

FridgeApp

INGENIERÍA DEL SOFTWARE

Entrega final

Grupo G4

Eduardo Ureña Toledano

NIA: 100329937

Santiago Vidal Carretero

NIA: 100346107

Álvaro González Muñoz

NIA: 100363552

Lucía Pérez Quintanar

NIA: 100363566

Página de estado del documento

Registro de cambios

Versión	Fecha	Autor	Resumen
0.1	11/10/2019	Lucía Pérez	Estructura inicial del documento, Introducción y primera versión de la Perspectiva del producto y el Alcance del software.
		Álvaro González	
		Eduardo Ureña	
0.15	12/10/2019	Eduardo Ureña	Se añaden algunas referencias y se actualizan los apartados “2.1 Perspectiva del Producto” y “2.2 Alcance del software”.
0.2	13/10/2019	Eduardo Ureña	Se completan los apartados “2.3 Capacidades generales” y “2.4 Restricciones generales”
0.3	13/10/2019	Santiago Vidal	Se completan los apartados “2.5 Características de los usuarios: roles y capacidades”, “2.6 Entorno operacional”, “2.7 Consideraciones éticas”, “3.2 Justificación de la clasificación de requisitos”, “3.3 Justificación de la plantilla de requisitos”.
0.4	13/10/2019	Lucía Pérez	Se completan los apartados “3.1 Vocabulario del dominio”, “3.1.1 Vocabulario del modelo de información” y “3.1.2 Vocabulario técnico”.
		Álvaro González	
0.5	13/10/2019	Santiago Vidal	Se realiza una primera versión de los requisitos funcionales.
		Lucía Pérez	
		Álvaro González	
		Eduardo Ureña	
0.6	13/10/2019	Eduardo Ureña	Se modifica el apartado “3.4 Requisitos funcionales” y se hace una primera versión del apartado “3.5 Requisitos no funcionales”.
0.7	14/10/2019	Santiago Vidal	Se actualizan los apartados “3.4 Requisitos funcionales” y “3.5 Requisitos no funcionales”.
0.8	14/10/2019	Eduardo Ureña	Se actualizan los apartados “3.1.1 Vocabulario del modelo de información”, “3.4 Requisitos funcionales” y “3.5 Requisitos no funcionales”.

0.9	14/10/2019	Álvaro González	Se completa el apartado “5.1 Consistencia entre los requisitos: conflictos, redundancias, acoplamientos” y se actualizan los apartados “3.4 Requisitos funcionales” y “3.5 Requisitos no funcionales”.
		Lucía Pérez	
1.0	14/10/2019	Lucía Pérez	Última revisión del documento antes de realizar la primera entrega.
		Álvaro González	
		Eduardo Ureña	
		Santiago Vidal	
1.1	21/11/2019	Lucía Pérez	Se hace una primera versión del modelo de información.
		Santiago Vidal	
1.2	23/11/2019	Eduardo Ureña	Se actualiza el modelo de información y se explican las clases y atributos en el apartado “4.1 Modelo de información”.
		Álvaro González	
		Lucía Pérez	
1.3	24/11/2019	Eduardo Ureña	Se termina el modelo de información y se incorpora al documento.
1.4	24/11/2019	Santiago Vidal	Se explica el modelo arquitectónico elegido en el apartado “4.2 Modelo de implementación” y se hace el modelo de implementación.
		Lucía Pérez	
		Álvaro González	
1.5	24/11/2019	Santiago Vidal	Se realiza la especificación del diseño de componentes en el apartado “4.3 Especificación del diseño de componentes”.
		Eduardo Ureña	

1.6	24/11/2019	Lucía Pérez	Se realizan los apartados “5.2 Trazabilidad requisitos – modelo de información (clases)” y “5.3 Trazabilidad requisitos – modelo de implementación (componentes)”.
2.0	25/11/2019	Eduardo Ureña	Última revisión del documento antes de realizar la segunda entrega.
2.1	23/12/2019	Eduardo Ureña	Se adapta el documento a la estructura de la entrega final, añadiendo tanto las revisiones recibidas como las realizadas, adaptándolas además al formato de nuestro documento.
2.2	24/12/2019	Eduardo Ureña	Se corrigen errores de ortografía y gramática expuestos por el grupo revisor.
2.3	26/12/2019	Eduardo Ureña	Se corrigen más errores especificados por el grupo revisor y se actualiza el modelo de información.
2.4	03/01/2020	Santiago Vidal	Se corrigen errores en la especificación del diseño de componentes, añadiendo una nueva operación “Consultar receta” y añadiendo las dependencias.
		Álvaro González	
2.5	04/01/2020	Lucía Pérez	Se crean los requisitos RF17 y RF18 y con ello se actualizan los apartados “5.2 Trazabilidad requisitos – modelo de información (clases)” y “5.3 Trazabilidad requisitos – modelo de implementación (componentes)”.
2.6	08/01/2020	Álvaro González	Se añaden “Vista” y “Datamaster” en el apartado 5.3.
2.7	08/01/2020	Lucía Pérez	Se rellena el “Anexo II. Respuesta a las revisiones recibidas”.
		Eduardo Ureña	
		Santiago Vidal	
3.0	08/01/2020	Eduardo Ureña	Última revisión del documento antes de realizar la entrega final.

Horas invertidas en el proyecto

Nombre	Individual	Equipo	Total
Lucía Pérez Quintanar	4	0	4
Eduardo Ureña Toledano	10	0	10
Santiago Vidal Carretero	4	2	6
Álvaro González Muñoz	3	2	5
TOTAL	21	4	25

Distribución de responsabilidades entre los miembros del equipo

Nombre	Aportación al documento
Lucía Pérez Quintanar	<ul style="list-style-type: none"> • Estructura del documento y diseño del documento. • Redacción de la introducción al documento. • Redacción de las definiciones del vocabulario. • Escritura de algunos de los requisitos presentes y últimas modificaciones de los ya existentes. • Elaboración de las relaciones entre requisitos funcionales y no funcionales. • Último repaso antes de la primera entrega. • Redacción de la explicación referente al modelo de implementación. • Realización de las tablas modelo de información (clases) y modelo de implementación (componentes). • Aportación de ideas para el modelo de información. • Actualización de los requisitos y de las tablas tablas modelo de información (clases) y modelo de implementación (componentes). • Escritura de respuestas del Anexo II.
Eduardo Ureña Toledano	<ul style="list-style-type: none"> • Estructura del documento y diseño del documento. • Escritura de las Referencias • Perspectiva del producto (Antecedentes y productos similares). • Aportaciones y reestructuramiento del alcance del software. • Redacción de las características y restricciones generales. • Escritura y modificación de algunos de los requisitos funcionales y no funcionales. • Pequeña aportación al vocabulario del modelo de

	<p>información.</p> <ul style="list-style-type: none"> ● Actualización del Modelo de información. ● Explicación de las clases y atributos del modelo de información. ● Aportación de ideas para la especificación del diseño de componentes. ● Adaptación del documento a la estructura de la entrega final. ● Corrección de errores de ortografía y gramática especificados por el grupo revisor. ● Corrección de demás errores expuestos por el grupo revisor. ● Reactualización del Modelo de información. ● Escritura de algunas respuestas del Anexo II. ● Último repaso antes de cada entrega.
Santiago Vidal Carretero	<ul style="list-style-type: none"> ● Estructura del documento y diseño del documento. ● Redacción de algunos de los requisitos funcionales y no funcionales. ● Vocabulario técnico. ● Restricciones generales. ● Características de los usuarios: roles y capacidades. ● Entorno operacional. ● Consideraciones éticas. ● Creación del diseño del modelo vista controlador ● Elaboración de las tablas para la especificación del diseño de componentes. ● Escritura de la respuesta a la segunda revisión en el Anexo II.
Álvaro González Muñoz	<ul style="list-style-type: none"> ● Estructura del documento y diseño del documento. ● Propósito del documento y Visión general. ● Perspectiva del producto y alcance del software. ● Recapitular el vocabulario del dominio y describir los conceptos. ● Justificar la clasificación de requisitos y aportación de requisitos. ● Elaboración de la consistencia entre los requisitos. ● Último repaso antes de la primera entrega. ● Realización del diagrama UML y explicación de las diferentes clases. ● Realización del modelo de arquitectura y redacción del apartado.

Tabla de contenido

1. Introducción	9
1.1 Propósito del Documento	9
1.2 Visión General del Documento	9
1.3. Referencias	10
2. Descripción General	10
2.1 Perspectiva del Producto	10
2.2 Alcance del software	11
2.3 Capacidades generales	11
2.4 Restricciones generales	11
2.5 Características de los usuarios: roles y capacidades	12
2.6 Entorno operacional	12
2.7 Consideraciones éticas	13
3.1 Vocabulario del dominio	13
3.1.1 Vocabulario del modelo de información	13
3.1.2 Vocabulario técnico	13
3.2 Justificación de la clasificación de requisitos	14
3.3 Justificación de la plantilla de requisitos	15
3.4 Requisitos funcionales	15
3.4.1 Requisitos sobre la administración de las cuentas de usuario	15
3.4.2 Requisitos sobre los alimentos	17
3.4.3 Requisitos sobre la administración de las recetas	19
3.4.4 Requisitos respecto al código QR	21
3.5 Requisitos no funcionales	22
4. Arquitectura	25
4.1 Modelo de información	25
4.2 Modelo de implementación	26
4.3 Especificación del diseño de componentes	28
4.3.1 Componente GestorUsuario	28
4.3.2 Componente GestorPreferencias	29
4.3.3 Componente GestorNevera	30
4.3.4 Componente DataMaster	31
4.3.5 Componente Vista	32
5. Lista y tablas	32
5.1 Consistencia entre los requisitos: conflictos, redundancias, acoplamientos	32
5.2 Trazabilidad requisitos – modelo de información (clases)	34
5.3 Trazabilidad requisitos – modelo de implementación (componentes)	35
Anexo I. Revisiones recibidas	36
Primera revisión	36
1. Aspectos positivos que aporta el documento:	36

2. Consideraciones generales sobre el formato del documento.	36
3. Consideraciones individuales sobre la adecuación al índice de la práctica.	37
3.1. Estructura del documento.	37
3.2. Contenido del documento.	38
3.3. Explicación con más detalle los errores principales.	39
Segunda revisión	40
1. Aspectos positivos que aporta el documento:	40
2. Consideraciones generales sobre el formato del documento.	40
3. Consideraciones individuales sobre la adecuación al índice de la práctica.	41
3.1. Estructura del documento.	41
3.2. Contenido del documento.	41
3.3. Explicación con más detalle de los errores principales.	42
Anexo II. Respuesta a las revisiones recibidas	43
Primera revisión	43
Segunda revisión	44
Anexo III. Revisiones enviadas	45
Primera revisión	45
1. Aspectos positivos que aporta el documento.	45
2. Consideraciones generales sobre el formato del documento.	45
2.1 Errores generales.	45
2.2 Errores de ortografía y gramaticales.	46
2.3 Presentación y legibilidad	47
3. Consideraciones individuales sobre adecuación al índice de la práctica.	47
3.1 Estructura del documento: ¿sigue los apartados del índice de la práctica?	47
3.2 Contenido del documento: ¿se corresponde cada uno de los apartados con la finalidad que el índice propone?	48
3.3 Explicación en más detalle de los errores principales	48
Segunda revisión	49
1. Aspectos positivos que aporta el documento.	49
2. Consideraciones generales sobre el formato del documento.	49
2.1 Errores generales.	49
2.2 Errores de ortografía y gramaticales.	49
2.3 Presentación y legibilidad	51
3. Consideraciones individuales sobre adecuación al índice de la práctica.	51
3.1 Estructura del documento: ¿sigue los apartados del índice de la práctica?	51
3.2 Contenido del documento: ¿se corresponde cada uno de los apartados con la finalidad que el índice propone?	51
3.3 Explicación en más detalle de los errores principales	52

1. Introducción

1.1 Propósito del Documento

El objetivo del siguiente documento es determinar el acuerdo entre accionistas y clientes en relación a la aplicación “FridgeApp” y la nevera “IntelFridge”, e irá destinado tanto a estos, como a los diseñadores de software y a los programadores.

