



**DESARROLLO DE APLICACIÓN PARA LA GESTIÓN DE DONACIONES DE  
ALIMENTOS - SHARE FOOD**

**ING. OWEIMAR ANDRES MENA HIDALGO  
ING. JORGE LUIS QUINTANA SUAREZ**

**PROYECTO DE GRADO PARA OPTAR POR EL TÍTULO DE ESPECIALISTA  
EN INGENIERÍA DE SOFTWARE**

**ASESORES:**

**PROFESORA DIANALIN NEME PRADA  
PROFESOR IVÁN RODRIGO ROMERO FLÓREZ  
PROFESOR ELKIN FERNEY QUINTERO GÓMEZ  
PROFESOR MIGUEL JAVIER FELIZZOLA FIGUEREDO**

**UNIVERSIDAD ANTONIO NARIÑO  
FACULTAD DE INGENIERÍA DE SISTEMAS  
BOGOTÁ D.C.**

**2020**

NOTA DE ACEPTACIÓN

---

---

---

---

---

Firma del jurado

---

Firma del jurado

---

Firma del jurado

## **Título del Proyecto**

DESARROLLO DE APLICACIÓN PARA LA GESTIÓN DE DONACIONES DE  
ALIMENTOS - SHARE FOOD

## Contenido

1.	INTRODUCCIÓN .....	11
2.	PLANTEAMIENTO DEL PROBLEMA .....	12
2.1	DESCRIPCIÓN DEL PROBLEMA.....	12
2.2	FORMULACIÓN DEL PROBLEMA .....	13
3.	OBJETIVOS .....	13
3.1	OBJETIVOS GENERAL.....	13
3.2	OBJETIVOS ESPECÍFICOS .....	14
4.	MARCO DE REFERENCIA .....	14
4.1	ESTADO DEL ARTE.....	14
4.1.1	Resumen del Estado del Arte.....	14
4.1.2	Bancos de alimentos en Colombia.....	16
4.1.3	Software para donación de alimentos .....	17
4.2	IMPACTO .....	18
4.3	COMPONENTE DE INNOVACIÓN .....	18
4.4	MARCO TEÓRICO .....	18
4.4.1	Software .....	18
4.4.1.1	Software de sistema .....	18
4.4.1.2	Software de programación .....	19
4.4.1.3	Software de aplicación .....	19
4.4.1.4	Aplicación Web .....	19
4.4.2	Ciclo de vida del Desarrollo de Software .....	20
4.4.3	Metodología de Desarrollo .....	22
4.4.3.1	2.3.2 Scrum .....	22
4.4.4	Patrones de Arquitectura.....	23
4.4.4.1	Capas.....	23
4.4.4.2	Modelo-Vista-Controlador.....	23
4.4.4.3	Microservicios.....	23

4.4.5	Patrones de diseño.....	24
4.4.5.1	Fachada (Facade) .....	24
5.	METODOLOGÍA.....	25
5.1	SCRUM.....	25
5.1.1	Roles .....	25
5.1.2	Resumen de los Eventos .....	26
5.1.2.1	Reunión de Planificación .....	26
5.1.2.2	Scrum Diario.....	26
5.1.2.3	Revisión de Sprint.....	26
5.1.2.4	Retrospectiva de Sprint .....	27
6.	PROCESO DE SOFTWARE.....	28
6.1	REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES .....	28
6.1.1	Estimación Puntos de Usuario .....	30
6.1.2	Atributos de calidad .....	31
6.2	DISEÑO Y ARQUITECTURA .....	32
6.2.1	Alcance y Restricciones .....	32
6.2.1.1	Alcance .....	32
6.2.1.2	Restricciones .....	36
6.2.1	Diagrama de Despliegue .....	36
6.2.1	Caso de uso Arquitecturalmente Relevante .....	37
6.2.2	Diagrama de Secuencia.....	37
6.2.3	Diagrama de Clases (Vista Lógica) .....	38
6.2.4	Vista de Desarrollo (Diagrama de Componentes) .....	39
6.2.5	Arquitectura de alto Nivel.....	39
6.3	CONSTRUCCIÓN .....	40
6.3.1	Lenguaje de programación.....	40
6.3.2	Herramientas de desarrollo utilizadas.....	40
6.3.3	Gestión del código .....	41
6.3.4	Modelo de base de datos .....	42
6.3.5	Diseño de la aplicación .....	42

6.3.5.1	Modelo .....	43
6.3.5.2	Controlador .....	44
6.3.5.3	Vista .....	44
6.4	PRUEBAS .....	45
6.4.1	Pruebas de código estático .....	45
6.4.2	Pruebas de integración .....	46
6.4.3	Pruebas Funcionales.....	46
6.4.4	Pruebas de Seguridad.....	47
6.5	INSTALACIÓN Y CONFIGURACIÓN .....	48
6.5.1	Servidor web .....	48
6.5.2	Servidor de base de datos.....	48
6.5.3	Aplicación WEB .....	49
7.	ANEXOS .....	53
8.	CONCLUSIONES .....	54
9.	REFERENCIAS.....	55

## Lista de Tablas

TABLA 1. LISTADO DE APLICACIONES PARA DONAR ALIMENTOS .....	17
TABLA 2. SPRINTS ASOCIADOS A HISTORIAS DE USUARIOS .....	27
TABLA 3. HISTORIA DE USUARIO DONANTE .....	28
TABLA 4. HISTORIA DE USUARIO REGISTRO DE USUARIOS .....	28
TABLA 5. HISTORIA DE USUARIO RECHAZAR DONACIÓN .....	29
TABLA 6. HISTORIA DE USUARIO PEDIR DONACIÓN .....	29
TABLA 7. HISTORIA DE USUARIO RETIRAR DONACIÓN .....	29
TABLA 8. HISTORIA DE USUARIO ADMINISTRAR APLICACIÓN .....	29
TABLA 9. HISTORIA DE USUARIO GEORREFERENCIAR DONACIÓN .....	30
TABLA 10. HISTORIA DE USUARIO GEORREFERENCIAR DONACIÓN .....	30
TABLA 11. HISTORIA DE USUARIO GENERAR REPORTE .....	30
TABLA 12. ESTIMACIÓN PUNTOS HISTORIAS DE USUARIO .....	31
TABLA 13. ATRIBUTO DE CALIDAD PORTABILIDAD .....	31
TABLA 14. ATRIBUTO DE CALIDAD USABILIDAD .....	31
TABLA 15. ATRIBUTO DE CALIDAD FUNCIONALIDAD .....	31
TABLA 16. ATRIBUTO DE CALIDAD SEGURIDAD .....	32
TABLA 17. LISTADO DE CASOS DE PRUEBA .....	46
TABLA 18. RESULTADO DE PRUEBAS .....	46
TABLA 19. RESULTADOS PRUEBAS DE SEGURIDAD .....	47

## Lista de Figuras

FIGURA 1. LISTADO Y MAPA DE LOS BANCOS DE ALIMENTOS .....	16
FIGURA 2. CICLO DE VIDA DEL SOFTWARE.....	21
FUENTE: ELABORACIÓN PROPIA .....	21
FIGURA 3. METODOLOGÍA DE TRABAJO SCRUM.....	22
FIGURA 4. ESQUEMA DEL PATRÓN FACADE .....	24
FIGURA 5. DIAGRAMA DE DESPLIEGUE.....	36
FIGURA 6. CASO DE USO ARQUITECTURALMENTE RELEVANTE.....	37
FIGURA 7. DIAGRAMA DE SECUENCIA.....	38
FIGURA 8. DIAGRAMA DE CLASES - VISTA LÓGICA .....	38
FIGURA 9. DIAGRAMA DE COMPONENTES.....	39
FIGURA 10. ARQUITECTURA DE ALTO NIVEL .....	40
FIGURA 11. ENTORNO DE DESARROLLO INTEGRADO VISUAL STUDIO .....	41
FIGURA 12. INTERFAZ AZURE DEVOPS .....	41
FIGURA 13. DIAGRAMA ENTIDAD RELACIÓN BASE DE DATOS.....	42
FIGURA 14. PROYECTOS DE LAS CAPAS DE LA APLICACIÓN .....	42
FIGURA 15. ARCHIVOS DE CLASES DEL MODELO.....	43
FIGURA 16. RELACIÓN DE LAS CLASES DE LA APLICACIÓN.....	43
FIGURA 17. ARCHIVOS DE LOS CONTROLADORES .....	44
FIGURA 18. ARCHIVOS DEL FONT-END .....	44
FIGURA 19. RESULTADOS SONAR CLOUD .....	45
FIGURA 20. RESULTADOS PRUEBAS HERRAMIENTA MANTIS .....	47
FIGURA 21. SERVIDOR DE APLICACIONES IIS .....	48
FIGURA 22. SERVIDOR SQL BASE DE DATOS .....	49
FIGURA 23. LOGIN APLICACIÓN SHAREFOOD.....	49



FIGURA 24.	FORMULARIO DE REGISTRO SHAREFOOD .....	50
FIGURA 25.	REGISTRAR DONACIÓN APLICACIÓN SHAREFOOD .....	50
FIGURA 26.	GEORREFERENCIACIÓN APLICACIÓN SHAREFOOD .....	51
FIGURA 27.	GESTIÓN DE LA BANDEJA APLICACIÓN SHAREFOOD .....	51
FIGURA 28.	DETALLES UNA DONACIÓN APLICACIÓN SHAREFOOD .....	52
FIGURA 29.	CONTROL DE DONACIONES APLICACIÓN SHAREFOOD .....	52

## **Resumen**

Este documento fue realizado como informe final de trabajo de grado de la Especialización en Ingeniería de Software, en el cual se presenta el diseño y desarrollo de una aplicación web multiplataforma siguiendo los principios, buenas prácticas, desarrollo seguro para aplicaciones web y calidad del servicio, aprendidos a lo largo de la especialización en Ingeniería de Software de la universidad UAN.

Basados en los principios de responsabilidad social y servicio a la comunidad, se orientó la finalidad de la aplicación desarrollada para que sea una herramienta web multiplataforma, gratuita para gestionar donaciones de alimentos, con una interfaz simple que permita a la población en general ofrecer y tomar alimentos en calidad de donación sin intervención de un tercero, como iniciativa social que permita en cierta medida disminuir al desperdicio de comida.

La aplicación fue desarrollada usando la metodología scrum implementada en el ecosistema en la nube de Microsoft denominado Azure Devops, asimismo las herramientas de desarrollo utilizadas fueron implementadas con la plataforma de desarrollo web de Microsoft, por sus múltiples herramientas que brindan un ambiente apropiado a las necesidades del proyecto.

El modelamiento fue propuesto usando el lenguaje de modelado unificado UML, para describir el comportamiento del sistema, la comunicación entre los componentes, y la forma que es desplegado en un entorno de producción.

## 1. INTRODUCCIÓN

En la actualidad el uso de herramientas tecnológicas y el internet se ha contribuido a la solución de problemas y en muchos casos estas soluciones se han convertido en proyectos disruptivos que cambian la forma de pensar y actuar de la humanidad y han impulsado el desarrollo y la calidad de vida de las personas en todo el mundo.

En el presente trabajo se realiza la descripción y la formulación del problema en donde se evidencia el gran desperdicio de alimentos y una población vulnerable carente de recursos alimenticios, cuya propuesta de solución se base en el diseño y desarrollo de una aplicación que permite la gestión de donaciones de alimentos, a modo de ofrecer y aceptar donaciones de manera descentralizada, para evitar el desperdicio de alimentos.

En este sentido para mejorar el proceso de rescate de alimentos se requiere pensar en aplicaciones innovadoras que permitan a las organizaciones y público en general tener acceso a los alimentos con mayor flexibilidad, menor esfuerzo y fácil acceso a través de internet, los cuales son planteado en los objetivos generales y específicos.

A través de la investigación realizada para el estado del arte, se puede conocer que los bancos de alimentos realizan una labor social para gestión de donaciones de comida, con personas voluntarias en su mayoría, pero que aún con este esfuerzo no se logra cubrir todas las necesidades.

En el marco teórico y metodología se definen los conceptos más relevantes sobre el software y su construcción, el ciclo de desarrollo de software, parones de arquitectura y patrones de desarrollo.

Seguidamente, se encuentra la sección principal del proyecto como es el proceso de software donde se describe detalladamente, cada una de las etapas del desarrollo hasta obtener un producto funcional que será puesto a prueba en la sustentación del proyecto.

## **2. PLANTEAMIENTO DEL PROBLEMA**

### **2.1 DESCRIPCIÓN DEL PROBLEMA**

De acuerdo con los datos obtenidos del DNP (Departamento Nacional de Planeación), en nuestro país se desperdician cada año alrededor de 9,7 millones de toneladas de alimentos, lo que equivale al 34% de la producción total, por otra parte 2,4 millones de la población no tienen asegurado una comida diaria. Al respecto, otro estudio desarrollado por la oficina de Prosperidad Social enfatizó que si se comparan los periodos 2004 - 2006 y 2016 – 2018, se puede observar una disminución de 2 millones de subalimentados.

Asimismo, de acuerdo con el informe de seguridad alimentaria de 2019 que reveló la FAO, entre los 28 países de la región que cuentan con datos, Colombia se encuentra en el puesto 20, de un listado donde Haití está en el primer lugar, resaltando que, de cada cinco, 10 personas eran afectadas por esa problemática. De los 170 países que fueron caracterizados, Colombia está en el puesto 102.

Aunque las cifras son bajas en relación con otros países de la misma categoría, Colombia siendo un país con tanto potencial alimentaria sus niveles deberían ser aún mucho más bajos, pero desafortunadamente no es así y existen demasiadas personas que sufren de desnutrición, aun inclusive en ciudades principales.

A pesar de la existencia de los “Bancos de Alimentos” como organizaciones solidarias sin ánimo de lucro, que contribuyen a reducir los índices de hambre y la desnutrición, no son suficientes debido a la gran cantidad de personas con necesidades alimentarias y no se alcanza a cubrir toda la población necesitada debido a limitaciones logísticas y también

priorizan la recepción de alimentos a grandes empresas o a donaciones representativas, dejando de aprovecharse un gran segmento de donantes como son los hogares y familias colombianas del común, que esta dispuestas a ayudar, ofreciendo sus alimentos en vez de permitir que se les dañe.

En el informe de gestión de la Asociación de Bancos de Alimentos de Colombia del año 2019, se reporta la cifra de desperdicio que 9.7 millones de toneladas de alimentos son tirados a la basura y que esta entidad rescata solamente el equivalente a 0.26% del total de los alimentos desperdiciados.

En este sentido y con el ánimo de contribuir a disminuir el desperdicio de alimentos y darle una destinación a disminuir el hambre en la población, necesita un aplicativo informático que pueda conectar a las personas dispuestas a donar sus productos alimenticios con las personas necesitadas de los mismo, sin ninguno tipo de intermediario o canal que haga más complicada la entrega y recepción de los alimentos.

## **2.2 FORMULACIÓN DEL PROBLEMA**

¿Como se podría desarrollar una aplicación informática que puede ser visualizada en múltiples dispositivos y que permita a la población en general, ofrecer y aceptar donaciones de alimentos?

## **3. OBJETIVOS**

### **3.1 OBJETIVOS GENERAL**

Desarrollar una aplicación web multiplataforma para la gestión de donaciones de alimentos, con una interfaz simple que permita a la población en general, bajo la primicia de responsabilidad social ofrecer y tomar donaciones de alimentos de modo descentralizado, como iniciativa para ayudar a disminuir al desperdicio de comida.

