



**Facultad de Economía y Negocios**  
**Escuela de Ingeniería Informática Empresarial**

**Propuesta de sistema para garantizar la calidad  
fitosanitaria de los alimentos en la logística de transporte  
con el fin de ser implementado con nanotecnología.**

**Autores: Lucas Mardones Vivanco**

**Pía Muñoz Núñez**

**Profesor guía: Leopoldo López Lastra**

**Proyecto de memoria para optar al título de INGENIERO INFORMÁTICO  
EMPRESARIAL**

**TALCA - CHILE 2021**

## CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su unidad de procesos técnicos certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Talca, 2023

# Índice de contenido

Índice de contenido.....	2
Índice de figuras.....	4
Resumen Ejecutivo .....	7
Abstract .....	8
Agradecimientos .....	9
Capítulo 1: Introducción.....	10
Capítulo 2: Marco teórico .....	13
2.1 Nanotecnología .....	13
2.1.1 Nanotecnología en la industria de los alimentos.....	13
2.2 Plataformas de creación electrónica.....	15
2.3 Sistemas de información web .....	18
2.3.1 Componentes .....	19
2.3.2 Ventajas y desventajas.....	19
2.3.3 Sistemas de información web en la industria de los alimentos.....	20
2.4 Regulaciones fitosanitarias.....	20
2.5 Calidad en el transporte de alimentos.....	21
2.5.1 Calidad .....	21
2.5.2 Transporte aéreo .....	22
2.5.3 Transporte marítimo.....	22
2.5.4 Transporte terrestre.....	22
2.5.5 Principales peligros en la seguridad de los alimentos durante el transporte	23
2.5.6 Controles preventivos para los riesgos en la seguridad de los alimentos en el transporte.....	24
Capítulo 3: Metodología.....	25
3.1 Metodologías Tradicionales .....	25
3.1.1 Modelo Cascada .....	25
3.1.2 Proceso unificado .....	25
3.1.3 Modelo en espiral .....	26
3.2 Metodologías Ágiles .....	26

3.2.1 Programación extrema (XP) .....	26
3.2.2 Scrum .....	26
3.2.3 Desarrollo basado en características (FDD) .....	27
3.2.4 Método de desarrollo del sistema dinámico .....	27
3.2.5 Desarrollo de software adaptativo (ASD) .....	27
3.2.6 Características de las metodologías ágiles .....	27
3.3 Metodología a ocupar .....	29
3.3.1 Design thinking.....	29
3.4 Herramientas a ocupar.....	31
3.4.1 Sistema Arduino .....	31
Capítulo 4: Resultados .....	36
4.1 Fases del proyecto .....	36
4.1.1 Empatizar.....	36
4.1.2 Definir.....	37
4.1.3 Idear .....	37
4.1.4 Prototipar .....	38
4.1.5 Testear .....	48
Capítulo 5: Análisis de resultados.....	56
Capítulo 6: Discusión .....	60
Capítulo 7: Conclusión.....	64
Capítulo 8: Bibliografía.....	65
Capítulo 9: Anexos .....	69

## Índice de figuras

<b>Figura 1 -Panificación del proyecto .....</b>	<b>35</b>
<b>Figura 2 -Conexión entre sensor SHT31 y placa Arduino 1 R3.....</b>	<b>39</b>
<b>Figura 3 -Conexión entre pantalla Lcd y placa Arduino 1 R3 .....</b>	<b>39</b>
<b>Figura 4 -Fotografía del sensor y la pantalla en función.....</b>	<b>40</b>
<b>Figura 5 -Vista del login del sistema .....</b>	<b>41</b>
<b>Figura 6 -Vista del módulo creado en las aplicaciones de Odoo .....</b>	<b>42</b>
<b>Figura 7 -Vista de sistema desde perspectiva del usuario.....</b>	<b>42</b>
<b>Figura 8 -Vista del registro de los datos obtenidos en la primera prueba de concepto .....</b>	<b>49</b>
<b>Figura 9 -Vista del módulo control de transporte .....</b>	<b>50</b>
<b>Figura 10 -Vista del Detalle de seguimiento correspondiente al módulo control de transporte .....</b>	<b>51</b>
<b>Figura 11 -Vista de los productos cargados en el módulo inventario.....</b>	<b>52</b>
<b>Figura 12 -Vista de las categorías de productos cargadas en el módulo Inventario .....</b>	<b>52</b>
<b>Figura 13 -Vista de los datos cargados al módulo de control de transportes.....</b>	<b>53</b>
<b>Figura 14 -Vista del primer código del Sistema Arduino.....</b>	<b>69</b>
<b>Figura 15 -Vista del primer código del Sistema Arduino.....</b>	<b>70</b>
<b>Figura 16 -Vista del primer código del Sistema Arduino.....</b>	<b>70</b>
<b>Figura 17. Vista de la clase de control .....</b>	<b>71</b>
<b>Figura 18. Código del Archivo XML correspondiente vista tipo Lista de la clase .....</b>	<b>71</b>
<b>Figura 19. Código del Archivo XML correspondiente vista tipo Formulario de la clase Control.....</b>	<b>72</b>

<b>Figura 20. Código del Archivo XML correspondiente vista tipo Formulario de la clase Control parte final .....</b>	<b>72</b>
<b>Figura 21. Vista del código de la clase de inventario... ..</b>	<b>73</b>
<b>Figura 22. Código del Archivo XML correspondiente vista tipo Listado de productos de la clase Control.....</b>	<b>74</b>
<b>Figura 23. Código del Archivo XML correspondiente vista tipo Formulario de la clase Control corregido.....</b>	<b>74</b>
<b>Figura 24. Código del Archivo XML correspondiente vista tipo Acciones de Categorías de la clase Control .....</b>	<b>75</b>
<b>Figura 25. Código del Archivo XML correspondiente vista tipo Listado de Categorías de la clase Control.....</b>	<b>76</b>
<b>Figura 26. Código del Archivo XML correspondiente al menú de la clase Control.....</b>	<b>77</b>
<b>Figura 27. Código del archivo PY de la vista de la clase de control corregida .....</b>	<b>77</b>
<b>Figura 28. Código del Archivo XML correspondiente vista tipo Listado de productos de la clase Control corregido... ..</b>	<b>78</b>
<b>Figura 29. Código del archivo XML correspondiente a la vista tipo Formulario de la clase de control corregida .....</b>	<b>78</b>
<b>Figura 30. Código del archivo XML correspondiente a la vista tipo Lista de detalle de venta la clase de control corregida .....</b>	<b>79</b>
<b>Figura 31. Código del archivo XML correspondiente a la vista tipo Formulario de detalle de venta de la clase de control corregida .....</b>	<b>79</b>
<b>Figura 32. Vista de la clase Control corregida .....</b>	<b>80</b>
<b>Figura 33. Vista de la clase DetalleControl .....</b>	<b>80</b>
<b>Figura 34. Código del Archivo XML correspondiente vista tipo Lista de la clase Control.....</b>	<b>80</b>
<b>Figura 35. Código del Archivo XML correspondiente vista tipo Formulario de la clase Control.....</b>	<b>81</b>
<b>Figura 36. Código del Archivo XML correspondiente vista tipo Lista de la clase DetalleControl.....</b>	<b>81</b>
<b>Figura 37. Código del Archivo XML correspondiente vista tipo Formulario de la clase</b>	

<b>DetalleControl.....</b>	<b>82</b>
<b>Figura 38. Modelo Relacional del sistema de información .....</b>	<b>82</b>

## **Resumen Ejecutivo**

La nanotecnología ha tomado una gran fuerza estos últimos años, debido a que puede generar múltiples soluciones a las diversas áreas que existen. En este caso en particular, tenemos que en la industria agroalimentaria cuenta con el uso de nanopartículas, nanocomposite, biosensores y nanotubos, entre otros, que ayudan a una pronta detección de residuos. Gracias a estas innovadoras ideas, se pueden generar otras en las diferentes etapas que tiene la industria alimentaria, como lo es la de transporte. Poder controlar las condiciones fitosanitarias de los alimentos cuando son transportados, no es algo que tenga una solución hoy en día, ya que la gran mayoría de las empresas toman las temperaturas manualmente cuando los productos llegan a su destino, provocando una reducción en los tiempos de trabajo y además una carga adicional para el trabajador encargado.

Por consiguiente, el objetivo de este proyecto es generar una propuesta o prototipo de sistema a base de sensores que midan la temperatura y humedad ambiente durante el transporte de alimentos, esto para que garanticen y controlen las condiciones fitosanitarias de los alimentos en la etapa de transporte. Adicional a lo anterior, la propuesta también incluye un sistema de información que guarde los datos que se van obteniendo de los sensores, todo esto de manera ordenada y clara para que pueda ser utilizado de una manera fácil y eficiente.

## **Abstract**

Nanotechnology has gained great strength in recent years, because it can generate multiple solutions to the various areas that exist. In this particular case, we have in the food industry the use of nanoparticles, nanocomposite, biosensors and nanotubes, among others, that help in the early detection of residues. Thanks to these innovative ideas, others can be generated in the different stages of the food industry, such as transportation. Being able to control the phytosanitary conditions of food when it is transported is not something that has a solution today, since the vast majority of companies take the temperatures manually when the products arrive at their destination, causing a reduction in working time and also an additional burden for the worker in charge.

Therefore, the objective of this project is to generate a proposal or prototype of a system based on sensors that measure the temperature and humidity during food transportation, in order to guarantee and control the phytosanitary conditions of the food during the transportation stage. In addition to the above, the proposal also includes an information system that stores the data obtained from the sensors, all this in an orderly and clear manner so that it can be used in an easy and efficient way.

## **Agradecimientos**

“Agradezco principalmente a mis padres y hermano que siempre me han dado su apoyo incondicional, a mi ahijada Amalia y a mi familia en general que siempre ha creído en mí y en mis proyectos, a mis amigos por ayudarme en momentos que se ha vuelto complicada la situación. No puedo dejar de agradecer a nuestro profesor guía por el apoyo que nos brindó en este proyecto, guiándonos de la mejor manera y haciendo sacar lo mejor de cada uno, ni tampoco puedo dejar afuera a mi compañero de tesis ya que sin duda fue un gran apoyo todo este tiempo y nada de este proyecto hubiera sido igual sin su compañía.”

**Pía Nelly Muñoz Núñez**

“En primer lugar agradezco a Dios por darme vida y salud, sin Él nada de esto sería posible. A mi familia por entregar lo mejor de ellos y por motivarme a superarme cada día. Sin duda mis amigos y amigas son fundamentales en este agradecimiento, ellos y ellas son quienes han estado conmigo en momentos donde la frustración abunda, así como también en momentos en donde ha habido victoria. Agradezco a mi compañera de tesis por brindar contención en momentos en donde sólo había miedo, realmente su apoyo fue fundamental. Nuestro profesor guía jugó un rol demasiado importante en este proyecto y por lo mismo quiero agradecer por su tiempo y paciencia con nosotros, y por plasmar su visión en este proyecto.”

**Lucas Ariel Mardones Vivanco**

## Capítulo 1: Introducción

Chile se destaca por ser un productor importante de productos alimentarios en el mundo y las exportaciones agropecuarias se han convertido en la segunda exportación del país solo superada por el cobre. En este contexto, para garantizar la inocuidad de los alimentos es fundamental mantener y/o controlar la temperatura de conservación y refrigeración de alimentos dentro de unos límites razonables. La cadena de frío nos certifica que esa inocuidad será totalmente garantizada sólo si la temperatura se mantiene bajo el control adecuado durante cada una de las fases por las que pasa el producto alimentario: producción, almacenamiento, transporte y venta.

En la fase de transporte las condiciones fitosanitarias pueden variar según la maquinaria de transporte. Esto deja en duda la trazabilidad del alimento ya que no se sabe si este tuvo alguna variación en alguno de sus indicadores fitosanitarios. Esta situación expone primeramente al consumidor a un riesgo de intoxicación, y en segundo lugar expone al alimento a tener una reducción en su tiempo de conservación o también, a tener una pérdida de calidad, lo cual implica la aparición de bacterias, pérdida nutricional y malos olores.

Los campos de la industria alimentaria y la nanociencia se han unido en este último tiempo para desarrollar nuevos productos y aplicaciones innovadoras, beneficiando la calidad y seguridad alimentaria, elaborando productos más saludables, más resistentes y de mayor durabilidad. Hoy en día la nanotecnología ha permitido mejorar la detección de pequeñas cantidades de sustancias nocivas, producir reflectores olfativos similares a los que poseen los seres humanos, optimizar los procesos de los alimentos utilizando enzimas que benefician los nutrientes de estos, envasados inteligentes, entre otros. Todas estas aplicaciones se han desarrollado e incluso utilizado en las diferentes etapas de la cadena de valor, pero cuando se trata de la etapa del transporte tenemos deficiencias en el proceso que se pueden eliminar incorporando tecnología. Por ejemplo, en ocasiones algunos productos que encontramos en las tiendas minoristas o grandes supermercados vienen en mal estado o se puede observar que no se mantuvo la cadena de frío. Hasta el momento no se ha podido desarrollar una solución que controle esta etapa de la cadena y provoca una deficiencia tecnológica a la hora de controlar las condiciones fitosanitarias en el transporte.