Por ende, se especificarán las diferentes versiones que se han ido actualizando de forma que todo aquel que lo lea entienda el proceso de elaboración del mismo. Además, poseerá una sección en la que se especificarán los términos utilizados para así hacerlo mucho más accesible.

1.2 Visión General del Documento

Descripción general: serie de subapartados en los que se realiza una introducción sobre el funcionamiento acordado de la aplicación y sus utilidades. Estos apartados son:

- Perspectiva del producto.
- Alcance del software.
- Capacidades generales.
- Restricciones generales.
- Características de los usuarios: roles y capacidades.
- Entorno operacional.

Requisitos: sección mucho más concreta en la que se especifican los requisitos de la aplicación, además del vocabulario utilizado a lo largo del documento con su correspondiente descripción de forma clara. Contiene también las subsecciones siguientes:

- Justificación de la clasificación de requisitos.
- Justificación de la plantilla de requisitos.
- Requisitos funcionales.
- Requisitos no funcionales.
- Vocabulario del dominio.

Arquitectura: serie de diagramas con los que se representarán los diferentes modelos presentes en la arquitectura del software. Consta de los siguientes apartados:

- Modelo de información.
- Modelo de implementación.
- Especificación del diseño de componentes.

Listas y tablas: en esta última sección se valorará la especificación entre los requisitos:

- Consistencia entre requisitos: conflictos, redundancias, acoplamientos.
- Trazabilidad requisitos – modelo de información (clases).
- Trazabilidad requisitos – modelo de implementación (componentes).

1.3. Referencias

[1] Universidad Carlos III de Madrid. Escuela Politécnica Superior, “Ingeniería de Software – Índice de la práctica”. Disponible en:

https://aulaglobal.uc3m.es/pluginfile.php/3276342/mod_resource/content/3/IS%201920%20C3%8Dndice%20de%20la%20pr%C3%A1ctica.pdf

[2] Universidad Carlos III de Madrid. Escuela Politécnica Superior, “Ingeniería del Software - Selección de propuestas vencedoras”. Disponible en:

https://aulaglobal.uc3m.es/pluginfile.php/3276390/mod_resource/content/3/Vencedores.pdf

[3] Gonzalo Génova Fuster, “Ingeniería del Software, un enfoque crítico (Curso 2019 – 2020) – Transparencias”. Disponible en:

https://aulaglobal.uc3m.es/pluginfile.php/3276340/mod_resource/content/3/IS%201718%20Transparencias.pdf

[4] Gonzalo Génova Fuster, “Ingeniería del Software, un enfoque crítico (Curso 2019 – 2020) – Guión de clase”. Disponible en:

https://aulaglobal.uc3m.es/pluginfile.php/3276341/mod_resource/content/4/IS%201718%20Guiones%20de%20clase.pdf

[5] “Frigoríficos inteligentes: cuando las neveras hacen mucho más que enfriar”. Disponible en:

<https://www.tecnologiadetatu.elcorteingles.es/actualidad/frigorificos-inteligentes-cuando-las-neveras-hacen-mucho-mas-que-enfriar/>

2. Descripción General

2.1 Perspectiva del Producto

El objetivo general de nuestro producto es que los usuarios puedan seguir una alimentación más equilibrada, saber los alimentos que tienen disponibles en la nevera sin necesidad de abrirla y, además aprender nuevas recetas.

Lo más parecido que existe en el mercado son unas neveras de LG, Samsung y Whirlpool, de las cuales, la de LG puede saber los alimentos que hay en su interior y la caducidad de estos; la de Samsung tiene Wi-Fi integrado y la capacidad de ver recetas; y la de Whirlpool tiene una aplicación disponible para smartphone o tablet, en la que se indican los productos existentes, además de sugerir recetas.

Nuestro producto busca combinar todas estas características, además de ofrecer otras nuevas, por lo tanto nuestro dispositivo será un gran avance en el campo de los refrigeradores.

2.2 Alcance del software

Neofridge Company S.A dispondrá de una gama de neveras inteligentes, denominadas "IntelFridge", las cuales tendrán disponibles una App complementaria, llamada "FridgeApp", que estará disponible tanto en IOS como en Android.

La combinación de "FridgeApp - IntelFridge", permitirá facilitar acciones del día a día, tales como proponer recetas que elaborar teniendo en cuenta los alimentos disponibles, o indicar que alimentos se necesitan para elaborar una determinada receta, además de proporcionar la fecha de caducidad de los alimentos.

Gracias al reconocedor de alimentos se podrán identificar cada uno de los elementos que se encuentran dentro de "IntelFridge", y al estar vinculado con FridgeApp, poder recordar en cualquier momento que alimentos debe ir a comprar el cliente en caso de que los necesite.

Además, el usuario podrá especificar si tiene alguna alergia o preferencia alimenticia, de manera que se puedan proponer las recetas más adecuadas para el usuario o usuarios de una aplicación.

2.3 Capacidades generales

Nuestro producto deberá ofrecer las siguientes funcionalidades:

- "IntelFridge" tendrá que poder escanear, tanto los alimentos que se encuentran en su interior como la caducidad de los mismos.
- FridgeApp deberá indicar los productos que hay en la nevera y la caducidad de los mismos.
- FridgeApp deberá ser capaz de guardar los datos del usuario junto a sus preferencias o alergias alimenticias.
- FridgeApp podrá recomendar recetas en base a los alimentos que hay en la nevera "IntelFridge" a la que esté conectada, teniendo en cuenta las restricciones alimenticias que haya puesto el usuario de la aplicación.
- FridgeApp dispondrá de una gran colección de recetas, gracias a la cual, será capaz de indicar qué ingredientes no disponibles en la nevera son necesarios para preparar otras recetas.
- FridgeApp podrá conectarse a la página web de "IntelFridge" para obtener las recetas.
- FridgeApp permitirá guardar al usuario sus recetas favoritas.

2.4 Restricciones generales

El funcionamiento de "IntelFridge" y FridgeApp está limitado por las siguientes restricciones:

- FridgeApp solo funcionará con la gama de neveras "IntelFridge" de NeoFridge.

- FridgeApp solo obtendrá recetas de la página web "de IntelFridge"
- FridgeApp requerirá de conexión a internet.
- FridgeApp se conectará a "IntelFridge" vía conexión Wi-Fi.
- Será necesario que el usuario se cree un perfil en FridgeApp para indicar sus restricciones alimenticias y guardar sus recetas favoritas.
- Los datos y preferencias del usuario se guardarán en el móvil donde esté instalada la aplicación FridgeApp.
- Se deberá poder crear más de un perfil por aplicación.

2.5 Características de los usuarios: roles y capacidades

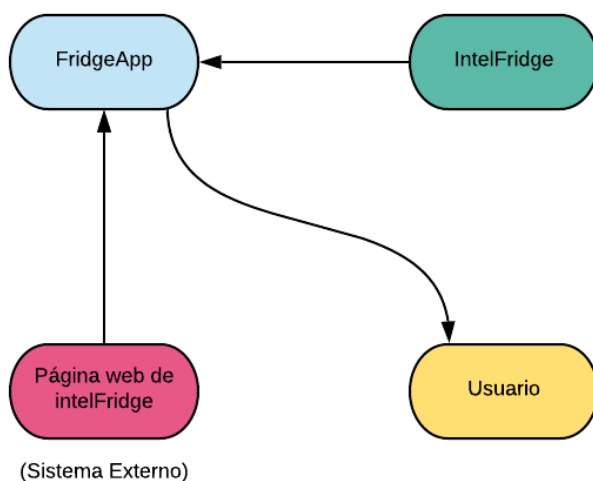
Rol	Capacidades
Usuario	Solicitar recetas en función de los alimentos disponibles.
	Consultar alimentos que se encuentran en la nevera y su caducidad.
	Indicar sus alergias o preferencias alimenticias.
Administrador	Verificar el correcto funcionamiento de la plataforma.
	Añadir valores nutricionales de los alimentos.

2.6 Entorno operacional

Se podrá utilizar la aplicación tanto en dispositivos móviles como en tabletas, Android versión 5.0 o superiores y/o IOS para versiones superiores a la 10.0.

La información de la aplicación se guardará únicamente en el dispositivo móvil.

La aplicación FridgeApp interactúa internamente con la gama de neveras "IntelFridge" y con la página web de FridgeApp de la cual descarga las recetas. Externamente no interactúa con nada ya que la única interacción externa posible sería el escaneo de los alimentos, pero de eso se encarga la nevera.



2.7 Consideraciones éticas

El sistema no crea ningún problema ético al solo tener acceso a los datos de inicio de sesión del usuario. El resto de datos, las recetas y los favoritos, se almacenan en el dispositivo móvil.

Resuelve problemas éticos respecto al consumo de ciertos alimentos, ya que podremos seleccionar preferencias alimenticias para que no se recomienden recetas que contengan alimentos que puedan perjudicar al usuario.

3. Requisitos

3.1 Vocabulario del dominio

3.1.1 Vocabulario del modelo de información

- **Neofridge:** Empresa que fabrica FridgeApp e "IntelFridge".
- **FridgeApp:** Aplicación complementaria de "IntelFridge".
- **IntelFridge:** Nevera inteligente capaz de escanear alimentos y es soporte de FridgeApp.
- **Nevera:** Aparato electrodoméstico para enfriar alimentos.
- **Receta:** Recomendación que las neveras "IntelFridge" ofrecen al usuario según el tipo de alimentación que el usuario haya configurado en la app "FridgeApp" y los ingredientes que tenga en la nevera.
- **Caducidad:** Momento en el que un alimento deja de ser adecuado para su consumición.
- **Dispositivo:** Soporte móvil electrónico en el que se podrá disponer de la aplicación "FridgeApp".
- **Usuario:** Persona a quien va dirigida tanto la aplicación FridgeApp como las neveras "IntelFridge".

3.1.2 Vocabulario técnico

- **App:** Aplicación informática diseñada para teléfonos móviles inteligentes.
- **Software:** Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.
- **Hardware:** Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.
- **Smartphone:** Dispositivo celular móvil que, entre muchas funciones, permite al usuario acceder a internet y gestionar sus aplicaciones.
- **Tablet:** Dispositivo más grande que el smartphone cuya característica principal es su mayor tamaño de pantalla .

- **Internet:** Red informática de nivel mundial que utiliza la línea telefónica para transmitir la información.
- **Wi-Fi:** Tecnología que permite la interconexión inalámbrica de dispositivos electrónicos.
- **Página Web:** Información electrónica capaz de contener texto, sonido, vídeo, programas, enlaces, imágenes y muchas otras cosas, adaptada para ser accedida mediante un navegador web.
- **IOS:** Sistema operativo para móviles desarrollado por Apple.
- **Android:** Sistema operativo para móviles basado en Linux y desarrollado por Google.
- **LG:** Compañía fabricante de dispositivos electrónicos.
- **Samsung:** Compañía fabricante de dispositivos electrónicos.
- **Whirlpool:** Compañía fabricante de dispositivos electrónicos.
- **Alergia:** Reacción inmunitaria del organismo frente a una sustancia generalmente inocua para el anfitrión, que se manifiesta por unos signos y síntomas característicos cuando este se expone a ella (por inhalación, ingestión o contacto cutáneo).
- **Escanear:** Pasar un texto, imagen u objeto a través de un escáner para convertirlo en un conjunto de datos procesables por una computadora o un sistema informático.
- **Google Play:** Plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android.
- **Apple Store:** Plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo IOS.
- **Perfil de usuario:** Cuenta que crea el usuario y que puede personalizar según sus restricciones alimenticias y los productos que tenga en su nevera.