### **3.2 OBJETIVOS ESPECÍFICOS**

- Análisis de requisitos para el desarrollo de la aplicación para gestión de donaciones de alimentos, que cumpla las necesidades de los usuarios.
- Plantear el diseño arquitectónico y metodología de desarrollo para la ejecución del proyecto.
- Desarrollar cada uno de los módulos que componen la aplicación para que sea usable y que incluya un componente de georreferenciación para las donaciones.
- Implementar un componente de administración, autenticación de usuarios y generación de estadística de las donaciones.
- Desarrollar pruebas a la aplicación que aporten seguridad y calidad al producto final.

## **4. MARCO DE REFERENCIA**

### **4.1 ESTADO DEL ARTE**

#### **4.1.1 Resumen del Estado del Arte**

Para muchas entidades encontrar soluciones a la desnutrición de la población y al desperdicio de alimentos, es hoy en día una preocupación a nivel mundial y cada vez los países son más conscientes que se deben implementar políticas y estrategias para mitigar estas problemáticas.

El Programa de las Naciones Unidas para el Desarrollo, celebrado en Río de Janeiro en 2012, trazó un plan de trabajo denominado los Objetivos de Desarrollo Sostenible (ODS), cuyo propósito ha sido crear un conjunto de objetivos mundiales relacionados con los desafíos ambientales, políticos y económicos con que se enfrenta nuestro mundo.

Los denominados ODM se plantearon como objetivos medibles acordados con alcance mundial con el fin de enfrentar la pobreza extrema y la hambruna, ayudar a prevenir enfermedades mortales y la posibilidad de ampliar la alfabetización primaria a la población infantil, entre otras prioridades del desarrollo.

El objetivo 2 de este programa, habla precisamente que para el 2030 exista “Hambre cero”, mediante la adopción de medidas para asegurar el buen funcionamiento de los mercados de productos básicos alimentarios, el uso de nuevas tecnologías, el acceso oportuno a información sobre las reservas de alimentos, ayudado a limitar la extrema volatilidad de los precios de los alimentos que permitan asegurar una sostenibilidad en el sistema de producción de alimentos.

No obstante, estas son estrategias para un mediano o largo plazo que no impiden que en el tiempo actual mueran personas a causa de la desnutrición extrema o presenten un peso con alto riesgo y continúe la brecha entre los que tienen tantos recursos que desperdician alimentos y los que no tienen acceso a una sana alimentación o completamente nada.

En nuestro país son escasas las leyes que existen para impedir estas condiciones, pero aun así son un punto de inicio para mejorar las condiciones:

- Ley 2010 de 2019 Reformas al Estatuto Tributario. Artículo 1.3.1.12.13. Exclusión del impuesto del IVA para los alimentos destinados al consumo humano donados a Bancos de Alimentos.
- Proyecto Acuerdo 017 de 2017, en el cual se establecen unos lineamientos para la reducción de la pérdida y desperdicio de alimentos en la ciudad capital.
- Decreto 1794 de 2013, artículo 4 y artículo 38, numeral 10. Exclusión del impuesto del IVA para los alimentos destinados al consumo humano donados a Bancos de Alimentos.

En este mismo sentido, una de las organizaciones impulsoras para reducir los índices de hambre y el desperdicio de comida en los países del mundo, son los bancos de alimentos,

fundados con el objetivo de llevar a cabo una labor social sin ánimo de lucro, abasteciéndose de las donaciones de alimentos por parte del sector agropecuario, comercial, industrial, hotelero, de restaurantes y/o personas naturales. Posteriormente se distribuyen estos alimentos a la población que se encuentra en situación de vulnerabilidad.

Las empresas y donantes particulares que cumplan con unos requisitos y obtengan un certificado de donación entregado por parte de los Bancos de Alimentos asociados se benefician por la deducción del 125% de renta, por la realización de donación en dinero o especie.

#### 4.1.2 Bancos de alimentos en Colombia

En Colombia la Asociación de Banco de Alimentos ABACO, lleva 10 años trabajando integrados, con 840 colaboradores de los cuales 601 son permanentes, en una red que actual cuenta con 19 bancos de alimentos registrados, que prestan sus servicios en la mayoría de las ciudades capitales y que han obtenido la calidad de miembros certificados internacionalmente de la red global de bancos de alimentos.

**Figura 1.** Listado y mapa de los bancos de alimentos



	CIUDAD	Nombre del Banco de Alimentos
1	BARRANQUILLA	BANCO ARQUIDIOCESANO DE ALIMENTOS
2	BOGOTÁ	FUNDACIÓN BANCO ARQUIDIOCESANO DE ALIMENTOS
3	BUCARAMANGA	FUNDACIÓN BANCO DE ALIMENTOS ARQUIDIOCESIS
4	CALI	FUNDACIÓN ARQUIDIOCESANA BANCO DE ALIMENTOS
5	CARTAGENA	BANCO ARQUIDIOCESANO DE ALIMENTOS
6	CARTAGO	FUNDACIÓN BANCO DIOCESANO DE ALIMENTOS
7	CÚCUTA	BANCO DIOCESANO DE ALIMENTOS MONSEÑOR OSCAR URBINA ORTEGA
8	IBAGUÉ	BANCO ARQUIDIOCESANO DE ALIMENTOS
9	MANIZALES	BANCO ARQUIDIOCESANO DE ALIMENTOS
10	MEDELLÍN	FUNDACIÓN BANCO ARQUIDIOCESANO DE ALIMENTOS
11	MEDELLÍN	FUNDACIÓN SACIAR
12	MONTERIA	BANCO DIOCESANO DE ALIMENTOS
13	NEIVA	FUNDACIÓN BANCO DIOCESANO DE ALIMENTOS
14	PASTO	BANCO DIOCESANO DE ALIMENTOS
15	PEREIRA	FUNDACIÓN CARITAS BANCO DE ALIMENTOS
16	POPAYÁN	ARQUIDIÓCESIS
17	SANTA MARTA	BANCO DIOCESANO DE ALIMENTOS
18	SINCELEJO	BANCO DE ALIMENTOS
19	VILLAVICENCIO	BANCO ARQUIDIOCESANO DE ALIMENTOS

Fuente: página web [www.abaco.org.co](http://www.abaco.org.co)



### 4.1.3 Software para donación de alimentos

Se realizó una búsqueda en internet de aplicaciones enfocadas a evitar el desperdicio de comida y se pudo evidenciar que existen múltiples aplicativos, en su mayoría para dispositivos móviles, de los cuales solo unos pocos tienen ámbito social y pertenecen a bancos de alimentos en otros países, otras aplicaciones fueron desarrolladas para explotar comercialmente el desperdicio de alimentos.

En Argentina hay una aplicación para inscribirte al banco de alimentos, acceder a videos instructivos, compartir las últimas noticias de la fundación y como ser voluntario.

Paraguay tiene una aplicación desarrollada por el principal el banco de alimentos enfocada a las necesidades de ese país, la cual permite registrar donaciones y suministrar información sobre campañas y actividades para evitar el desperdicio en este país.

En España hay otra aplicación llama Banco de Alimentos Burgos, igualmente enfocada a su propio banco de alimentos, es muy sencilla y solo solo ofrece información de esta asociación.

Nombre Aplicación	Origen	Plataforma	Alcance
<i>Banco de Alimentos</i>	Argentina	Móvil	Facilitar donaciones al banco de alimentos
<i>Banco de alimentos</i>	Paraguay	Móvil	Facilitar donaciones al banco de alimentos, ofrecer noticias sobre actividades.
<i>Banco de Alimentos Burgos</i>	España	Móvil	Ayudar al proceso de donaciones del banco de alimentos.
<i>Nadie Sin Su Ración Diaria</i>	España	Móvil	Donación de alimentos y fidelización de cliente cadena de tiendas.
<i>Too Good To Go</i>	España	Móvil	Adquisición de excedentes alimentarios a un precio muy reducido
<i>Geev</i>	Francia	Móvil	Gestión de donación de alimentos y bienes en general
<i>Food for All</i>	Estados Unidos	móvil	Conecta a los clientes con los restaurantes una hora antes de que cierren, para obtener descuentos de hasta el 80%
<i>NoFoodWasted</i>	Estados Unidos	móvil	Genera alertas a los usuarios, de alimentos en tiendas que se acercan a su fecha de caducidad y con precios de una fracción de su costo original.

Tabla 1. Listado de aplicaciones para donar alimentos

## **4.2 IMPACTO**

Con este proyecto se establecerá un canal que amplíe la cobertura de donaciones de alimentos, para que más colombianos se unan a evitar el desperdicio y se haga uso responsable de los alimentos que no necesitan o que por descuido llegan a la fecha de caducidad y son arrojados a la basura, tanto en empresas del sector agroindustrial, minoristas, mayoristas, restaurantes, como en los hogares en todo el país.

## **4.3 COMPONENTE DE INNOVACIÓN**

La aplicación para la gestión de donaciones de alimentos estará desarrollada con forme a las necesidades para agilizar el proceso de publicar y solicitar donaciones, usando componentes de georreferenciación, que permitan poner a disposición un aplicativo moderno, con experiencia de usuario, que pueda ser visualizado en diferentes dispositivos móviles y computadores.

## **4.4 MARCO TEÓRICO**

En este capítulo se pretende abordar a manera de resumen, los temas más relevantes relacionados con los procesos y tecnologías empleadas en la construcción de una aplicación, en cumplimiento de los objetivos propuestos en este trabajo, enfocado a la gestión de donación de alimentos.

### **4.4.1 Software**

La palabra software tiene como definición un programa o conjunto de programas para computadora, para procesamiento de datos, conjunto de procedimientos o lista de instrucciones que permiten realizar tareas en un equipo informático. Normal mente se utiliza esta palabra para llamar de manera general a los programas que hacen parte de un sistema informático. El software cubre toda la parte intangible o diferentes de lo físico en un sistema de cómputo.

#### **4.4.1.1 Software de sistema**

Es un tipo de software o programa encargado de permitir la ejecución de las aplicaciones requeridas para que el sistema funcione adecuadamente. A veces se confunde con el sistema

operativo, pero este también incluye múltiples herramientas para optimización, controladores de dispositivo y los servidores, entre otros componentes adicionales.

#### **4.4.1.2 Software de programación**

Se refiere a las herramientas de software que permiten el desarrollo de nuevos programas de cómputo. Este tipo de herramientas no son a menudo utilizadas por usuarios finales sino por programadores. Entre el software de programación, se destacan los más importantes, los siguientes:

- Compiladores (programas convierten a lenguaje de maquina).
- Intérpretes (software que "traduce" los lenguajes de programación, sin compilar).
- Editores de código (se emplean para escribir el código fuente).

#### **4.4.1.3 Software de aplicación**

Un software de aplicación es un programa de computador diseñado para facilitar realizar tareas específicas, ya sea en computadores, tablets o celulares. El software de aplicación más populares son los de ofimática como Microsoft Office, los videojuegos, los programas de diseño CAD, el software utilizado para telecomunicaciones, aplicaciones de productividad empresarial CRM, ERP, BI SGA, o educativos, son algunos ejemplos de este tipo de programas.

#### **4.4.1.4 Aplicación Web<sup>1</sup>**

Las aplicaciones web son las herramientas de software que los usuarios utilizan para acceder a un servidor web a través de internet o de una intranet usando un navegador web. En otras palabras, es un programa que se codifica usando lenguaje denominado HTML HyperText Markup Language (Lenguaje de Marcas De Hipertexto) interpretado por los navegadores, donde la tarea de presentar las interfaces la realiza directamente el navegador.

---

<sup>1</sup> Tomado de la página web [https://es.wikipedia.org/wiki/Aplicación\\_web](https://es.wikipedia.org/wiki/Aplicación_web)

Las aplicaciones web son muy populares por la practicidad al utilizar el navegador web como cliente ligero, a la independencia del sistema operativo específico, de igual modo la facilidad para actualizarse y mantener aplicaciones web sin la necesidad de distribuir e instalar software a todos los usuarios. Existen aplicaciones como correos electrónicos, drives en la nube, redes sociales, tiendas en línea y el streaming como Netflix, YouTube o Deezer que son ejemplos muy conocidos de aplicaciones web.

#### 4.4.2 Ciclo de vida del Desarrollo de Software

La definición de ciclo de vida del software describe las etapas involucradas para el desarrollo de aplicaciones de software, desde la fase de análisis de requisitos hasta la fase de despliegue, concretando las distintas actividades que se requieren para garantizar que el producto tenga la calidad deseada y cumpla con todos los requisitos funcionales y no funcionales, el uso de tecnologías, metodologías de desarrollo, patrones de arquitectura a emplear. No todos los ciclos de vida del desarrollo implementan la misma cantidad de pasos algunas se centran en la seguridad siendo más rigurosos en la definición de riesgos, otros en las pruebas.

La implementación del ciclo de vida del desarrollo se aplica en los proyectos debido a que es muy costoso corregir los errores detectados durante la fase de implementación o despliegue. Mediante el ciclo de vida, se permite que los errores sean detectados tempranamente, permitiendo a los desarrolladores dirigir sus esfuerzos en la calidad del software y al cumplimiento del proceso de negocio. El ciclo de vida básico para el desarrollo del software consta de los siguientes pasos:

**Análisis de requisitos:** Es el proceso y estudio de las necesidades de los usuarios con el fin de determinar la definición de los requisitos del producto de software a desarrollar. En esta etapa el desarrollador y el cliente plantear las funciones y comportamiento del sistema de manera específica evitando las ambigüedades.

**Diseño:** Se realiza la definición general de cada componente de la aplicación, las consideraciones de la infraestructura tecnológica que soportará el software. Se definen las entidades y diseñan los casos de uso y diagramas UML, obteniendo un modelo cercano al todo el software a construir.

**Desarrollo:** Se toman el diseño del sistema y se realiza la implementación de la aplicación en un lenguaje de programación, metodología, frameworks y buenas prácticas, para crear las funciones, componente o módulos, que se integrarán para finalmente obtener un producto terminado.

**Pruebas:** Se realizan las pruebas de cada funcionalidad, modulo, componte, pantalla, reporte o visualización, con el fin de corregir posibles errores o bugs no detectados en el momento de la construcción de la aplicación.

**Despliegue:** Por lo general el desarrollo de las aplicaciones se realiza en ambientes locales y una vez sea completado y pase satisfactoriamente todas las pruebas y expectativas del cliente o usuario final debe ser instalado o desplegado y puesto en producción.

*Figura 2. Ciclo de vida del software*



*Fuente: Elaboración propia*

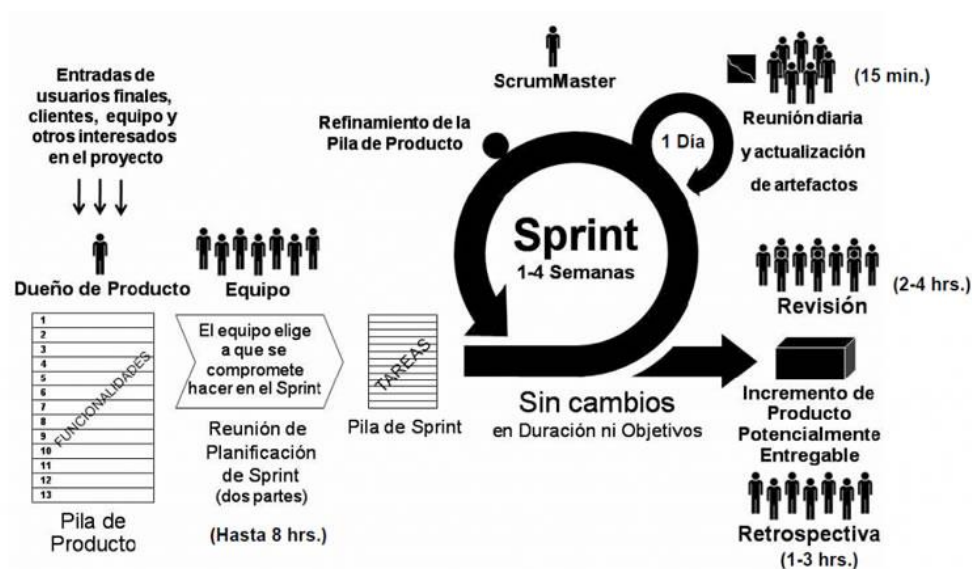
### 4.4.3 Metodología de Desarrollo

#### 4.4.3.1 2.3.2 Scrum

La metodología Scrum es definida como un marco de trabajo o framework para la gestión de proyectos utilizada a menudo por equipos que manejan proyectos complejos. Es decir, se trata de una metodología de trabajo ágil que tiene como finalidad la entrega de valor en períodos cortos de tiempo y para ello se basa en tres pilares: la transparencia, inspección y adaptación. Esto permite al cliente, junto con su equipo comercial, insertar el producto en el mercado pronto, rápido y empezar a obtener ventas. Las características más sobresalientes que de la metodología son:

- Gestión permanente de las expectativas del cliente.
- Resultados rápidos.
- Flexibilidad y adaptación.
- pronto retorno de inversión.
- Mitigación de riesgos.
- Productividad y calidad.
- Alineación entre cliente y equipo.
- Equipo motivado.