La nanotecnología tiene diversas aplicaciones en la industria de los alimentos, una de estas aplicaciones es poder asegurar que las condiciones fitosanitarias se mantengan durante el transporte marítimo y terrestre de los bienes. Si bien, en un futuro no muy lejano, la nanociencia busca crear microsensores para la gestión y control de los alimentos o micro etiquetas inteligentes que permitan localizar, inspeccionar y monitorear en tiempo real cada envase, este estudio busca facilitar la creación de estas aplicaciones, desarrollando un prototipo que facilita la información al cliente del cumplimiento de las condiciones fitosanitarias del alimento, beneficiando así la etapa de transporte de la cadena.

Por lo que se realizó una investigación aplicada, donde se propuso un sistema que, en base a nanosensores, asegure la calidad fitosanitaria de los alimentos durante la logística de transporte.

Para llevar a cabo esto se analizaron los riesgos fitosanitarios que se efectúan a lo largo de la fase de transporte de los alimentos, identificando las distintas variables que influyen en las condiciones fitosanitarias de los productos. En segundo lugar, se propuso una solución prototipada con sensores basados en Arduino. Estos sensores obtuvieron las lecturas máximas y mínimas de cada variable fitosanitario (temperatura, humedad, presión). En tercer lugar, se mostraron y guardaron los datos obtenidos desde los sensores, con el fin de identificar las variaciones en cada indicador.

En primera instancia, esta investigación tuvo una metodología principalmente bibliográfica, debido a que tuvo como objetivo describir las características de las tecnologías aplicadas a la logística de transporte de alimentos; a través de un método de análisis donde se dividieron los elementos y se pasaron a estudiar detenidamente. En segunda instancia, debido a que se desarrolló un prototipo de sistema, la metodología utilizada fue Design Thinking la cual nos permitió lograr una maqueta o prototipo lo más cercano a la realidad de la solución deseada. De esta manera pudimos testear o probar el prototipo logrado a través de una prueba de concepto.

El objetivo general del proyecto fue desarrollar un prototipo de sistema que garantice la calidad y seguridad de los alimentos durante la logística de transporte. Y los objetivos específicos fueron:

- Investigar sobre los riesgos fitosanitarios en la logística de transporte y del estado de arte de los sensores de nanotecnología.
- Proponer un sistema que contenga los sensores y que permita rescatar la información de ellos.
- Realizar una prueba de concepto del sistema con sensores basados en Arduino.

## **Capítulo 2: Marco teórico**

En el presente capítulo se abordarán los temas relacionados con la investigación, describiendo en primer lugar la Nanotecnología y su aplicación en la industria de los alimentos, las regulaciones fitosanitarias, y los distintos sistemas a usar para el correcto desarrollo de la solución propuesta la cual consiste en un sistema que asegure la calidad fitosanitaria de los alimentos durante la logística de transporte.

### **2.1 Nanotecnología**

La nanotecnología se considera como el estudio y la manipulación de materia y sistemas a escala nanométrica. Sin embargo, hoy en día existen variadas definiciones sobre la Nanotecnología, siendo las más acertadas aquellas que se refieren a la escala de tamaño. Por ejemplo, según la Royal Society [1] se define a la nanociencia como el estudio de los fenómenos y manipulación de materiales a escalas atómicas, moleculares y macromoleculares. En cambio, se define como nanotecnología al diseño, caracterización, producción y aplicación de estructuras, dispositivos y sistemas para controlar la forma y el tamaño a escala nanométrica.

Debido a que, previo al surgimiento de la nanotecnología ya había diversas disciplinas que trabajan con materiales a escala nanométrica, a algunos autores se les hace difícil clasificar la nanotecnología como una actividad nueva. Por lo mismo prefieren referirse a la Nanotecnología como un conjunto de disciplinas que operan a la misma escala y que hoy, gracias a instrumentos innovadores se pueden ampliar las fronteras de conocimiento sobre este tema [2].

Lo atractivo de la nanotecnología es que las propiedades de la materia cambian a escala nanométrica porque tienen una superficie más grande que los materiales a macro escala, esto permite que se modifiquen significativamente las características observadas a escala macroscópica de la materia o sistemas [3].

#### **2.1.1 Nanotecnología en la industria de los alimentos**

La nanotecnología se introduce en la industria de los alimentos a través del proceso de Pasteurización, con el objetivo de eliminar las bacterias de descomposición de los

alimentos, las cuales tienen un tamaño de mil nanómetros. Con esto se inició una revolución en el procesado de alimentos y en el control de calidad de los mismos Jafari, S. (2021).

### **2.1.2 Nanotecnología en el procesamiento de alimentos**

El procesamiento de alimentos es la transformación de ingredientes crudos en alimentos con el fin de hacerlos comercializables y con una larga vida útil. Los alimentos procesados suelen ser menos aptos al deterioro temprano que los alimentos frescos y son más adecuados para el transporte a larga distancia desde la fuente hasta el consumidor. Estos alimentos se hacen más efectivos con la incorporación de la nanotecnología en la actualidad debido a que los sistemas de administración de nano cápsulas juegan un papel importante en el sector de procesamiento [4].

### **2.1.3 Nanotecnología en el envasado de alimentos**

El envasado tiene una gran importancia en la conservación de los alimentos para que sean comercializables. Las innovaciones en el envasado han dado lugar a un envasado de calidad y un enfoque amigable para el consumidor al determinar la vida útil, el embalaje biodegradable, entre otros. Una de las investigaciones más conocidas, es el uso de las nanopartículas en los envases para que puedan obtener mayor tiempo el gas de las bebidas carbonatadas. De igual forma, el desarrollo de la “lengua electrónica” de Kraft, donde la plataforma de nanosensores ultramente sensibles a los gases de salen de los alimentos cuando comienzan a perder su calidad y hace que un indicador cambie de color.

### **2.1.4 Nanotecnología en la protección de barrera de los alimentos**

Los productos alimenticios se conservan manteniéndolos en una atmósfera inerte y con poco oxígeno para inhibir el crecimiento y el deterioro microbiano, por lo que el material utilizado debe ser impermeable a los gases. Estos materiales mantienen la humedad alejada de productos como los snacks, conservando por más tiempo su textura crujiente y, por supuesto, son cruciales cuando hablamos de aumentar la vida

útil de productos que tienen una duración limitada como lácteos, carnes y pescados. La nanotecnología ofrece nanocompuestos poliméricos que se utilizan como matriz polimérica en envases de alimentos para obtener una mayor calidad. Cabe destacar que se han desarrollado diversas investigaciones en nanocompuestos a partir de células y nanotubos de carbono, ya que el envasado juega un papel fundamental en la comercialización de productos [4].

## **2.2 Plataformas de creación electrónica**

Las plataformas de desarrollo electrónico cumplen un rol fundamental a la hora de reducir los tiempos para el desarrollo de alguna solución o prototipo de una forma más sencilla, que incluso, en ocasiones se convierte en la base del proyecto final. Y es que estas plataformas consisten en placas que ya traen consigo una serie de funciones implementadas debido a que la persona que elabora estas plataformas ya soluciona los problemas básicos para el desarrollo del producto, lo cual permite al desarrollador enfocarse netamente en las prestaciones del diseño del prototipo o solución.

### **2.2.1 Microprocesadores y Microcontroladores**

Para entender estos dos conceptos, primero que todo debemos saber cómo funciona una CPU. Esta se encarga de interpretar funciones obtenidas desde la memoria, con el fin de ejecutarlas. Por lo tanto, la CPU se considera como el cerebro del microcomputador, es decir, el microprocesador; circuito encargado de efectuar las funciones descritas anteriormente.

En cuanto a los microcontroladores, estos se crearon o aparecieron posterior a los microprocesadores y con la finalidad de cubrir una gran variedad de aplicaciones en las cuales el microcontrolador tiene que efectuar tareas puntuales al menor costo posible. En estas aplicaciones, el microcontrolador es el encargado de ejecutar un programa almacenado de forma permanente en una memoria, este programa almacena datos que interactúan con ciertas líneas de entrada y salida [5].

### **2.2.2 Arquitectura de los microcontroladores**

Inicialmente, todos los microcontroladores tenían la arquitectura clásica de Van Neumann, pero ahora en el presente, la mayoría de los microcontroladores poseen la arquitectura Harvard.

La arquitectura clásica de Van Neumann posee solo una memoria principal, esta se encarga de almacenar datos e instrucciones. Para acceder a esta memoria se necesita un sistema de buses único (direcciones, datos y control). En cambio, la arquitectura de Harvard posee dos memorias independientes con el fin de poseer instrucciones y datos de forma separada, es decir, una memoria está destinada a las instrucciones y otra, a los datos [5].

En su mayoría las placas se componen de microcontroladores, circuitos y elementos electrónicos que en conjunto facilitan técnicas básicas de tal manera que el desarrollador solo debe seleccionar la plataforma que más cumpla con los requisitos para el proyecto. Estas plataformas son el elemento más importante del mundo del internet de las cosas, debido a que esto proporciona la conexión con otros dispositivos de internet de las cosas para transferir la información. A continuación, se mencionan las plataformas y herramientas más utilizadas que ayudan al desarrollo de un proyecto.

- **NodeMCU**

Esta plataforma se utiliza en proyectos de internet de las cosas, puesto que tienen un chip llamado ESP8266 que incorporan Wifi y un procesador de 32 bits, esto lo convierte en uno de los chips más integrados dentro de las tecnologías debido a que requiere de pocos circuitos externos. Con esto la plataforma puede crear una red o conectarse a una existente [6].

- **Raspberry Pi**

Esta plataforma ya es denominada como una minicomputadora. Para su funcionamiento se requiere una tarjeta SD con mínimo 8GB para poder instalar el sistema operativo. Una vez instalado se puede controlar los puertos HDMI que incluye la Raspberry con salida de audio y vídeo desde Python. En la plataforma se puede utilizar para navegar por internet, ver correos, reproducir videos, para mensajería instantánea, también como un ordenador portable, pero no portátil porque depende de la red electrónica, pantalla y teclado. Cabe señalar que ha recibido críticas

por esto, pero esta plataforma tiene un buen rendimiento y a un precio no elevado [6].

- **Beagleboard**

Esta placa es de hardware abierto que corre bajo un sistema operativo Linux y cuenta con diferentes entradas, salidas y un conector microHDMI, pero es común programar la placa a través de la consola.

Es importante destacar que existen varias versiones de la plataforma, algunas de ellas son Beaglebone Blue que está más ligada a trabajos de robótica, debido a que contiene motores y sensores de acelerómetro y giroscopio. Otra versión es la Beaglebone AI que se utiliza para proyectos de Inteligencia Artificial.

Beagleboard está más dirigido a usuarios avanzados dentro de la industria de la programación puesto que las otras placas no pueden correr proyectos tan pesados, al contrario de Beagleboard que incluso se puede poner en funcionamiento al sacarlo de la caja [6].

- **STM**

A menudo se confunde con Arduino debido a su gran parecido en lo que es el diseño de la placa pero la diferencia entre estas dos plataformas es considerable. Esto se debe a la gran diferencia en el rendimiento en procesamiento de datos y en la eficiencia energética de STM. Esta plataforma cuenta con placas con tecnología inalámbrica, Bluetooth, Lora y Sigfox, los cuales son componentes esenciales del Internet de las cosas [6].

- **Arduino**

Es un open hardware y debido a esto han salido varias réplicas de esta plataforma. Hoy en día es muy demandada, sobre todo en el ámbito estudiantil. Cabe destacar que estas réplicas permiten las mismas funciones que las placas originales.

Arduino tiene una variedad de modelos, entre ellos encontramos el modelo Arduino Uno, el modelo Nano que brinda una placa más pequeña, el modelo DUE que brinda más potencia de procesamiento, entre otros.

Al igual que otras plataformas, Arduino puede recibir información de distintos dispositivos de entrada y proyectarla a través de distintos dispositivos de salida. Arduino también cuenta con un editor de código propio con el fin de subir el programa desarrollado a las distintas placas [6].

### **Ventajas de utilizar Arduino**

Una de las tantas ventajas de utilizar Arduino es su hardware abierto, su bajo costo y la facilidad que tiene de poder programar y crear proyectos de menor complejidad a comparación de las otras plataformas, como es el ejemplo de Beagleboard que se especializa en trabajos robóticos y programación más avanzada. Pero lo más importante es mencionar que el sistema Arduino que escogimos es el óptimo para el proyecto que se desea realizar, debido a que se utiliza mucho en las primeras fases como prototipo, antes de llegar al producto final.