3.2 Justificación de la clasificación de requisitos

La clasificación de requisitos se compone de dos tipos de requisitos, los requisitos funcionales y los requisitos no funcionales.

Los requisitos funcionales son el conjunto de requisitos que abarca todas las capacidades del sistema, es decir, lo que es capaz de lograr.

Los requisitos funcionales estarán agrupados conforme a la función principal que satisfacen:

- Administración de la cuenta de usuario
- Requisitos sobre los alimentos
- Requisitos sobre la administración de las recetas
- Requisitos respecto al código QR

Los requisitos no funcionales, por el contrario, se encargan de establecer las limitaciones del sistema.

3.3 Justificación de la plantilla de requisitos

Tanto los requisitos funcionales como los no funcionales mantendrán el formato de campos que establecemos con la siguiente plantilla, siendo todos los campos obligatorios.

Identificador	Título del requisito
Versión	Fecha en la que ha sido modificado el requisito por última vez.
Autor/Autores	Nombre de la última persona que ha modificado el requisito.
Descripción	Breve descripción del requisito.
Necesidad	Indica si el requisito es obligatorio u opcional.
Pruebas	Pruebas de verificación que se deben realizar para comprobar el correcto funcionamiento del requisito.

Los campos “Versión” y “Autores” sirven para identificar la fecha y autoría del requisito en cuestión, mientras que “Descripción” e “Necesidad” facilitan una idea fundamental del requisito y la importancia que tiene éste en el sistema. El campo “Pruebas” establece las acciones que hay que realizar para comprobar que el requisito es correcto.

3.4 Requisitos funcionales

3.4.1 Requisitos sobre la administración de las cuentas de usuario

RF03	Crear cuenta de usuario
Versión	13/10/2019
Autor/Autores	Santiago Vidal Carretero
Descripción	<p>El usuario deberá crearse una cuenta en FridgeApp. Los campos necesarios para la creación de dicha cuenta serán:</p> <ul style="list-style-type: none">• Nombre de usuario.• Contraseña.• Correo electrónico.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none">• Crear una cuenta con todos los campos rellenados correctamente y comprobar que se puede crear satisfactoriamente.• Crear una cuenta con un nombre de usuario ya existente. (no debe permitirse)

	<ul style="list-style-type: none"> ● Crear una cuenta con un correo electrónico previamente registrado. (no debe permitirse) ● Crear una cuenta con un formato de correo electrónico erróneo. (no debe permitirse)
--	--

RF04	Modificar cuenta de usuario
Versión	13/10/2019
Autor/Autores	Santiago Vidal Carretero
Descripción	<p>Se permite al usuario poder modificar algunos campos de su cuenta como, por ejemplo:</p> <ul style="list-style-type: none"> ● Cambiar nombre de usuario. ● Cambiar contraseña. ● Cambiar el correo electrónico.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Comprobación de guardado correcto al cambiar alguno de los datos anteriormente mencionados. Se debe comprobar en la información de la cuenta. ● No debe de permitirse poner un nombre ya existente al cambiar el nombre de usuario. ● No debe permitirse el cambio de correo electrónico a un formato incorrecto.

RF05	Borrar cuenta de usuario
Versión	13/10/2019
Autor/Autores	Santiago Vidal Carretero
Descripción	FridgeApp permitirá borrar la cuenta del usuario. Todos los datos vinculados a dicha cuenta de usuario serán borrados.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Borrar la cuenta de un usuario. ● Probar que se puede crear una cuenta con el nombre del usuario que se ha borrado. ● Probar que se puede crear una cuenta con el correo electrónico del usuario que se ha borrado.

RF06	Ventana de diálogo para borrar la cuenta
Versión	13/10/2019
Autor/Autores	Santiago Vidal Carretero
Descripción	Al borrar una cuenta de usuario, el sistema debe mostrar una ventana de diálogo para confirmar la eliminación de la cuenta.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Borrar la cuenta de usuario, verificar que salta la ventana de confirmación y que, al aceptar, la cuenta se borra correctamente. ● Comprobar si al no aceptar la ventana de confirmación la cuenta no se borra.

RF17	Conexión con la página web
Versión	04/01/2020
Autor/Autores	Lucía Pérez Quintanar
Descripción	Un usuario con la aplicación FridgeApp deberá poder conectarse a la página web de "IntelFridge".
Necesidad	Obligatorio
Pruebas	Una vez descargada la aplicación FridgeApp en un dispositivo con conexión a internet se comprueba que puede acceder a la página web de "IntelFridge" y puede registrarse con su cuenta de usuario.

3.4.2 Requisitos sobre los alimentos

RF01	Reconocimiento de alimentos
Versión	13/10/2019
Autor/Autores	Álvaro González Muñoz
Descripción	El sistema deberá ser capaz de identificar los diferentes alimentos que se encuentran en el interior de "IntelFridge" mediante los sensores que se hayan instalado.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Se introduce un alimento (plátano) en la "IntelFridge". ● Se comprueba que la aplicación muestra un plátano en los alimentos que contiene la aplicación.

RF02	Reconocimiento de la fecha de caducidad de los alimentos
Versión	13/10/2019
Autor/Autores	Eduardo Ureña Toledano
Descripción	El sistema deberá ser capaz de identificar la fecha de caducidad de cada alimento que se encuentre en el interior de "IntelFridge" mediante los sensores que se hayan instalado.
Necesidad	Obligatorio
Pruebas	Comprobar que en cada alimento escaneado se puede ver un campo en el que se indica la caducidad del alimento.

RF07	Notificación sobre la caducidad de alimentos
Versión	13/10/2019
Autor/Autores	Lucía Pérez Quintanar
Descripción	FridgeApp deberá enviar una notificación al dispositivo móvil de qué alimento o serie de alimentos van a caducar pronto.
Necesidad	Obligatorio
Pruebas	Se introducirá un alimento con una fecha de caducidad próxima (inferior a 2 días) y se comprobará que se envía la notificación al dispositivo móvil, indicando el alimento que va a caducar.

RF08	Valores mínimos de alimentos
Versión	13/10/2019
Autor/Autores	Santiago Vidal Carretero
Descripción	El usuario podrá establecer la cantidad mínima de un alimento en la "IntelFridge".
Necesidad	Obligatorio
Pruebas	Se establece la cantidad mínima del alimento, verificar que se ha guardado la cantidad mínima.

RF09	Notificación sobre insuficiencia de alimento
Versión	13/10/2019
Autor/Autores	Santiago Vidal Carretero
Descripción	FridgeApp notificará al usuario a través del dispositivo móvil acerca de una cantidad inferior a la cantidad mínima del alimento.
Necesidad	Obligatorio
Pruebas	Una vez establecida la cantidad mínima de un alimento, probar a retirar ese alimento hasta alcanzar menos del mínimo y comprobar que se envía la notificación acerca del alimento.

RF10	Seleccionar preferencias alimenticias
Versión	13/10/2019
Autor/Autores	Lucía Pérez Quintanar
Descripción	<p>El usuario podrá añadir sus preferencias alimenticias o alergias. De este modo se podrán especificar:</p> <ul style="list-style-type: none"> ● Alimentos nunca consumidos (ideal para personas con restricciones alimentarias o alergias). ● Alimentos pocos consumidos (por cuestiones económicas o por otros factores como reacciones alérgicas en grandes cantidades).
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● El sistema no debe dejar seleccionar el mismo tipo de alimento para las dos opciones de “Nunca consumidos” y “Poco consumidos”. ● Comprobación de que las recetas recomendadas no contengan alguno de los alimentos marcados como “Nunca consumidos”.

3.4.3 Requisitos sobre la administración de las recetas

RF11	Recetas recomendadas
Versión	13/10/2019
Autor/Autores	Eduardo Ureña Toledano

Descripción	FridgeApp tiene que poder recomendar recetas teniendo en cuenta los alimentos que hay en "IntelFridge" y las restricciones alimenticias del usuario.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Buscar en FridgeApp una sección donde aparezcan recetas disponibles. ● Comprobar que todos los ingredientes necesarios para elaborarla están en "IntelFridge".

RF12	Ingredientes necesarios
Versión	13/10/2019
Autor/Autores	Eduardo Ureña Toledano
Descripción	FridgeApp tiene que ser capaz de indicar que ingredientes faltan para poder preparar una determinada receta.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Buscar en FridgeApp la sección donde aparezcan todas las recetas guardadas en FridgeApp. ● Comprobar que para cada una de ellas se indican los ingredientes que faltan para poder prepararla.

RF13	Guardar recetas favoritas
Versión	13/10/2019
Autor/Autores	Eduardo Ureña Toledano
Descripción	Debe ser posible guardar recetas como favoritas.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Buscar en FridgeApp una receta cualquiera. ● Marcar la receta como favorita. ● Comprobar que aparece la receta elegida como favorita.

RF15	Borrar recetas de favoritas
Versión	13/10/2019
Autor/Autores	Santiago Vidal Carretero

Descripción	Debe ser posible borrar recetas de favoritas.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Buscar en FridgeApp una receta marcada como favorita. ● Borrar la receta de favoritos. ● Comprobar que la receta no aparece como favorita.

RF16	Botón de recetas favoritas
Versión	1/1/2020
Autor/Autores	Álvaro González Muñoz
Descripción	Botón que al pulsarlo mostrará una lista de las recetas favoritas.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Buscar en FridgeApp el botón de recetas favoritas. ● Muestra la lista de recetas favoritas.

RF18	Descarga de recetas
Versión	04/01/2020
Autor/Autores	Lucía Pérez Quintanar
Descripción	La aplicación FridgeApp deberá poder descargar de la página web de “IntelFridge” las recetas disponibles.
Necesidad	Obligatorio
Pruebas	Una vez descargada la aplicación FridgeApp en un dispositivo con conexión a internet y conectado a la página web de “IntelFridge”, se comprueba que se descargan las recetas disponibles.

3.4.4 Requisitos respecto al código QR

RF14	Código QR
Versión	14/10/2019
Autor/Autores	Álvaro González Muñoz
Descripción	Cada nevera debe contener un código QR con el cual se identificará independientemente del resto.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● El código QR es reconocido por FridgeApp.

3.5 Requisitos no funcionales

RNF01	Exclusiva compatibilidad de FridgeApp con "IntelFridge"
Versión	13/10/2019
Autor/Autores	Eduardo Ureña Toledano
Descripción	FridgeApp funcionará con la gama de neveras "IntelFridge" de Neofridge y con ninguna otra nevera.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Probar a conectar FridgeApp a una nevera "IntelFridge" y comprobar la correcta conexión. ● Probar a conectar FridgeApp a los diferentes modelos de neveras inteligentes existentes en el mercado con capacidad de conexión Wi-Fi y comprobar que no se conecta.

RNF02	Conexión a internet necesaria
Versión	13/10/2019
Autor/Autores	Eduardo Ureña Toledano
Descripción	FridgeApp necesita tener conexión a internet para acceder a las recetas y poder conectarse a "IntelFridge". Será necesario que aparezca un mensaje indicativo de que no hay conexión en caso de no haberla.
Necesidad	Obligatorio
Pruebas	Probar a entrar en FridgeApp sin tener acceso a internet, y comprobar que sale el mensaje correspondiente.

