuario puede solicitar un plan a un profesional.
<b>Actores</b>	Usuario, Profesional( actor secundario)
<b>Beneficios</b>	Poder seguir la alimentación de acuerdo a un plan de alimentación elegido por un profesional
<b>Disparador</b>	El usuario hace clic en el icono de administrador
<b>Precondiciones</b>	Debe haber usuarios y profesionales ya registrados en el sistema
<b>Postcondiciones</b>	Tras realizar la solicitud esta es enviada al profesional correspondiente
<b>Curso Normal</b>	1. Selecciona un profesional 2. Hace clic en el botón aceptar, se muestra "se ha pedido el plan de alimentación"

1

#### 7.5 Caso de uso 5: Consulta con profesional



<b>Nombre</b>	<b>Consulta con profesional</b>
<b>ID</b>	CU_005
<b>Descripción</b>	En el menú de usuarios, dentro de estas opciones esta Consulta con Profesional. La misma deriva en una lista de profesionales con los que se ha entablado conversaciones, donde el usuario selecciona a quien le quiere realizar la consulta, la redacta y la envía
<b>Actores</b>	Usuario, Profesional(actor secundario)
<b>Beneficios</b>	Poder mantener una conversación con el profesional que elegiste
<b>Disparador</b>	El usuario hace clic en consulta directa
<b>Precondiciones</b>	Debe haber usuarios y profesionales registrados.
<b>Postcondiciones</b>	Luego de enviado el mensaje él mismo le llega al profesional
<b>Curso Normal</b>	<ol style="list-style-type: none"> <li>1. Se presiona en redactar</li> <li>2. Se selecciona el destinatario</li> <li>3. Ingresa un asunto</li> <li>4. Ingresa un mensaje</li> <li>5. Presiona en enviar, mensaje "Se ha enviado la consulta"</li> </ol>
<b>Curso Alternativo</b>	1.1 En caso de ya haber mensajes en la bandeja de entrada puede seleccionar uno, vuelve a 3

## 7.6 Caso de uso 6: Realizar Plan de Alimentación

<b>Nombre</b>	<b>Realizar Plan de Alimentación</b>
<b>ID</b>	CU_006
<b>Descripción</b>	Como profesional dentro de las opciones del menú se encuentra Realizar un Plan de Alimentación a un usuario que lo haya solicitado. Ingresando qué alimentos se deben ingerir
<b>Actores</b>	Profesional, Usuario( actor secundario)
<b>Beneficios</b>	Poder ayudar a un usuario a tener una mejor dieta
<b>Disparador</b>	El profesional hace clic en el botón Realizar Plan
<b>Precondiciones</b>	Debe haber profesionales y usuarios registrados; al menos un usuario debe haberle solicitado un plan al profesional y un elemento registrado
<b>Postcondiciones</b>	Luego de realizado el plan es enviado al usuario correspondiente
<b>Curso Normal</b>	<ol style="list-style-type: none"> <li>1. Selecciona un usuario de una lista</li> <li>2. Clic en realizar plan</li> <li>3. Selecciona el día de la semana</li> <li>4. Clic en agregar</li> <li>5. Selecciona elemento deseado</li> <li>6. Clic en agregar, mensaje "Se agregó correctamente la comida"</li> <li>7. Si hay comidas agregadas puede seleccionar una</li> <li>8. Clic en quitar comida</li> <li>9. Clic en enviar plan, mensaje "Se ha enviado correctamente el plan"</li> </ol>
<b>Curso Alternativo</b>	<p>1.1 Entrada: No selecciona ningún usuario</p> <p>Respuesta del Sistema: Mensaje de error "Seleccione un usuario para realizar el plan"</p>

### 7.7 Caso de uso 7: Registrar Alimento

<b>Nombre</b>	<b>Registrar Alimento</b>
<b>ID</b>	CU_007
<b>Descripción</b>	El administrador en el menú de administrador tiene la opción de registrar un alimento
<b>Actores</b>	Administrador
<b>Beneficios</b>	Para el manejo de el sistema necesitamos tener alimentos registrados
<b>Disparador</b>	El administrador hace clic en Registrar Alimento
<b>Precondiciones</b>	El administrador debe estar registrado en el sistema
<b>Postcondiciones</b>	Tras registrar el alimento el mismo puede utilizado para la elaboración de planes o ingerido por usuarios
<b>Curso Normal</b>	1. Ingresa el nombre 2. Selecciona el tipo de alimento 3. Selecciona los principales nutrientes 4. Da clic en enviar, mensaje "Alimento registrado correctamente"
<b>Curso Alternativo</b>	1.1 Entrada: No se ingresa un nombre Respuesta del Sistema: Mensaje de error "El nombre no puede ser vacío"  4.1 No se ingresaron correctamente los campos. Mensaje de error "Error al ingresar el alimento" y además se muestran todos los mensajes de error correspondientes

En el caso del registro de alimentos no se valida el tipo de alimento ya que hay un alimento predeterminado. Tampoco se valida que se seleccione uno de los principales nutrientes, este es un fallo reportado en reportes de defectos.