### **Desventajas de utilizar Arduino**

Son pocas las desventajas que podemos encontrar en la plataforma Arduino, ya que en los últimos años ha aumentado su popularidad y con ello se ha perfeccionado su uso y los elementos que hacen que esta plataforma sea bien completa para algunos trabajos de programación. Pero como la programación tiene sus pro y contra, en este caso, tenemos que el Arduino al ya venir ensamblado le resta flexibilidad a la creación de los proyectos, por ejemplo, el programador se ve obligado a usar espacios acordes a los espacios PCB de Arduino y para poder hacerlo de otra manera se debe hacer un PCB desde cero trabajando con un microcontrolador diferente al que viene en la plataforma.

## **2.3 Sistemas de información web**

Los sistemas de información web son un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no son creados por el desarrollador [7]. Una de las principales características de estos sistemas es que

permiten el acceso a la información mediante cualquier dispositivo que posea un navegador web y conexión a internet.

### **2.3.1 Componentes**

- **Cliente:** Por lo general son programas (navegadores) que permiten al usuario tener una acción mutua con el servidor con la finalidad de solicitar la información deseada.
- **Servidor:** Al igual que el cliente, el servidor también es un programa pero que tiene como objetivo esperar permanentemente las solicitudes de conexión o información que efectúan los clientes.

### **2.3.2 Ventajas y desventajas**

Los sistemas de información web, tienen muchas diferencias con otros tipos de sistemas, lo que lo hacen muy beneficioso tanto para las empresas que lo utilizan, como para los usuarios que operan en el sistema. A continuación, se mencionan las ventajas descritas por [8].

- El problema de gestionar el código en el cliente se reduce drásticamente debido a que todos los cambios, tanto de interfaz como de funcionalidad, que se deseen realizar a la aplicación se efectúa convirtiendo el código que habita en el servidor web.
- Se previene problemas de inseguridad en las actualizaciones, ya que no hay clientes con diferentes versiones de la aplicación.
- No se requiere adquirir ni tampoco instalar herramientas adicionales para los clientes.
- Independencia de plataforma

Sin embargo, a pesar de estas ventajas, los sistemas de información web tienen una deficiencia a nivel de desarrollo porque la programación en la web presenta muchas limitaciones debido a su lenguaje HTML, por ejemplo, el escaso repertorio de controles disponibles para crear formularios.

### **2.3.3 Sistemas de información web en la industria de los alimentos**

La industria de los alimentos hace años viene experimentando una transformación digital a través de robots, avances en agricultura de precisión y sistemas de apoyo para la toma de decisiones. Algunas de las tecnologías involucradas en esta transformación digital son: Cloud of computing, Big data, Blockchain, Robótica e Inteligencia Artificial. Se pronostica que las operaciones de cada proceso se integrarán automáticamente en la cadena alimentaria a través de tecnologías semánticamente activas hasta el consumidor final [9].

## **2.4 Regulaciones fitosanitarias**

Las medidas sanitarias y fitosanitarias son todas las leyes, reglamentos y procedimientos establecidos por varios países, los cuales son necesarios para proteger la vida y salud de los seres humanos.

La inspección fitosanitaria de productos controlados, es decir productos que requieren un certificado fitosanitario para poder ser importado a un determinado país o región, necesita principalmente una auditoría para poder verificar si el país exportador cumple con los requisitos de importación del país importador.

Estas medidas sanitarias tienen que garantizar que los alimentos sean microbiológicamente sanos, libres de plagas y que tengan una larga vida útil, para satisfacer las necesidades del consumidor. La contaminación puede pasar en las diferentes etapas de la cadena de valor, ya sea almacenamiento, transporte o venta [10].

Un estudio que comparaba estándares para las exportaciones de frutas frescas de Chile [11], reveló que los países latinoamericanos tienen regulaciones fitosanitarias más estrictas, mientras que los países asiáticos son más estrictos en las medidas de calidad y los países europeos en los requisitos de límites máximos de residuos.

La creciente popularidad de las normas privadas sobre seguridad alimentaria, calidad de los alimentos, el impacto ambiental y social de los sistemas agroalimentarios han generado preocupaciones sobre cómo afectarían a los países en desarrollo y la gobernación de la cadena de valor agropecuaria en un sentido más amplio. Por lo tanto, es necesario reconocer la

estrecha relación que hay entre las regulaciones públicas y los estándares privados y la forma en que estos pueden seguir evolucionando y afectar el comercio [12].

Chile ha establecido un marco institucional capaz de apoyar al sector exportador para cumplir la alta demanda de los requisitos del país o región de destino. El ministerio de Agricultura y el Servicio Agrícola y Ganadero, responsables de estas regulaciones están ayudando a los exportadores y productores con los procedimientos de certificación de aspectos sanitarios y fitosanitarios. Mientras que el sector privado trabaja junto con el sector público en programas e instrumentos que ayuden a alcanzar los estándares de exportación a los productores y exportadores. Los programas se enfocan en los aspectos técnicos, riego, monitoreo de plagas, entre otros; así también con las buenas prácticas agrícolas y sistemas de aseguramiento de la calidad [13].

## **2.5 Calidad en el transporte de alimentos**

Se estima que cada año se transportan alrededor de 200 mil millones de toneladas de alimentos en el mundo, el 35% del transporte se efectúa por vía terrestre mientras que el 60 % se efectúa por la vía marítima, siendo solamente un 5% de los alimentos transportados por la vía aérea. Debido a que estos productos son destinados al consumo humano, el transporte de alimentos involucra velar y dar un gran enfoque en la seguridad y calidad de los productos [14].

### **2.5.1 Calidad**

La calidad, en el contexto del transporte de alimentos, tiene que ver en primera instancia con las propiedades físicas de los alimentos y también, en segunda instancia, tiene que ver con cómo el consumidor percibe el alimento, generalmente se implican aspectos microbianos, pero también aspectos como el sabor o la textura del alimento.

Trienekens y Zuurbier [15] detallan que el proceso de aseguramiento de la calidad debería estar muy involucrado en lo que es la producción y distribución de alimentos debido a que, en la industria de los alimentos, la calidad de estos es de suma importancia.

### **2.5.2 Transporte aéreo**

El transporte aéreo se utiliza cada vez más para productos perecederos de alto valor como las carnes. Sin embargo, aunque esta vía de transporte ofrece un método rápido para mercados distantes, hay muchos problemas porque el producto está desprotegido de refrigeración durante gran parte del viaje, aunque esté revestido con algún material aislante [16].

### **2.5.3 Transporte marítimo**

La necesidad de preservar los alimentos durante el transporte por vía marítima fue lo que impulsó el desarrollo de la refrigeración mecánica y el comercio internacional moderno de alimentos.

La gran mayoría de los contenedores destinados al transporte marítimo tienen una longitud entre 6 y 12 metros, y soportan alrededor de 26 toneladas de productos. Estos contenedores se caracterizan por ser aislados o refrigerados; los contenedores aislados utilizan unidades de refrigeración de tipo enchufe o pueden conectarse directamente a un sistema de manipulación de aire de una bodega o muelle. En cambio, los contenedores refrigerados incorporan aislamiento y tienen unidades de refrigeración incorporadas en su estructura [17].

### **2.5.4 Transporte terrestre**

Los sistemas diseñados para transporte terrestre pueden variar desde pequeñas furgonetas no aisladas que suministran alimentos a los puntos de ventas y consumidores, hasta contenedores de 12 metros, refrigerados y diseñados para rutas de larga distancia.

El aumento de los servicios de entrega a domicilio, donde se efectúan entregas de productos que requieren diferentes temperaturas, está introduciendo una nueva complejidad en la entrega local por la vía terrestre. En una encuesta realizada en el año 2006 [17] se encontró que un producto refrigerado puede ser sometido a hasta 50 aberturas de puertas permitiendo la entrada directa de calor desde el exterior, lo cual es perjudicial para los productos refrigerados.

El sistema de transporte de alimentos es super complejo, por lo tanto, si un problema se produce fuera de este sistema y no se detecta, el riesgo es mayor porque es muy probable que este problema se extienda a muchos distribuidores, minoristas y finalmente a los consumidores.

Existen empresas que organizan internamente el transporte de sus productos, así como también hay otras empresas que utilizan proveedores de logística externos con el fin de subcontratar la contratación del transporte. A estos proveedores de logística externos se les conoce por “3PL”. Un 3PL es una empresa que brinda servicios de logística para parte, o a veces, toda la cadena de suministro de una empresa de alimentos. Sin embargo, todo indica que las mayores preocupaciones durante el transporte de alimentos son la manipulación y el sabotaje, el abuso de temperatura y la contaminación cruzada.

Los factores de riesgo de contaminación de los alimentos incluyen malas prácticas en la producción, temperaturas inadecuadas, áreas de carga insalubre, contenedores de envío en mal estado, malos hábitos de los empleados y las condiciones de las carreteras. Por esto, las regulaciones federales indican que los productos alimenticios deben protegerse contra la contaminación física, química y microbiana durante el almacenamiento y transporte de alimentos [18].

A menudo puede que se confunda la seguridad de los alimentos con la calidad de los mismos, si bien se relacionan los dos conceptos, estos funcionan de manera distinta. Mientras que la calidad de los alimentos se relaciona con un proceso continuo de degradación, la seguridad de los alimentos consiste en un modelo binario en donde se decide si el producto está apto o no para su respectivo consumo [14].

#### **2.5.5 Principales peligros en la seguridad de los alimentos durante el transporte**

- Falta de seguridad en las instalaciones de transporte y almacenamiento
- Prácticas inadecuadas de espera para productos alimenticios en espera de envío o inspección.
- Control de temperatura o refrigeración inadecuada de los alimentos
- Manejo inadecuado de unidades de transporte y almacenamiento para evitar la contaminación cruzada.
- Prácticas de carga inadecuadas.

- Control de plagas ineficiente en unidades de transporte.
- Manejo y seguimiento inadecuado de productos rechazados, recuperados, reelaborados, devueltos o destinados a su eliminación.

#### **2.5.6 Controles preventivos para los riesgos en la seguridad de los alimentos en el transporte**

- Correcto empaque de los productos alimentarios y las unidades de transporte.
- Uso adecuado de los equipos de refrigeración.
- Dispositivos de control.
- Registro de temperatura.
- Procedimientos de carga y descarga adecuados según la unidad de transporte.
- Seguridad física adecuada para las instalaciones y unidades de transporte.
- Uso de tecnologías de rastreo.
- Saneamiento y mantenimiento de unidades de transporte.
- Programas de control de plagas.
- Buena comunicación entre expedidor, transportista y receptor.
- Etiquetado, señalización e instrucciones adecuadas para el transportista.

## Capítulo 3: Metodología

La metodología es la manera en cómo el proyecto será descrito, donde se aplican técnicas y métodos para el desarrollo de este. Por lo que se selecciona la metodología que esté más acorde al tema y que asegure un proceso eficiente y eficaz. A continuación, se mencionan las metodologías existentes y la que seleccionamos para nuestro proyecto.

### 3.1 Metodologías Tradicionales

Al principio el desarrollo de software era totalmente artesanal, pero cuando la necesidad de una mejoría en los procesos y en las soluciones de los proyectos, tuvieron que adaptar metodologías de otras áreas ya existentes al desarrollo de software. En este periodo de adaptación contiene el desarrollo dividido secuencialmente y mejoraba las necesidades en los campos de software [19].

Estas metodologías consisten en un conjunto de pasos que están de manera secuencial, como por ejemplo definir los requisitos, crear las soluciones, planificar, entre otros. Al comenzar un proyecto, la metodología tradicional necesita definir y documentar los requisitos. Existen varios modelos dentro de la metodología tradicional, los cuales son utilizados por la comunidad de desarrolladores y los pasaremos a definir a continuación:

#### 3.1.1 Modelo Cascada

Este modelo contiene etapas que consisten en un conjunto ya definido de actividades y entregables que deben estar cumplidos antes de comenzar con la fase siguiente. Las etapas se nombran de forma diferente, pero la idea de la primera fase es que tiene que intentar capturar lo que el sistema hará y los requisitos del software. La segunda etapa va a determinar cómo será el diseño del software. Luego la tercera etapa consiste en que los desarrolladores escriban el código, en la cuarta etapa se prueba el sistema y finalmente se capacita y se documenta [20].

#### 3.1.2 Proceso unificado

Este modelo se planifica en flujos de trabajo unificado y se realizan de manera iterativa e incremental. Algunas características de este proceso es que utiliza la arquitectura de componentes para crear un sistema que sea maleable, a su vez promueve la reutilización del software y que sea automáticamente comprensible.

Es importante verificar la calidad del software en un proyecto y el proceso unificado ayuda a organizar el control de calidad y evaluar los procesos que involucran a todos los miembros del equipo.

### **3.1.3 Modelo en espiral**

Otro modelo de desarrollo de software de gran importancia es el modelo espiral, el cual mezcla los elementos de diseño y creación de prototipos en las fases correspondientes, donde se juntan las ventajas de conceptos de arriba hacia abajo y de abajo hacia arriba. Este modelo es definido por Barry Boehm, centrado en la experiencia con conceptos del modelo de cascada que se aplicaron a importantes proyectos de software. Las fases principales de este modelo en espiral es establecer los objetivos, evaluar y reducir los riesgos, el desarrollo y validación y finalmente la planificación.