RNF03	Conexión Wi-Fi
Versión	14/10/2019
Autor/Autores	Eduardo Ureña Toledano y Álvaro González Muñoz
Descripción	FridgeApp se podrá conectar a "IntelFridge" únicamente vía conexión Wi-Fi.
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Se activa el Wi-Fi en "IntelFridge" y se intenta encontrar el router Wi-Fi. ● Se comprueba la correcta conexión de internet con "IntelFridge". ● Se escanea el código QR de "IntelFridge" desde el dispositivo donde está instalado FridgeApp. ● Se comprueba la correcta sincronización con la nevera.

RNF04	Tiempo de respuesta
Versión	14/10/2019
Autor/Autores	Santiago Vidal Carretero
Descripción	FridgeApp deberá tener un tiempo de respuesta de menos de 1 segundo.
Necesidad	Obligatorio
Pruebas	Se irá pasando por todas las funcionalidades de FridgeApp comprobando que el tiempo de respuesta sea menor o igual al establecido.

RNF05	Plataformas
Versión	14/10/2019
Autor/Autores	Santiago Vidal Carretero
Descripción	FridgeApp deberá funcionar en las plataformas Android (mínimo 5.0) e iOS (mínimo 10.0)
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Descargar FridgeApp en un dispositivo Android y comprobar que funciona. ● Descargar FridgeApp en un dispositivo iOS y comprobar que funciona.

RNF06	Dispositivos móviles
Versión	14/10/2019
Autor/Autores	Santiago Vidal Carretero
Descripción	FridgeApp deberá presentar versiones para móviles y tablets
Necesidad	Obligatorio
Pruebas	<ul style="list-style-type: none"> ● Comprobar el correcto funcionamiento en móvil. ● Comprobar el correcto funcionamiento en tablet.

RNF07	Distribución de FridgeApp
Versión	14/10/2019
Autor/Autores	Santiago Vidal Carretero
Descripción	FridgeApp se distribuirá por medio de Google Play en el caso de Android y por medio de Apple Store en caso de IOS.

Necesidad	Obligatorio
Pruebas	Comprobación de la existencia de la aplicación en ambas plataformas y descargar la aplicación.

RNF08	Cámara
Versión	14/10/2019
Autor/Autores	Álvaro González Muñoz
Descripción	Los dispositivos en los que se descarga FridgeApp deben disponer de cámara para escanear el código QR de la "IntelFridge" a la que se quiera conectar.
Necesidad	Obligatorio
Pruebas	Comprobar el correcto funcionamiento de la cámara.

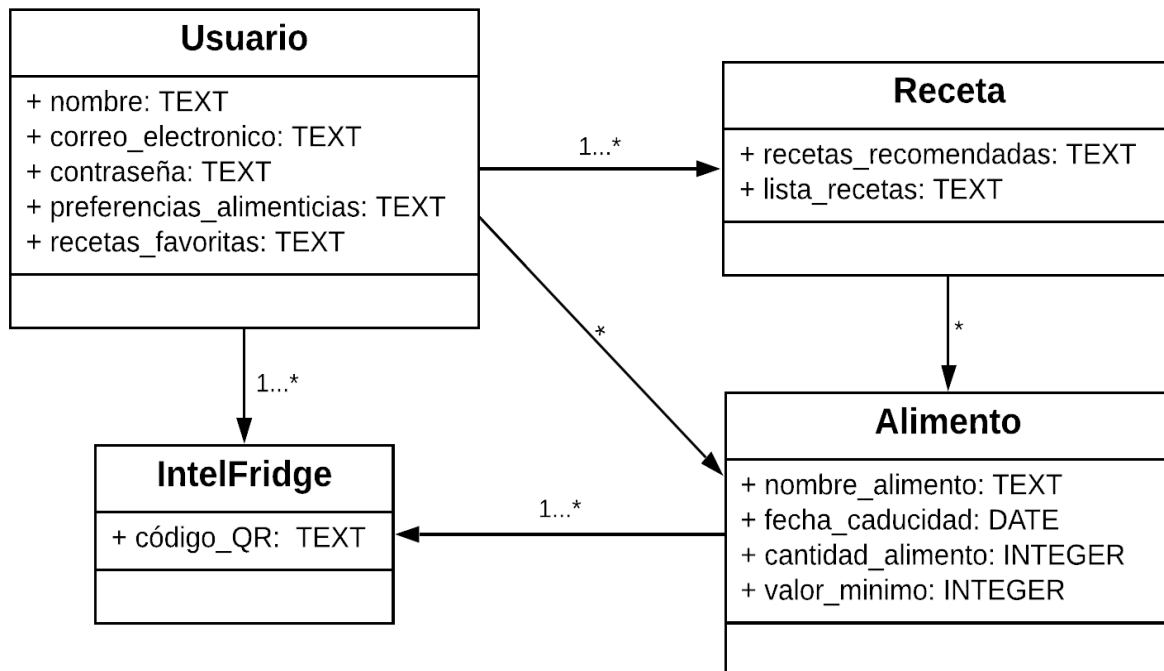
RNF09	Multisesión
Versión	14/10/2019
Autor/Autores	Lucía Pérez Quintanar
Descripción	FridgeApp debe ser capaz de registrar varios "IntelFridge" en un mismo dispositivo.
Necesidad	Obligatorio
Pruebas	Un mismo usuario con la misma cuenta se ha conectado a diferentes "IntelFridge".

RNF10	Multidispositivo
Versión	14/10/2019
Autor/Autores	Lucía Pérez Quintanar
Descripción	Diferentes cuentas de usuario de la aplicación FridgeApp deben poder conectarse a un mismo "IntelFridge".
Necesidad	Obligatorio
Pruebas	Diferentes personas con sus respectivos SmartPhone y cuentas de usuario han podido conectarse al mismo "IntelFridge".

4. Arquitectura

4.1 Modelo de información

Para definir la arquitectura del sistema a implementar, lo primero que haremos será desarrollar un diagrama de clases UML mediante el cual se podrá entender todo el sistema en su conjunto. Más tarde, explicaremos cada una de las clases contenidas en el modelo. Modelado del diagrama: Lucidchart (<https://www.lucidchart.com/>).

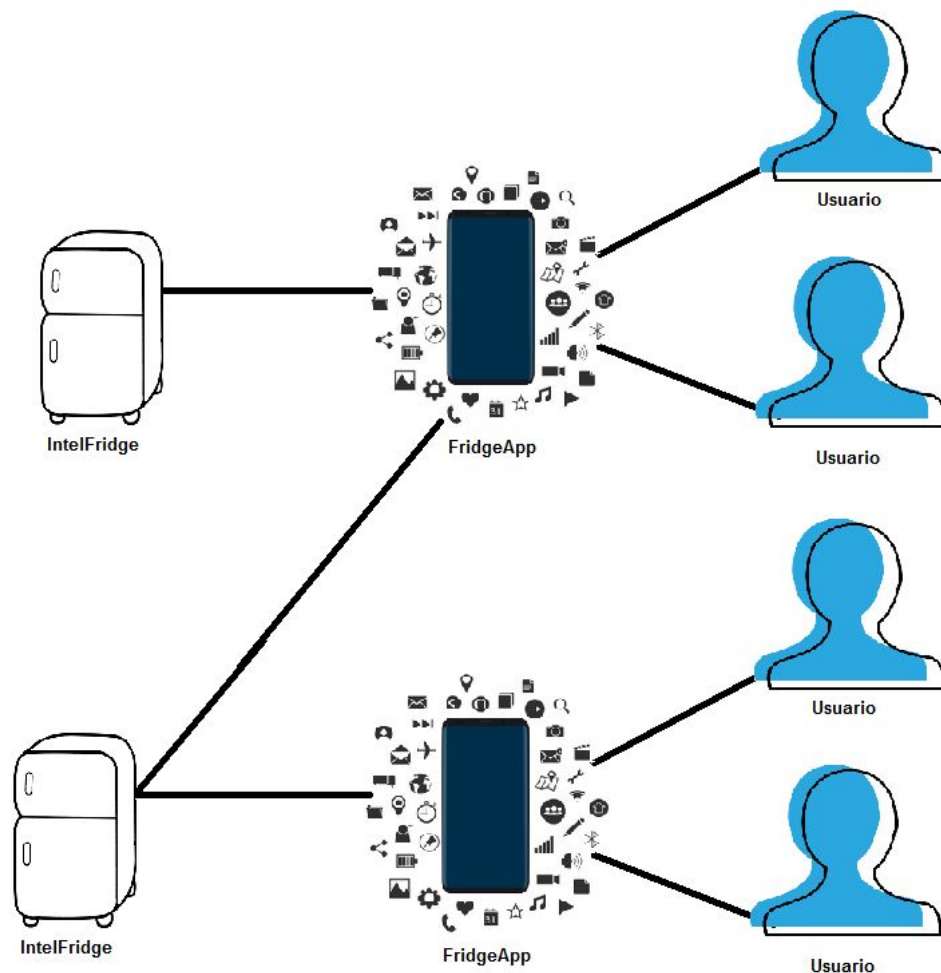


- **Usuario:**
 - Definición: Hace referencia a la cuenta de usuario creada para hacer uso de nuestra aplicación FridgeApp.
 - Atributos:
 - **nombre:** Tipo de dato TEXT que servirá para identificar a cada usuario.
 - **correo_electronico:** Tipo de dato TEXT que hace referencia a la dirección de correo electrónico que se vincula a la cuenta de FridgeApp.
 - **contraseña:** Tipo de dato TEXT que hace referencia al código que es necesario para iniciar sesión en la cuenta de la app.
 - **preferencias_alimenticias:** Tipo de dato TEXT que hace referencia a los alimentos que el usuario ha marcado como “nunca consumidos” o como “poco consumidos”.
 - **Recetas_favoritas:** Tipo de dato TEXT que hace referencia a las recetas que el usuario ha marcado como favoritas.

- **Alimento:**
 - Definición: Hace referencia a los diferentes alimentos que existen.
 - Atributos:
 - alimentos_disponibles: Tipo de dato TEXT que hace referencia a los alimentos que se encuentran disponibles dentro de una nevera IntelFridge.
 - fecha_caducidad: Tipo de dato DATE que hace referencia a las fechas de caducidad de los alimentos, habiendo una por cada alimento disponible en la nevera.
 - cantidad_alimento: Tipo de dato INTEGER que hace referencia al número de unidades de un determinado alimento.
 - valor_minimo: Tipo de dato INTEGER que hace referencia a las cantidades mínimas de cada alimento que el usuario quiere que haya en una nevera IntelFridge.
- **Receta:**
 - Definición: Hace referencia a los distintos apartados de recetas que podremos encontrar en nuestra app.
 - Atributos:
 - Recetas_recomendadas: Tipo de dato TEXT que hace referencia a la lista de recetas que la aplicación FridgeApp debe recomendar al usuario en función de los alimentos que se encuentran en las IntelFridge a las que esté conectada.
 - lista_recetas: Tipo de dato TEXT que hace referencia a la lista completa de recetas que ofrece la aplicación, en las cuales aparecerá un aviso con los ingredientes que faltan para elaborar cada una de ellas.
- **IntelFridge:**
 - Definición: Nevera inteligente que, a través de sus atributos, podrá conectarse y dar funcionalidad a la aplicación FridgeApp.
 - Atributos:
 - codigo_QR: Tipo de dato TEXT que sirve para identificar cada nevera IntelFridge.

4.2 Modelo de implementación

Para nuestro modelo arquitectónico hemos establecido las relaciones básicas que se llevan a cabo entre los diferentes dispositivos implicados en la aplicación y los usuarios, implementando este esquema que se encuentra a continuación.