## 8.2. Casos de prueba

### Registro Usuario desde administrador:

Se asume que hay un administrador registrado en el sistema y que el mismo tiene su sesión abierta

CP1.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Camila	Se acepta correctamente	Se acepta correctamente
Apellido: Burgueño	Se acepta correctamente	Se acepta correctamente
Nombre de usuario: camibur	Se acepta correctamente	Se acepta correctamente
Nacionalidad: Uruguay	Se acepta correctamente	Se acepta correctamente
Fecha de nacimiento: 28/09/1999	Se acepta correctamente	Se acepta correctamente
Altura: 163 cm	Se acepta correctamente	Se acepta correctamente
Peso: 60 kg	Se acepta correctamente	Se acepta correctamente
Sexo: Femenino	Se acepta correctamente	Se acepta correctamente
Restricciones: Intolerante a la lactosa	Se acepta correctamente	Se acepta correctamente
Preferencias: Vegetariano	Se acepta correctamente	Se acepta correctamente
Se agrega una foto	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el usuario se registro correctamente	Se muestra un mensaje diciendo que el usuario se registro correctamente

CP2.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Belen	Se acepta correctamente	Se acepta correctamente
Apellido: (vacío)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Nombre de usuario: belu	Se acepta correctamente	Se acepta correctamente
Nacionalidad: Chilena	Se acepta correctamente	Se acepta correctamente
Fecha de nacimiento: 08/12/1999	Se acepta correctamente	Se acepta correctamente
Altura: 160 cm	Se acepta correctamente	Se acepta correctamente
Peso: 55 kg	Se acepta correctamente	Se acepta correctamente
Sexo: Femenino	Se acepta correctamente	Se acepta correctamente
Restricciones:(vacío)	Se acepta correctamente	Se acepta correctamente
Preferencias: (vacío)	Se acepta correctamente	Se acepta correctamente
No se agrega una foto	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que hubo un error al registrar el usuario	Se muestra un mensaje diciendo que hubo un error al registrar el usuario
Se ingresa el apellido: Martinez	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el usuario se registro correctamente	Se muestra un mensaje diciendo que el usuario se registro correctamente

CP3.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Pepe	Se acepta correctamente	Se acepta correctamente
Apellido: Perez	Se acepta correctamente	Se acepta correctamente
Nombre de usuario: camibur	Se avisa que el nombre de usuario es no válido	Se avisa que el nombre de usuario es no válido
Nacionalidad: Rusa	Se acepta correctamente	Se acepta correctamente
Fecha de nacimiento: (No se selecciona ninguna)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Altura: 257 cm	Se espera que al registrar se avise que esta fuera del rango	Se espera que al registrar se avise que esta fuera del rango
Peso: 60 kg	Se acepta correctamente	Se acepta correctamente
Sexo: Masculino	Se acepta correctamente	Se acepta correctamente
Restricciones: Celíaco, Hipertenso	Se acepta correctamente	Se acepta correctamente
Preferencias: (vacío)	Se acepta correctamente	Se acepta correctamente
Se agrega una foto	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que hay un error al registrar el usuario	Se muestra un mensaje diciendo que hay un error al registrar el usuario
Se cambia la altura a una dentro del rango: 190 cm	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que hay un error al registrar el usuario	Se muestra un mensaje diciendo que hay un error al registrar el usuario
Se cambia el nombre de usuario a uno que no haya estado registrado anteriormente: pepi123	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el usuario se registro correctamente	Se muestra un mensaje diciendo que el usuario se registro correctamente

**Registro Profesional desde administrador:**

Se asume que hay un administrador registrado en el sistema y que el mismo tiene su sesión abierta

CP4.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Joselen	Se acepta correctamente	Se acepta correctamente
Apellido: Cecilia	Se acepta correctamente	Se acepta correctamente
Nombre de usuario: choche	Se acepta correctamente	Se acepta correctamente
Fecha de nacimiento: 04/02/2000	Se acepta correctamente	Se acepta correctamente
Título de profesional: Nutricionista	Se acepta correctamente	Se acepta correctamente
Fecha de graduación: 13/09/2018	Se acepta correctamente	Se acepta correctamente
Nacionalidad: Sudafricana	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el profesional se registró correctamente	Se muestra un mensaje diciendo que el profesional se registró correctamente

CP5.



Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Federico	Se acepta correctamente	Se acepta correctamente
Apellido: Novelli	Se acepta correctamente	Se acepta correctamente
Nombre de usuario: (vacío)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Se agrega nombre de usuario: choche	Se avisa que el alias de usuario no puede ser repetido	Se avisa que el alias de usuario no puede ser repetido
Fecha de nacimiento: (vacía)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Título de profesional: Doctor	Se acepta correctamente	Se acepta correctamente
Fecha de graduación: (vacía)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Nacionalidad: Perú	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el profesional no se pudo registrar correctamente	Se muestra un mensaje diciendo que el profesional no se pudo registrar correctamente
Se agrega fecha de nacimiento: 28/11/1994	Se acepta correctamente	Se acepta correctamente
Se agrega fecha de graduación: 15/12/2017	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el profesional se pudo registrar correctamente	Se muestra un mensaje diciendo que el profesional se pudo registrar correctamente

**Registro de alimento desde administrador:**

Se asume que hay un administrador registrado en el sistema y que el mismo tiene su sesión abierta

CP6.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Yogurt light	Se acepta correctamente	Se acepta correctamente
Tipo de alimento: Lácteos	Se acepta correctamente	Se acepta correctamente
Principales nutrientes: Vitaminas	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el alimento se registró correctamente	Se muestra un mensaje diciendo que el alimento se registró correctamente

CP7.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: (vacío)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Tipo de alimento: Carne Roja	Se acepta correctamente	Se acepta correctamente
Principales nutrientes: Fibra, Proteínas	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el alimento no se registró correctamente	Se muestra un mensaje diciendo que el alimento no se registró correctamente
Se ingresa nombre del alimento: Albóndigas	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el alimento se registró correctamente	Se muestra un mensaje diciendo que el alimento se registró correctamente

#### **Agregar comida a dieta desde usuario**

Se asume un usuario registrado y con la sesión iniciada en el sistema

CP8.