*Figura 3. Metodología de trabajo scrum*



Fuente: página web <https://www.antevenio.com/blog/2020/02/que-es-la-metodologia-scrum/>

#### **4.4.4 Patrones de Arquitectura**

Se puede interpretar como patrón arquitectónico a una solución general y reutilizable basada en un esquema u organización estructural fundamental para un problema común en el desarrollo de software. Los patrones arquitectónicos son similares al patrón de diseño de software, pero este tiene un alcance más amplio.

##### **4.4.4.1 Capas**

La arquitectura en capas es un estilo que permite separar los roles de la aplicación en distintas capas perfectamente definidas y en la práctica la mayoría de las veces es implementado usando 4 capas: presentación, negocio, persistencia y base de datos. Las capas se adecuan de forma horizontal de manera que cada capa solo puede comunicarse con la capa que esta inmediatamente por debajo y sucesivamente de esta manera. Este estilo tiene la ventaja de permitir separar las responsabilidades, es fácil de desarrollar, probar y mantener.

##### **4.4.4.2 Modelo-Vista-Controlador**

Es un patrón de diseño de software también conocido como patrón MVC, que separa una aplicación en 3 partes: el Modelo, que contiene la funcionalidad y los datos básicos, la vista, muestra la información al usuario por medio de formularios web (por lo general se definen más de una vista) y el controlador, encargado de escuchar los eventos originados en la vista. Las tres partes están estrechamente interconectadas pero separadas, también desacopla los componentes y permite la reutilización eficiente del código. Este patrón es muy fácil de implementar y viene a modo de framework con el IDE de Visual Studio.

##### **4.4.4.3 Microservicios**

Consiste en construir una aplicación como un conjunto de pequeños servicios, los cuales se ejecutan en su propio proceso o instancia y se comunican con mecanismos ligeros. Para esto cada microservicio se encarga de implementar una funcionalidad independiente y completa de la lógica del negocio. Cada servicio se despliega de manera independiente. Tiene como ventaja que cada microservicio puede estar programado en un lenguaje de programación diferente y usar una tecnología de base de datos distinta.

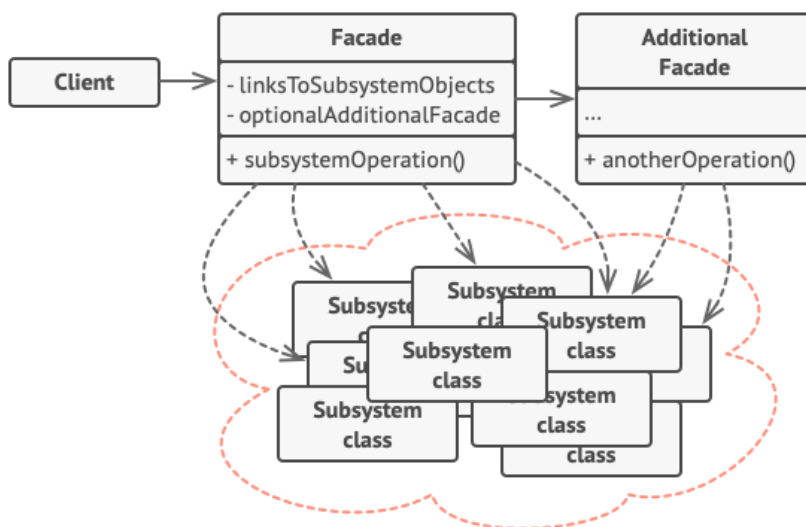
#### 4.4.5 Patrones de diseño

Un patrón de diseño se puede definir como una solución a un problema recurrente en la construcción de software, a un nivel más específico, del cual se debe haber comprobado su efectividad resolviendo problemas similares, es reutilizable y es aplicable a diferentes problemas. Los patrones más importantes se agrupan en los denominados patrones GoF (Gang Of Four), presentados por los desarrolladores Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides - Addison Wesley.

##### 4.4.5.1 Fachada (Facade)

El patrón Facade trabaja agrupando las interfaces de un conjunto de objetos, en una única interface para que ese conjunto sea más fácil de usar por parte de un cliente. En otras palabras, este patrón busca ocultar toda la complejidad de múltiples subsistemas al entregarle la operación concreta y evitarnos entrar a interactuar directamente o conocer el conocimiento técnico y de funcionamiento de los subsistemas.

*Figura 4. Esquema del patrón Facade*



Fuente: página web <https://refactoring.guru/es/design-patterns/facade>



## 5. METODOLOGÍA

Para el desarrollo de la aplicación web el equipo de trabajo escogió como herramienta para la gestión del proyecto, la metodología scrum, porque se ajusta a las necesidades y restricciones, es flexible y permite incorporar cambios continuos a lo largo del desarrollo de la aplicación sin que se impacte de manera significativa su construcción. En scrum es posible que para cada iteración se muestre un avance tangible del proyecto o mínimo producto viable, que pueda ser aprobado sin que surja la necesidad de esperar hasta completar totalmente la aplicación.

### 5.1 SCRUM

#### 5.1.1 Roles

Normalmente en proyectos de construcción de software donde se utiliza la metodología scrum, cada uno de los roles son asumidos por distintas personas, para nuestro caso al solo ser dos integrantes del grupo de trabajo, tuvimos que asumir varios roles, para conservar la estructura y esencia de la metodología.

**Product Owner:** Comunica la visión del producto a desarrollar al equipo de trabajo, igualmente debe representar al cliente en cuanto a la especificación de requisitos y priorización. Es quien tiene la mayor responsabilidad en el proyecto y asume las consecuencias positivas o negativas.

Por ser un proyecto de emprendimiento, la idea de crear una aplicación para la gestión de donaciones de alimentos surge del equipo de trabajo, por lo tanto, la especificación de las necesidades fue desarrollada con base en la visión del equipo y la función de los bancos de alimentos llevada al plano digital.

**Responsables:** Oweimar Andrés Mena y Jorge Luis Quintana

**Scrum Máster:** Actúa como enlace entre el Product Owner y el equipo, su función no es la de dirigir al equipo son que se encarga de evitar cualquier barrera que impida lograr los objetivos en cada sprint.

**Responsables:** Jorge Luis Quintana

**Scrum Team:** Son las personas responsables de hacer el trabajo de construcción de la aplicación y por lo general están conformados por 7 miembros que conforman un equipo cross funcional, entre quienes habría ingenieros de software, desarrolladores, arquitectos de software, analistas de sistemas, testers y diseñadores de interfaces de usuario.

**Miembros:** Oweimar Andrés Mena y Jorge Luis Quintana

## **5.1.2 Resumen de los Eventos**

### **5.1.2.1 Reunión de Planificación**

Para cada uno de los 4 sprints definidos se desarrollaron reuniones para analizar las prioridades y necesidades del proyecto, en cuanto a las funcionalidades incorporadas al producto de software, de igual forma se verificaron y validaron los avances obtenidos. En la primera reunión se definieron los requisitos funcionales y no funcionales con base en la información que ya se había levantado, se aclararon dudas por parte del equipo.

Las reuniones fueron realizadas por video conferencias atendiendo las recomendaciones de distanciamiento social por la emergencia sanitaria del COVID-19, con una duración promedio de 3 horas aproximadamente.

### **5.1.2.2 Scrum Diario**

Estas reuniones se realizaron los días en que se trabajó en el desarrollo de la aplicación, siendo lo más fiel posible al manifiesto de la metodología empleada y con el fin de facilitar la transferencia de información y colaboración entre los miembros del equipo, con el fin de aprovechar el tiempo y mejorar la productividad. Como el equipo de desarrollo estaba conformado por dos estudiantes, cada uno inspeccionaba el trabajo del otro con el fin de cambiar de visión y dar ideas para solucionar los inconvenientes que se iban presentando.

### **5.1.2.3 Revisión de Sprint**

En esta reunión de carácter informal, normalmente el equipo de trabajo presenta al cliente los requisitos de la aplicación ya completados en el sprint, en forma de incremento o de mínimo producto viable de toda la configuración final, pero al tratarse del proyecto con

características de emprendimiento, en esta reunión evaluábamos la pertinencia del alcance logrado en el sprint con base en los objetivos específicos definidos.

#### 5.1.2.4 Retrospectiva de Sprint

Estas reuniones se efectúan para mejorar de manera continua la productividad y la calidad del producto que está desarrollando, analizando cómo ha sido la forma de trabajo, la motivación del equipo, el desempeño de las actividades. En esta reunión encontrábamos varios problemas que se nos presentaron por el poco tiempo que nos quedaba debido a los compromisos laborales y académicos con la especialización que adelantamos.

En el segundo y tercer sprint, el incremento de producto no fue desarrollado con la velocidad esperada y se requirió implementar algunas acciones de mejoras y disponer de tiempo adicional para avanzar y lograr los objetivos propuestos.

#### Tablero de Sprints

<b>Sprint 1</b> Del 17/08/20 al 06/09/20	<b>Sprint 2</b> Del 07/09/20 al 27/09/20	<b>Sprint 3</b> Del 28/09/20 al 18/10/20	<b>Sprint 4</b> Del 19/10/20 al 08/11/20
1. Planteamiento de requisitos	8. Arquitectura de alto nivel	13. Módulo hacer donación	18. Filtros y Búsquedas
2. Construcción historias de usuario	9. Formulación de Modelo-Entidad-	14. Modulo pedir donación	19. Diseño UI
3. Diseño modelo de requerimientos	10. Construcción base de datos	15. Modulo administración	20. Reportes de donaciones
4. Diagrama de clases	11. Módulo de acceso	16. Componente de mapa geográfico	21. Pruebas
5. Caso de uso relevante	12. módulo de registro	17. Notificaciones	22. Construcción términos legales
6. Diagrama de Secuencia			
7. Diagrama de componentes			

Tabla 2. Sprints asociados a historias de usuarios

## 6. PROCESO DE SOFTWARE

### 6.1 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

Historia de Usuario	
<b>Numero:</b> HU01	<b>Usuario:</b> Donante, Administrador, Solicitante de donación.
<b>Nombre historia:</b> Acceso a la aplicación	
<b>Priorización:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de usuario:</b> 20	<b>Número de Iteración:</b> 2
<b>Desarrollador:</b> Jorge Quintana	
<b>Descripción:</b> Permitir a los usuarios registrados el ingreso a la aplicación, garantizando el correcto inicio y presentación de las opciones correspondientes según el perfil asignado al usuario, digitando un usuario y contraseña válidos. Ya el usuario debe estar creado previamente y habilitado.	

Tabla 3. Historia de usuario donante

Historia de Usuario	
<b>Numero:</b> HU02	<b>Usuario:</b> Donante, Administrador, Solicitante de donación.
<b>Nombre historia:</b> Registrar usuarios	
<b>Priorización:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de usuario:</b> 30	<b>Número de Iteración:</b> 2
<b>Desarrollador:</b> Oweimar Mena	
<b>Descripción:</b> Permitir a los usuarios crear una cuenta de usuario para hacer uso de las funciones de la aplicación, registrando datos de nombres, apellidos, ubicación de residencia. Generar una solicitud de recuperación de contraseñas. Es necesario diligenciar todos los campos que se encuentren marcadas con el símbolo de “*”. Para registrar automáticamente con el geoposicionador es necesario tener habilitado el GPS en el dispositivo.	

Tabla 4. Historia de usuario registro de usuarios

Historia de Usuario	
<b>Numero:</b> HU03	<b>Usuario:</b> Donante
<b>Nombre historia:</b> Hacer donación	
<b>Priorización:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de usuario:</b> 50	<b>Número de Iteración:</b> 3
<b>Desarrollador:</b> Jorge Quintana	
<b>Descripción:</b> Permitir a los usuarios la creación de una lista de alimentos para donación. se debe permitir incluir la cantidad, la ubicación geográfica registrada, título de la donación, una descripción, horario para la recolección, una fotografía y temporalidad de la donación. Otra de las funciones que se incluyen en este módulo, es la de poder rechazar solicitud de tomar donación de algún usuario, por motivos de que no se llevó a un acuerdo satisfactorio para efectuar la entrega de los alimentos.	

Tabla 5. Historia de usuario registro de usuarios

Historia de Usuario	
<b>Numero:</b> HU04	<b>Usuario:</b> Donante
<b>Nombre historia:</b> Rechazar donación	

<b>Priorización:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de usuario:</b> 10	<b>Número de Iteración:</b> 3
<b>Desarrollador:</b> Oweimar Mena	
<b>Descripción:</b> El donante podrá rechazar la solicitud o petición de una donación efectuada por un solicitante de donación por incompatibilidades en su solicitud o porque no hay común acuerdo. La donación podrá ser rechazada solamente por el usuario que la creo la donación inicialmente.	

Tabla 6. Historia de usuario rechazar donación

Historia de Usuario	
<b>Numero:</b> HU05	<b>Usuario:</b> Solicitante de donación.
<b>Nombre historia:</b> Pedir donación	
<b>Priorización:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de usuario:</b> 30	<b>Número de Iteración:</b> 3
<b>Desarrollador:</b> Oweimar Mena	
<b>Descripción:</b> Esta función está diseñada para los usuarios que necesitan los alimentos en calidad de donación, ya que pueden mirar las donaciones cercanas a su ubicación actual por medio del componente de georreferenciación.	

Tabla 7. Historia de usuario pedir donación

Historia de Usuario	
<b>Numero:</b> HU06	<b>Usuario:</b> Solicitante de donación.
<b>Nombre historia:</b> Retirar donación	
<b>Priorización:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de usuario:</b> 10	<b>Número de Iteración:</b> 3
<b>Desarrollador:</b> Jorge Quintana	
<b>Descripción:</b> Permitir a los Donantes retirar una donación de manera manual por imprecisión en la información o porque se presentó alguna novedad. La donación podrá ser retirada solamente por el usuario que la creo inicialmente.	

Tabla 8. Historia de usuario retirar donación

Historia de Usuario	
<b>Numero:</b> HU07	<b>Usuario:</b> Administrador.
<b>Nombre historia:</b> Administrar aplicación	
<b>Priorización:</b> Media	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de usuario:</b> 40	<b>Número de Iteración:</b> 3
<b>Desarrollador:</b> Oweimar Mena	
<b>Descripción:</b> Permitir que el usuario administrador controle y monitoree las donaciones que se realizan, generar reportes, recibir PQRS. Permitir cancelar usuarios que incumplan las políticas y retirar lista de donaciones por inconsistencias e información errada.	