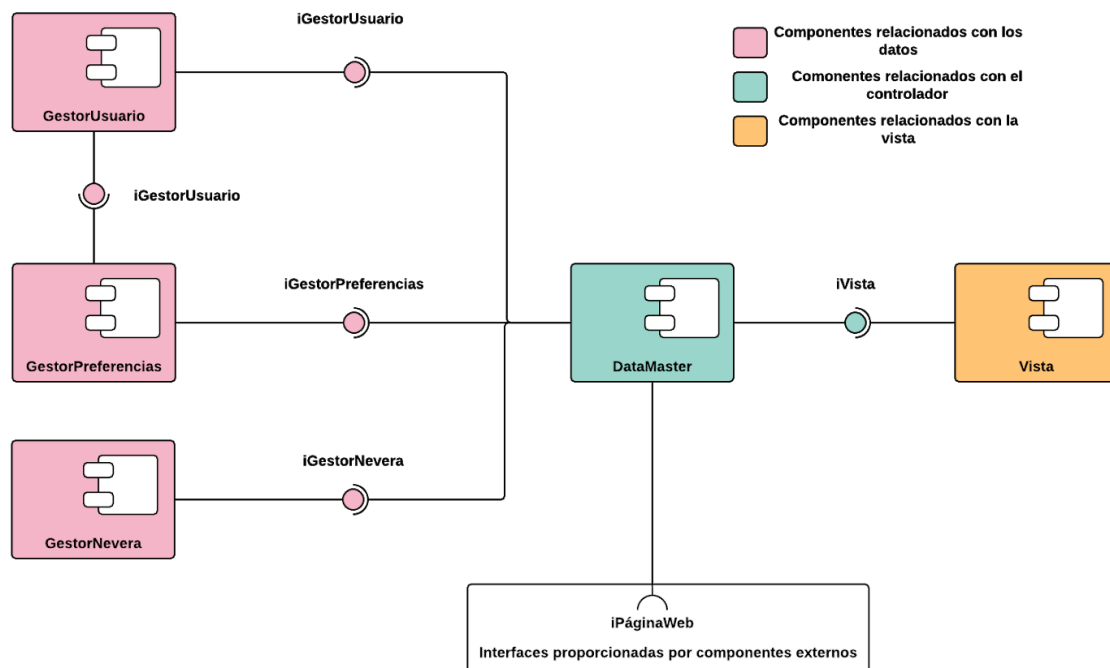


Siguiendo los criterios de simplicidad, extensibilidad, modificabilidad y eficiencia, el modelo de arquitectura que hemos considerado más adecuado ha sido el MVC (Modelo - Vista - Controlador), pues facilita la reutilización de código, además de su mantenimiento y desarrollo. Por otro lado, este modelo permite una mayor independencia entre operaciones realizadas y las vistas de la interfaz, de modo que si en algún momento se requiriera realizar algún tipo de modificación o actualización en algún componente del modelo, se podría realizar de manera individual en el componente afectado, sin tener que tocar ninguno de los otros.

Las interfaces definidas han sido “iGestorUsuario”, “iGestorPreferencias” y “iGestorNevera”, que serán utilizadas para la información relacionada con el usuario, las preferencias alimenticias y las recetas favoritas de cada usuario y las funciones de las neveras IntelFridge respectivamente, y las cuales dan soporte a los componentes “GestorUsuario”, “GestorPreferencias” y “GestorNevera”. “iVista” será otra interfaz que se encargará de mostrar por pantalla los elementos correspondientes, la cual dará soporte al componente “Vista”.

Por último, el componente DataMaster usará los métodos de “iGestorUsuario”, “iGestorPreferencias”, “iGestorNevera” e “iVista”, gestionando su funcionamiento y

actuando dependiendo de lo que ocurra en la aplicación. Modelado del diagrama: Lucidchart (<https://www.lucidchart.com/>)



4.3 Especificación del diseño de componentes

4.3.1 Componente GestorUsuario

Nombre	GestorUsuario	
Tipo	Ejecutable	
Propósito	Se encarga de controlar la creación, modificación y eliminación de cuentas de usuario.	
Dependencias	-	
Interfaces	iGestorUsuario	
Contrato de operaciones	eliminar(usuario)	
	Descripción	El usuario pasado por parámetro será eliminado de la aplicación.
	Precondición	El usuario a eliminar debe existir.
	Postcondición	La cuenta de usuario es eliminada de la aplicación.
	modificar(usuario)	
	Descripción	El usuario pasado por parámetro será modificado en la aplicación.
	Precondición	El usuario a modificar debe existir.

	Postcondición	La cuenta de usuario es modificada según los datos del usuario pasado por parámetros.
	crear(usuario)	
	Descripción	El usuario pasado por parámetro será guardado en la aplicación.
	Precondición	El usuario a guardar no debe existir.
	Postcondición	Se añade a la aplicación un usuario con los datos del usuario pasado por parámetro.

4.3.2 Componente GestorPreferencias

Nombre	GestorPreferencias	
Tipo	Ejecutable	
Propósito	Se encarga de controlar las preferencias alimenticias del usuario y añadir, borrar o consultar recetas favoritas	
Dependencias	iGestorUsuario	
Interfaces	iGestorPreferencias	
Contrato de operaciones	añadir(preferencia)	
	Descripción	La preferencia alimenticia pasada por parámetro será guardada en la aplicación.
	Precondición	El alimento o tipo de alimentos a los que hace referencia la preferencia deben existir.
	Postcondición	Se añade la preferencia alimenticia.
	modificar(preferencia)	
	Descripción	Se modifica la preferencia alimenticia pasada por parámetro y se guarda en la aplicación.
	Precondición	La preferencia debe existir.
	Postcondición	Se guarda la preferencia modificada.
	eliminar(preferencia)	
	Descripción	Se elimina la preferencia alimenticia pasada por parámetro de la aplicación.
	Precondición	La preferencia debe existir.
	Postcondición	La preferencia es eliminada de la aplicación.
	añadir(favorito)	
	Descripción	La receta pasada seleccionada será guardada como favorita en la aplicación.
	Precondición	La receta debe existir y no estar marcada como favorita.
	Postcondición	Se añade la receta a favoritas.

	eliminar(favorito)	
	Descripción	La receta pasada seleccionada será eliminada de favoritas en la aplicación.
	Precondición	La receta debe existir y estar marcada como favorita.
	Postcondición	La receta queda eliminada de favoritas.
	consultarFavorito(favorito)	
	Descripción	Se muestra una lista con las recetas marcadas como favoritas.
	Precondición	La receta debe existir y estar marcada como favorita.
	Postcondición	Las recetas quedan mostradas al usuario.
	consultarReceta(receta)	
	Descripción	Se muestra una lista con todas las recetas disponibles.
	Precondición	La receta debe de existir.
	Postcondición	Las recetas quedan mostradas al usuario.

4.3.3 Componente GestorNevera

Nombre	GestorNevera	
Tipo	Ejecutable	
Propósito	Se encarga de gestionar la información recibida por el sensor de la nevera, tanto la cantidad de alimentos que hay en su interior como controlar el tiempo que lleva un alimento en su interior.	
Dependencias	-	
Interfaces	iGestorNevera	
Contrato de operaciones	actualizar(alimento)	
	Descripción	Cuando se introduzca o se retire un alimento en la nevera IntelFridge, se actualizan los valores de dicho alimento en la aplicación.
	Precondición	El alimento debe existir.
	Postcondición	La cantidad del alimento queda actualizada en la aplicación.
	mostrarCaducidad(alimento)	
	Descripción	Muestra la fecha de caducidad del alimento.
	Precondición	El alimento debe tener fecha de caducidad.
	Postcondición	Se muestra la fecha de caducidad.

	avisarCaducidad(alimento)	
	Descripción	Compara la fecha de caducidad del alimento con la fecha de entrada en la nevera y si caduca en menos de 2 días, avisa al usuario mediante una notificación indicando que el alimento va a caducar y cuando va a caducar.
	Precondición	El alimento debe existir, debe de tener fecha de caducidad.
	Postcondición	Se envía una notificación al usuario indicando el alimento que va a caducar.
	establecerMínimo(alimento)	
	Descripción	Se selecciona un alimento y se determina la cantidad mínima de ese alimento.
	Precondición	El alimento debe de existir.
	Postcondición	Se guarda la cantidad mínima en la aplicación.
	avisarEscasez(alimento)	
	Descripción	Se avisará al usuario mediante una notificación si se excede la cantidad mínima. Se especificará qué alimento ha llegado al mínimo.
	Precondición	El alimento debe de existir.
	Postcondición	Se produce el aviso de escasez del alimento.
	eliminarMínimo(alimento)	
	Descripción	Se elimina la cantidad mínima que debe haber de ese alimento.
	Precondición	El alimento debe de existir, el mínimo de alimento debe de existir.
	Postcondición	Se elimina la cantidad mínima de la aplicación.

4.3.4 Componente DataMaster

Nombre	DataMaster
Tipo	Controlador
Propósito	Se encarga de decidir el comportamiento de la aplicación en base a una serie de eventos o condiciones.
Dependencias	iGestorUsuario, iGestorPreferencias, iGestorNevera, iVista
Interfaces	iGestorUsuario, iGestorPreferencias, iGestorNevera, iVista

4.3.5 Componente Vista

Nombre	Vista	
Tipo	Ejecutable	
Propósito	Se encarga de controlar aquello que se muestra y cómo se muestra.	
Dependencias	-	
Interfaces	iVista	
Contrato de operaciones	mostar(pantalla)	
	Descripción	Muestra en la interfaz de la aplicación lo que pasa en el controlador.
	Precondición	Hay algo que mostrar.
	Postcondición	Se muestra correctamente lo que pasa en el controlador.

5. Lista y tablas

5.1 Consistencia entre los requisitos: conflictos, redundancias, acoplamientos

El análisis de la consistencia entre requisitos ayuda a poder visualizar las diferentes relaciones existentes entre los requisitos declarados anteriormente.

Para poder analizar las relaciones se utiliza una tabla que declara 3 tipos de relaciones entre los requisitos, la relación de conflicto, cuando dos requisitos se contradicen, la relación de redundancia, cuando dos requisitos significan lo mismo, y por último la relación de acoplamiento, que nos indicará cuando un requisito depende de otro o añade información.

Requisito	Conflicto	Redundancia	Acoplamiento
RF01	-	-	RF02, RF07, RF09
RF02	-	-	RF01, RF07
RF03	-	-	RF01, RF04, RF05, RF10, RF11, RF13, RF15, RF16
RF04	-	-	RF03, RF05, RF10, RF11, RF13, RF15
RF05	-	-	RF03, RF04, RF06, RF10, RF11, RF13, RF15, RF16
RF06	-	-	RF03, RF04, RF05
RF07	-	-	RF01, RF02, RF12, RF07, RF06, RF14

RF08	-	-	RF01, RF02, RF03, RF05, RF14
RF09	-	-	RF01, RF08, RF14
RF10	-	-	RF03, RF04, RF11
RF11	-	-	RF10, RF03, RF04, RF17, RF18, RF12, RF13, RF15, RF16
RF12	-	-	RF14, RF01, RF17, RF18, RF09, RF10
RF13	-	-	RF18, RF17, RF03, RF15, RF16, RF04, RF05, RF10, RF11
RF14	-	-	RF01, RF02, RF07, RF08, RF09, RF12
RF15	-	-	RF13, RF16, RF10, RF17, RF18, RF03, RF04, RF05
RF16	-	-	RF13, RF15, RF03, RF04, RF05
RF17	-	-	RF18, RF13, RF15, RF16
RF18	-	-	RF17, RF13, RF15, RF16, RF03, RF04, RF05
RNF01	-	-	-
RNF02	-	-	-
RNF03	-	-	-
RNF04	-	-	-
RNF05	-	-	-
RNF06	-	-	-
RNF07	-	-	-
RNF08	-	-	-
RNF09	-	-	-
RNF10	-	-	-

5.2 Trazabilidad requisitos – modelo de información (clases)

En la siguiente tabla se mostrará la dependencia entre los requisitos funcionales y los componentes descritos en el modelo de información.