## **3.2 Metodologías Ágiles**

En la actualidad, las empresas que deciden efectuar una transformación digital optan por un proceso que les brinde flexibilidad e inmediatez, es por esto que los métodos de desarrollo de software cada vez intentan ofrecer a través de la rapidez y agilidad. A continuación, se describen las principales metodologías ágiles que se están implementando en la actualidad.

### **3.2.1 Programación extrema (XP)**

La programación extrema se caracteriza por ciclos cortos de desarrollo, planificaciones incrementales, retroalimentación continua, dependencia de la comunicación y diseño evolutivo. Se dice que los programadores le dedican cierta cantidad de minutos a cada una de las características mencionadas, muchas veces al día. Es por esto que el término “extremo” proviene de llevar estas prácticas de sentido común a niveles extremos.

### **3.2.2 Scrum**

Scrum es un proceso iterativo y se caracteriza en plantear cómo deben funcionar los miembros del equipo a fin de producir una flexibilidad del sistema en un entorno

cambiante. En otras palabras, Scrum se caracteriza por brindar prácticas y herramientas de gestión en sus diferentes fases.

### **3.2.3 Desarrollo basado en características (FDD)**

Este enfoque basado en características, a diferencia de las otras metodologías no cubre todas las etapas del proceso de desarrollo de software, este se orienta en las etapas de diseño y construcción. Los procesos de estas etapas se desarrollan ágilmente y se adaptan rápidamente a los cambios tardíos y las necesidades comerciales. El enfoque contiene un seguimiento de los progresos de los informes y a la vez entregables de manera tangible.

### **3.2.4 Método de desarrollo del sistema dinámico**

Esta metodología es una combinación y una extensión del desarrollo de aplicaciones y prácticas de un proceso iterativo. Lo fundamental de este método es fijar el tiempo y los recursos a ocupar, para luego acoplar las funciones.

### **3.2.5 Desarrollo de software adaptativo (ASD)**

La metodología de desarrollo de software adaptativo fue diseñada para proyectos de alta velocidad y cambios debido a que, en un entorno empresarial impredecible y rápido en movimiento, no siempre es posible planificar con éxito.

### **3.2.6 Características de las metodologías ágiles**

Las metodologías ágiles son reconocidas como impulsoras del éxito de los proyectos y junto con esto, por su enfoque en la eficacia y maniobrabilidad de forma intensa. A continuación, se describen las principales características de las metodologías ágiles:

- **Orientado a las personas:** Para las metodologías ágiles, las personas son el factor más importante. Si las personas en el proyecto son lo suficientemente buenas, pueden efectuar casi cualquier proceso y cumplir con sus tareas.
- **Adaptado:** Estas personas que integran el grupo de trabajo, no le temen a los cambios. La metodología ágil realiza cambios en todas las etapas del proyecto, mediante estos cambios es donde el equipo entiende y aprende más sobre lo que necesitará para satisfacer el mercado.

- **Conformidad con lo real:** Para la metodología ágil la conformidad se valora en los resultados reales y no cuando está en el plan detallado. Los proyectos bajo esta metodología se basan en la conformidad con el valor comercial. En cada ciclo de trabajo se va agregando un valor comercial para el producto final. Para el equipo, la decisión final si se ha agregado valor comercial o no, lo dan los usuarios finales y no los desarrolladores.
- **Equilibrio entre flexibilidad y planificación:** La planificación es importante, pero el gran problema de los proyectos de software es que no se puede saber con precisión el futuro por las variables que se deben de tener en consideración. La mejor opción es planificar en periodos muy cortos de plazo. Lo relevante de esta característica es que las decisiones pueden cambiar fácilmente, lo cual genera que no se tome la importancia necesaria para hacer las cosas bien.
- **Proceso empírico:** Es un proceso no lineal, estos proyectos de software definidos como empírico se caracterizan porque solo se puede iniciar y permitir que se ejecute hasta finalización.
- **Enfoque descentralizado:** El desarrollo ágil de software extiende la toma de decisiones a los desarrolladores, sin embargo, esto no quiere decir que los desarrolladores tomen el papel o rol de la dirección ya que la administración aún es necesaria para afrontar los obstáculos que se interponen en el camino. La extensión de la toma de decisiones consiste en reconocer la experiencia del equipo de desarrollo para tomar decisiones sin su permiso.
- **Sencillez:** Los equipos que implementan estas metodologías ágiles siempre toman el camino más simple y acorde a sus objetivos. La razón de esta conducta es que será mucho más fácil cambiar el diseño si es necesario en una fecha posterior.
- **Colaboración:** Las metodologías ágiles requieren que los clientes sean retroalimentados de forma regular y frecuente debido a que estos trabajan en estrecha relación con el equipo de desarrollo. Cabe mencionar que la colaboración constante entre los miembros del equipo es esencial.

- **Pequeños equipos autoorganizados:** El desarrollo ágil es más difícil con equipos más grandes, por lo mismo un proyecto promedio tiene sólo nueve personas, lo cual hace que el equipo tenga más alcance a la mayoría de los procesos ágiles básicos.

Esta metodología se centra fundamentalmente en los resultados y su respectiva calidad, más que en las tareas o procesos utilizados para llevar a cabo el proyecto. Para gestionar esta metodología, es de suma importancia fomentar la comunicación en lugar de decirle a la gente qué hacer, esto con el fin de que los integrantes entreguen respuestas o ideas más creativas al equipo.

Por último, es importante recalcar que la metodología ASD no se presenta específicamente como una metodología para el desarrollo de software sino como un enfoque o actitud que debe tomar una organización al momento de aplicar procesos ágiles.

### **3.3 Metodología a ocupar**

#### **3.3.1 Design thinking**

En su mayoría consolidan esta metodología hacia una postura de diseño y lo vinculan mucho con la creatividad y racionalidad, dado que se consideran las necesidades de los usuarios, mediante las ideas, pensamientos creativos y finalizar con la experimentación y el pensamiento analítico.

El proceso de un pensamiento lógico cumple un rol esencial cuando hay conocimiento sobre el cual se debe construir, pero que exige un pensamiento creativo cuando es ya necesario generar un nuevo conocimiento o plantear los problemas de un modo diferente [21].

El Design Thinking se diferencia de los otros enfoques del diseño, puesto que se centra más que nada en el proceso que en el lugar del producto, es decir, se enfoca en resolver el problema, pero no comienza necesariamente teniendo una solución previa.

Esta metodología tiene dos principales características, la primera es que está muy centrado al usuario, debido a que las necesidades de este se observan y se añaden

constantemente a un proceso de diseño promovido por el conocimiento y la creatividad. La segunda característica es que es iterativo, ya que se corrige y se vuelve a evaluar, en una constante retroalimentación. Además, es social y comunicativo, pues se aplica a proyectos que se realizan en equipo, donde los integrantes de este generan ideas y toman decisiones juntos [21].

Se afirma que el Design Thinking es una forma de utilizar las capacidades de todos los miembros del equipo, pero a menudo se pasa por alto en las prácticas de resolución de problemas más tradicionales [22].

El proceso de Design Thinking tiene 5 acciones fundamentales:

- **Empatizar:** Esta etapa comienza captando las necesidades que tiene el usuario del proyecto que se va a realizar
- **Definir:** Durante esta etapa se filtra la información recopilada durante la fase de empatía y se toma lo que realmente aporta valor y que servirá para alcanzar las nuevas perspectivas interesantes.
- **Idear:** Esta etapa tiene como objetivo generar muchas opciones para la problemática. En esta etapa, las actividades facilitan el pensamiento expansivo que consiste en una lluvia de ideas. Todo puede ser importante y a menudo las ideas que salen de lo normal son las que generan resultados más innovadores.
- **Prototipar:** Por medio de un prototipo se construye un modelo “rápido” que facilitara a dar forma a lo que anteriormente era una idea o concepto. A partir de esta etapa ya existe algo físico, algo que se puede visualizar. Este prototipo también se puede tratar de algo virtual en caso de una aplicación informática.
- **Testear:** Esta etapa se enlaza con la fase anterior. Luego se aplica el prototipo con la ayuda del supuesto usuario, por el cual está orientado el proyecto. Una vez obtenido los comentarios, estos se incorporan para mejorar el resultado que se busca.

El Design Thinking puede destinarse en cualquier sector, y en general sirve para:

- Resolver de problemas de forma creativa e innovadora
- Diseñar y desarrollar productos o servicios
- Rediseñar procesos de negocios
- Emprender y crear empresas (Startups)
- Crear un Plan B de vida
- Diseñar crear una presentación de negocios
- Diseñar cursos virtuales u online

### **3.4 Herramientas a ocupar**

#### **3.4.1 Sistema Arduino**

Arduino consiste en una plataforma basada en un hardware y software libres, flexibles y fáciles de usar para los creadores y desarrolladores, y su fin la creación electrónica. La plataforma permite la creación de diferentes tipos de microordenadores y los desarrolladores pueden proporcionar diferentes tipos de uso [23]. Es decir, Arduino es una placa con circuitos integrados a los que se le pueden grabar instrucciones a través de la programación y realizan una acción concreta dependiendo de lo que haya programado.

Arduino fue creado en el 2005 por el estudiante Massimo Banzi del instituto IVRAE en Italia, con la intención de integrar y facilitar el aprendizaje integrado de computación y electrónica para los estudiantes de ese entonces. Debido a su fácil acceso, bajo costo y la popularidad que alcanzó en el instituto, se inició su distribución y comercialización tanto en Italia como en el resto de los países.

En comparación con otros microcontroladores, las placas de arduino son mucho más accesibles debido a que Arduino IDE, el software de la placa, es compatible en múltiples sistemas operativos, como lo son Windows, Linux y Mac OSX. A su vez tiene un ambiente de programación sencillo y directo, fácil de usar para los usuarios. Por lo tanto, hay muchos otros microcontroladores y plataformas disponibles pero que son más complicados de programar en comparación con Arduino que ofrece algunas ventajas respecto a otros sistemas [24].

### 3.4.1.1 Características básicas de Arduino

El microcontrolador de Arduino cuenta con una interfaz de entrada, la cual es una conexión donde se puede conectar la placa a diferentes tipos de periféricos, que son dispositivos que proyectan información hacia el exterior o interior del ordenador, por ejemplo, pueden ser cámaras, teclados, sensores.

La información de los periféricos se traslada al microcontrolador donde se encarga de procesar los datos que lleguen. A su vez cuenta con una interfaz de salida, que se encarga de llevar la información procesada en el Arduino a otros periféricos, por ejemplo, pueden ser pantallas donde se muestran los datos obtenidos. (Fernández, 2018)

- **Hardware:** Es una placa electrónica que se puede ensamblar o construir debido a que se encuentran los planos electrónicos y licencias en internet. Hoy en día la placa cuenta con un único microcontrolador que lleva a cabo la comunicación y las instrucciones para poder ser ejecutadas, a diferencia de las primeras placas de Arduino que tenían un microcontrolador para ser programado y otro para la comunicación.

La placa Arduino “Uno” cuenta con:

- Conector USB que genera la comunicación, la toma de datos y a su vez provee una fuente para alimentar al Arduino, pero que al ser de baja corriente no es suficiente para alimentar motores de grandes potencias.
- Para que la placa tenga funcionalidad y pueda alimentar los circuitos se necesita un regulador 5V de voltaje
- Plug de conexión para la fuente de alimentación externa donde el voltaje que se proporciona directamente y debe estar entre los 6 V y 18V.
- Puerto de conexiones con la funcionalidad de RESET que se forman por 6 pines de conexión, lo cual permite reiniciar el microcontrolador al enviarle un cero.

- Puerto de entrada análogas que es el lugar donde se conectan las salidas de los sensores análogos cuando reciben voltaje entre cero y cinco voltios directos.
  - Microcontrolador Atmega328 que se utiliza para reducir el peso y obtener un poco más de espacio.
  - Botón de reset que autoriza a reiniciar el microcontrolador y reiniciar el programa.
  - Pines de programación ICSP se utilizan para programar los microcontroladores que se encuentran sobre los circuitos sin la necesidad de tener que retirarlos de su sitio.
  - Leds de recepción y transmisión que se encienden cuando la tarjeta se comunica con el ordenador.
  - Puertos de conexiones de pines de entrada o salidas digitales, estos puertos deben contar con la configuración como entrada o salida incluida en el programa para permitir su uso controlado.
  - Led pin 13 que indica el estado en que se encuentra el Arduino.
  - Chip de comunicación que permite la comunicación de serial a USB.
- 
- **Software:** Arduino se puede programar a través del uso de su propio lenguaje basado en C/C++ pero, aun así, debido a que esta plataforma utiliza la comunicación mediante transmisión de datos en formato serie, es posible utilizar otros tipos de lenguaje para su desarrollo. Algunos de los lenguajes de programación son Java, Processing, Python, Mathematica, Perl, Visual Basic, entre otros. El entorno de desarrollo es sencillo y de código abierto donde puede descargarse gratuitamente y funciona en los sistemas operativos de Windows, Mac OS X y Linux.