Tabla 9. Historia de usuario administrar aplicación

Historia de Usuario	
<b>Numero:</b> HU08	<b>Usuario:</b> Donante
<b>Nombre historia:</b> Georreferenciar donación	
<b>Priorización:</b> Alta	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de usuario:</b> 50	<b>Número de Iteración:</b> 3

<b>Desarrollador:</b> Jorge Quintana	
<b>Descripción:</b> Usando el API de Google Maps, de manera gráfica se debe permitir geolocalizar la posición de los donantes y las donaciones. Uso de este componente para listar las donaciones cercanas al solicitante de donación. El GPS de dispositivo debe estar encendido para la toma de las coordenadas por parte de la aplicación.	

*Tabla 10. Historia de usuario georreferenciar donación*

Historia de Usuario	
<b>Numero:</b> HU09	<b>Usuario:</b> Donante, Administrador, Solicitante de donación.
<b>Nombre historia:</b> Aplicar filtros y búsqueda	
<b>Priorización:</b> Baja	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de usuario:</b> 40	<b>Número de Iteración:</b> 4
<b>Desarrollador:</b> Oweimar Mena	
<b>Descripción:</b> La aplicación debe permitir aplicar filtros y búsquedas a las donaciones para identificar más eficientemente los elementos que interesen a los usuarios. Los filtros y búsquedas se podrán hacer por ubicación, fecha, título de la donación y tiempo de expiración.	

*Tabla 11. Historia de usuario georreferenciar donación*

Historia de Usuario	
<b>Numero:</b> HU10	<b>Usuario:</b> Donante, Administrador
<b>Nombre historia:</b> Generar reportes	
<b>Priorización:</b> Baja	<b>Riesgo de desarrollo:</b> Medio
<b>Puntos de usuario:</b> 30	<b>Número de Iteración:</b> 4
<b>Desarrollador:</b> Jorge Quintana	
<b>Descripción:</b> Se debe permitir al usuario donante y administrador generar reportes sobre las donaciones efectuadas, con fines estadísticos, pudiendo consolidar información de: Cantidad de donaciones efectuadas Lugares donde más hay donaciones Cantidad de donaciones tomadas Cantidad de donaciones retiradas Cantidad de donaciones no solicitadas Cantidad de donaciones no aprobadas.	

*Tabla 12. Historia de usuario generar reportes*

### 6.1.1 Estimación Puntos de Usuario

El equipo de desarrollo usando las tarjetas formato de Fibonacci valores (0, 0.5, 1, 2, 3, 5, 8, 13, 20, 40, 100.) efectúa la estimación de punto de historia para medir el esfuerzo que llevaría realizar cada una de las historias de usuario definidas en el levantamiento de información.

Historia de Usuario	Miembro		Estimación media	Prioridad
	Oweimar Mena	Jorge Quintana		
HU01	20	20	20	Alta

HU02	20	40	30	Media
HU03	50	50	50	Media
HU04	10	10	10	Media
HU05	25	35	30	Alta
HU06	10	10	10	Alta
HU07	30	50	40	Media
HU08	50	50	50	Media
HU09	50	30	40	Baja
HU10	20	30	30	Baja
Total	285	325	310	

Tabla 13. Estimación puntos historias de usuario

### 6.1.2 Atributos de calidad

Atributo de Calidad:	Portabilidad		
	ID	Descripción	Prioridad
Adaptación	P01	La aplicación debe ser capaz de adaptarse a varios dispositivos móviles, tabletas y a un entorno de escritorio sin agregar mayor complejidad.	Media, media
Capacidad de Instalación	P02	La aplicación debe poder ser iniciado o ejecutado en una computadora sin en convenientes de instalación o configuración.	Media, media

Tabla 14. Atributo de calidad portabilidad

Atributo de Calidad:	Usabilidad		
	ID	Descripción	Prioridad
Operabilidad	U01	La aplicación debe ser intuitiva y de fácil operar por los usuarios.	Alta baja
Accesibilidad	U02	El sistema debe ser fácilmente accesible por los usuarios, a través una conexión de internet con una velocidad promedio.	Alta, media

Tabla 15. Atributo de calidad Usabilidad

Atributo de Calidad:	Funcionabilidad		
	ID	Descripción	Prioridad
Adecuado	F01	El sistema justarse a las necesidades de los usuarios para registrar y solicitar los alimentos en donación.	Alta, alta
Conformidad	F02	Cumple con los criterios y buenas prácticas del desarrollo y terminal legales.	Alta, alta

Tabla 16. Atributo de calidad funcionalidad

Atributo de Calidad:	Seguridad		
	ID	Descripción	Prioridad
Autenticidad	S01	El sistema debe permitir que los usuarios autenticuen mediante un usuario y contraseña.	Alta, alta
Integridad	S02	Los datos ingresados de las donaciones deben reflejarse exactamente, en el listado	Alta, alta

		de alimentos disponibles para ser tomados.	
Confiabilidad	S03	El sistema debe ser seguro para sus usuarios, que no exista riesgo de pérdida de información privada.	Alta, alta

Tabla 17. Atributo de calidad seguridad

## 6.2 DISEÑO Y ARQUITECTURA

Partiendo de las necesidades de software y restricciones de tiempo, recursos y basados en un análisis del estado del arte; para el desarrollo de la aplicación para la gestión de donaciones Share Food, en el back-end se utilizó el patrón de arquitectura en capas y MVC (modelo, vista y controlador), el patrón de desarrollo Facade, en las interfaces graficas del front-end se utilizó razor pages, bootstrap, JQuery y JavaScript.

Por parte del equipo de desarrollo se seleccionaron estas configuraciones ya que, por sus características más notables como rápido desarrollo, empleo de menos desarrolladores, más fácil de probar, conveniente para desarrollos pequeños y por la curva de aprendizaje que el equipo de desarrollo ya tenía, nos permitía cumplir con los objetivos propuestos y reducir la deuda técnica que podría surgir a lo largo del desarrollo.

De ser necesario esta arquitectura, permitiría aceptar más conexiones de usuarios a través de un escalamiento vertical agregando más capacidad al hardware (memoria RAM y procesamiento) y/o escalamiento horizontal, creando nuevas instancias del aplicativo y desplegándolas en nuevos servidores.

### 6.2.1 Alcance y Restricciones

#### 6.2.1.1 Alcance

Está dirigido al diseño y construcción de una aplicación informática especificada en los objetivos del proyecto, para la gestión de donaciones, que permita a los donantes crear listas de alimentos y definir unos atributos para que las personas necesitadas puedan acceder a ellos fácilmente y se pueda establecer un traspaso de estos recursos alimenticios. La aplicación está diseñada para ser publicada en internet y pueda ser accedida por medio de un navegador web. La aplicación web, contará con las siguientes características y funcionalidades:



- **Módulo de autenticación (Login)**

Permitirá acceder a la aplicación mediante un usuario y una contraseña, igualmente se podrá validar la autenticación mediante el uso de una cuenta registrada en algunas de las plataformas de Google + y Facebook.

- **Módulo de Registro**

Para poder hacer uso de las funciones de la aplicación es necesario estar registrado, lo cual se puede hacer suministrando datos básicos, como nombres, apellidos, ubicación mediante geoposicionamiento con el mapa de Google. Este módulo igualmente permitirá recuperar las contraseñas olvidadas.

- **Módulo para Donación**

El donante podrá hacer la donación mediante la creación de una lista de alimentos que incluya la cantidad, la ubicación geográfica mediante el mapa de Google, indicando un título de la donación, una descripción, horario para la recolección, una fotografía de los elementos y cuánto tiempo estará disponible la donación. Asimismo, podrá retirar una donación de manera manual por imprecisión en la información o porque se presentó alguna novedad.

Otra de las funciones que se incluyen en este módulo, es la de poder rechazar solicitud de tomar donación de algún usuario, por motivos de que no se llevó a un acuerdo satisfactorio para efectuar la entrega de los alimentos.

- **Modulo Pedir Donación**

Este módulo fue diseñado para que los usuarios que necesitan los comida en calidad de donación pueden visualizar las ofertas cercanas a su ubicación actual por medio del módulo de georreferenciación.

- **Modulo Administración**

Este módulo se incorpora a la aplicación con el fin de controlar y monitorear las donaciones que se realizan, generar reportes, recibir PQRS, asimismo, cancelar usuarios y retirar

donaciones porque violación de los términos o efectuar procedimientos no acordes con el objetivo de la aplicación.

- **Notificaciones**

Notificación de donación solicitada

Notificación de aprobación de donación

Notificación de donación cercana a su ubicación

Notificación de donación expirada

- **Filtros y Búsquedas**

Filtro donaciones cercanas (< 10 km)

Donaciones recientes

Filtro tipo de donación

Filtro periodo de expiración donación

- **Diseño Interfaces de Usuario**

Diseño de botones

Diseño de listas desplegables

Diseño de menús

Paleta de colores

- **Reportes de Donaciones**

Los usuarios podrán consultar y visualizar la cantidad de listas de donaciones creadas y tomadas, para efecto estadísticos.

- **Pruebas**

Para garantizar que la aplicación

Pruebas de Código Estático

Pruebas de Integración

Pruebas de Seguridad

## ▪ Construcción Términos y Condiciones

### Condiciones de uso

#### a. Propietario de esta página web

Este aplicativo web es de propiedad y su operación está a cargo de ShareFood y ofrece a los navegantes la posibilidad de hacer y solicitar donaciones de alimentos. Accediendo a la aplicación web, usted da su aprobación de haber leído, entendido y aceptado estar sujeto a la aplicación de estos Términos.

#### b. ¿Quién puede usar tu página web; ¿Cuáles son los requisitos para crear una cuenta?

Para usar este aplicativo web y/o recibir los servicios, deberás tener una edad de 18 años, o la mayoría de edad legal en tu país. No tendrás permitido utilizar este aplicativo web y/o recibir servicios, si hacerlo está prohibido en tu país de origen o en virtud de cualquiera de las leyes o regulaciones en cada caso.

#### c. Posesión de propiedad intelectual, derechos de autor y logos.

El Servicio prestado y todos los elementos incluidos o transferidos, incluyendo, sin limitación, el software, las imágenes, el texto, los gráficos, el logotipo, derechos de autor, son la propiedad exclusiva de ShareFood. Salvo que se indique explícitamente en este documento, no se considerará que nada en estos Términos de una licencia en o bajo ninguno de dichos Derechos de Propiedad Intelectual, estas aceptando no vender, alquilar, licenciar, distribuir, modificar, reproducir, copiar, exhibir, editar, transmitir, adaptar o crear software derivado de los mismos.

#### d. Derecho a suspender o cancelar la cuenta de usuario

Podemos terminar o suspender de manera permanente o temporal tu acceso al servicio del aplicativo sin previo aviso y responsabilidad por cualquier razón o si a nuestra sola determinación tú violas alguna disposición de estos Términos o cualquier ley o regulación aplicable. Tu estas en la posibilidad de finalizar el uso y solicitar que se cancele tu cuenta y/o cualquier servicio en cualquier momento, esto sin ningún perjuicio.

#### e. Limitación de responsabilidad

En la máxima medida permitida por la ley aplicable, ShareFood no asume responsabilidad alguna por (i) errores, errores o inexactitudes de contenido; (ii) lesiones en la persona o propiedad, de cualquier naturaleza, como resultado del acceso o uso del servicio; y (iii) cualquier acceso no autorizado o el uso de nuestros servidores y/o toda la información personal almacenada en los mismos.

Igualmente, en ninguno de los casos ShareFood será el responsable por daños indirectos, punitivos, incidentales, entre otros daños por pérdida de beneficios, buena voluntad, uso, datos u otras pérdidas intangibles, que surjan de o estén relacionadas con el uso o la imposibilidad de utilizar el aplicativo.

### **Política de Privacidad y el Tratamiento de Datos (Aviso)**

Las políticas de privacidad y tratamiento de la información han sido diseñadas para poner en conocimiento de los usuarios, que su información se colecta por medio de la utilización de la aplicación ShareFood. Cómo es usada, compartida y opciones con respecto a sus datos.

El usuario nos entrega ciertos datos los cuales son considerados personales por la legislación colombiana conforme a la ley 1581 de 2012. Los datos recabados serán tratados directamente por la administración de la aplicación web ShareFood, quien a su vez será encargada y responsable del tratamiento, aplicando las disposiciones legales vigentes.

Los datos aportados cuentan con los estándares de seguridad y serán guardados en la base de datos de usuarios de ShareFood en un servidor de base de datos SQLServer Express, la cual tiene como finalidad el correcto funcionamiento de la aplicación en cuestión.

Se realizará el tratamiento de los datos suministrados por los usuarios conforme a la Ley, con la absoluta reserva hacia los datos y la no divulgación de los mismos. La información recolectada en los procesos de donación de alimentos, objetivo principal de esta aplicación, podría ser utilizada para generar estadística para ayudar a mejorar el proceso de donación de alimentos.

ShareFood puede llegar a almacenar la siguiente información personal, por medio de la aplicación: Nombre, nombre de usuario, contraseña: para fines de autenticación en la aplicación, ciudad, teléfono celular, correo electrónico, ubicación geográfica y datos de usuario a redes sociales, cuando se configure la cuenta para acceder por medio de estas plataformas.

### 6.2.1.2 Restricciones

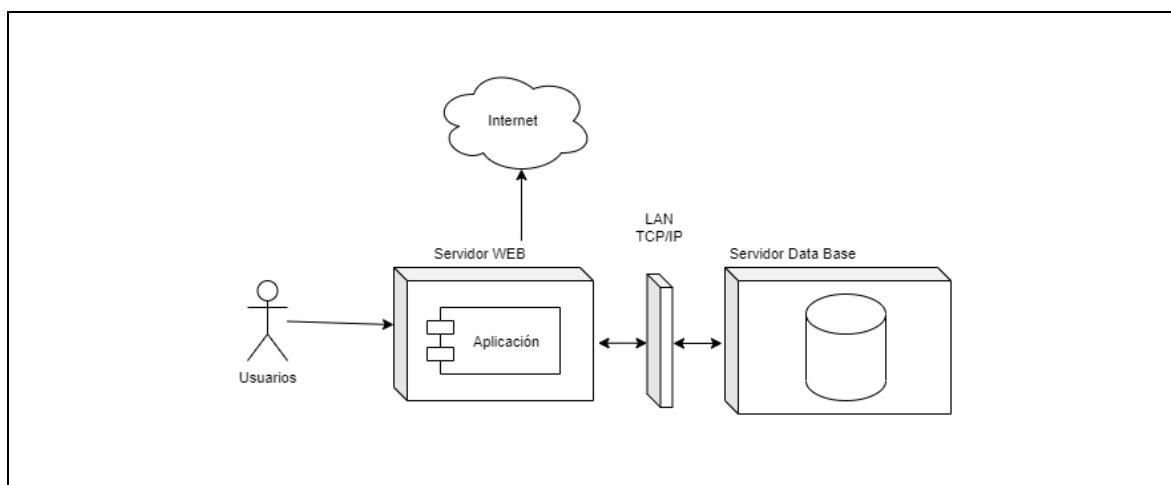
**Tiempo:** El tiempo empleado para todos el diseño y construcción fue el periodo académico de la especialización en ingeniería de software.

**Recursos:** Desde el inicio del planteamiento y solución del problema, el equipo de trabajo definió que se emplearían herramientas de desarrollo y gestión libres, como también recursos tecnológicos propios, tales como equipos de cómputo y tiempo para las distintas etapas del proyecto.

### 6.2.1 Diagrama de Despliegue

A continuación, se muestra el diagrama de Despliegue que pretende mostrar la relación entre los componentes de hardware que se manejan frente al sistema.