Entrada	Resultado esperado	Resultado obtenido
Se oprime el botón agregar comida	Se despliega la lista con los alimentos registrados en el sistema	Se despliega la lista con los alimentos registrados en el sistema
Se selecciona uno de los alimentos registrados (Yogurt light en este caso) y se oprime el botón agregar comida	Se muestra un mensaje de que el alimento fue agregado correctamente al plan	Se muestra un mensaje de que el alimento fue agregado correctamente al plan
Se selecciona la opción de quitar alimento de la dieta marcando el alimento agregado	Se quita correctamente de la lista de alimentos del plan	Se quita correctamente de la lista de alimentos del plan

CP9.

Entrada	Resultado esperado	Resultado obtenido
Se oprime el botón agregar comida	Se despliega la lista con los alimentos registrados en el sistema	Se despliega la lista con los alimentos registrados en el sistema
No se seleccione ningún alimento de la lista	Se espera que se avise que se tiene que seleccionar un alimento para proseguir	Se avisa que se agregó correctamente el alimento

CP10.

### **Consulta dieta desde usuario:**

Se asume que hay un usuario registrado y con la sesión iniciada en el sistema

CP11.

Entrada	Resultado esperado	Resultado obtenido
Se oprime el botón borrar sin seleccionar ninguna consulta	Se avisa que se debe seleccionar una consulta	Se avisa que se debe seleccionar una consulta
Se oprime el botón redactar	Se despliega una ventana para redactar el mensaje	Se despliega una ventana para redactar el mensaje
Se elige como destinatario: Choche	Se acepta correctamente	Se acepta correctamente
Asunto: Pedir dieta	Se acepta correctamente	Se acepta correctamente
Mensaje: "Solicito dieta"	Se acepta correctamente	Se acepta correctamente
Se oprime el botón enviar	Se avisa que el mensaje fue enviado correctamente	Se avisa que el mensaje fue enviado correctamente

CP12.

Entrada	Resultado esperado	Resultado obtenido
Se oprime el botón redactar	Se despliega una ventana para redactar el mensaje	Se despliega una ventana para redactar el mensaje
Se elige como destinatario: Fede	Se acepta correctamente	Se acepta correctamente
Asunto: (vacío)	Se avisa que el asunto está vacío	Se avisa que el asunto está vacío
Mensaje: (vacío)	Se avisa que el mensaje está vacío	Se avisa que el mensaje está vacío
Se oprime el botón enviar	Se avisa que el mensaje no puede ser enviado correctamente	Se avisa que el mensaje no puede ser enviado correctamente
Se agrega Asunto: Consulta	Se acepta correctamente	Se acepta correctamente
Se agrega Mensaje: "Puedo consumir albóndigas el domingo?"	Se acepta correctamente	Se acepta correctamente
Se oprime el botón enviar	Se avisa que el mensaje fue enviado correctamente	Se avisa que el mensaje fue enviado correctamente

#### **Pedir plan desde usuario:**

Se asume que hay un usuario registrado y con la sesión iniciada en el sistema

CP13.

Entrada	Resultado esperado	Resultado obtenido
Se selecciona el profesional al que se le pedirá el plan de alimentación: choche	Se acepta correctamente	Se acepta correctamente
Se oprime el botón aceptar	Se avisa que se ha pedido el plan de alimentación correctamente	Se avisa que se ha pedido el plan de alimentación correctamente

#### **Realizar plan desde profesional:**

Se asume un profesional registrado y con la sesión iniciada en el sistema

CP14.

Entrada	Resultado esperado	Resultado obtenido
Se oprime el botón realizar plan sin seleccionar a ningún usuario que lo haya solicitado	Se pide que se seleccione un usuario para realizar el plan	Se pide que se seleccione un usuario para realizar el plan
Se selecciona uno de los usuarios que hayan solicitado el plan (en este caso camibur) y se oprime el botón realizar plan	Se despliega la venta para poder realizar el plan del usuario	Se despliega la venta para poder realizar el plan del usuario
Se elige el día de la semana: martes	Se acepta correctamente	Se acepta correctamente
Se oprime el botón agregar comida	Se despliega una ventana con los alimentos registrados	Se despliega una ventana con los alimentos registrados
Se agrega el alimento: Albóndigas	Se acepta correctamente	Se acepta correctamente
Se oprime el botón agregar comida	Se avisa que se agregó correctamente la comida a la dieta	Se avisa que se agregó correctamente la comida a la dieta

CP15.

Entrada	Resultado esperado	Resultado obtenido
Se selecciona uno de los usuarios que hayan solicitado el plan (en este caso camibur) y se oprime el botón realizar plan	Se despliega la venta para poder realizar el plan del usuario	Se despliega la venta para poder realizar el plan del usuario
Se elige el día de la semana: jueves	Se acepta correctamente	Se acepta correctamente
Se oprime el botón agregar comida	Se despliega una ventana con los alimentos registrados	Se despliega una ventana con los alimentos registrados
Se agrega el alimento: Jugo de pomelo	Se acepta correctamente	Se acepta correctamente
Se oprime el botón agregar comida	Se avisa que se agregó correctamente la comida a la dieta	Se avisa que se agregó correctamente la comida a la dieta
Se marca la comida Jugo de pomelo y se selecciona el botón quitar comida	Se avisa que la comida se sacó correctamente de la dieta	Se avisa que la comida se sacó correctamente de la dieta

#### **Consulta directa profesional:**

Se asume un profesional registrado y con la sesión iniciada en el sistema

CP16.