Las áreas de utilidad más importante del programa son:

- Menú donde en la parte de herramientas de este se puede configurar el programa para que pueda notificarse con la placa de Arduino.
- Botones comunes que son accesos rápidos de acciones que se encuentran disponibles en el menú, como por ejemplo el botón de verificación, de carga, nuevo, abrir, guardar y monitor serial.
- Editor de texto, en esta área se escribe la implementación para que pueda ser cargada en la placa Arduino. El programa consta de 3 partes. La primera permite incluir librerías y declaraciones o variables globales. La segunda es el método “`sestup()`” el cual se encarga de iniciar los dispositivos que se encuentren conectados a la placa y se ejecuta al iniciar el sistema. Finalmente, la tercera parte consiste en el método “`loop()`” donde se ejecuta el código repetitivamente mediante un modo bucle.
- Áreas de mensajes que muestran la situación del sistema cuando se usa uno de los botones comunes.
- Consola de texto que es donde podrán aparecer con más detalles los eventos del área de mensajes.

### **3.4.1.2 Dispositivos acoplables a Arduino**

Para poder conseguir y proyectar los datos que queremos obtener del órgano central del Arduino, es necesario contar con dispositivos, tales como sensores y pantallas que ayuden a recoger los datos y mostrar los resultados. Estos son periféricos de entrada y salida como se mencionaba anteriormente, donde obtienen información del exterior y la trasladan al microordenador para que puedan procesar estos datos y sean proyectados a través del periférico de salida.

- **Sensor:** dispositivo que transforma magnitudes físicas o químicas y en variables eléctricas. Estas variables pueden ser de temperatura,

intensidad lumínica, distancia, humedad, pH, etc. La salida del sensor es un dato útil para un sistema de medición. Por lo tanto, el sensor es un dispositivo de entrada ya que siempre será un intermediario entre la variable física y el sistema de medida.

- **Pantalla LCD:** Son dispositivos digitales de salida, su función consiste en recibir las instrucciones del microordenador y proyectar la información. Estas pantallas son compatibles con cualquier hardware libre y las placas de proyecto Arduino.

Hoy en día encontramos en el comercio tres tipos de pantalla LCD:

- LCD de líneas
- LCD por puntos
- Display OLED-LED.TFT

Los pines que se necesitan en una placa de Arduino para conectar las pantallas LCD:

- GND y VCC
- Contraste
- RS
- RW
- Pines de D0 a D7
- Dos pines para Blacklight

## Capítulo 4: Resultados

En el presente capítulo se muestran los resultados obtenidos de todas las fases del proyecto. Sin embargo, es necesario recordar que la metodología utilizada en el proyecto es Desing Thinking, por lo cual la estructura del proyecto se fundamenta en esta metodología. A continuación, se dará a conocer la planificación del proyecto, explicando cómo se trabajó en cada una de las fases de la metodología utilizada

Fases	Descripción	Abril				Mayo				Junio				Julio				Agosto				Septiembre				Octubre				Noviembre				Diciembre			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Investigar la calidad y la nanotecnología en el transporte de alimentos	■	■	■	■	■	■	■	■	■	■	■	■																								
2	Conectar la placa de Arduino con los módulos (Sensor SHT31 y Pantalla LCD)												■	■																							
3	Desarrollar las funciones del prototipo del sistema Arduino IDE													■																							
4	Desarrollar prototipo de sistema de información para registrar los datos obtenidos desde el sistema de Arduino													■	■																						
5	Realizar primera prueba de concepto y analizar funcionamiento del sistema														■																						
6	Desarrollar las correcciones del prototipo del sistema con Arduino IDE															■																					
7	Desarrollar las correcciones del sistema de información																■																				
8	Realizar segunda prueba de concepto y analizar funcionamiento del sistema corregido																	■																			
9	Desarrollar las correcciones finales del sistema con Arduino IDE																																				
10	Desarrollar las correcciones finales del sistema que guarda los datos obtenidos																																				
11	Realizar prueba final de concepto																																				
12	Redactar libro de tesis	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				

Figura 1. Planificación del proyecto - Fuente (creación propia,2021)

### 4.1 Fases del proyecto

En esta sección se describirán las fases de la metodología Desing Thinking las cuales son:

#### 4.1.1 Empatizar

La variación en las distintas condiciones fitosanitarias que sufren los productos congelados y refrigerados durante el transporte es una problemática que enfrentan día a día las empresas pertenecientes a la industria de alimentos. Tal problemática deja en duda la trazabilidad de estos productos debido a que pierden su calidad, lo cual se traduce en aparición de bacterias, pérdida nutricional y malos olores, afectando principalmente a los consumidores finales.

Para captar las necesidades del usuario, generamos contacto con un grupo de manipuladoras de alimentos que trabajan en un colegio de la ciudad de Talca, recibiendo, cocinando y repartiendo alimentos a una gran cantidad de alumnos. En

primera instancia se consultaron los tipos de productos alimenticios con los que trabajan, a lo que ellas nos indicaron que trabajaban con productos congelados, refrigerados y no perecibles. Por consiguiente, se consultó si ellas han notado que productos congelados o refrigerados pierden su cadena de frío, la respuesta fue un rotundo sí. Y es que productos que pierden su cadena de frío se reciben semana a semana en cada colegio o liceo, sobre todo en las épocas de primavera y verano. Finalmente, las manipuladoras de alimentos nos comentaron que ellas son las que registran la temperatura de cada producto congelado y refrigerado en un cuaderno, por lo cual toda esta información termina perdiéndose.

#### **4.1.2 Definir**

En consecuencia, a partir de la fase anterior, se definió que tanto los consumidores finales, como así también quienes fabrican y transportan alimentos, necesitan una herramienta que dé certeza de que las condiciones fitosanitarias se respetan en todo el trayecto de transporte que experimentan los productos alimenticios. Esto debido a que, sobre todo los productos congelados y refrigerados, experimentan rupturas en la cadena de frío durante su transporte ya sea a almacenes minoristas, mayoristas, colegios o liceos.

#### **4.1.3 Idear**

En esta fase del proyecto se formularon distintas posibles soluciones al problema descrito anteriormente. A continuación, se detallan cada una de las alternativas de solución:

- **Envases inteligentes:** La creación de envases inteligentes a través de sensores que detectan el incremento sustancial de bacterias en los productos alimenticios es una de las posibles soluciones en la cual tanto el consumidor, como el fabricante o intermediario, pueden saber en qué estado se encuentra el producto alimenticio. Es una solución que está en boca de todos actualmente, sobre todo en lo que se conoce como la Industria 4.0.

La ventaja de los envases inteligentes es notoria debido a que la seguridad general del producto aumenta significativamente y los desperdicios de alimentos se minimizan a la vez.

En la actualidad existe una gran variedad de sensores y biosensores que se pueden aplicar a estos envases inteligentes, sin embargo, su desarrollo y producción involucra un alto costo. Pero la desventaja principal es que estos envases inteligentes no mantienen un control y monitoreo de la cadena de frío durante el transporte de los productos.

- **Sistema de monitorización:** Es importante, primero que todo, recalcar que los sistemas de monitorización principalmente se enfocan en obtener información del entorno a través de sensores para posteriormente analizar y registrar estos datos a través de una interfaz.

En este caso, el sensor a utilizar idealmente sería uno que obtenga la temperatura y humedad del entorno, el cual corresponde a la maquinaria utilizada para el transporte de alimentos (camiones, camionetas, barcos, etc.). A su vez, este sistema contaría con una pantalla para proyectar las máximas y mínimas de tanto la temperatura como la humedad. Con esto podríamos saber los rangos de temperatura y humedad que experimentaron los alimentos transportados y en efecto saber si la cadena de frío sufrió alguna ruptura o no. Por consiguiente, estos rangos se registrarían en un sistema de información con la finalidad de poder analizarlos en un futuro y tomar decisiones con respecto al transporte de productos.

Algunas de las ventajas de los sistemas de monitoreo son la rotunda mejora en la toma de decisiones, previenen los problemas antes de que estos lleguen a los consumidores finales y a su vez permiten aprender de circunstancias pasadas con el fin de maximizar el rendimiento actual. Sin embargo, la principal desventaja de los sistemas de monitorización es la implementación de los mismos. La implementación de sistemas, la mayoría de las veces afecta a las empresas debido a la resistencia al cambio que éstas tengan.

#### **4.1.4 Prototipar**

Luego de idear y describir las distintas alternativas de solución, decidimos que la solución más conveniente e ideal a la problemática descrita es el desarrollo de un

sistema de monitoreo, A continuación, se describen las distintas etapas que se llevaron a cabo en el desarrollo del prototipo de la solución elegida.

- **Conexión de la placa Arduino Uno con los módulos correspondientes**

Como se mencionó anteriormente, para crear el prototipo de un sistema que lea la temperatura y humedad ambiente durante el proceso de transporte de alimentos, se utilizó una placa Arduino Uno R3 como microcontrolador, un módulo de entrada que corresponde al sensor SHT31 y un módulo de salida correspondiente a una Pantalla LCD.

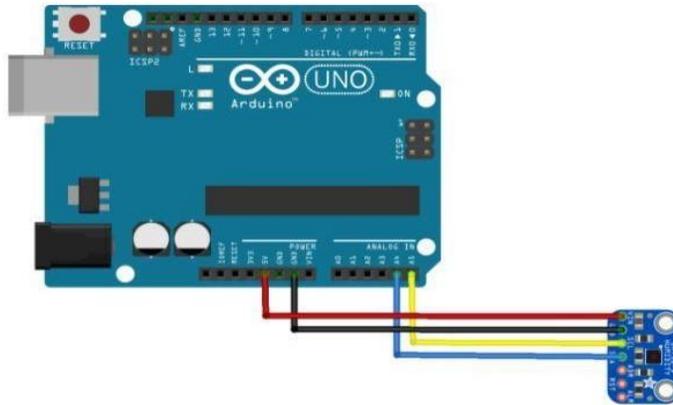
La conexión entre los módulos y la placa Arduino se llevó a cabo a través del protocolo de comunicación i2c. Este protocolo posee una arquitectura que consiste en dos tipos de roles:

- **Maestro:** La mayoría de las veces la placa Arduino cumple este rol, el cual se caracteriza por iniciar y coordinar la comunicación
- **Esclavo:** Estos dispositivos se mantienen a la espera de que un dispositivo maestro se comunique con ellos. En nuestro caso, el sensor SHT31 y la pantalla LCD cumplen este rol.

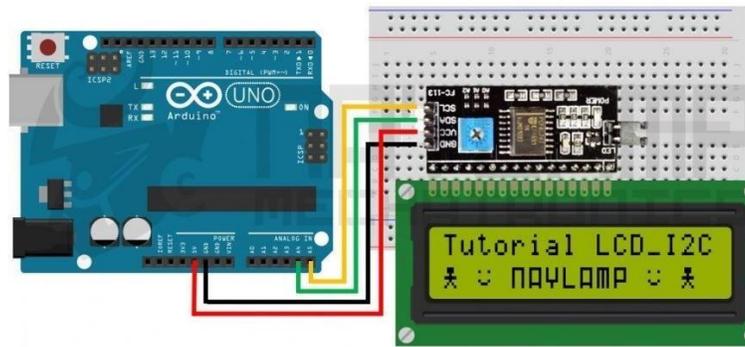
Para la conexión entre los módulos y la placa se necesitaron 4 pines por cada módulo:

- **5V:** Es el pin que le da el voltaje a los módulos.
- **GND:** Corresponde al pin de referencia de tierra. Este pin es común en todos los dispositivos de un circuito.
- **SDA:** Este pin es parte del protocolo de comunicación i2c y permite el intercambio de datos entre la placa y los módulos.
- **SCL:** Este pin se emplea como señal de reloj y también es parte del protocolo de comunicación i2c

A continuación, se detallan las conexiones del sensor SHT31 y la pantalla LCD con la placa Arduino Uno R3 en las siguientes figuras 2 y 3:



*Figura 2. Conexión entre Sensor SHT31 y placa Arduino Uno R3 (Fuente: <http://arduinolearning.com>).*



*Figura 3. Conexión entre Pantalla LCD y placa Arduino Uno R3 (Fuente: <https://naylampmechatronics.com>).*

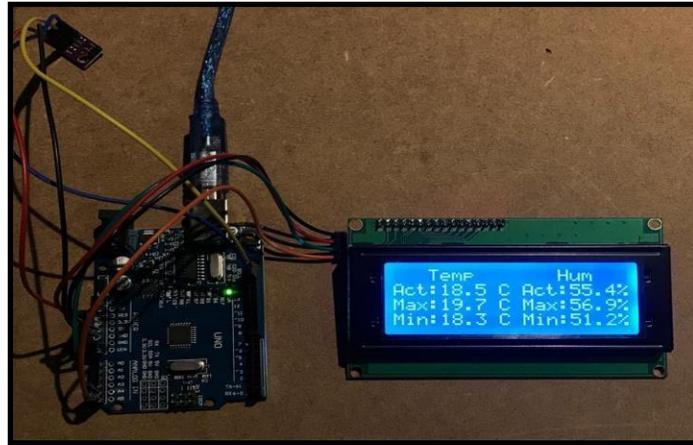
#### - **Desarrollo del sistema en Arduino**

Para el desarrollo del prototipo del sistema con Arduino fue necesario utilizar la plataforma Arduino IDE con el fin de programar todas las funciones que se requiere tener en el dispositivo que mida la temperatura ambiente en el transporte de los alimentos. Para el correcto funcionamiento de los módulos utilizados, se tuvo que descargar la librería correspondiente al sensor SHT31 (Adafruit\_SHT31) y a la pantalla LCD (LiquidCrystal\_I2C).