**Figura 5. Diagrama de despliegue**



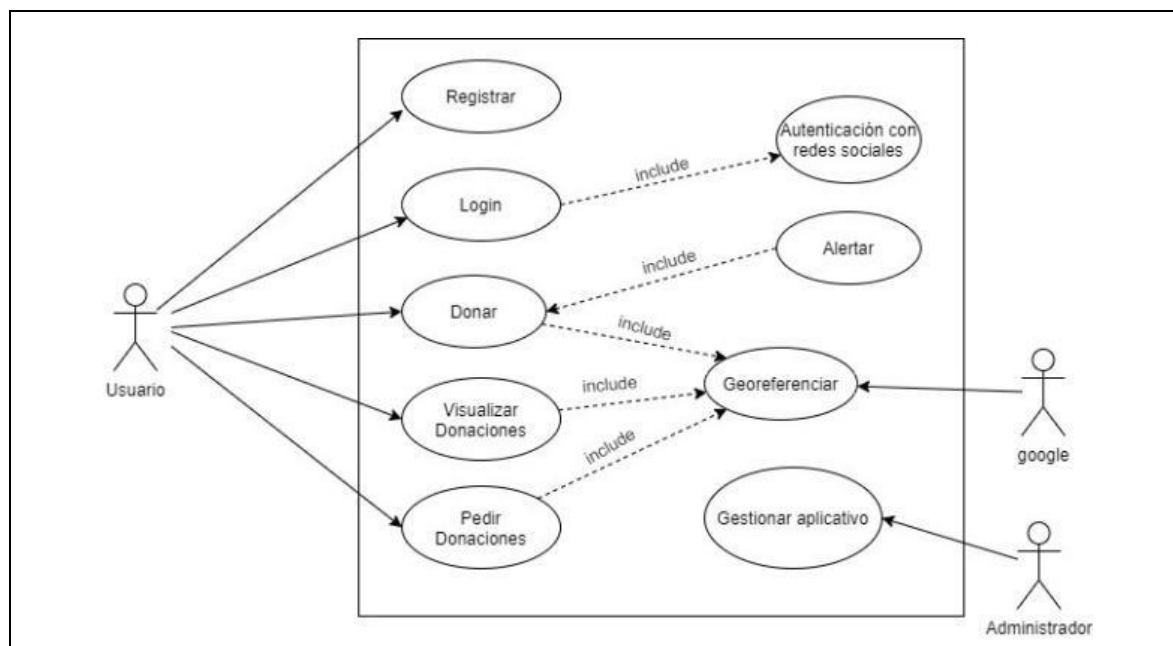
Título: Diagrama general de componentes	ID: 01	Profundidad: 2	Nomenclatura: UML
Arquitecto: Integrante proyecto de grado especialización Ing. De Software		Fecha: 25-10-2020	Versión: 1.0

Fuente: Elaboración propia

### 6.2.1 Caso de uso Arquitecturalmente Relevante

En el siguiente grafico se muestra el caso uso principal de la aplicación, relativo al proceso de donar y solicitar donaciones de alimentos.

**Figura 6. Caso de uso Arquitecturalmente Relevante**



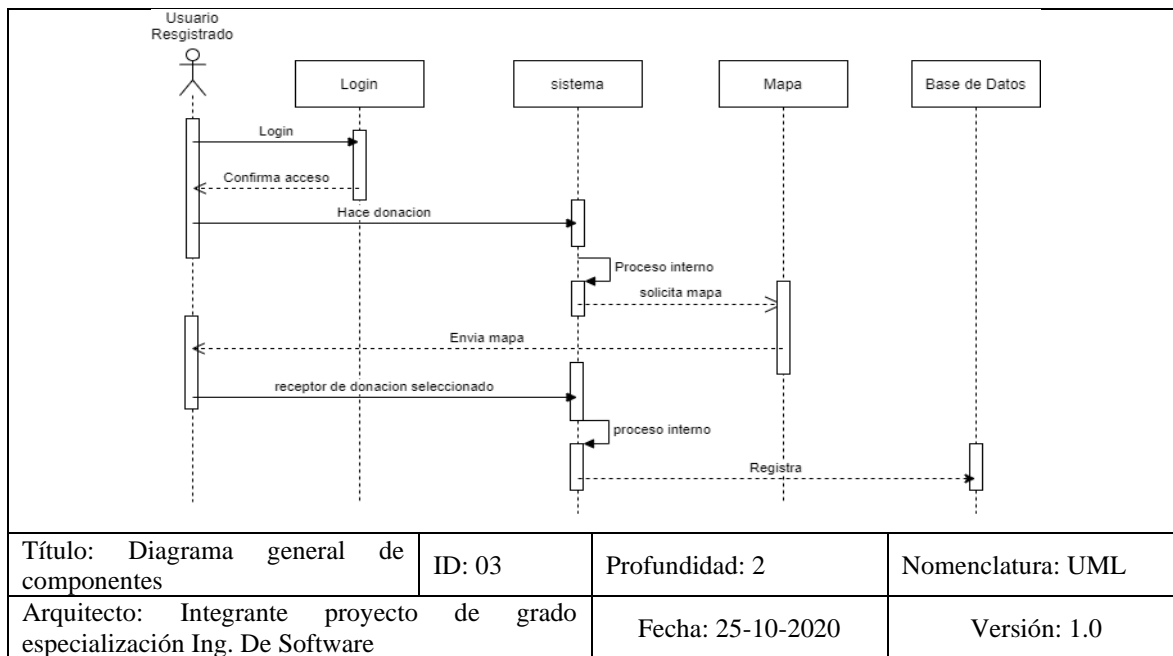
Título: Caso de uso Arquitecturalmente Relevante	ID: 02	Profundidad: 2	Nomenclatura: UML
Arquitecto: Integrante proyecto de grado especialización Ing. De Software		Fecha: 25-10-2020	Versión: 1.0

Fuente: Elaboración propia

### 6.2.2 Diagrama de Secuencia

Se relaciona el diagrama de secuencia, que demuestra la relación entre los diferentes objetos del sistema, iniciando por el login hasta finalizar el registro de una donación de alimentos.

**Figura 7. Diagrama de Secuencia**

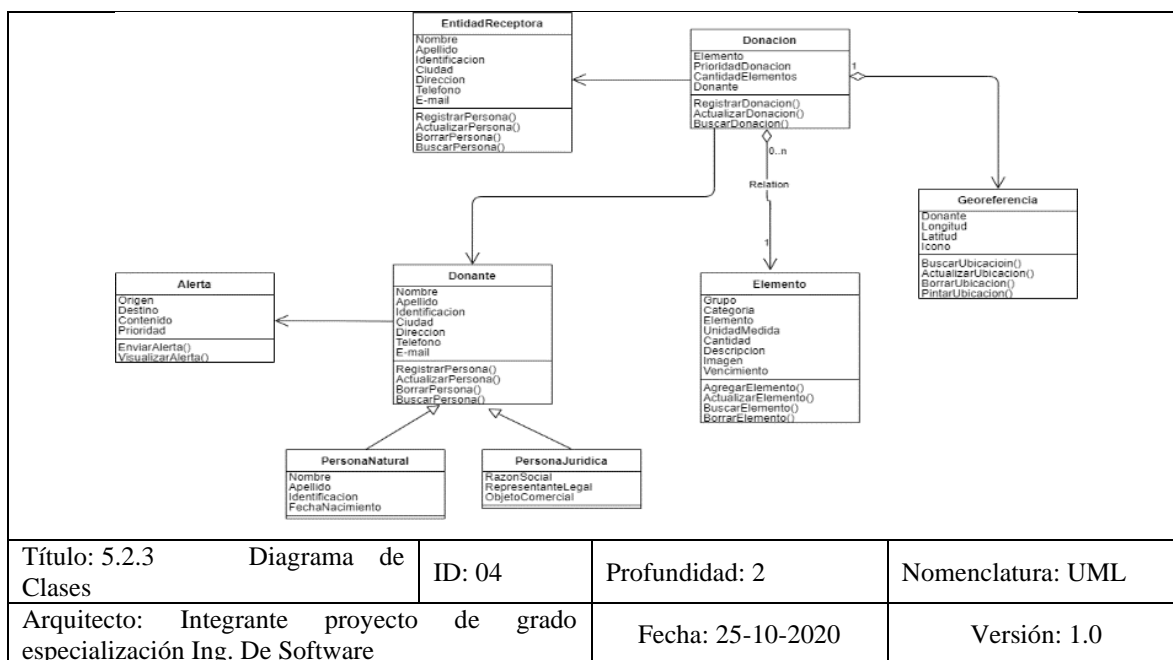


Fuente: Elaboración propia

### 6.2.3 Diagrama de Clases (Vista Lógica)

A continuación, se muestra el diagrama con las principales clases, que describe la estructura del sistema.

**Figura 8. Diagrama de Clases - Vista Lógica**

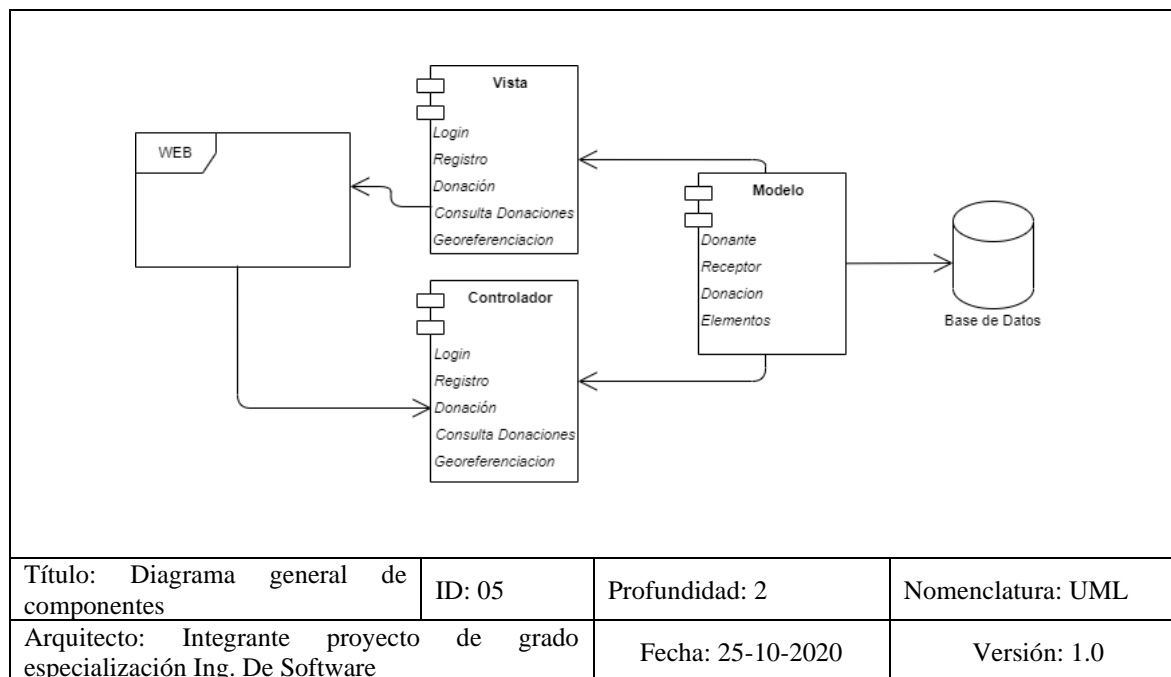


Fuente: Elaboración propia

### 6.2.4 Vista de Desarrollo (Diagrama de Componentes)

A continuación, se describe el diagrama de componentes, que muestra, como su nombre lo indica, los diferentes componentes independientes de software en que se divide el sistema.

**Figura 9. Diagrama de componentes**



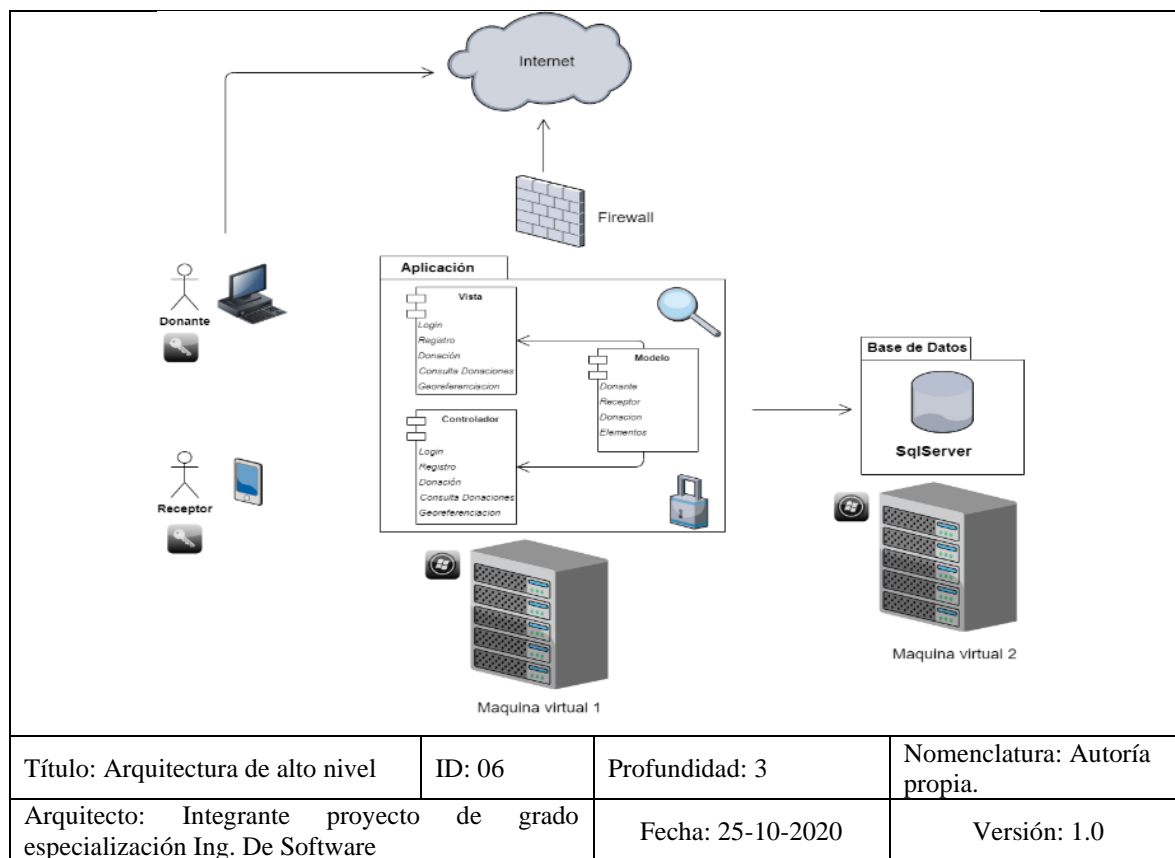
Fuente: Elaboración propia

### 6.2.5 Arquitectura de alto Nivel

Para la representación de este escenario se tuvieron en cuenta varios aspectos, con el fin de mostrar al stakeholder (usuarios finales), en términos generales el funcionamiento del sistema, como interactúan los diferentes componentes y como se puede tener acceso al sistema desde cualquier dispositivo conectado a internet. La base de datos y la aplicación web son desplegadas desde dos máquinas virtuales, con el fin de ahorrar recursos de hardware y dado el caso de necesitarse más capacidad, se pueden desplegar dos instancias de la aplicación y balancearlas para aceptar más peticiones de los usuarios.

A nivel de seguridad, el usuario y la contraseña se representa por una llave, el acceso al sistema con un candado, haciendo analogía al mundo real. Igualmente, el firewall está en medio del sistema e internet, para protegerlo de ataque desde el exterior.