Entrada	Resultado esperado	Resultado obtenido
Se selecciona el mensaje que se desea leer	Se muestra el mensaje	Se muestra el mensaje
Se desea enviar un mensaje de respuesta, se oprime el botón redactar	Se muestra una ventana para poder redactar el mensaje con el usuario por defecto al que se le va a contestar	Se muestra una ventana para poder redactar el mensaje con el usuario por defecto al que se le va a contestar
Asunto: Dieta solicitada	Se acepta correctamente	Se acepta correctamente
Mensaje: "Se adjunta dieta solicitada"	Se acepta correctamente	Se acepta correctamente
Se oprime el botón enviar	Se avisa que se envió correctamente el mensaje	Se avisa que se envió correctamente el mensaje

CP17

Entrada	Resultado esperado	Resultado obtenido
Se selecciona el mensaje que se desea borrar	Se borra el mensaje	Se borra el mensaje
Se selecciona el mensaje que se desea leer	Se muestra el mensaje	Se muestra el mensaje
Se desea enviar un mensaje de respuesta, se oprime el botón redactar	Se muestra una ventana para poder redactar el mensaje con el usuario por defecto al que se le va a contestar	Se muestra una ventana para poder redactar el mensaje con el usuario por defecto al que se le va a contestar
Asunto: (vacío)	Se avisa que no se puede dejar el campo vacío	Se avisa que no se puede dejar el campo vacío
Mensaje: (vacío)	Se avisa que no se puede dejar el campo vacío	Se avisa que no se puede dejar el campo vacío
Se oprime el botón enviar	Se avisa que no se envió correctamente el mensaje	Se avisa que no se envió correctamente el mensaje

#### Registro Administrador:

CP18.

Entrada	Resultado esperado	Resultado Obtenido
Usuario: lola	Se acepta correctamente	Se acepta correctamente
Contraseña: 12345678	Se acepta correctamente	Se acepta correctamente
Se elige el botón registrarme	Se muestra un mensaje donde se muestra que fue registrado exitosamente	Se muestra un mensaje donde se muestra que fue registrado exitosamente

CP19.



Entrada	Resultado esperado	Resultado Obtenido
Usuario: pedro	Se acepta correctamente	Se acepta correctamente
Contraseña: 123	Se pide que la contraseña tenga por lo menos 8 caracteres	Se pide que la contraseña tenga por lo menos 8 caracteres
Contraseña:123hola123	Se acepta correctamente	Se acepta correctamente
Se elige el botón registrarme	Se muestra un mensaje donde se muestra que fue registrado exitosamente	Se muestra un mensaje donde se muestra que fue registrado exitosamente

#### **Inicio sesión de administrador:**

Se asume por lo menos un administrador registrado en el sistema

CP20.

Entrada	Resultado esperado	Resultado Obtenido
Usuario: pedro	Se acepta correctamente	Se acepta correctamente
Contraseña:123hola123	Se acepta correctamente	Se acepta correctamente
Se elige el botón iniciar sesión	Se inicia sesión correctamente	Se inicia sesión correctamente

#### **Inicio sesión usuario:**

Se asume al menos un usuario registrado en el sistema

CP21.

Entrada	Resultado esperado	Resultado Obtenido
Nombre: maria y se oprime el botón iniciar sesión	Se avisa que el usuario no está registrado en el sistema	Se avisa que el usuario no está registrado en el sistema

CP22

Entrada	Resultado esperado	Resultado Obtenido
Nombre: camibur y se oprime el botón iniciar sesión	Se inicia sesión correctamente	Se inicia sesión correctamente

#### **Inicio sesión profesional:**

Se asume al menos un profesional registrado en el sistema

CP23.

Entrada	Resultado esperado	Resultado Obtenido
Nombre: maria y se oprime el botón iniciar sesión	Se avisa que el profesional no está registrado en el sistema	Se avisa que el profesional no está registrado en el sistema

CP24.

Entrada	Resultado esperado	Resultado Obtenido
Nombre: chohce y se oprime el botón iniciar sesión	Se inicia sesión correctamente	Se inicia sesión correctamente

### 8.2.1. Casos de prueba

#### **Registro Usuario desde administrador:**

Se asume que hay un administrador registrado en el sistema y que el mismo tiene su sesión abierta

CP1.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Camila	Se acepta correctamente	Se acepta correctamente
Apellido: Burgueño	Se acepta correctamente	Se acepta correctamente
Nombre de usuario: camibur	Se acepta correctamente	Se acepta correctamente
Nacionalidad: Uruguay	Se acepta correctamente	Se acepta correctamente
Fecha de nacimiento: 28/09/1999	Se acepta correctamente	Se acepta correctamente
Altura: 163 cm	Se acepta correctamente	Se acepta correctamente
Peso: 60 kg	Se acepta correctamente	Se acepta correctamente
Sexo: Femenino	Se acepta correctamente	Se acepta correctamente
Restricciones: Intolerante a la lactosa	Se acepta correctamente	Se acepta correctamente
Preferencias: Vegetariano	Se acepta correctamente	Se acepta correctamente
Se agrega una foto	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el usuario se registro correctamente	Se muestra un mensaje diciendo que el usuario se registro correctamente

CP2.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Belen	Se acepta correctamente	Se acepta correctamente
Apellido: (vacío)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Nombre de usuario: belu	Se acepta correctamente	Se acepta correctamente
Nacionalidad: Chilena	Se acepta correctamente	Se acepta correctamente
Fecha de nacimiento: 08/12/1999	Se acepta correctamente	Se acepta correctamente
Altura: 160 cm	Se acepta correctamente	Se acepta correctamente
Peso: 55 kg	Se acepta correctamente	Se acepta correctamente
Sexo: Femenino	Se acepta correctamente	Se acepta correctamente
Restricciones:(vacío)	Se acepta correctamente	Se acepta correctamente
Preferencias: (vacío)	Se acepta correctamente	Se acepta correctamente
No se agrega una foto	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que hubo un error al registrar el usuario	Se muestra un mensaje diciendo que hubo un error al registrar el usuario
Se ingresa el apellido: Martinez	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el usuario se registro correctamente	Se muestra un mensaje diciendo que el usuario se registro correctamente