En primera instancia se creó un método para inicializar tanto la pantalla LCD como el sensor SHT31. Luego se crearon las variables máximas y mínimas tanto de la temperatura como la humedad, para seguidamente crear un método

con el fin de leer la temperatura y humedad proveniente del sensor, y luego ir guardando estos datos según la variable que corresponda. Finalmente, dentro del último método mencionado se programan las funciones para que la pantalla LCD proyecte las variables anteriormente mencionadas con sus respectivas actualizaciones.

En la figura número 4 se muestra el prototipo funcionando:



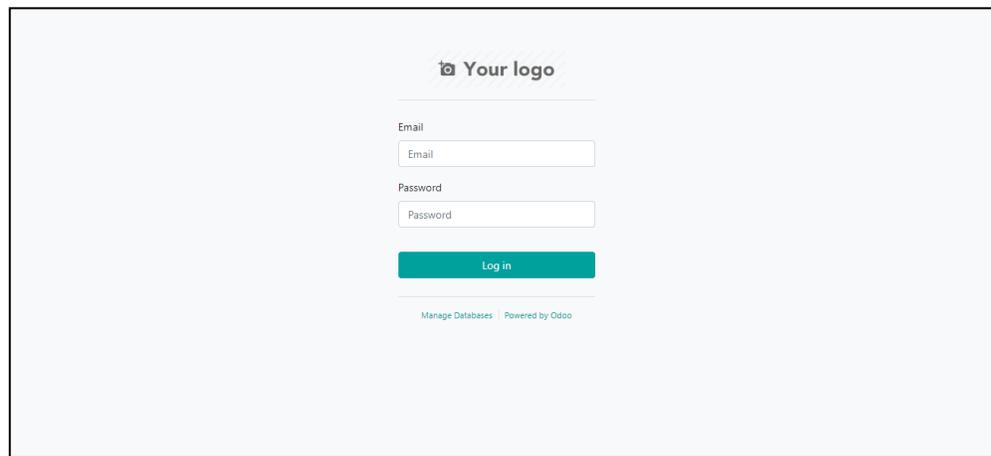
*Figura 4. Fotografía del sensor y la pantalla en función- Fuente (creación propia,2021)*

#### - **Desarrollo del sistema de información**

Para comenzar esta fase tuvimos que establecer ciertas herramientas que son necesarias para desarrollar el sistema de información. En primera instancia seleccionamos el framework Odoos anteriormente conocido como OpenERP, debido a sus características de sistema de ERP integrado de código abierto y catalogada como una herramienta más eficaz de gestión empresarial de hoy en día. Odoos trabaja en un entorno de desarrollo llamado OpenObject, el cual permite crear de manera rápida las aplicaciones de gestión orientadas a objetos en Python y con una base de datos PostgreSQL dando a que es el sistema de gestión de base de datos que ofrece la gran mayoría de la sintaxis SQL. Adicionalmente, para desarrollar el sistema se utilizó Pylance que es un servidor de lenguaje de Python, también XML como lenguaje de marcas extensibles el cual ordena la información y luego HTML la manera en que

será presentada en el navegador. Como editor de texto tenemos a Visual Studio Code, donde Microsoft (2021) señala que es un editor poderoso y ágil que cuenta con soporte de JavaScript, TypeScript y Node.js, asimismo cuenta con extensiones para lenguajes C ++, C #, Java, Python.

Una vez instalado Odoo en Ubuntu 18.04 con los respectivos paquetes de desarrollo de Python, la base de datos, WKHTML y las dependencias de Odoo, procedemos a crear el módulo a través de un código en Visual Studio Code. Compilamos el código y en primera instancia nos señala que tenemos que crear un login, tal como no se muestra en la figura 5.



*Figura 5. Vista del login del sistema - Fuente (creación propia- 2021)*

Una vez registrados los datos se selecciona el módulo creado que se encuentra en el menú de las aplicaciones de Odoo, como se muestra en la figura 6.

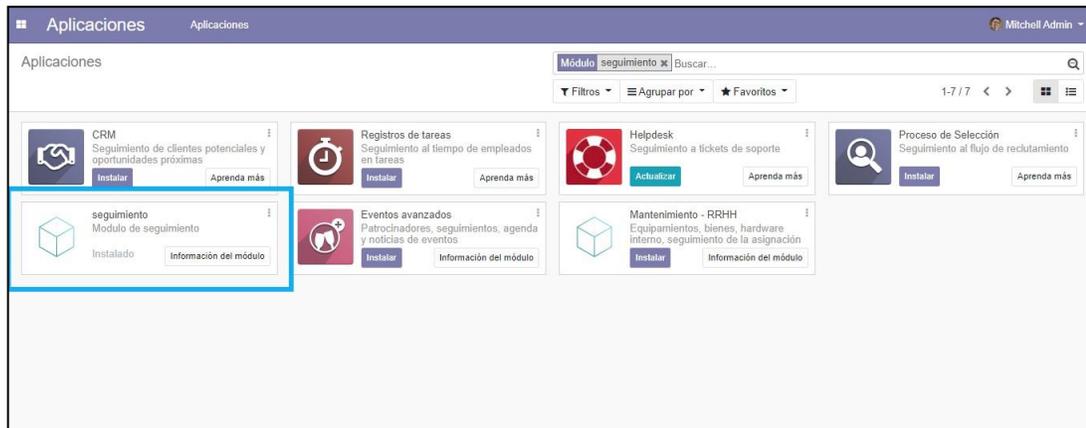


Figura 6. Vista del módulo creado en las aplicaciones de Odoo - Fuente (creación propia-2021)

Luego de instalar el módulo se mostrará en pantalla lo que generó la estructura del código, donde se visualizará el control del alimento en el cual el usuario debe registrar el nombre de los alimentos, la fecha de llegada, la temperatura máxima y mínima al igual que la humedad, tal como se muestra en la figura 7. Previamente a esto la persona que quiera entrar a este sistema debe registrarse e ingresar como un usuario a la plataforma, donde se muestra en la figura 5.

The image shows the 'Seguimiento' form in Odoo. The header includes 'Seguimiento' and 'Seguimientos'. Below the header, there are 'Guardar' and 'Descartar' buttons. The main content area is titled 'Control de producto' and contains a form with the following fields:

Nombre del producto	<input type="text"/>
Fecha de llegada	<input type="text"/>
Temp Max	<input type="text" value="0,00"/>
Temp Min	<input type="text" value="0,00"/>
Hum Max	<input type="text" value="0,00"/>
Hum Min	<input type="text" value="0,00"/>

*Figura 7. Vista del sistema desde la perspectiva del usuario - Fuente (creación propia,2021)*

- **Corrección del sistema Arduino**

Luego de realizar la primera prueba de concepto, los resultados obtenidos fueron presentados al profesor guía de este proyecto, el cual se mostró conforme con el resultado del sistema Arduino. Sin embargo, en el código del sistema se encontraron falencias en cuanto al orden en el que se escribió, por lo que se nos sugirió comentar cada uno de los métodos empleados en el código, a fin de que se especificara de correcta forma el fin con el que se utilizaron los métodos.

El código estaba escrito como se observa en las figuras 14, 15 y 16 del anexo. En primera instancia se inicializa la pantalla LCD con el comando `LiquidCrystal_I2c lcd(0x27,20,4)` para que luego se encienda la luz de esta. Inmediatamente se inicializa el sensor SHT 31 con la línea de código `Adafruit_SHT31 =Adafruit_SHT31()` y se crean las variables para guardar las máximas y mínimas de la temperatura y de la humedad.

Se comienza a leer la temperatura y humedad actual que está captando el sensor, luego se crea un método con el fin de guardar la temperatura máxima y mínima en su respectiva variable, al igual que la humedad. En seguida se selecciona el cursor donde queremos que se indique la temperatura, proyectando el encabezado a través de `lcd.print`, del mismo modo podemos indicar la temperatura actual con el mismo comando.

Con el comando `lcd.print` podemos indicar la temperatura máxima y mínima proyectando su variable y su título respectivamente, al igual que con la humedad.

- **Corrección del sistema de información**

En cuanto al sistema de información creado para guardar los datos obtenidos desde el sistema Arduino, se llegó a la conclusión de que se debía corregir la

estructura del sistema de información, con el fin de poder ingresar y guardar correctamente tanto los productos seleccionados, como los datos correspondientes a los mismos (temperatura y humedad durante el transporte).

En un comienzo, se había creado un módulo en Odoo llamado “seguimiento”, en este módulo se registraban todos los datos del transporte, productos, temperatura y humedad, todo en conjunto.

En las figuras 17, 18, 19 y 20 del Anexo se presenta la primera versión del código correspondiente al módulo “seguimiento”. En primera instancia tenemos lo que corresponde a la clase Control, desarrollada en lenguaje Python. En esta clase se guardan los datos del producto transportado y su respectiva temperatura y humedad

En segundo lugar, tenemos el desarrollo de las denominadas “vistas” del sistema, estas vistas fueron desarrolladas en lenguaje XML, utilizando los mismos campos creados en la clase “Control”. Sin embargo, esta estructura estaba lejos de estar a la altura de un sistema de información, por lo tanto, decidimos crear otro módulo aparte del módulo de “seguimiento”.

El nuevo módulo creado se llama “inventario” y tiene la finalidad de guardar todos los productos en sus respectivas categorías, separados totalmente del módulo de “seguimiento”. Esto quiere decir que cuando debamos ingresar los datos correspondientes al transporte de alimentos, los productos que podrán ser seleccionados son los que están ingresados en el módulo de “inventario”.

En lo que respecta al código de este nuevo módulo, se crearon las clases “Producto” y “CategoriaProducto” como se muestra en las figuras 21, 22, 23, 24, 25 y 26 del Anexo.

A continuación, se muestra el código corregido, en donde se creó una nueva clase llamada “DetalleControl” la cual contiene todos los datos de lo que anteriormente era la clase llamada “Control”. Y ahora la clase “Control” solo contiene los datos del cliente al cual se le hacen llegar los productos transportados, y la fecha del transporte. De la misma forma, las vistas de las

clases fueron actualizadas y el código quedó como se muestran en las figuras 28, 29, 30 y 31 del anexo.

#### - **Corrección final del sistema de información**

Durante la ejecución de la segunda prueba de concepto observamos que, por cada producto que se transportaba, se tenía que indicar la temperatura y humedad máxima y mínima en el sistema, siendo un proceso abrumador en el caso de transportar una gran cantidad de productos. También, el sistema no daba la opción de especificar en qué área de transporte (Congelado o Refrigerado) iban los productos seleccionados. Debido a esta falla en el funcionamiento eficiente del sistema, se corrigió específicamente el módulo de “Control de transportes”. Para esto se efectuaron modificaciones tanto en el archivo Python como en el archivo XML del módulo. En la figura 32 se muestran los cambios efectuados tanto en el archivo .py como en el archivo .xlm en Visual Studio Code

En la clase Control anteriormente solo se tenía la información sobre el cliente al cual se le transportaban los productos y la fecha del transporte. Con la corrección desarrollada se llegó a la conclusión de incluir la temperatura y humedad máxima y mínima, así como también el área de transporte, es decir, área congelada o refrigerada.

En la clase de DetalleControl por lo tanto se eliminaron los atributos de temperatura y humedad máxima y mínima con el objetivo de no detallarlas cada vez que ingresemos un producto en el módulo de Control de transporte.

Como se cambiaron tanto las clases Control y DetalleControl desde los archivos Python, lo mismo se hizo con los archivos XML que son los que crean las vistas de dichas clases en el sistema. El código XML se puede observar en las figuras 34, 35, 36, 37 del anexo.

## Modelo Relacional del sistema de información

El modelo relacional del sistema de información consta de cuatro tablas:

- Producto
- CategoríaProducto
- Control
- DetalleControl.

En cuanto a los atributos, la tabla de producto tiene:

- id\_producto: Corresponde al atributo principal de cada producto, el ID propio de cada uno de ellos.
- name\_producto: Corresponde al nombre de cada producto.
- date\_elaboration: Corresponde a la fecha de elaboración de cada producto.
- date\_expiration: Corresponde a la fecha de expiración del producto.
- id\_categoria: Corresponde al ID principal de la tabla de categoría producto, debido a que los datos de esta última tabla ayudan a identificar el nombre de la categoría del producto, ya que un producto puede tener solo una categoría y una categoría puede tener muchos productos, por lo tanto, es necesaria la identificación en la tabla.