**Figura 10. Arquitectura de alto Nivel**



Fuente: Elaboración propia

## 6.3 CONSTRUCCIÓN

### 6.3.1 Lenguaje de programación

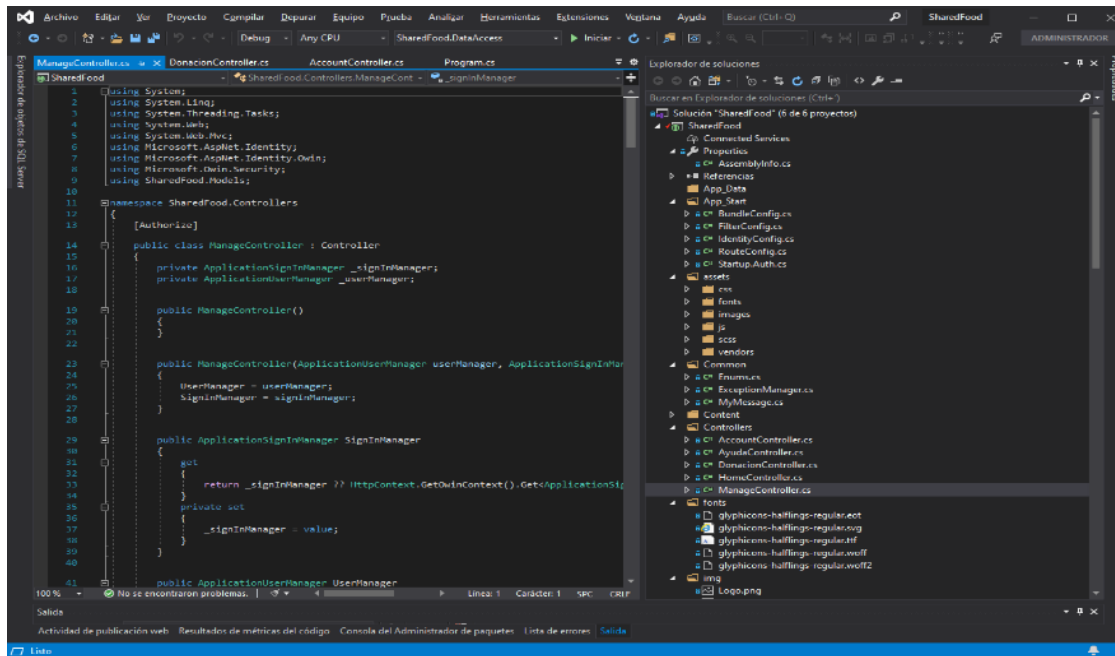
- Para el backend se utilizó el lenguaje de programación C# (C sharp)
- Para el frontend el lenguaje Javascript
- Para la base de datos el lenguaje SQL con el motor SQLServer Express de Microsoft.

### 6.3.2 Herramientas de desarrollo utilizadas

Para la codificación de la aplicación se usó el IDE Visual Studio en su versión 2019, por su versatilidad y gran cantidad de herramientas que hacen más cómodo, práctico y seguro el desarrollo de aplicaciones.



**Figura 11. Entorno de Desarrollo Integrado Visual Studio**

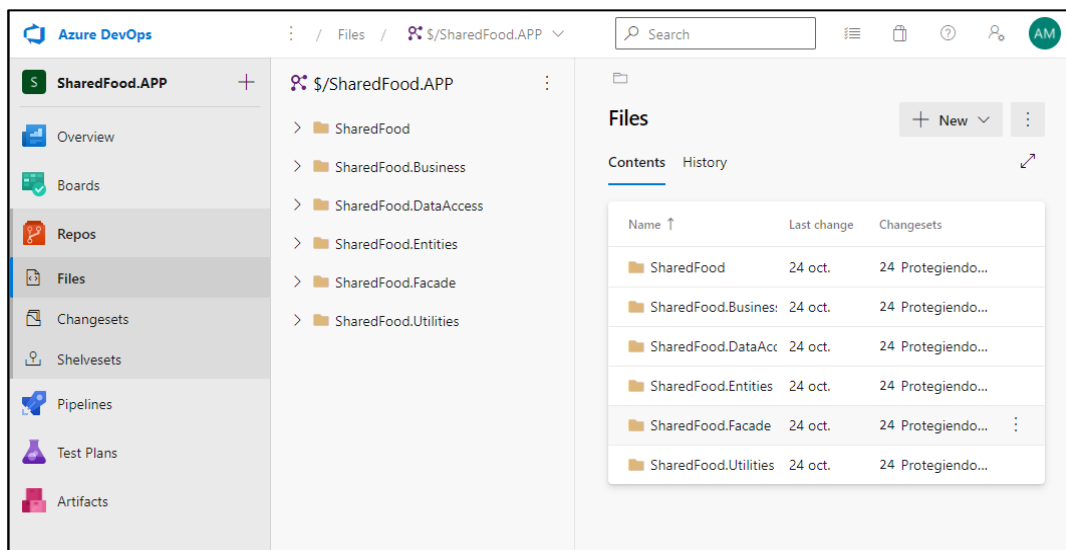


Fuente: Elaboración propia captura pantalla

### 6.3.3 Gestión del código

Se utilizó el repositorio en Azure DevOps, para almacenar el código fuente y llevar las versiones del proyecto.

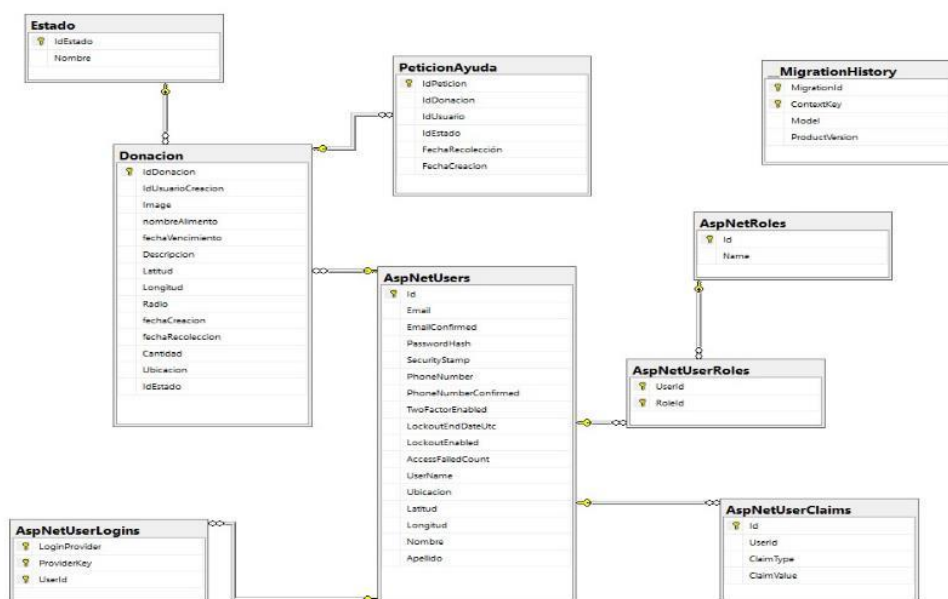
**Figura 12. Interfaz Azure DevOps**



Fuente: Elaboración propia captura pantalla

### 6.3.4 Modelo de base de datos

**Figura 13.** *Diagrama Entidad Relación Base de Datos*

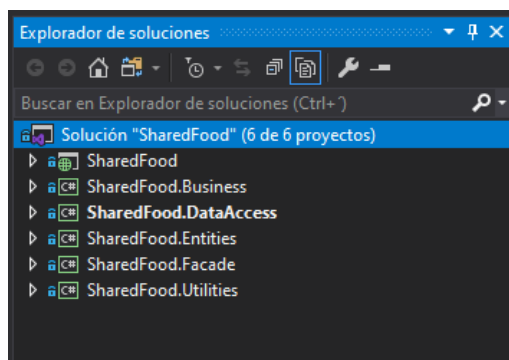


*Fuente: Elaboración propia con el modelador de SQLServer Express*

### 6.3.5 Diseño de la aplicación

Estructuración de la aplicación se llevó a cabo en capas y en la capa principal ShareFood se implemente el MVC (Modelo Vista controlador). En visual estudio cada capa se define como un proyecto diferente y la comunicación entre las mismas se efectúa mediante la referencia de las dependencias entre los proyectos, capa por capa.

**Figura 14.** *Proyectos de las capas de la aplicación*

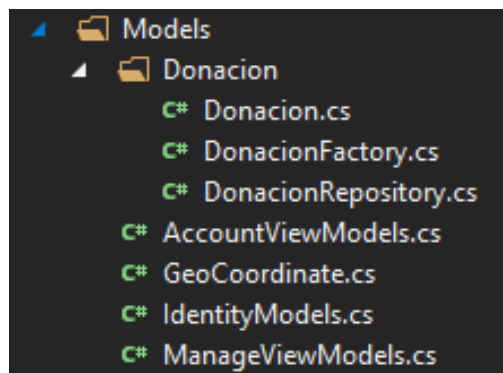


*Fuente: Elaboración propia captura pantalla de Visual Studio.*

### 6.3.5.1 Modelo

Las clases implementadas definidas como el modelo de la aplicación de acuerdo con el patrón de desarrollo MVC.

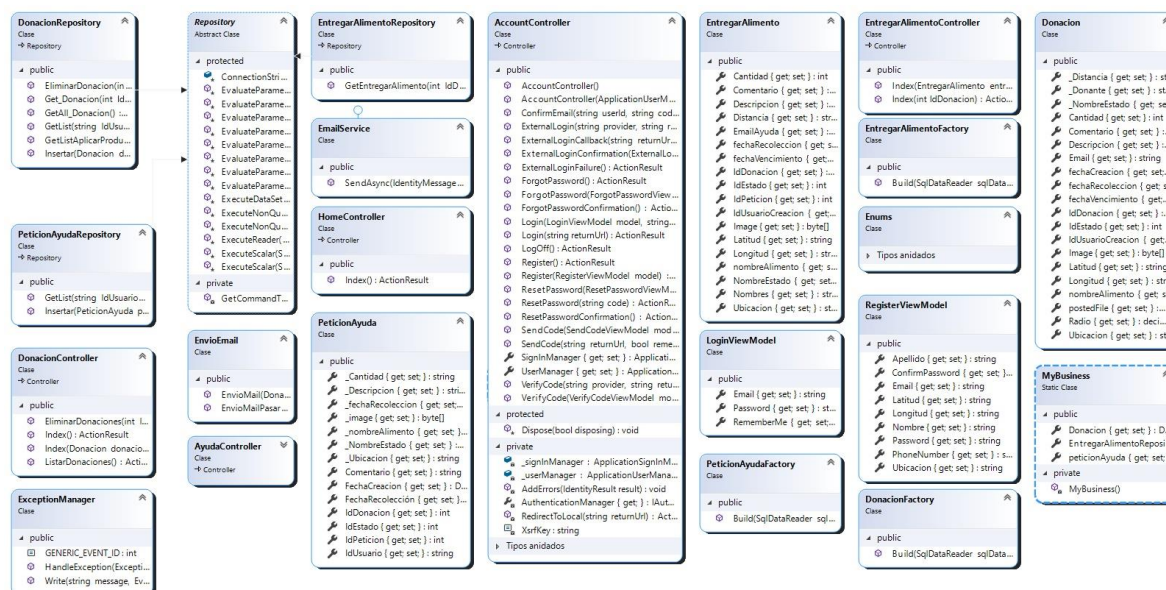
Figura 15. Archivos de clases del Modelo



Fuente: Elaboración propia captura pantalla de Visual Studio.

Igualmente haciendo uso de la herramienta de visualización de clases de Visual Studio, se generan el listado de clases codificadas en el proyecto, las cuales hacen parte del Back-end.

Figura 16. Relación de las Clases de la aplicación

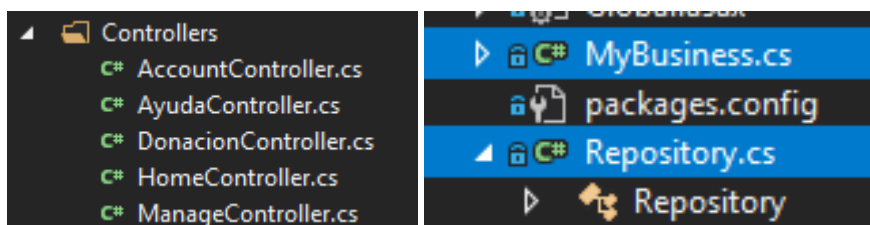


Fuente: Elaboración propia captura pantalla de Visual Studio.

### 6.3.5.2 Controlador

Las clases implementadas definidas como los controladores de la aplicación de acuerdo con el patrón de desarrollo MVC, que se comunicaran con las vistas o formularios del Font-end.

**Figura 17. Archivos de los Controladores**

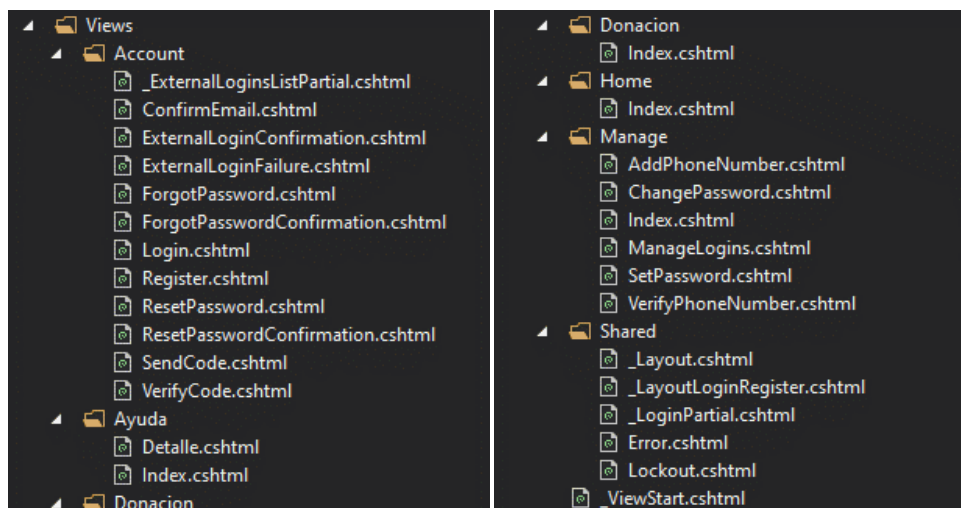


Fuente: Elaboración propia captura pantalla de Visual Studio.

### 6.3.5.3 Vista

Los documentos mixtos de html, C# y razor con extensión “.cshtml”, implementados como las vistas o formularios del Font-end de la aplicación de acuerdo al patrón de desarrollo MVC.

**Figura 18. Archivos del Font-end**



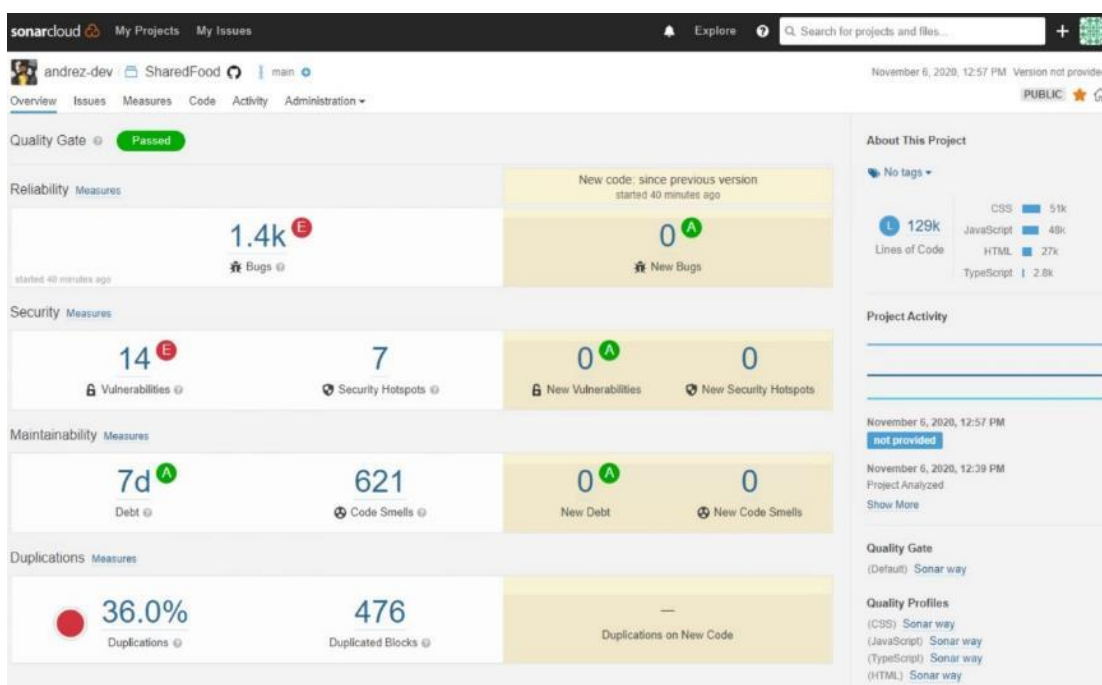
Fuente: Elaboración propia captura pantalla de Visual Studio.