CP3.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Pepe	Se acepta correctamente	Se acepta correctamente
Apellido: Perez	Se acepta correctamente	Se acepta correctamente
Nombre de usuario: camibur	Se avisa que el nombre de usuario es no válido	Se avisa que el nombre de usuario es no válido
Nacionalidad: Rusa	Se acepta correctamente	Se acepta correctamente
Fecha de nacimiento: (No se selecciona ninguna)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Altura: 257 cm	Se espera que al registrar se avise que esta fuera del rango	Se espera que al registrar se avise que esta fuera del rango
Peso: 60 kg	Se acepta correctamente	Se acepta correctamente
Sexo: Masculino	Se acepta correctamente	Se acepta correctamente
Restricciones: Celíaco, Hipertenso	Se acepta correctamente	Se acepta correctamente
Preferencias: (vacío)	Se acepta correctamente	Se acepta correctamente
Se agrega una foto	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que hay un error al registrar el usuario	Se muestra un mensaje diciendo que hay un error al registrar el usuario
Se cambia la altura a una dentro del rango: 190 cm	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que hay un error al registrar el usuario	Se muestra un mensaje diciendo que hay un error al registrar el usuario
Se cambia el nombre de usuario a uno que no haya estado registrado anteriormente: pepi123	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el usuario se registro correctamente	Se muestra un mensaje diciendo que el usuario se registro correctamente

**Registro Profesional desde administrador:**

Se asume que hay un administrador registrado en el sistema y que el mismo tiene su sesión abierta

CP4.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Joselen	Se acepta correctamente	Se acepta correctamente
Apellido: Cecilia	Se acepta correctamente	Se acepta correctamente
Nombre de usuario: choche	Se acepta correctamente	Se acepta correctamente
Fecha de nacimiento: 04/02/2000	Se acepta correctamente	Se acepta correctamente
Título de profesional: Nutricionista	Se acepta correctamente	Se acepta correctamente
Fecha de graduación: 13/09/2018	Se acepta correctamente	Se acepta correctamente
Nacionalidad: Sudafricana	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el profesional se registró correctamente	Se muestra un mensaje diciendo que el profesional se registró correctamente

CP5.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Federico	Se acepta correctamente	Se acepta correctamente
Apellido: Novelli	Se acepta correctamente	Se acepta correctamente
Nombre de usuario: (vacío)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Se agrega nombre de usuario: choche	Se avisa que el alias de usuario no puede ser repetido	Se avisa que el alias de usuario no puede ser repetido
Fecha de nacimiento: (vacía)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Título de profesional: Doctor	Se acepta correctamente	Se acepta correctamente
Fecha de graduación: (vacía)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Nacionalidad: Perú	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el profesional no se pudo registrar correctamente	Se muestra un mensaje diciendo que el profesional no se pudo registrar correctamente
Se agrega fecha de nacimiento: 28/11/1994	Se acepta correctamente	Se acepta correctamente
Se agrega fecha de graduación: 15/12/2017	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el profesional se pudo registrar correctamente	Se muestra un mensaje diciendo que el profesional se pudo registrar correctamente



**Registro de alimento desde administrador:**

Se asume que hay un administrador registrado en el sistema y que el mismo tiene su sesión abierta

CP6.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: Yogurt light	Se acepta correctamente	Se acepta correctamente
Tipo de alimento: Lácteos	Se acepta correctamente	Se acepta correctamente
Principales nutrientes: Vitaminas	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el alimento se registró correctamente	Se muestra un mensaje diciendo que el alimento se registró correctamente

CP7.

Entrada	Resultado Esperado	Resultado Obtenido
Nombre: (vacío)	Se espera que al momento de querer registrar se avise que el campo está vacío	Se espera que al momento de querer registrar se avise que el campo está vacío
Tipo de alimento: Carne Roja	Se acepta correctamente	Se acepta correctamente
Principales nutrientes: Fibra, Proteínas	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el alimento no se registró correctamente	Se muestra un mensaje diciendo que el alimento no se registró correctamente
Se ingresa nombre del alimento: Albóndigas	Se acepta correctamente	Se acepta correctamente
Se oprime el botón Aceptar	Se muestra un mensaje diciendo que el alimento se registró correctamente	Se muestra un mensaje diciendo que el alimento se registró correctamente

#### **Agregar comida a dieta desde usuario**

Se asume un usuario registrado y con la sesión iniciada en el sistema

CP8.

Entrada	Resultado esperado	Resultado obtenido
Se oprime el botón agregar comida	Se despliega la lista con los alimentos registrados en el sistema	Se despliega la lista con los alimentos registrados en el sistema
Se selecciona uno de los alimentos registrados (Yogurt light en este caso) y se oprime el botón agregar comida	Se muestra un mensaje de que el alimento fue agregado correctamente al plan	Se muestra un mensaje de que el alimento fue agregado correctamente al plan
Se selecciona la opción de quitar alimento de la dieta marcando el alimento agregado	Se quita correctamente de la lista de alimentos del plan	Se quita correctamente de la lista de alimentos del plan

CP9.