De la tabla CategoríaProducto, obtenemos los siguientes atributos:

- id\_categoria: Corresponde al identificador único de la categoría de producto.
- name\_categoria: Corresponde al nombre de la categoría de producto.
- cantidad: Corresponde a la cantidad de productos que se encuentran dentro de la categoría.

Así mismo, la tabla de control tiene los siguientes atributos:

- id\_control: Corresponde al atributo principal de la tabla.

- cliente: Corresponde al nombre del cliente al cual se le transportan los productos.
- fecha: Corresponde a la fecha en que el transporte de productos se efectuó.
- area: Corresponde al área en donde se transportan los productos (congelada o refrigerada).
- temp\_min: Corresponde a la temperatura mínima que se obtuvo durante el transporte de los productos.
- temp\_max: Corresponde a la temperatura máxima que se obtuvo durante el transporte de los productos.
- hum\_min: Corresponde a la humedad mínima que se obtuvo durante el transporte de los productos.
- hum\_max: Corresponde a la humedad máxima que se obtuvo durante el transporte de los productos.

Con respecto a la tabla de DetalleControl, tenemos los siguientes atributos:

- id\_control: Corresponde a toda la información obtenida desde la tabla Control.
- id\_producto: Corresponde a toda la información obtenida desde la tabla Producto.
- cantidad: Corresponde a la cantidad de productos transportados.
- La tabla DetalleControl es donde se ven los datos de las tablas de producto y control, por lo tanto, debe tener los ID de cada uno de ellos para generar el detalle.

#### **4.1.5 Testear**

- **Primera prueba de concepto**

La prueba de concepto según Guiltinan, Paul y Madden (Gerencia de Marketing Colombia 1999, página 206) genera un método para poder medir el interés de un comprador que tiene en un producto en específico, antes de

desarrollar el prototipo final. En consecuencia, el objetivo de esta prueba es desarrollar un estimativo de la función que tiene el sistema, la cual sería medir la temperatura y humedad de diferentes tipos de alimentos transportados de un lugar específico a otro. Todo esto con la finalidad de validar el proyecto y que en un futuro cada alimento lleve su código QR con la información del cumplimiento de la cadena de frío.

La primera prueba de concepto fue llevada a cabo en dos trayectos, los días 10 y 11 del mes de julio. El punto de inicio del trayecto fue el centro de abastecimientos de Talca (CREA), lugar donde se efectuó la compra de distintos tipos de verduras y frutas; tales como tomate, zanahoria, lechuga, manzanas y plátanos. Posterior a eso, el prototipo del sistema, el cual consiste en el sensor que recoge los datos de temperatura y humedad ambiente, se instaló en el maletero del auto junto con los alimentos comprados. El lugar para transportar los alimentos fue elegido debido a la similitud que tiene con los contenedores y camiones que transportan los alimentos, y más que nada por el hecho de que la temperatura y humedad del maletero no se vieran alteradas por la climatización del auto.

Los puntos finales de los dos trayectos fueron las casas de cada uno de los integrantes del proyecto respectivamente, y el tiempo de transporte fue de 20 minutos aproximadamente.

Durante el primer viaje se registró una temperatura máxima de 18,1°C y una temperatura mínima de 11,5°C. En cuanto a la humedad, se registró una máxima de 60,7% y una mínima de 55,2%. Con respecto al segundo viaje, la temperatura máxima que se registró fue de 17,2°C y la temperatura mínima fue de 13,9°C. La humedad máxima registrada fue de 60,3% y la mínima fue de 57,1%. Todos estos datos fueron proyectados a través de la pantalla lcd que forma parte del sistema.

Finalmente, se da por terminada la primera prueba de concepto a las 14:15 horas del día 11 de julio, por lo que se procede a registrar los datos obtenidos en el sistema de información como se presenta en la figura 8.



<input type="checkbox"/>	Nombre del producto	Fecha de llegada	Temp Max	Temp Min	Hum Max	Hum Min
<input type="checkbox"/>	Tomates	10/07/2021	18,10	11,50	60,70	55,20
<input type="checkbox"/>	Lechugas	10/07/2021	18,10	11,50	60,70	55,20
<input type="checkbox"/>	Paltas	10/07/2021	18,10	11,70	60,70	55,20
<input type="checkbox"/>	Carne de vacuno	11/07/2021	17,20	13,90	60,30	57,10
<input type="checkbox"/>	Zanahoria	11/07/2021	17,20	13,90	60,30	57,10

*Figura 8. Vista del registro de los datos obtenidos en la primera prueba de concepto - Fuente (creación propia- 2021)*

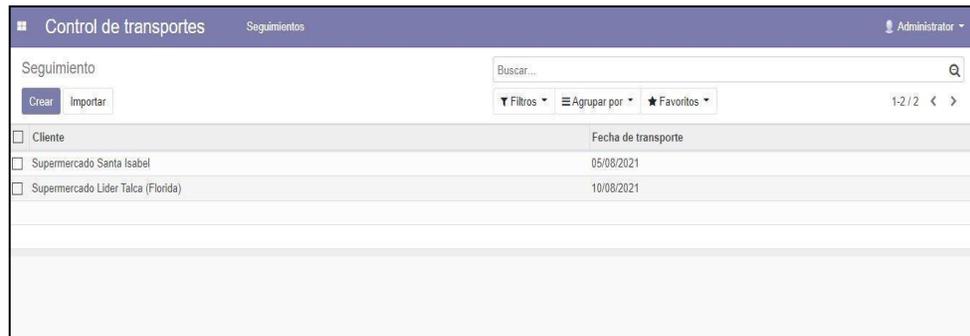
Se considera que el prototipo cumple con la funcionalidad que se desea para el sistema, sin embargo, es de suma importancia recalcar que la temperatura y humedad que se registraron dependen netamente de la condición climática del día en que se realizó la prueba y también de las condiciones del lugar donde se transportan los alimentos.

#### - **Segunda prueba de concepto**

La segunda prueba de concepto resultó ser distinta a la primera, esto se debió a las correcciones efectuadas tanto en el sistema que mide la temperatura y humedad, como en el sistema de información en el cual se registran todos los datos obtenidos de cada transporte de alimentos que se efectúe. Sin embargo, el cambio más significativo se dio en el sistema de información. Por lo tanto, la segunda prueba de concepto se dividió en dos fases; en la primera fase se realizó una medición de temperatura y humedad con el sistema creado en arduino, teniendo resultados similares a los de la primera prueba de concepto y en la segunda fase de la prueba se cargaron más de cien datos en el sistema de información destinado a registrar y guardar datos. Los datos cargados fueron, en su mayoría, productos y categorías de productos correspondiente al módulo Inventario, en tanto los registros hechos en el módulo Control de

transportes corresponden a las series de veces que los investigadores han hecho distintas compras en distintos supermercados u otros lugares.

A continuación, se muestran en la figura 9 las vistas correspondientes a la carga de datos efectuada en el sistema de información.

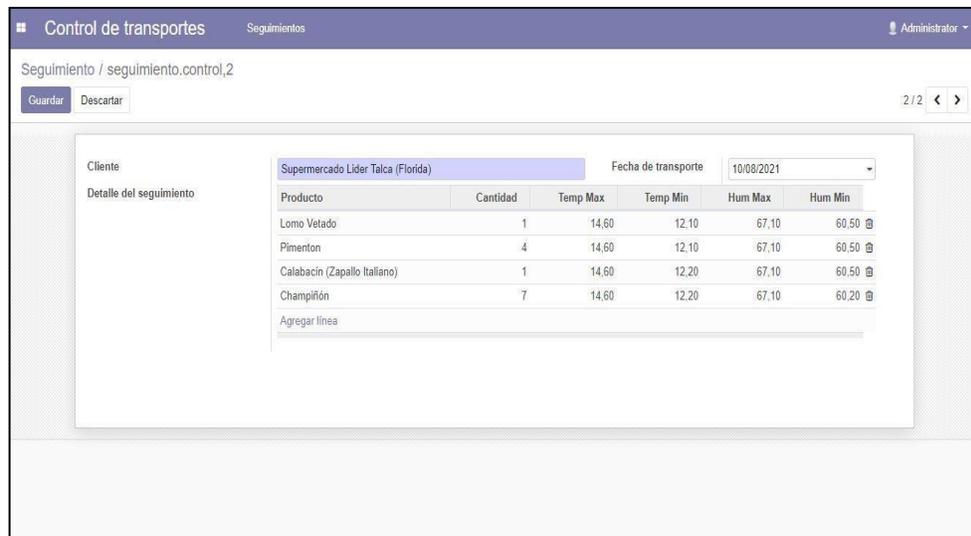


The screenshot shows a web application interface for 'Control de transportes' with a 'Seguimientos' sub-section. The interface includes a search bar, 'Crear' and 'Importar' buttons, and a table with columns for 'Cliente' and 'Fecha de transporte'. The table contains two entries: 'Supermercado Santa Isabel' with a date of '05/08/2021' and 'Supermercado Líder Talca (Florida)' with a date of '10/08/2021'. The interface also features a 'Filtros' dropdown, 'Agrupar por' and 'Favoritos' options, and a page indicator '1-2/2'.

Cliente	Fecha de transporte
<input type="checkbox"/> Supermercado Santa Isabel	05/08/2021
<input type="checkbox"/> Supermercado Líder Talca (Florida)	10/08/2021

*Figura 9. Vista del módulo Control de Transportes - Fuente (creación propia- 2021)*

En el módulo llamado Control de transportes es en donde se registran todos los transportes de alimentos realizados. En primera instancia se observan solo lo que es el cliente al cual se transportan los productos y su respectiva fecha. A diferencia de la primera iteración del sistema de información, se ha creado un detalle de seguimiento en el cual se registran todos los productos transportados, cada uno con sus respectivos rangos de temperatura y humedad, como lo podemos visualizar en la figura número 10.



*Figura 10. Vista del Detalle de seguimiento correspondiente al módulo Control de Transportes - Fuente (creación propia- 2021)*

Para poder crear un detalle de seguimiento, se tuvo que desarrollar un módulo llamado Inventario. Como se detalló anteriormente, este módulo se encarga de registrar y guardar los productos que posteriormente serán transportados. Para el correcto funcionamiento de este módulo, se cargaron más de 100 productos, cada uno con su respectiva categoría, como se visualiza en la siguiente figura 11.

Nombre	Fecha de elaboración	Fecha de vencimiento	Categoría
Lomo Vetado	01/08/2021	29/08/2021	Carnes
Pimenton	09/07/2021	30/09/2021	Verduras
Alfajor	10/07/2021	13/08/2021	Pastelería
Alitas de pollo	17/07/2021	08/08/2021	Carnes
Ameja	14/07/2021	12/08/2021	Pescado
Arroz con leche	11/07/2021	08/08/2021	Lácteo
Berenjena	01/07/2021	03/08/2021	Fruta
Brazo de reina	27/07/2021	06/08/2021	Pastelería
Brocoli	10/07/2021	15/08/2021	Verdura
Calabacín (Zapallo Italiano)	27/07/2021	03/08/2021	Verduras
Calamar	18/07/2021	08/08/2021	Pescado
Camarón	01/08/2021	17/08/2021	Pescado
Cebolla	28/07/2021	09/08/2021	Verduras
Centolla	17/07/2021	11/08/2021	Pescado
Cerezas	09/07/2021	09/08/2021	Fruta
Champiñón	12/07/2021	03/08/2021	Verduras
Cheesecake	01/07/2021	16/08/2021	Pastelería
Chorrito	23/07/2021	14/08/2021	Pescado

*Figura 11. Vista de los productos cargados en el módulo Inventario - Fuente (creación propia- 2021)*

En el módulo de inventario se visualizan dos pestañas en la parte superior izquierda de la vista. En la pestaña “Productos” están registrados los alimentos anteriormente mencionados y en la pestaña “Categorías” se encuentran todas las categorías de los alimentos actualmente disponibles para ser transportados.

Cada producto alimenticio es registrado con su respectiva fecha de elaboración y vencimiento, con el fin de saber dentro de qué fecha el alimento puede ser transportado.

Nombre	Total Productos
Carnes	4
Verduras	4
Pescado	5
Verdura	1
Lácteo	1
Congelados	0
Fiambre	1
Pastelería	3
Fruta	2

Figura 12. Vista de las categorías de productos cargadas en el módulo Inventario - Fuente (creación propia- 2021)

- **Prueba de concepto final**

Esta última prueba de concepto realizada el día 28 de septiembre se realizó con éxito gracias a las correcciones de las pruebas de concepto anteriores, donde el sistema de información nuevamente fue modificado para obtener un producto final que cumpla con todas las condiciones. Por consiguiente, en la última prueba de concepto se realizaron las mediciones de las temperaturas y humedad de varios productos, tales como verduras, frutas y carnes, a través del sistema arduino que creamos donde los resultados fueron similares a las pruebas anteriormente realizadas, por tanto, consideramos que este sistema funciona de manera exitosa y que cumple con el objetivo del proyecto. Por otro lado, tenemos el sistema de información donde modificamos el módulo de inventario, cargando datos ficticios y que corresponden al detalle de los seguimientos de todos los transportes que se van a realizar. Se cargaron 100 datos correspondiente al módulo de control de transporte en los cuales se detalla el nombre de los clientes, las temperaturas máxima y mínima, al igual que la humedad, la fecha y finalmente el área donde corresponde el producto. A continuación, se muestra en la figura 13 las vistas correspondientes a los datos cargados en el módulo de control de transporte.