## 6.4 PRUEBAS

### 6.4.1 Pruebas de código estático

Para la comprobación de pruebas de código estático, que permiten verificar la calidad del software, encontrar bugs, “code smells”, entre otras características para asegurar que se escriba de código limpio, se utilizó la herramienta Sonar Cloud, en un ambiente en la nube, cuyos resultados se pueden observar en la siguiente figura.

**Figura 19. Resultados Sonar Cloud**



Fuente: Elaboración propia captura pantalla de SonarCloud.

*“Quality Gate: Quality Gate es un conjunto de condiciones booleanas basadas en medidas. Le ayuda a saber inmediatamente si su proyecto está listo para producción. Si su estado actual no es Aprobado, verá qué medidas causaron el problema y los valores necesarios para aprobar”.*

A pesar de que, durante las pruebas, el software SonarCloud arrojó varios indicadores de “code smell” y métricas a mejorar en el código, el valor “Quality Gate” marcado con verde “Passed”, indica que el código puede estar listo para pasar a producción. El alcance de este trabajo no contemplaba la refactorización, por lo cual estos indicadores podrían ser corregidos en otra fase del proyecto.

### 6.4.2 Pruebas de integración

Posterior la codificación y pruebas de código, el equipo de desarrollo verificó que los componentes de software funcionaran armónicamente, mediante el examen de las diferentes interfaces que componen la aplicación, identificando posibles fallas o problemas.

#### *Descripción de las actividades a realizar*

# Caso de Prueba	Componente	Descripción de lo que se Probará	Prerrequisitos
CP001	Login de Acceso a la aplicación	Impedir el acceso sin no se proporciona un usuario y contraseña validos	Haber creado usuario y contraseña en la aplicación
CP002	Pantalla para crear lista de donación.	Crear una lista completa de una donación de alimentos	Estar registrado y logueado en la aplicación
CP003	Pantalla para tomar donaciones	Tomar una donación de alimentos.	Que existan listas de donaciones en la aplicación.
CP004	Pantalla para administrar la aplicación	Visualizar todas las listas de donaciones, generar reportes y eliminar una lista de donación al azar.	Que existan listas de donaciones.

Tabla 18. Listado de casos de prueba

#### *Resultado*

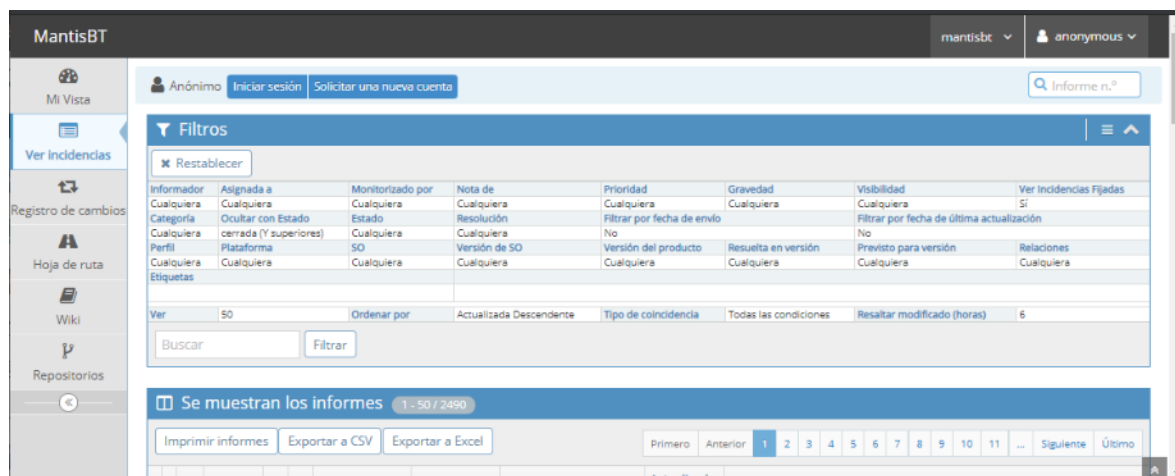
# Caso de Prueba	¿Estatus Ok?	Observaciones
CP001	Ok	Ninguna
CP002	Ok	Ninguna
CP003	Ok	Ninguna
CP004	Ok	Ninguna

Tabla 19. Resultado de pruebas

### 6.4.3 Pruebas Funcionales

Para la gestión y control de las pruebas de funcionamiento de la aplicación Share Food, se usó la herramienta gratuita MantisBT, con la cual se pueden crear los casos de incidencias y seguimiento a las mismas, y realizar el cierre respectivo al finalizar la etapa.

**Figura 20. Resultados pruebas herramienta mantis**



Fuente: Elaboración propia captura pantalla de Mantis.

#### 6.4.4 Pruebas de Seguridad

Se utilizaron las técnicas de desarrollo seguro y se implantaron los controles basado en el proyecto de seguridad OWAST Top 10. Para las pruebas se seleccionaron los principales riesgos de seguridad y se utilizaron diversas herramientas para detectar la existencia de fallas en la aplicación. La relación de vulnerabilidad conocidas seleccionadas para ser probadas se relacionan a continuación.

OWASP	Procedimiento	Resultado
A1 - Inyección	Uso de herramienta de seguridad de hacking ético OWASP ZAP.	No se encontraron vulnerabilidades
A2 - Pérdida de Autenticación	Uso de herramienta de seguridad de hacking ético OWASP ZAP.	No se encontraron vulnerabilidades
A7 - Secuencia de Comandos en Sitios Cruzados (XSS)	Uso de herramienta de seguridad de hacking ético OWASP ZAP.	No se encontraron vulnerabilidades
A9 - Componentes con vulnerabilidades conocidas	Uso de herramienta de seguridad de hacking ético OWASP ZAP.	No se encontraron vulnerabilidades
A10 - Registro y Monitoreo Insuficientes	Uso de herramienta de seguridad de hacking ético OWASP ZAP.	No se encontraron vulnerabilidades

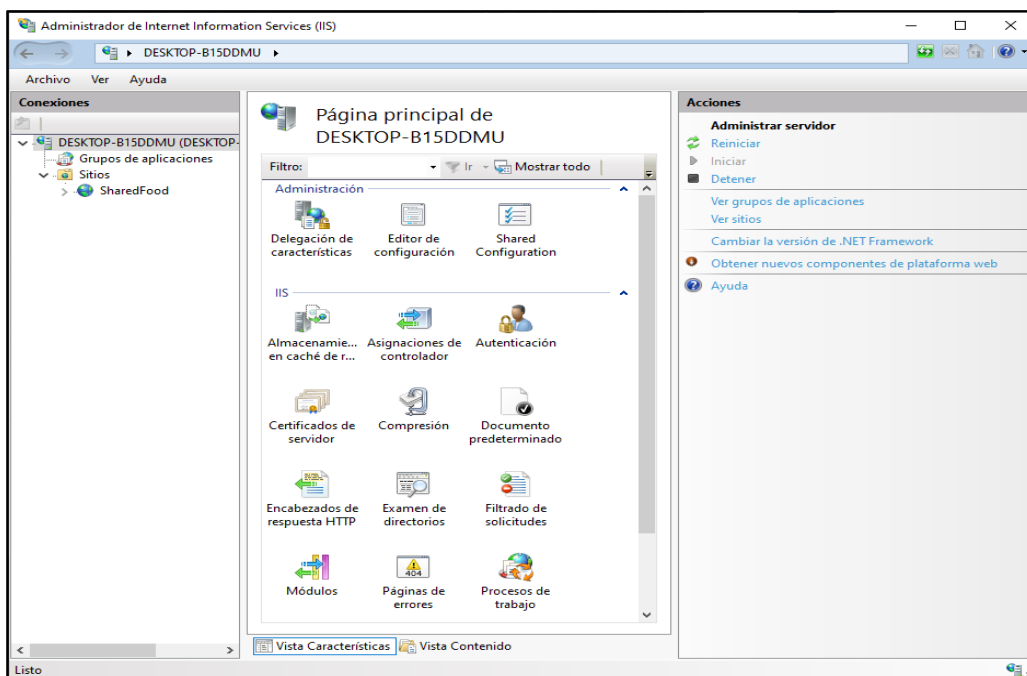
Tabla 20. Resultados pruebas de seguridad

## 6.5 INSTALACIÓN Y CONFIGURACIÓN

### 6.5.1 Servidor web

Para las pruebas y sustentación del proyecto ante los jurados de la universidad, la instalación de la aplicación web Share Food, se llevó a cabo en una computadora de escritorio, con el software de servidor IIS Internet Información Server, por la compatibilidad que ofrecía teniendo en cuenta que se utilizaron lenguajes y herramientas de Microsoft en el desarrollo del software.

*Figura 21. Servidor de aplicaciones IIS*



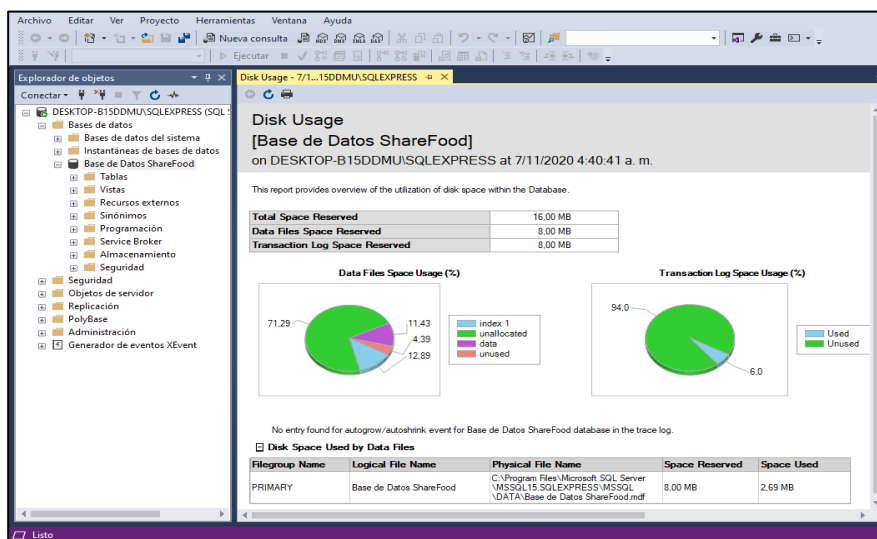
*Fuente: Elaboración propia captura pantalla de Internet Information Services .*

### 6.5.2 Servidor de base de datos

Como persistencia de datos para la aplicación, se utilizó el motor de base de datos SQL Server en su edición express, la cual es libre para ser usada tanto en desarrollo, pruebas y producción, pero con una limitante de espacio para almacenamiento. Lo cual no es impedimento para el proyecto en su alcance inicial.



**Figura 22. Servidor SQL base de datos**



Fuente: Elaboración propia captura pantalla de Visual Studio.

### 6.5.3 Aplicación WEB

Como resultado de las actividades del ciclo de desarrollo de software, siguiente los principios de código limpio, patrones de arquitectura y desarrollo, metodología de gestión scrum, software seguro, entre otros atributos de calidad, aprendidos en la especialización, se finalizó el diseño y construcción del aplicativo web para la gestión de donaciones de alimentos, con resultados satisfactorios. Las principales interfaces graficas del aplicativo se mostrarán a continuación.

**Figura 23. Login aplicación ShareFood**

Share Food  
find home-cooked food near you

Inicia sesión con

Or

Correo Electrónico

Contraseña

Recuérdame Olvidó su contraseña?

No soy un robot

**INICIAR SESIÓN**

¿Necesitas una cuenta? [¡Regístrate!](#)

Copyright © 2020. Todos los derechos reservados.

Fuente: Elaboración propia captura pantalla aplicación ShareFood.

**Figura 24. Formulario de registro ShareFood**

Share Food  
 And here we could find our joy

Registro

Correo Electrónico  
 jorge.luis.quintana@gmail.com.co

Contraseña  
 .....

Confirmar Contraseña  
 .....

Nombres

Apellidos

Teléfono

Ubicación

Latitud y Longitud  
 Latitud Longitud

Recuérdame

Regístrate

Fuente: Elaboración propia captura pantalla aplicación ShareFood.

**Figura 25. Registrar donación aplicación ShareFood**

Inicio Contacto Search

Hola por2011ghotmail.com!Cerrar sesión

Sube tu tarjeta

Arroz Diana x 5 kilos.jpg (12.21 KB)

Arroz Diana x 5 kilos.jpg Remove Borrar

Nombre del Alimento

Fecha de Vencimiento

Ubicación

Latitud y Longitud  
 Latitud Longitud

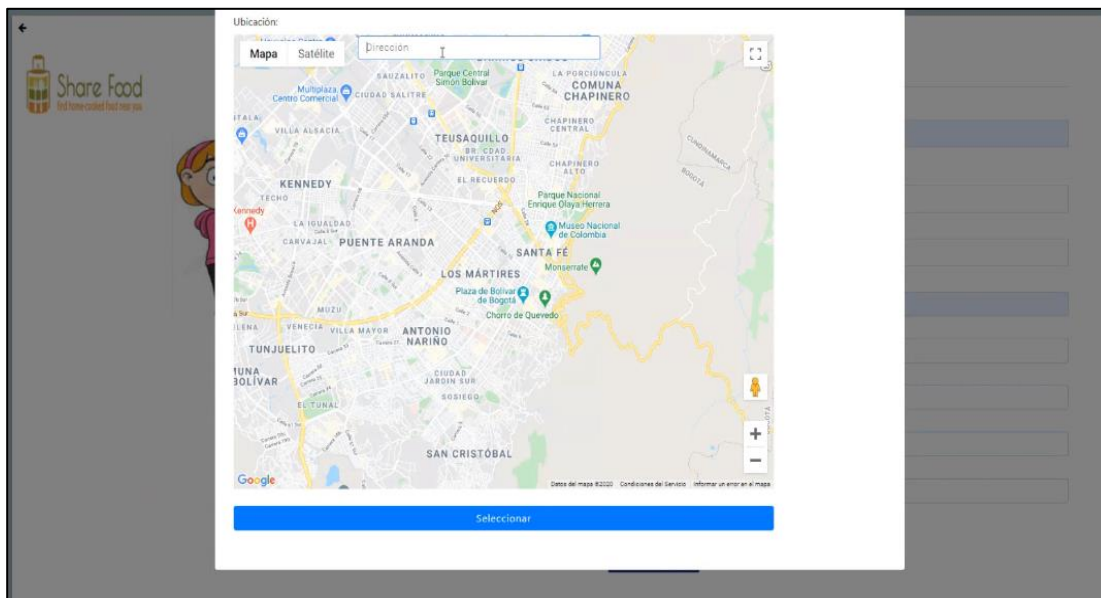
Fecha de recolección

Cantidad

Cantidad del producto

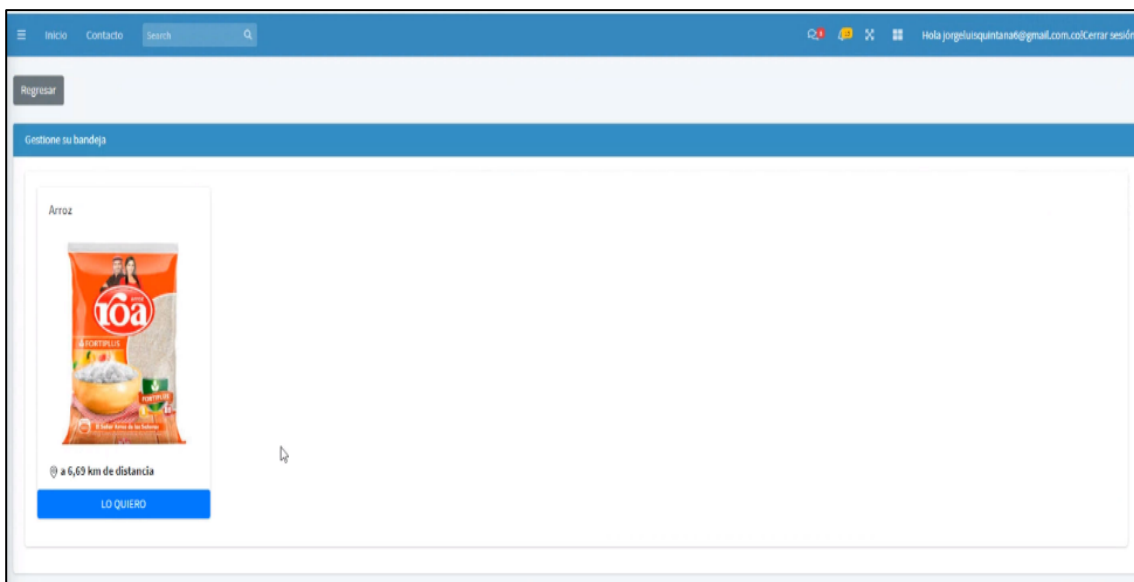
Fuente: Elaboración propia captura pantalla aplicación ShareFood.