Requisito	Cuenta usuario	FridgeApp	Recetas	IntelFridge	Alimentos
RF01					
RF02					
RF03					
RF04					
RF05					
RF06					
RF07					
RF08					
RF09					
RF10					
RF11					
RF12					
RF13					
RF14					
RF15					
RF16					
RF17					
RF18					

5.3 Trazabilidad requisitos – modelo de implementación (componentes)

En la siguiente tabla se mostrará la dependencia entre los requisitos funcionales y los componentes descritos en el modelo de implementación.

Requisito	GestorDatos	GestorPreferencias	GestorNevera	Datamaster	Vista
RF01					
RF02					
RF03					
RF04					
RF05					
RF06					
RF07					
RF08					
RF09					
RF10					
RF11					
RF12					
RF13					
RF14					
RF15					
RF16					
RF17					
RF18				-	

Anexo I. Revisiones recibidas

Primera revisión

1. Aspectos positivos que aporta el documento:

Para comenzar, consideramos que visualmente es un documento bien estructurado, que nos permite hacer una primera división de conceptos a tratar. Los títulos están bien destacados y las subdivisiones son correctas.

Las tablas insertadas contienen la información necesaria y bien explicada, son conceptuales y de acorde a los colores iniciales escogidos.

La descripción de los requisitos es bastante concreta y está bien explicada, a su vez, se acompaña de unas pruebas correctas y variadas, para abarcar un mayor rango de posibilidades.

La tabla de acoplamientos es extensa y muestra bien las relaciones, se ha intentado reflejar la mayor información posible en ella de manera satisfactoria.

Enumerar los tipos de vocabulario hacen que sea mucho más sencillo para el lector, a la vez que es una forma práctica para hacer una enumeración.

La extensión también es la adecuada, ya que no es ni demasiado larga, ni demasiado corta. Aporta la información necesaria sin extenderse demasiado.

2. Consideraciones generales sobre el formato del documento.

En general, la presentación es muy positiva; está bien escrito, es legible, y entendible. Sin embargo, hay algunos errores de escritura que creemos que son importantes a la par que fáciles de corregir. Dichos errores, tanto de ortografía como de coherencia o entendimiento son los siguientes:

p.7, abrirla, y además >>> abrirla y, además,

p. 8, que >>> qué

p.8, Además el usuario >>> Además, el usuario

En la página 8, en el primer punto del apartado 2.3 , “tendrá que poder escanear los alimentos que se encuentran en su interior y la caducidad de los mismos.”. Consideramos que faltaría un verbo para especificar qué ocurre con “la caducidad de los mismos”, como por ejemplo, “... y determinar la caducidad de los mismos. “

En la página 10, encontramos redundancia e incoherencia en “Resuelve problemas éticos respecto al consumo de ciertos alimentos. Dado que la aplicación tendrá la opción de eliminar ciertos alimentos de las dietas.” Creemos que el punto de separación entre ambas oraciones sobra; lo sustituiríamos por una coma; y cambiaríamos la segunda vez que aparece “ciertos alimentos” ya que se encuentran demasiado seguidos.

p.10, Por ejemplo se podrán... >>> Por ejemplo, se podrán...

p.10, Dispositivo celular móvil que entre muchas funciones permite ... >>>

Dispositivo celular móvil que, entre muchas funciones, permite...

p.13, Probar a crear >>> Crear

En la página 13, en el requisito con id = RF04, “Al cambiar el nombre de usuario, poner un nombre ya existente no puede permitirse.” Falta signos de puntuación en esta oración para poder entenderla completamente.

p.14, “Borrar la cuenta de usuario.” No nos queda claro de qué usuario se trata. Con añadir el artículo “un”, se resuelve.

p.16, de el alimento >>> del alimento

En la página 18, pruebas del requisito RF13, no quedan claras las pruebas realizadas. “Hacer lo necesario para ponerla como favorita.” . Faltaría especificar, qué es lo necesario para poner una receta como favorita.

A lo largo del documento, los términos “IntelFridge” y “FridgeApp”, aparecen tanto entrecomillados como no entrecomillados, y no nos queda claro el propósito o importancia de las comillas.

3. Consideraciones individuales sobre la adecuación al índice de la práctica.

3.1. Estructura del documento.

El documento se adapta a la estructura requerida. La primera parte es la menos extensa de todo el documento, como es lógico, aunque, a su vez, consideramos que el propósito podría ser un poco más detallado, y las referencias más variadas. La visión general está bien estructurada, bien numerada y se distinguen bien los distintos apartados, además de estar completa.

La segunda parte necesita que haya bastante contenido teórico, ya que es donde se debe escribir los distintos aspectos del producto. No tiene por qué haber una relación directa, pero mayormente mientras más texto haya, más detallada estará la descripción si se hace correctamente.

El apartado de los requisitos abarca la mayoría del documento, ya entraremos con más detalle en los errores principales, pero a grandes rasgos es bastante completo.

Por último, la tabla del apartado cinco está completa en su mayoría y cumple con lo exigido.

3.2. Contenido del documento.

Siguiendo la línea del apartado anterior, podemos afirmar que cada apartado cumple con su finalidad, siempre y cuando, tengamos en cuenta en qué pueden solaparse algunos apartados entre sí.

No obstante, en algunos de los apartados encontramos algún que otro error que se podría corregir. Mencionaremos dichos apartados, con lo cual, los que no hayan sido nombrados, opinamos que están correctamente y sin ningún fallo digno de enmendar.

Como ya hemos dicho, el propósito del documento es una introducción al proyecto, pero puede ser más extensa. Por otro lado, no queda bien definido un aspecto:

→ “Especificar el perfil del lector al que va dirigido el documento.”

En el apartado 2.3 ha faltado lo siguiente:

→ “También deberá describir el proceso al que el software dará soporte: la descripción de este proceso justifica y hace comprensibles los requisitos que más tarde se especificarán de modo completo, y permite realizar un análisis crítico de los mismos.”

En el apartado 2.5 Características de los usuarios: roles y capacidades, la tabla se entiende correctamente y los roles contemplados son acertados, sin embargo, falta decir algo sobre el tipo general de interfaz; si es una interfaz clásica de ordenador o si requiere un terminal específico.

El apartado 2.6, a pesar de ser entendible por la manera en la que está redactado, consideramos que un apoyo visual mejoraría la comprensión plena del entorno operacional, ya sea con un diagrama de contexto, para resumir interfaces externas, o un diagrama de actividad para mostrar cómo se integra la actividad externa con el sistema informático.

Por último, la parte 5.1 está completa, aunque se pueden apreciar muchas partes en blanco en las relaciones, y si nos fijamos, el enunciado explicaba “ La tabla de doble entrada tiene el peligro de ser excesivamente dispersa, por lo que a menudo las tablas simplificadas de pocas columnas son una solución más conveniente ”, por lo que debería mostrar menos relaciones.

3.3. Explicación con más detalle los errores principales.

El principal error que encontramos en el apartado de consistencias de requisitos, ya que como podemos observar en la tabla, la mayoría de requisitos no se acoplan con otros requisitos. Es decir, en el desarrollo de requisitos es muy complicado que no estén acoplados unos con otros.

El identificador del requisito debería ser sólo un número entero, no debería incluir “RF” o RNF”.

En el requisito RF01, las pruebas no parecen pruebas, parecen, de nuevo, la descripción del requisito.

Convendría añadir en la plantilla de requisitos un campo que indique el estado del requisito, pudiendo ser: pendiente, validado o cancelado.

Consideramos, que en la tabla de consistencia entre requisitos, existe una relación de redundancia entre RFN05 y RFN07.

A continuación pondremos una serie de opciones que se nos han ocurrido que se podrían tener en cuenta para la aplicación:

En el documento, página 18, pruebas del RF11, pone lo siguiente:

- Buscar en FridgeApp una sección donde aparezcan recetas disponibles.
- Comprobar que todos los ingredientes necesarios para elaborarla están en IntelFridge.

Por lo tanto, entendemos que para las recetas sólo tendría en cuenta los alimentos que contiene en la nevera, no aquellos que estén en la despensa. Estaría bien que se tuviera en cuenta esos alimentos de alguna manera.

No se menciona nada del almacenaje de la información que se maneja en la app, estaría bien incluir bases de datos en los requisitos no funcionales.

Estaría bien que se consideraran diferentes roles en la familia, es decir, que si un miembro de la familia fuera vegetariano, vegano o fuera intolerante a la lactosa y los demás no, ofreciera recetas adaptadas para esa persona.

Segunda revisión

1. Aspectos positivos que aporta el documento:

En este apartado comentaremos los aspectos positivos que aporta la segunda parte del documento, es decir, a partir del punto 4, ya que lo anterior ya fue revisado.

Para comenzar, consideramos que visualmente es un documento bien estructurado, que nos permite hacer una primera división de conceptos a tratar. Los títulos están bien destacados y las subdivisiones son correctas.

Los esquemas tanto del modelo conceptual como del modelo de implementación son visualmente claros y agradables, sin muchas flechas, lo cual haría del esquema un esquema confuso, y con colores, que aporta comodidad al lector.

La subdivisión de cada clase del modelo conceptual es muy clara y apropiada. Cada clase está bien explicada con sus diferentes apartados de definición y atributos.

Las tablas insertadas de la definición de cada componente contienen la información necesaria y bien explicada, no se olvidan ni del tipo, ni del propósito, de las interfaces ni del contrato de operaciones; son conceptuales y de acorde a los colores iniciales escogidos.

La tabla de acoplamientos es extensa y muestra bien las relaciones, se ha intentado reflejar la mayor información posible en ella de manera satisfactoria.

La extensión también es la adecuada, ya que no es ni demasiado larga, ni demasiado corta. Aporta la información necesaria sin extenderse demasiado.

2. Consideraciones generales sobre el formato del documento.

En general, la presentación es muy positiva; está bien escrito, es legible, y entendible. Sin embargo, hay algunos errores de escritura que creemos que son importantes a la par que fáciles de corregir. Dichos errores, tanto de ortografía como de coherencia o entendimiento son los siguientes:

- p.25: lo primero que haremos, será...>>> lo primero que haremos será...
- p.26 en definición de IntelFridge: Nevera inteligente que a través de sus atributos podrá... >>> Nevera inteligente que, a través de sus atributos, podrá...
- p.29 en la precondición de añadir(preferencia): El usuario debe de existir >>> creemos que se refieren a que "la preferencia debe existir"
- Lo mismo ocurre en eliminar(favorito), en consultar(favorito) y en añadir(favorito); la precondición no está clara, o no se corresponde con la operación.

- p.30 en la descripción de actualizar (alimento): Cuando se introduzca o se retire un alimento en la nevera IntelFridge se actualizan >>> Cuando se introduzca o se retire un alimento en la nevera IntelFridge, se actualizan

3. Consideraciones individuales sobre la adecuación al índice de la práctica.

3.1. Estructura del documento.

El documento se adapta a la estructura requerida.

La parte 4 contiene los apartados requeridos en el índice de la práctica: el modelo de información, el modelo de implementación y la especificación del diseño de componentes (cada uno con su propósito, dependencias, interfaces y contrato de operaciones). La visión general está bien estructurada, bien numerada y se distinguen bien los distintos apartados, además de estar completa.

Por último, la parte 5, contiene las tres tablas o listas requeridas: 5.1 Consistencia entre los requisitos: conflictos, redundancias, acoplamientos; 5.2 Trazabilidad requisitos – modelo de información (clases) y 5.3 Trazabilidad requisitos – modelo de implementación (componentes).