Entrada	Resultado esperado	Resultado obtenido
Se oprime el botón agregar comida	Se despliega la lista con los alimentos registrados en el sistema	Se despliega la lista con los alimentos registrados en el sistema
No se seleccione ningún alimento de la lista	Se espera que se avise que se tiene que seleccionar un alimento para proseguir	Se avisa que se agregó correctamente el alimento

CP10.

### **Consulta dieta desde usuario:**

Se asume que hay un usuario registrado y con la sesión iniciada en el sistema

CP11.

Entrada	Resultado esperado	Resultado obtenido
Se oprime el botón borrar sin seleccionar ninguna consulta	Se avisa que se debe seleccionar una consulta	Se avisa que se debe seleccionar una consulta
Se oprime el botón redactar	Se despliega una ventana para redactar el mensaje	Se despliega una ventana para redactar el mensaje
Se elige como destinatario: Choche	Se acepta correctamente	Se acepta correctamente
Asunto: Pedir dieta	Se acepta correctamente	Se acepta correctamente
Mensaje: "Solicito dieta"	Se acepta correctamente	Se acepta correctamente
Se oprime el botón enviar	Se avisa que el mensaje fue enviado correctamente	Se avisa que el mensaje fue enviado correctamente

CP12.

Entrada	Resultado esperado	Resultado obtenido
Se oprime el botón redactar	Se despliega una ventana para redactar el mensaje	Se despliega una ventana para redactar el mensaje
Se elige como destinatario: Fede	Se acepta correctamente	Se acepta correctamente
Asunto: (vacío)	Se avisa que el asunto está vacío	Se avisa que el asunto está vacío
Mensaje: (vacío)	Se avisa que el mensaje está vacío	Se avisa que el mensaje está vacío
Se oprime el botón enviar	Se avisa que el mensaje no puede ser enviado correctamente	Se avisa que el mensaje no puede ser enviado correctamente
Se agrega Asunto: Consulta	Se acepta correctamente	Se acepta correctamente
Se agrega Mensaje: "Puedo consumir albóndigas el domingo?"	Se acepta correctamente	Se acepta correctamente
Se oprime el botón enviar	Se avisa que el mensaje fue enviado correctamente	Se avisa que el mensaje fue enviado correctamente

#### **Pedir plan desde usuario:**

Se asume que hay un usuario registrado y con la sesión iniciada en el sistema

CP13.

Entrada	Resultado esperado	Resultado obtenido
Se selecciona el profesional al que se le pedirá el plan de alimentación: choche	Se acepta correctamente	Se acepta correctamente
Se oprime el botón aceptar	Se avisa que se ha pedido el plan de alimentación correctamente	Se avisa que se ha pedido el plan de alimentación correctamente

#### **Realizar plan desde profesional:**

Se asume un profesional registrado y con la sesión iniciada en el sistema

CP14.

Entrada	Resultado esperado	Resultado obtenido
Se oprime el botón realizar plan sin seleccionar a ningún usuario que lo haya solicitado	Se pide que se seleccione un usuario para realizar el plan	Se pide que se seleccione un usuario para realizar el plan
Se selecciona uno de los usuarios que hayan solicitado el plan (en este caso camibur) y se oprime el botón realizar plan	Se despliega la venta para poder realizar el plan del usuario	Se despliega la venta para poder realizar el plan del usuario
Se elige el día de la semana: martes	Se acepta correctamente	Se acepta correctamente
Se oprime el botón agregar comida	Se despliega una ventana con los alimentos registrados	Se despliega una ventana con los alimentos registrados
Se agrega el alimento: Albóndigas	Se acepta correctamente	Se acepta correctamente
Se oprime el botón agregar comida	Se avisa que se agregó correctamente la comida a la dieta	Se avisa que se agregó correctamente la comida a la dieta

CP15.

Entrada	Resultado esperado	Resultado obtenido
Se selecciona uno de los usuarios que hayan solicitado el plan (en este caso camibur) y se oprime el botón realizar plan	Se despliega la venta para poder realizar el plan del usuario	Se despliega la venta para poder realizar el plan del usuario
Se elige el día de la semana: jueves	Se acepta correctamente	Se acepta correctamente
Se oprime el botón agregar comida	Se despliega una ventana con los alimentos registrados	Se despliega una ventana con los alimentos registrados
Se agrega el alimento: Jugo de pomelo	Se acepta correctamente	Se acepta correctamente
Se oprime el botón agregar comida	Se avisa que se agregó correctamente la comida a la dieta	Se avisa que se agregó correctamente la comida a la dieta
Se marca la comida Jugo de pomelo y se selecciona el botón quitar comida	Se avisa que la comida se sacó correctamente de la dieta	Se avisa que la comida se sacó correctamente de la dieta

#### **Consulta directa profesional:**

Se asume un profesional registrado y con la sesión iniciada en el sistema

CP16.

Entrada	Resultado esperado	Resultado obtenido
Se selecciona el mensaje que se desea leer	Se muestra el mensaje	Se muestra el mensaje
Se desea enviar un mensaje de respuesta, se oprime el botón redactar	Se muestra una ventana para poder redactar el mensaje con el usuario por defecto al que se le va a contestar	Se muestra una ventana para poder redactar el mensaje con el usuario por defecto al que se le va a contestar
Asunto: Dieta solicitada	Se acepta correctamente	Se acepta correctamente
Mensaje: "Se adjunta dieta solicitada"	Se acepta correctamente	Se acepta correctamente
Se oprime el botón enviar	Se avisa que se envió correctamente el mensaje	Se avisa que se envió correctamente el mensaje

CP17

Entrada	Resultado esperado	Resultado obtenido
Se selecciona el mensaje que se desea borrar	Se borra el mensaje	Se borra el mensaje
Se selecciona el mensaje que se desea leer	Se muestra el mensaje	Se muestra el mensaje
Se desea enviar un mensaje de respuesta, se oprime el botón redactar	Se muestra una ventana para poder redactar el mensaje con el usuario por defecto al que se le va a contestar	Se muestra una ventana para poder redactar el mensaje con el usuario por defecto al que se le va a contestar
Asunto: (vacío)	Se avisa que no se puede dejar el campo vacío	Se avisa que no se puede dejar el campo vacío
Mensaje: (vacío)	Se avisa que no se puede dejar el campo vacío	Se avisa que no se puede dejar el campo vacío
Se oprime el botón enviar	Se avisa que no se envió correctamente el mensaje	Se avisa que no se envió correctamente el mensaje

#### Registro Administrador:

CP18.