**Figura 26. Georreferenciación aplicación ShareFood**



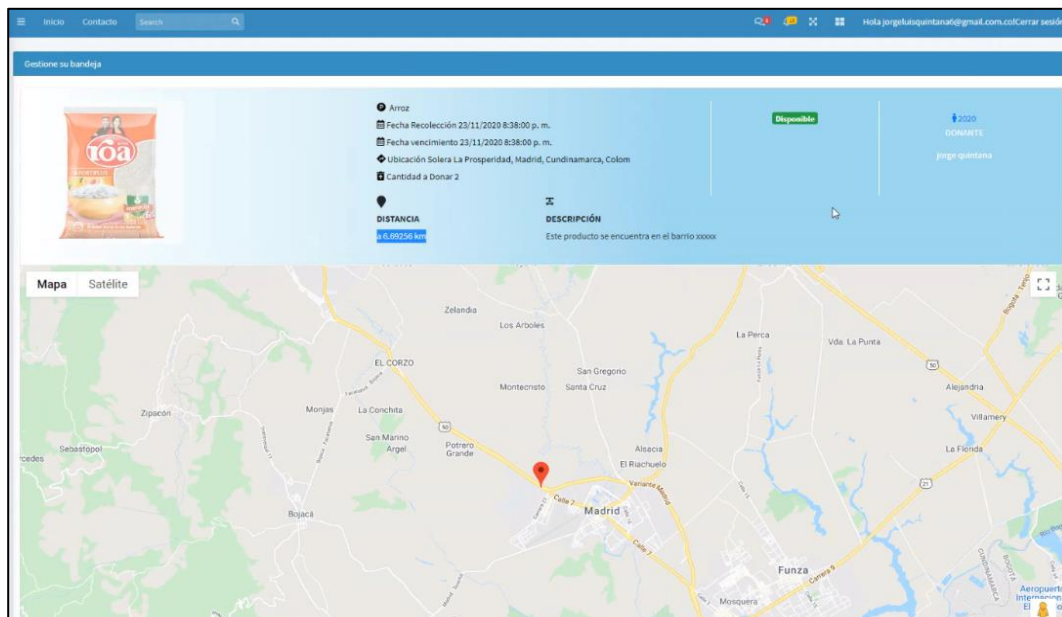
Fuente: Elaboración propia captura pantalla aplicación ShareFood.

**Figura 27. Gestión de la bandeja aplicación ShareFood**



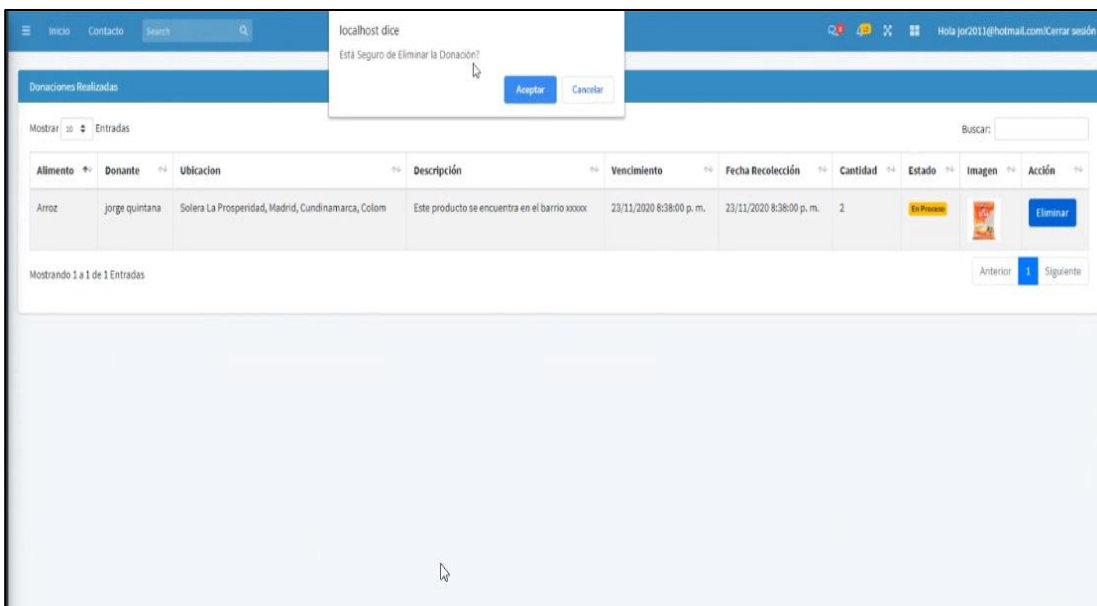
Fuente: Elaboración propia captura pantalla aplicación ShareFood.

**Figura 28. Detalles una donación aplicación ShareFood**



Fuente: Elaboración propia captura pantalla aplicación ShareFood.

**Figura 29. Control de donaciones aplicación ShareFood**



Fuente: Elaboración propia captura pantalla aplicación ShareFood.

## **7. ANEXOS**

1. Identificación de riesgos y controles de la aplicación web Share Food - Norma ISO 27002.

## 8. CONCLUSIONES

Como resultado del compromiso y dedicación del equipo de trabajo, se logró completar el producto final del proyecto de grado que consistía en desarrollar una aplicación web para la gestión de donaciones de alimentos, usando la metodología ágil para proyectos de software scrum adaptada a nuestras necesidades, teniendo en cuenta que todo el equipo estaba conformado únicamente por dos personas, lo cual representó un reto importante para nosotros la distribución de los roles, el cumplimiento del manifiesto de la agilidad y la culminación de todas las actividades.

La implementación del desarrollo usando el patrón de arquitectura por capas y modelo-vista controlador (MVC) nos permitió avanzar más rápidamente en la construcción de los distintos componentes de la aplicación, al requerir menos complejidad y mejor organización de las funciones lógicas de negocio en cada capa.

En la fase de pruebas se utilizó la herramienta de análisis de código estático SonarCloud, que nos permitió identificar debilidades en el código fuente y a su vez nos suministró información de cómo mejorar estas debilidades encontradas, proporcionándonos mejores prácticas, sin embargo, no se consideró en el alcance una fase de refactorización y mejoras del código por el poco tiempo que se contaba para desarrollar el proyecto.

## 9. REFERENCIAS

FAO, Banco de Alimentos, Artículo Tomado de la página web:

<http://www.fao.org/colombia/noticias/detailevents/en/c/BancodatosdelDepartamentoNacional/articulos>.

CENIRED, Artículo Periodístico, Tomado de la página web:

<http://www.cenired.org.co/index.php/servicios-cenired/66-en-colombia-2-4-millones-aun-padecen-el-flagelo-del-hambre-fao>.

PNUD, Objetivos de Desarrollo Sostenible (ODS). Tomado de la página web:

<https://www.undp.org/content/undp/es/home/sustainable-development-goals/background.html>

IEBS, (2017). Definición y características del Scrum Master, tomado de la página web:

<https://www.iebschool.com/blog/definicion-y-caracteristicas-del-scrum-master-agile-scrum/>

Buitrago, J: Informe de Gestión, Asociación de Banco de Alimentos ABACO, 2019.

FOWLER, Martin, Patterns of Enterprise Application Architecture (The Addison-Wesley Signature Series)– Nov 2002.

Proyectosagiles. (2008). [www.proyectosagiles.org](http://www.proyectosagiles.org). Recuperado el 2017, de <https://proyectosagiles.org/que-es-scrum>

Scrum Manager. (2016). Scrummanager. Recuperado el 2017, de [http://www.scrummanager.net/files/scrum\\_manager.pdf](http://www.scrummanager.net/files/scrum_manager.pdf)

Sharp, John: Microsoft Visual C# 2013 Step by Step. Pearson Education, 2013

ICONTEC, Guía Técnica Colombiana GTC-ISO/IEC 27002 de 2013

**ANEXO 1**

IDENTIFICACIÓN DE RIESGOS Y CONTROLES DE LA APLICACIÓN WEB SHARE FOOD  
NORMA ISO 27002.

**TABLA DE IDENTIFICACIÓN DE CONTROLES**

Dominio	14. ADQUISICIÓN, DESARROLLO Y MANTENIMIENTO DE SISTEMAS
<b>Objetivo del control</b>	<b>14.1 REQUISITOS DE SEGURIDAD DE LOS SISTEMAS DE INFORMACIÓN</b> Asegurar que la seguridad de la información sea una parte integral de los sistemas de información durante todo el ciclo de vida. Esto incluye también los requisitos para sistemas de información que prestan servicios sobre redes públicas.*
<b>Control</b>	<b>14.1.1 Análisis y especificación de requisitos de seguridad de la información</b> Los requisitos relacionados con seguridad de la información se deberían incluir en los requisitos para nuevos sistemas de información o para mejoras a los sistemas de información existentes.*
	<b>Implementación</b> En la fase de análisis de requisitos del ciclo del software para el desarrollo de la aplicación Share Food se deben tener en cuenta los requisitos no funcionales relacionados con la seguridad, más específicamente en con los criterios de Autenticidad Integridad y Confiabilidad.
<b>Control</b>	<b>14.1.2 Seguridad de servicios de las aplicaciones en redes públicas</b> La información involucrada en los servicios de aplicaciones que pasan sobre redes públicas se debería proteger de actividades fraudulentas, disputas contractuales y divulgación y modificación no autorizadas.*
	<b>Implementación</b> En el desarrollo de la aplicación se deben incorporar componentes de seguridad para el login, para evitar ataques de denegación de servicios por medio de bots, el uso del protocolo cifrado https, asimismo el acceso a las funciones de la aplicación solo las podrá tener los usuarios debidamente registrados y autenticados.
<b>Objetivo del control</b>	<b>14.2 SEGURIDAD EN LOS PROCESOS DE DESARROLLO Y DE SOPORTE</b> Asegurar de que la seguridad de la información esté diseñada e implementada dentro del ciclo de vida de desarrollo de los sistemas de información.*
<b>Control</b>	<b>14.2.1 Política de desarrollo seguro</b> Se deberían establecer y aplicar reglas para el desarrollo de software y de sistemas, a los desarrollos dentro de la organización.*
	<b>Implementación</b> Para mejorar la seguridad durante el desarrollo se deben implementar y definir ambientes de desarrollo seguro, con el uso de una plataforma para almacenar de manera segura el código fuente y la gestión de las versiones.  Uso de una metodología para la gestión del proyecto que aporte versatilidad, agilidad y seguimiento periódico al desarrollo.  Uso de lenguajes de programación, entornos de desarrollo y frameworks que ofrezcan seguridad.
<b>Control</b>	<b>14.2.2 Procedimientos de control de cambios en sistemas</b> Los cambios a los sistemas dentro del ciclo de vida de desarrollo se deberían controlar mediante el uso de procedimientos formales de control de cambios.*
	<b>Implementación</b> Usar una herramienta para la gestión del código desarrollado, que permita realizar rollback en dado caso que se presenten problemas con la implementación de nuevas versiones de la aplicación.



**ANEXO 1**

**IDENTIFICACIÓN DE RIESGOS Y CONTROLES DE LA APLICACIÓN WEB SHARE FOOD  
NORMA ISO 27002.**

	Dichas herramientas, permiten llevar registro de los cambios de las versiones realizadas a lo largo del ciclo del desarrollo.
<b>Control</b>	<b>14.2.3 Revisión técnica de las aplicaciones después de cambios en la plataforma de Operación</b>
	Cuando se cambian las plataformas de operación, se deberían revisar las aplicaciones críticas del negocio, y someter a prueba para asegurar que no haya impacto adverso en las operaciones o seguridad de la organización.*
	<b>Implementación</b>
	Para la puesta en producción del aplicativo, se considera la creación de un protocolo que permita identificar las necesidades en cuanto a hardware y procedimientos para su configuración, de modo que se afecte el funcionamiento desde un servidor local a un servidor en producción.
<b>Control</b>	<b>14.2.4 Restricciones en los cambios a los paquetes de software</b>
	Se deberían desalentar las modificaciones a los paquetes de software, los cuales se deben limitar a los cambios necesarios, y todos los cambios se deberían controlar estrictamente.*
	<b>Implementación</b>
	Se debe implementar un registro para llevar información de los paquetes de terceros implementados con el fin de verificar que riesgos o vulnerabilidad puedan tener las cuales pueden ser verificadas y subsanadas verificándolas en una base de datos de CVE (Common Vulnerabilities and Exposures).
<b>Control</b>	<b>14.2.5 Principios de construcción de sistemas seguros</b>
	Se deberían establecer, documentar y mantener principios para la construcción de sistemas seguros, y aplicarlos a cualquier actividad de implementación de sistemas de información.*
	<b>Implementación</b>
	Para la codificación del software del aplicativo se requiere seguir con las buenas prácticas SOLID y principios y patrones de para el desarrollo evitando escribir código con "code smell" y "bugs". Igualmente usar herramientas que permitan verificar los principios de código limpio.
<b>Control</b>	<b>14.2.8 Pruebas de seguridad de sistemas</b>
	Durante el desarrollo se deberían llevar a cabo pruebas de funcionalidad de la seguridad.*
	<b>Implementación</b>
	Con base el levantamiento de información o requisitos del sistema, identificar las pruebas que se le va a realizar al código y al aplicativo ya en funcionamiento. Realizar pruebas de seguridad OWSAP y pruebas funcionales, para verificar que se cumplan todos los objetivos de la aplicación.

\*Controles tomados de la norma técnica ISO/IEC 27002 de 2013.