3.2. Contenido del documento.

Siguiendo la línea del apartado anterior, podemos afirmar que cada apartado cumple con su finalidad.

Si también es cierto que en algunos de los apartados encontramos algún que otro error que se podría corregir. Mencionaremos dichos apartados en el siguiente punto, 3.3 Explicación con más detalle de los errores principales; con lo cual, los que no hayan sido nombrados, opinamos que están correctamente y sin ningún fallo digno de enmendar.

3.3. Explicación con más detalle de los errores principales.

Los principales errores que hemos encontrado son los siguientes:

- En el contrato de operaciones del componente gestorPreferencias, hay un método consultarFavoritos(favorito) con el que se puede buscar aquellas recetas que el usuario haya indicado como favoritas. Sería conveniente que se añadiera un método que pudiera buscar cualquier receta no solo poder buscar aquellas recetas que tienes asignadas como favoritas.
- En el diagrama de componentes se debería considerar añadir un componente llamado "Nevera" que pueda recibir aquellos datos que se almacenan en el gestorNevera. Es decir, cuando por ejemplo se actualice un alimento en la nevera, método actualizarAlimento(alimento), o actualizarCantidad(alimento, num) si añadimos, por ejemplo, un plátano nuevo; y dicho componente reciba los datos.
- En el contrato de operaciones del componente gestorUsuario, en la operación crear(usuario), la precondition "El usuario a guardar debe existir" debería ser "El usuario a guardar no debe existir".
- En el componente gestorNevera no aparecen las dependencias de dicho componente.
- En el modelo de información, en la clase "Alimentos" se incluyen las preferencias alimenticias, creemos que ese atributo debería pertenecer a la clase "Usuario". A su vez, creemos que estaría bien añadir en la clase "Alimentos" un atributo sobre familias de alimentos, como, por ejemplo, frutas, verduras...
- Por último, tenemos una sugerencia, no un error, que creemos que facilitaría el entendimiento del sistema y que podríais incluir. Sería añadir una clase "Recetas", con su función "consultarRecetas" y demás, para cuando quieras buscar una receta, saber dónde encontrarla, porque si no, ¿cómo funciona esa parte?

Anexo II. Respuesta a las revisiones recibidas

Primera revisión

Algunas de las sugerencias de corrección propuestas por nuestros compañeros no han sido tomadas en cuenta a la hora de la elaboración de la versión definitiva de la práctica. Los motivos de nuestra decisión han sido los siguientes:

1. **El identificador del requisito debería ser sólo un número entero, no debería incluir “RF” o RNF”.**

Dejamos el nombre de los requisitos tal y como los habíamos llamado originalmente debido a que los requisitos funcionales y no funcionales, al estar precedidos de las letras RF y RNF respectivamente, creemos que son más fáciles de identificar y distinguir, sobre todo en lo que a la tabla de consistencia de requisitos se refiere.

2. **Convendría añadir en la plantilla de requisitos un campo que indique el estado del requisito, pudiendo ser: pendiente, validado o cancelado.**

Consideramos que únicamente el campo obligatorio u opcional es necesario.

3. **En el apartado 2.5 Características de los usuarios: roles y capacidades, la tabla se entiende correctamente y los roles contemplados son acertados, sin embargo, falta decir algo sobre el tipo general de interfaz; si es una interfaz clásica de ordenador o si requiere un terminal específico.**

Consideramos que el apartado 2.5 está completo tal cual está, ya que exponemos lo que se pide: Los roles y las capacidades de cada rol.

4. **En el documento, página 18, pruebas del RF11, pone lo siguiente:**

- **Buscar en FridgeApp una sección donde aparezcan recetas disponibles.**
- **Comprobar que todos los ingredientes necesarios para elaborarla están en IntelFridge.**

Por lo tanto, entendemos que para las recetas sólo tendría en cuenta los alimentos que contiene en la nevera, no aquellos que estén en la despensa. Estaría bien que se tuviera en cuenta esos alimentos de alguna manera.

FridgeApp recomienda recetas según las preferencias configuradas por el usuario, indicando qué alimentos faltarían en la nevera para realizarlas. Consideramos este hecho suficiente y por ello no seguimos la sugerencia dada.

5. **No se menciona nada del almacenaje de la información que se maneja en la app, estaría bien incluir bases de datos en los requisitos no funcionales.**

Aunque no seguimos esta sugerencia en su totalidad, sí que vimos necesario añadir algún lugar desde donde la app pudiera acceder a información. Por tanto, decidimos que FridgeApp descargara recetas desde una página web.

6. **Estaría bien que se consideraran diferentes roles en la familia, es decir, que si un miembro de la familia fuera vegetariano, vegano o fuera intolerante a la lactosa y los demás no, ofreciera recetas adaptadas para esa persona.**

No hemos seguido esta propuesta debido a que un mismo IntelFridge puede conectarse a más de un dispositivo móvil que tenga FridgeApp, por lo que todo miembro de la familia que tenga descargada la app puede configurar los alimentos que no quiere consumir, sin influir las preferencias de ninguno de los integrantes de la familia a los otros. Desde ese momento, toda receta que se le sugiera a cada miembro de la familia estará adaptada a su configuración de alimentos personal.

Segunda revisión

Hemos corregido todos los errores destacados por nuestros compañeros, e incluso implementado sugerencias propuestas por estos mismos. Es por esto que en la segunda revisión no podemos destacar ninguna corrección no tomada.

Anexo III. Revisiones enviadas

Primera revisión

1. Aspectos positivos que aporta el documento.

- Desarrolla correctamente el propósito general de la aplicación y su funcionalidad principal.
- La plantilla empleada para definir los requisitos es clara y concisa y contiene los atributos esenciales para que los requisitos puedan entenderse de forma correcta.
- La estructura del documento es clara y presenta apartados específicos para el contenido que se desarrolla en cada uno de ellos.
- En el entorno operacional aporta un esquema que hace mucho más visual la interacción entre el usuario, la nevera, la base de datos y la aplicación.

2. Consideraciones generales sobre el formato del documento.

2.1 Errores generales.

2.1.1. Página 9, Apartado 2.6 Entorno operacional. En el esquema de este apartado se muestra que la nevera detecta cambios y se comunica con la base de datos para actualizarla. La aplicación es la que se debería de comunicar con la base de datos. Por lo tanto la nevera no se debe de comunicar con la base de datos.

2.1.2. Página 9, Apartado 2.7 Consideraciones éticas. Una de las consideraciones éticas que presentan es la elección de no representar las marcas de ningún producto. Más tarde en el requisito RF-07 se habla sobre organizar los productos según su marca.

2.1.3. Página 9, Apartado 3.1.1 Vocabulario del modelo de información. El vocabulario que se describe está incompleto. Este apartado define todos los términos significativos y específicos del problema que aparecen en los requisitos. Se debería añadir el término “nevera inteligente” o “nevera”, los cuáles aparecen muchas veces. Además sería necesario usar tan solo uno de los dos, ya que significan lo mismo en el contexto de la aplicación.

2.1.4. Página 9, Apartado 3.1.1 Vocabulario del modelo de información. Se habla sobre carencias nutricionales, pero se podrían añadir excesos nutricionales, ya que se habla sobre ello en el RF-22.

2.1.5. Página 10, Apartado 3.1.2 Vocabulario técnico. Consta de términos típicamente referidos al diseño o a las tecnologías empleadas en el sistema. Encontramos el vocabulario escaso. Faltarían términos como software, smartphone, Wi-Fi, iOS o android.

2.1.6. Página 11, Apartado 3.4.1 Requisitos centrados en la conectividad. Requisito con título RF-01: “Conectividad con el frigorífico”. La prueba realizada a la hora de comprobar la efectividad de este requisito es demasiado general. Debería ser más específica,

comprobando si la aplicación funciona de la manera esperada si se introduce en la nevera alimento en específico, como una manzana.

2.1.7. Página 12, Apartado 3.4.1 Requisitos centrados en el almacenamiento del inventario. Requisito con título RF-03: “Almacenamiento del inventario”. La prueba realizada a la hora de comprobar la efectividad de este requisito es demasiado general. Debería detallarse más exhaustivamente los pasos seguidos a la hora de comprobar los productos de la nevera, por ejemplo, introduciendo un cartón de leche y comprobando entonces si la aplicación se ha actualizado correctamente.

2.1.8. Página 12, Apartado 3.4.1 Requisitos centrados en el almacenamiento del inventario. Requisito con título RF-04: “Organización del inventario”. La prueba realizada a la hora de comprobar la efectividad de este requisito es demasiado general. Una sugerencia sería especificar qué opción elige el usuario y en base a ello, se tendrían que realizar más pruebas.

2.1.9. Página 13, Apartado 3.4.1 Requisitos centrados en el almacenamiento del inventario. Requisito con título RF-06: “Organización cronológica del inventario”. La prueba realizada a la hora de comprobar la efectividad de este requisito es demasiado general. Tal y como ocurría en el punto 2.1.8., una sugerencia sería especificar la opción elegida por el usuario.

2.2 Errores de ortografía y gramaticales.

2.2.1. Página 5, destinario >>> destinatario.

2.2.2. Página 5, donde ser recogen >>> donde se recogen.

2.2.3. Página 7, acepta futura actualizaciones >>> acepta futuras actualizaciones.

2.2.4. Página 7, funcionalidad las cuales, si tienen aplicaciones parecidas >>> funcionalidad que sí tienen aplicaciones parecidas.

2.2.5. Página 7, Notificara >>> Notificará.

2.2.6. Página 7, aconsejara >>> aconsejará.

2.2.7. Página 8, notificara >>> notificará.

2.2.8. Página 8, facilitara >>> facilitará.

2.2.9. Página 8, funcionara >>> funcionará.

2.2.10. Página 8, necesitara >>> necesitará.

2.2.11. Página 8, funcionara >>> funcionará.

2.2.12. Página 8, encargara >>> encargará.

2.2.13. Página 9, acarren >>> acarreen.

2.2.14. Página 10, las distintas variable >>> las distintas variables.

- 2.2.15.** Página 10, en cuanto a requisitos refiere >>> en cuanto a requisitos se refiere.
- 2.2.16.** Página 11, tener conectados varias neveras >>> tener conectadas varias neveras.
- 2.2.17.** Página 14, al seleccionar la dieta aplicación es capaz >>> al seleccionar la dieta la aplicación es capaz.
- 2.2.18.** Página 15, la almacena la destaca y añade >>> la almacena, la destaca y la añade.
- 2.2.19.** Página 15, a almacenado >>> ha almacenado.
- 2.2.20.** Página 15, mandara >>> mandará.
- 2.2.21.** Página 16, Se introducirán un determinado número >>> Se introducirá un determinado número.
- 2.2.22.** Página 16, en función al inventario >>> en función del inventario.
- 2.2.22.** Página 18, tener implementada las funcionalidades >>> tener implementadas las funcionalidades.

2.3 Presentación y legibilidad

La presentación es, por lo general, muy correcta, adecuándose al formato y tipo de texto presentado.

En lo que respecta a la legibilidad, existen apartados en los que no es del todo legible lo que está redactado. Los errores observados se citarán a continuación:

- 2.3.1.** En el apartado 3.3 justificación de la plantilla de requisitos, el ID debería de ser el primer campo de la tabla para un reconocimiento más inmediato de cada requisito. Ya que este campo es el más identificativo. Por esto creemos que la mejor distribución para la plantilla sería id, título, fecha, autor, descripción, importancia, pruebas y precondiciones. Siguiendo ese orden.
- 2.3.2.** En el apartado 3.4.1, en el requisito RF-01 se repite en el mismo párrafo la expresión “en todo momento”, lo cual es totalmente innecesario.
- 2.3.3.** En el apartado 3.4.2, en el requisito RF-03, la palabra “esta” tal y como está expresada, produce ambigüedad al no saber con certeza si se refiere a la aplicación o a la nevera.
- 2.3.4.** En el apartado 3.5.2, en el requisito RNF-03 hay un error de sintaxis, en la frase “error operativos que admite la aplicación, comprobará la correcta”. La forma correcta sería aquella con “y” o “.”.