Entrada	Resultado esperado	Resultado Obtenido
Usuario: lola	Se acepta correctamente	Se acepta correctamente
Contraseña: 12345678	Se acepta correctamente	Se acepta correctamente
Se elige el botón registrarme	Se muestra un mensaje donde se muestra que fue registrado exitosamente	Se muestra un mensaje donde se muestra que fue registrado exitosamente

CP19.

Entrada	Resultado esperado	Resultado Obtenido
Usuario: pedro	Se acepta correctamente	Se acepta correctamente
Contraseña: 123	Se pide que la contraseña tenga por lo menos 8 caracteres	Se pide que la contraseña tenga por lo menos 8 caracteres
Contraseña:123hola123	Se acepta correctamente	Se acepta correctamente
Se elige el botón registrarme	Se muestra un mensaje donde se muestra que fue registrado exitosamente	Se muestra un mensaje donde se muestra que fue registrado exitosamente

#### **Inicio sesión de administrador:**

Se asume por lo menos un administrador registrado en el sistema

CP20.

Entrada	Resultado esperado	Resultado Obtenido
Usuario: pedro	Se acepta correctamente	Se acepta correctamente
Contraseña:123hola123	Se acepta correctamente	Se acepta correctamente
Se elige el botón iniciar sesión	Se inicia sesión correctamente	Se inicia sesión correctamente

#### **Inicio sesión usuario:**

Se asume al menos un usuario registrado en el sistema

CP21.

Entrada	Resultado esperado	Resultado Obtenido
Nombre: maria y se oprime el botón iniciar sesión	Se avisa que el usuario no está registrado en el sistema	Se avisa que el usuario no está registrado en el sistema

CP22

Entrada	Resultado esperado	Resultado Obtenido
Nombre: camibur y se oprime el botón iniciar sesión	Se inicia sesión correctamente	Se inicia sesión correctamente

#### **Inicio sesión profesional:**

Se asume al menos un profesional registrado en el sistema

CP23.



Entrada	Resultado esperado	Resultado Obtenido
Nombre: maria y se oprime el botón iniciar sesión	Se avisa que el profesional no está registrado en el sistema	Se avisa que el profesional no está registrado en el sistema

CP24.

Entrada	Resultado esperado	Resultado Obtenido
Nombre: chohce y se oprime el botón iniciar sesión	Se inicia sesión correctamente	Se inicia sesión correctamente

## 9. Reporte de defectos

### 9.1. Definición de categoría de defectos

A lo largo de la implementación del sistema, se presentaron distintas complicaciones y dificultades. Muchas de estas fueron corregidas y otras tantas por diversos motivos no.

Estos defectos o fallos se pueden categorizar en distintos criterios de categorización de defectos, estos serian:

La primera categorización es en cuanto al lugar en donde se presento el problema.

Defectos de dominio:

Los defectos de dominio consisten en distintas dificultades en cuanto a la organización y decisiones de diseño de las clases. También en los problemas que tuvimos a la hora de la implementación y manejo de las mismas, y sobre un buen uso de los estándares de codificación.

Defectos de interfaz:

Los defectos de la interfaz consisten en distintas dificultades que se presentaron a la hora de realizar la implementación de la misma y todas las interacciones con el usuario.

Defectos de usabilidad:

Los defectos de la usabilidad consisten en distintos fallos y dificultades que surgieron a la hora de hacer la interfaz. Por ejemplo, faltas de ortografía, desprolijidades en la interfaz, mala interacción con el usuario y problemas en general sobre las decisiones de diseño de la interfaz.

Defectos de repositorio:

Al ser este una nueva herramienta de trabajo, tuvimos distintas dificultades sobre un buen uso del mismo.

En cuanto a la gravedad de los defectos también podemos agruparlos en 3 categorías:

Alta:

Esta categoría incluye todos los defectos que fue fundamental arreglarlos para obtener un buen manejo del sistema. También incluye aquellos defectos que por algún motivo su resolución llevo mas tiempo del estimado.

Media:

Esta categoría incluye defectos que si bien su resolución fue fundamental para un buen funcionamiento del sistema, su resolución fue medianamente sencilla o se pudo seguir trabajando mientras se le buscaba una solución

Baja:

Esta categoría incluye defectos encontrados a lo largo del desarrollo, que pueden ser considerados como detalles del desarrollo de la aplicación. Por ejemplo, en el diseño de la interfaz. También agregamos una simple categorización para diferenciar las fallas que ya fueron arregladas y las que no.

## 9.2. Defectos encontrados por iteración

Defectos encontrados a lo largo del desarrollo de sistema, casos de uso, casos de prueba y pruebas funcionales:

### **Defectos encontrados en los paneles de la interfaz**

#### **Panel Editar Perfil de Profesional**

- Si no ingreso ningún dato en el campo nombre titulo profesional, aparece un label en fecha de graduación que dice el nombre de usuario no puede ser vacío”.

-Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

- Si no ingresó ningún dato en la fecha de nacimiento, no se muestra nada indicando el fallo en el ingreso.

Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

- Si no ingresó nada en la fecha de graduación, no se muestra nada indicando el fallo en el ingreso.

Nivel: Baja

Categoría: Usabilidad

Estado : Sin Arreglar

- Al presionar el botón aceptar, en caso de que se halla ingresado algún dato mal, no se activan los labels indicando los fallos de cada campo.

Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

#### **Panel Consulta Directa Profesional desde Profesional**

- El botón borrar es mas chico que el label que diría “Borrar”, y esto no deja ver

su texto.

Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

### **Panel Consulta Profesional desde Profesional**

- En la bandeja de mensajes de entrada no se lee completamente el texto de quien mando el mensaje

Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

### **Panel Realizar plan alimentación desde Profesional**

- Deja enviar el plan vacío

Nivel: Alta

Categoría: Usabilidad

Estado : Arreglado

### **Panel Redactar Consulta Directa profesional desde Profesional**

- Al presionar el botón enviar, no se le notifica al usuario que se envió el mensaje por medio de ningún label.

Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

### **Panel Agregar Comidas a Plan**

- Al presionar el botón agregar comida, sin importar que este agregue comidas o no, no se le informa al usuario lo que esta sucediendo en el sistema.

Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

- Al agregar comida y oprimir el botón Agregar comida, el panel cambia de tamaño a uno muy chico que no permite ver el resto de los elementos

Nivel: Alto

Categoría: Usabilidad

Estado : Arreglado

- Cuando no se selecciona ninguna comida para agregar y se aprieta el botón Agregar comida se notifica igual que se agrego correctamente

Nivel: Medio

Categoría: Usabilidad

Estado : Arreglado

### **Panel Editar Perfil Usuario**

- Al presionar el botón aceptar, si la altura o el peso son erróneos, si el focuslost de

cada campo no se activa, los labels indicando el error no se mostraran.

Nivel: Baja

Categoría: Usabilidad

Estado : Sin Arreglar

- Los focuslost de los campos no cambian los labels de error.

Nivel: Baja

Categoría: Usabilidad

Estado : Sin Arreglar

### **Panel Registro de Usuario**

- Si ingreso un dato invalido en la caja de peso o altura, luego de corregirlas, los labels indicando los errores no se corrigen.

Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

- Si no se agrega una fecha no se muestra ningún mensaje

Nivel: Alta

Categoría: Usabilidad

Estado : Arreglado

### **Panel Registro Profesional**

- Si no se ingresa un apellido no se notifica

Nivel: Alta

Categoría: Usabilidad

Estado : Arreglado

- El mensaje Alias no vacío no se lee entero

Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

- Si no se agrega una fecha no se muestra ningún mensaje

Nivel: Alta

Categoría: Usabilidad

Estado : Arreglado

- El mensaje de titulo vacío no se lee entero

Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

### **Panel Registro Alimento**

- No se utilizo el enum Ninguna para especificar que no se identifico ningún nutriente principal.

Nivel: Baja

Categoría: Usabilidad

Estado : Arreglado

#### **Fallos detectados durante el desarrollo del sistema:**

- No se utilizo el enum tipoAlimento en Alimento para filtrar a la hora de agregar un alimento a una dieta o plan de alimentación.

Nivel: Baja

Categoría: Dominio

Estado : Sin Arreglar

- No se verifica que la contraseña tenga números y letras (acepta contraseñas que tenga solo letras)

Nivel: Alta

Categoría: Usabilidad

Estado : Arreglado

- Encontramos faltas de ortografía

Nivel: Baja

Categoría: Usabilidad

Estado:Arreglado

- Como se menciono anteriormente, los registros tenían una metodología la cual causaba una mal función, esta no dejaba registrar mas de 1 objeto al sistema consecutivamente. Además, si había un campo con datos inválidos, el mismo no dejaba registrar

Nivel: Alta

Categoría: Interfaz

Estado: Arreglado

- Al abrir las consultas directas, tanto desde usuario como de profesional, esta dejaba redactar un mensaje sin validar que hubiera destinatarios registrados, si continuábamos la acción y enviábamos el mensaje, el programa se caía por falta de manejo de excepciones.

Nivel: Media

Categoría: Interfaz

Estado: Arreglado

- Utilizando las pruebas unitarias en conjunto con el JaCoCo, arreglamos los equals de las clases en las fueron redefinidos y les dimos cobertura.

Nivel: Media

Categoría: Dominio

Estado: Arreglado

- Encontramos problemas en la visibilidad de los atributos de los paneles ya que eran todos públicos.

Nivel: Baja

Categoría: Usabilidad

Estado: Arreglado

- Arreglamos bordes de botones transparentes.

Nivel: Bajo

Categoría: Usabilidad

Estado: Arreglado

- Colores a nivel de la interfaz incorrectos ya que no eran consistentes con el resto de la misma.

Nivel: Bajo

Categoría: Usabilidad

Estado: Arreglado

- Desincronización entre la lista que se muestra en la dieta diaria al momento de quitar una comida de la misma.

Nivel: Medio.

Categoría: Interfaz

Estado: Arreglado

# Bibliografía

- [1] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, eight ed. McGraw-Hill, 2014.
- [2] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2015.
- [3] Universidad ORT Uruguay. (2013) Documento 302 - Facultad de Ingeniería. [Online]. Available: <https://www.ort.edu.uy/fi/documentos/normas-especificas-para-la-presentacion-de-trabajos-finales-de-carrera.pdf>