3. Consideraciones individuales sobre adecuación al índice de la práctica.

3.1 Estructura del documento: ¿sigue los apartados del índice de la práctica?

Sí, la estructura del documento del equipo G1 cumple de forma satisfactoria con el índice de la práctica. Todos los puntos están incluidos y además se disponen en el orden propuesto.

3.2 Contenido del documento: ¿se corresponde cada uno de los apartados con la finalidad que el índice propone?

Sí, el contenido de la práctica corresponde con lo descrito en los apartados del índice.

3.3 Explicación en más detalle de los errores principales

3.3.1. En el apartado 2.5 características de los usuarios: roles y características, solo es necesario un usuario. Aunque un usuario ocasional no utilicen todas las funcionalidades de la aplicación, el usuario ocasional y usuario habitual pueden acceder a las mismas funciones.

3.3.2. En el apartado 2.5 características de los usuarios: roles y características, hace falta un rol de administrador que introduzca recetas y dietas.

3.3.3. En el apartado 3.2 justificación de la clasificación de los requisitos. Se debería pasar la clasificación de los requisitos funcionales a un esquema. Es más fácil de visualizar y se añade el nombre completo.

- Requisitos centrados en la conectividad.
- Requisitos centrados en el almacenamiento del inventario.
- Requisitos centrados en las dietas presentes en la aplicación.
- Requisitos centrados en avisos al usuario.
- Requisitos centrados en las recetas.
- Requisitos centrados en las recomendaciones al usuario.
- Requisitos centrados en el mantenimiento de la aplicación.

3.3.4. En el apartado 3.2 justificación de la clasificación de los requisitos. Se debería añadir una clasificación para los requisitos no funcionales.

3.3.5. En las restricciones generales se dice: “La aplicación funcionará para versiones Android 5.0 o superior, iOS 9.0 o superior.” pero más tarde en el requisito RNF-02 Se habla sobre: “La aplicación podrá ejecutarse en dispositivos Android 4.0 o superior y iOS 7 o superior.”

3.3.6. En las restricciones generales se dice: “Windows 7 o superior” pero luego en el requisito RNF-03 se dice: “La aplicación podrá ejecutarse en ordenadores cuyo sistema operativo sea Windows 8 o superior.”

3.3.7. En el requisito RNF-03 se dice: “La aplicación podrá ejecutarse en ordenadores cuyo sistema operativo sea Windows 8 o superior, y macOS.” pero en las restricciones generales no se dice nada sobre que la aplicación deba funcionar en macOS.

3.3.8. RF-07 descripción mal redactada. Se podría reemplazar por: Se añadirá una serie de productos de diferentes marcas a una nevera inteligente y se comprobará si la aplicación es capaz de organizar los productos según su marca.

3.3.9. RF-22 descripción mal redactada. Se podría reemplazar por: Se analiza el contenido nutricional del inventario actual de la nevera y se recomiendan productos que mejoren el valor nutricional del inventario.

3.3.10. Acoplamiento de RF-08 con RF-01. Consideramos que dicho acoplamiento no se da, ya que aunque se necesite la conexión a internet para almacenar las dietas, se trata de dos cuestiones muy diferenciadas.

3.3.11. Acoplamiento de RF-13 con RF-01. Consideramos que este acoplamiento tampoco se da, ya que de forma similar al punto 3.3.10, aunque se necesite conexión a internet para determinar si hay escasez de algún alimento, se trata de dos cuestiones muy diferenciadas.

Segunda revisión

1. Aspectos positivos que aporta el documento.

- Visualmente es un documento limpio y ordenado, que facilita su corrección al estar adecuadamente dividido en apartados bien diferenciados.
- El esquema del modelo de implementación es conciso y claro, facilitando la visualización de los componentes.
- Las clases que forman el esquema del modelo conceptual están correctamente explicadas y justificadas al tener diversos apartados en los que se especifican sus características y atributos.
- La especificación del diseño de componentes contiene en cada tabla la información necesaria para justificar el modelo conceptual elegido.
- La tabla de acoplamientos es clara y reconocible, facilitando su comprensión de forma correcta.

2. Consideraciones generales sobre el formato del documento.

2.1 Errores generales.

2.1.1. Si se usan las comillas para alguna palabra, cómo “Fridgeapp”, se deben de utilizar a lo largo del documento.

2.1.2. Página 7. Intercalan el uso de comillas dobles y comillas simples. ‘smartphones’ >>> “smartphones” y ‘APP’ >>> “APP”

2.1.3. Apartado 4.3 Especificación del diseño de componentes. Creemos que tanto las descripciones, precondiciones y postcondiciones deben de ser más concisas.

2.1.4. Apartado 4.3 Especificación del diseño de componentes. Las postcondiciones describen el resultado obtenido al finalizar la operación. En la mayoría de contratos

(especialmente “Gestor_Información” y “Gestor_Notificaciones”), las postcondiciones indican procesos que se llevan a cabo durante la operación.

2.2 Errores de ortografía y gramaticales.

2.2.1. Página 25, índica >>> indica.

2.2.2. Página 25, en casa de estar cerca >>> en caso de estar cerca.

2.2.3. Página 25, función asociada_ >>> función asociada_.

2.2.4. Página 25, que le permite >>> que le permita.

2.2.5. Página 25, En cuanto a la clase alimento recolectaremos >>> En cuanto a la clase alimento, recolectaremos.

2.2.6. Página 26, actuales del inventario utilizaremos >>> actuales del inventario, utilizaremos.

2.2.7. Página 26, tiempo predefinido_ >>> tiempo predefinido_.

2.2.8. Página 27, es la base de datos_su función principal >>> es la base de datos_Su función principal.

2.2.9. Página 27, En el caso de los componentes hemos >>> En el caso de los componentes, hemos.

2.2.10. Página 28, Recibe una dieta la cual >>> Recibe una dieta, la cual.

2.2.11. Página 29, desmarcar >>> desmarcarla.

2.2.12. Página 29, parámetros >>> parámetro.

2.2.13. Página 30, la receta_la primera >>> la receta_La primera.

2.2.14. Página 30, La cuanta >>> La cuenta.

2.2.15. Página 30, quedara >>> quedará.

2.2.16. Página 33, recomendación_de esta forma >>> recomendación_De esta forma.

2.2.17. Página 33, menor de la cantidad >>> menor que la cantidad.

2.2.18. Página 34, la terminal como >>> la terminal, como.

2.2.19. Página 34, Consistencia entre requisito >>> Consistencia entre requisitos.

2.2.20. Página 28, descripción de “añadirRecetaFavorita(Receta)”. Faltan comas para que la frase tenga sentido. Otra opción sería cambiar la descripción a: “Permite al usuario marcar una receta como favorita”.

2.2.21. Página 33, La recomendación quedará activa notificando al usuario cuando sea necesario >>> La recomendación quedará activa, notificando al usuario cuando sea necesario.

2.2.22. Página 28, descripción de “abandonarDieta(Dieta)”. Descripción más concisa, “Se desmarca la receta como favorita.

2.2.23. Página 29, postcondición de “consultarReceta(NombreReceta) : Receta”. ¿Qué se devuelve una vez encontrada?. Sería más correcto decir “una vez encontrada se muestra la receta al usuario”.

2.2.24. Página 29, descripción de “eliminarRecetaFavorita (NombreReceta)”. Descripción más concisa, “La receta seleccionada por el usuario se eliminará de favoritos”.

2.2.24. Página 30, postcondición de “solicitarReceta() : Receta”. Se describen pasos que se realizan durante la operación, no el resultado final una vez acabada.

2.2.25. Página 32, descripción de “activarRecomendación(Recomendación)”. Eliminar “si este lo desea”

2.2.26. Página 36, “Gesto_Inventario” >>> “Gestor_Inventario”.

2.2.27. Página 36, “Gesto_Notificaciones” >>> “Gestor_Notificaciones”.

2.3 Presentación y legibilidad

La presentación es, por lo general, muy correcta, adecuándose al formato y tipo de texto presentado.

En lo que respecta a la legibilidad, existen apartados en los que no es del todo legible lo que está redactado. Los errores observados se citarán a continuación:

2.3.1. En el apartado 4.1 modelo de información, las flechas que unen las distintas clases, no son las correspondientes a UML.

2.3.2. En el apartado 4.1 modelo de información, las clases “Alimento” y “Recetas” podrían estar algo más separadas. De esta forma quedarán más claras las relaciones “0...N” y “1...N”.

2.3.3. En el apartado 4.1, utilización de palabras coloquiales, que se podrían sustituir por unas más adecuadas dado el entorno en el que se desarrolla el documento. Por ejemplo “nick”.

2.3.3. En el apartado 4.3, en el apartado de “Contratos de operaciones” que se ubica dentro de cada tabla, el nombre de cada contrato tiene el mismo formato que los atributos del mismo, dificultando su localización dentro de la tabla.

2.3.4. El texto contenido en las tablas del documento tiene un espaciado entre palabras excesivo. Creemos que puede ser debido al uso la opción de texto “justificar”. Alineándolo a la izquierda sería más legible.

3. Consideraciones individuales sobre adecuación al índice de la práctica.

3.1 Estructura del documento: ¿sigue los apartados del índice de la práctica?

Sí, la estructura del documento del equipo G1 cumple de forma satisfactoria con el índice de la práctica. Todos los puntos están incluidos y además se disponen en el orden propuesto.

3.2 Contenido del documento: ¿se corresponde cada uno de los apartados con la finalidad que el índice propone?

Sí, el contenido de la práctica corresponde con lo descrito en los apartados del índice.

3.3 Explicación en más detalle de los errores principales

3.3.1. En el apartado 4.1 modelado de información, no se recogen las cantidades necesarias de cada alimento para la receta.

3.3.2. En el apartado 4.1 modelado de información, falta el atributo “Receta_favorita” en la clase “Receta”.

3.3.3. En el apartado 4.2 modelo de implementación, se podría especificar con que está relaciono cada componente (controlador, vista, datos) para una mejor comprensión del modelado.

3.3.4. En el contrato de operación “activarRecomendación(Recomendación)”. En la postcondición, la parte de “notificando al usuario cuando sea necesario” formaría parte de otro contrato de operación.

3.3.5. En el contrato de operación “crearAviso(Alimento, Cantidad)”. En la descripción, la parte de “En el momento en el que dicho alimento baje de la cantidad especificada, la aplicación mandará un aviso al usuario.” formaría parte de otro contrato de operación.

3.3.6. Según el documento, el contrato de operación “crearAviso(Alimento, Cantidad)” tiene la capacidad de crear el aviso y de mandar el aviso al usuario. Se debería de eliminar de las postcondiciones la parte sobre el envío del aviso, ya que eso pertenece a otra operación.

3.3.7. En “Gestor_Notificaciones” falta por crear una operación para mandar avisos al usuario.

3.3.8. En “Trazabilidad de requisitos - modelo de información (clases)” el RF-03 debería tener dependencia con la columna “alimento”.

3.3.9. En “Trazabilidad de requisitos - modelo de información (clases)” el RF-07 debería guardar dependencia con la columna “dieta”.

3.3.10. En “Trazabilidad de requisitos - modelo de información (clases)” tanto el RF-08 como el RF-19 deberían guardar dependencia con la columna “inventario”.

3.3.11. En “Trazabilidad de requisitos - modelo de información (clases)” el RF-14 debería guardar dependencia con la columna “receta”.