Cliente	Area de transporte	Fecha de transporte	Temp Max	Temp Min	Hum Max	Hum Min
<input type="checkbox"/> Agromir S A	Refrigerado	02/10/2021	6,00	3,00	61,00	43,00
<input type="checkbox"/> Mayov Industria	Refrigerado	10/01/2022	5,00	2,00	60,00	46,00
<input type="checkbox"/> Brisas del campo	Refrigerado	13/11/2021	9,00	1,00	66,00	47,00
<input type="checkbox"/> Terra libre	Refrigerado	16/11/2021	5,00	1,00	61,00	42,00
<input type="checkbox"/> OrganiX	Refrigerado	22/02/2022	11,00	1,00	63,00	43,00
<input type="checkbox"/> La cabaña	Congelado	20/11/2021	7,00	3,00	62,00	42,00
<input type="checkbox"/> McCain	Congelado	11/10/2021	12,00	4,00	66,00	42,00
<input type="checkbox"/> Minuto verde	Refrigerado	14/01/2022	9,00	1,00	68,00	48,00
<input type="checkbox"/> Frutos del Maipo	Refrigerado	25/02/2022	6,00	3,00	61,00	50,00
<input type="checkbox"/> Not Burger	Congelado	25/12/2021	9,00	2,00	60,00	53,00
<input type="checkbox"/> La Crianza	Congelado	26/02/2022	5,00	3,00	70,00	54,00
<input type="checkbox"/> Super Pollo	Congelado	07/10/2021	5,00	2,00	69,00	55,00
<input type="checkbox"/> Mastodonte	Refrigerado	05/10/2021	5,00	2,00	70,00	43,00
<input type="checkbox"/> Mister	Refrigerado	09/11/2021	13,00	3,00	61,00	47,00
<input type="checkbox"/> Receta del abuelo	Congelado	05/11/2021	7,00	2,00	62,00	41,00
<input type="checkbox"/> La Preferida	Congelado	19/02/2022	14,00	4,00	68,00	53,00
<input type="checkbox"/> Sadia	Congelado	22/02/2022	13,00	1,00	63,00	54,00
<input type="checkbox"/> King	Refrigerado	11/02/2022	13,00	2,00	62,00	41,00

*Figura 13. Vista de los datos cargados al módulo de control de transportes-  
Fuente (creación propia- 2021)*

Para concluir esta última prueba de concepto que realizó, podemos decir que ambos sistemas funcionan correctamente, donde cada uno cumple su función y a su vez con el objetivo del proyecto que es poder verificar si las temperaturas y humedad de diferentes productos están en los rangos correspondiente, y ordenar esta información en un sistema de manera detallada para un mejor control de parte de las empresas que quieran tener este sistema.

## Capítulo 5: Análisis de resultados

En el presente capítulo se analizarán los resultados obtenidos a lo largo de todo el desarrollo del sistema, especificando las diferentes relaciones y generalizaciones que tienen dichos resultados. También se abordarán los aspectos no resueltos, con el fin de no ocultar los límites de este.

Para dar contexto, partiremos resumiendo todo el proceso de desarrollo del sistema para luego explicitar las relaciones y generalizaciones que se obtuvieron en los resultados.

Para desarrollar este sistema fue creado con la finalidad de garantizar la calidad fitosanitaria de los alimentos en la logística de transporte, se utilizó la metodología Design Thinking ya que, el objetivo de esta tesis es crear un prototipo de sistema. Recordemos que la metodología Design Thinking se diferencia de los otros enfoques del diseño, puesto que se centra más que nada en el proceso que en el lugar del producto, es decir, se enfoca en resolver el problema, pero no comienza necesariamente teniendo una solución previa.

En un comienzo, para captar las necesidades de los usuarios se hizo contacto con un grupo de manipuladoras de alimentos, de este contacto se pudieron identificar los productos que generalmente se transportan, los cuales son productos congelados, refrigerados y no perecibles. También, en esta fase, debido a los productos identificados, también se identificaron las distintas zonas (refrigerado/congelado) en las que se transportan los alimentos puntualmente en camiones.

Debido a estas necesidades, se definió que tanto los consumidores finales, como así también quienes fabrican y transportan alimentos, necesitan una herramienta que dé certeza de que las condiciones fitosanitarias se respetan en todo el trayecto que experimentan los productos alimenticios. Por lo tanto, se procedió a idear distintas alternativas de solución a este problema, de las cuales se eligió desarrollar un sistema de monitorización tanto de humedad como de temperatura.

Durante la fase de prototipado se desarrolló tanto el sistema que obtiene la temperatura y humedad ambiente, así como también el sistema de información desarrollado con el fin de registrar estos datos, específicamente las mínimas y máximas tanto de la temperatura como de la humedad. Estos dos sistemas fueron desarrollados en plataformas distintas, el primero

se desarrolló netamente con Arduino mediante un sensor el cual obtiene datos de temperatura y humedad, y el segundo se desarrolló en Odoo gracias los lenguajes HTML y Python. Cabe destacar que una vez avanzado el prototipo, se empezó a itinerar esta etapa con la etapa de testeo, llevando a cabo tres pruebas de concepto de las cuales se obtuvieron correcciones de cada una.

Con las iteraciones mencionadas se observó una notable evolución en el sistema general, sobre todo en lo que respecta al sistema de información desarrollado con la finalidad de registrar los datos obtenidos de los sensores. Esta evolución se debe a que, en un comienzo, se había creado un módulo en Odoo llamado “Control de transporte”, en este módulo se registraban todos los datos del transporte: productos, temperatura y humedad, todo en conjunto. Sin embargo, esta estructura estaba lejos de estar a la altura de un sistema de información, por lo tanto, decidimos crear otro módulo aparte del módulo de “Control de transporte”. El nuevo módulo creado se llama “Inventario” y tiene la finalidad de guardar todos los productos en sus respectivas categorías, separados totalmente del módulo de “seguimiento”.

En la primera prueba de concepto se observó que el sistema desarrollado en Arduino cumplía a la perfección con lo que se requería: obtener las máximas y mínimas de tanto la temperatura como la humedad, por lo tanto, en las siguientes pruebas de concepto solamente se realizaron cargas de datos en el sistema de información destinado a registrar los datos obtenidos de los sensores. En la segunda prueba de concepto se realizó un registro de más de cien datos correspondiente a productos y categorías de productos, esto en el módulo de Inventario respectivamente. En la tercera prueba de concepto se realizó otro registro de datos, pero esta vez en el módulo de Control de transportes. Fueron cien registros los que se efectuaron en este módulo y son registros que pertenecen netamente a la información que se obtiene de cada transporte: Cliente al que se le transportan los productos, área en que se transportan los productos (refrigerado/congelado), fecha de transporte, temperatura máxima y mínima, humedad máxima y mínima.

A continuación, se detallan las ventajas y desventajas del sistema desarrollado:

- **Ventajas**

Una de las principales ventajas del sistema desarrollado se encuentra en el registro de los datos en el ERP Odoo.

Al desarrollar este sistema en la plataforma Odoo, se puede integrar con diferentes funciones, tales como gestión de proyectos, comercio electrónico, almacén, CRM, facturación, fabricación, contabilidad e inventario. Justamente fue con esta última función o módulo que integramos nuestro módulo de Control de transporte. Así, el usuario puede comenzar usando módulos básicos y, en un futuro, cuando la empresa de expanda, puede seguir agregando más módulos, ya sea desarrollando, debido a que el código fuente se encuentra totalmente disponible, o integrando los existentes. Toda esta integración permite que el usuario pueda personalizar el software acorde a los requerimientos de la organización, con el fin de suplir las necesidades más específicas de la empresa, sobre todo en estos tiempos de ambientes empresariales tan cambiantes debido a las nuevas tecnologías que van desarrollándose.

En cuanto a la interfaz de usuario, el sistema posee una apariencia simple pero aun así es un sistema extremadamente fácil de usar.

Las funciones de búsqueda que posee el sistema en el cual se registran los datos, se han diseñado a modo de conveniencia general, así los usuarios fácilmente pueden filtrar sus búsquedas como, por ejemplo; por fecha, por temperatura o humedad, máxima o mínima, entre otros.

Nuestro sistema, debido a que se trata de un prototipo, requiere de una implementación a muy bajo costo, ya que el hardware usado para desarrollar el sistema pertenece a la familia de microcontroladores de Arduino.

Por último, tenemos que, a través de la metodología para diseñar prototipos, está la ventaja de poder corregir los errores, modificar e incluso implementar nuevas funciones para un mejor resultado, tanto para la empresa como para el cliente.

## - **Desventajas**

A pesar de todas las ventajas mencionadas, el sistema desarrollado tiene sus límites. Una de las principales desventajas del sistema desarrollado es que los datos obtenidos de los sensores no se registran automáticamente en el sistema destinado a registrar estos datos. Por lo tanto, todos los registros, hasta el momento, se deben realizar de manera manual por un empleado designado para esa tarea.

Anteriormente se mencionó que el usuario puede comenzar usando módulos básicos y, en un futuro, cuando la empresa se expanda, puede seguir agregando más módulos, ya sea desarrollando, debido a que el código fuente se encuentra totalmente disponible, o integrando los existentes. Sin embargo, a medida que la empresa u organización se expanda, necesitará invertir en los módulos o aplicaciones que desea desarrollar o implementar, por lo que se puede llegar a pagar excesivamente más de lo necesario, resultando en una desventaja.

Otra desventaja que podemos mencionar es cuando el dispositivo falla, ya que no hay una solución prevista para este problema, debido a que el proyecto está desarrollado de manera local, lo que quiere decir, que solo funciona con conexión USB, por consiguiente, si se produjera una falla del dispositivo en el transporte podría ser la falta de energía del artefacto que toma la temperatura y humedad.

## Capítulo 6: Discusión

En el presente capítulo se relacionan los resultados obtenidos del sistema desarrollado con la teoría de la investigación detallada en los primeros capítulos del libro.

En primer lugar, para relacionar los resultados obtenidos con la teoría de la investigación, tenemos que recordar que el objetivo principal de este proyecto es desarrollar un prototipo de sistema que garantice la calidad y seguridad de los alimentos durante la logística de transporte. Dado que el objetivo general es desarrollar solo un prototipo de sistema, se utilizó Arduino para desarrollar la plataforma que permite captar tanto la humedad y temperatura, lo que nos indica que la herramienta ocupada para captar estos indicadores es un sensor (SHT31) como se detalló en el capítulo tres (3). Por lo tanto, dado que en el presente proyecto solo se desarrolló un prototipo de sistema, no se usó ninguna herramienta perteneciente a la nanotecnología. Sin embargo, la diferencia entre sensores y nanosensores solamente es el tamaño del sensor, por lo que, en un futuro, cuando los nanosensores sean más accesibles comercialmente, se podría desarrollar un sistema que contenga herramientas pertenecientes a la nanotecnología con la misma metodología utilizada en el presente proyecto.

Así también, como se sabe, para el registro de los datos captados por el sensor, se desarrolló un módulo en Odoos llamado “Control de transporte”, dando como resultado una plataforma totalmente accesible para que el usuario pueda registrar las mínimas y máximas de cada indicador (temperatura y humedad). Sin embargo, para que el usuario pueda registrar estos datos en la plataforma debe tener un perfil creado, lo cual resulta en una limitante para el desarrollo del proyecto ya que no podemos explicitar la información del transporte de alimentos con los clientes finales (consumidores), debido a que estos no estarían registrados en la plataforma de Odoos.

En lo que respecta a la calidad en el transporte de alimentos, el sistema desarrollado en el presente proyecto contribuye a mejorar la calidad en la logística de los productos alimenticios debido a que, si hablamos de calidad, tiene que ver en primera instancia con las propiedades físicas de los alimentos y también, en segunda instancia, tiene que ver con cómo el consumidor percibe el alimento. El hecho de que nuestro sistema esté desarrollado con la intención de captar y registrar las máximas y mínimas de tanto la humedad como la temperatura, es con el fin justamente de resguardar las propiedades físicas de los alimentos

y también que, en un futuro, el consumidor pueda percibir y dar por hecho que el alimento sostuvo sus condiciones fitosanitarias durante toda la fase de transporte.

Recordemos que tal y como se indica en el capítulo dos (2), cada año se transportan alrededor de 200 mil millones de toneladas de alimentos en el mundo, el 35% del transporte se efectúa por vía terrestre mientras que el 60 % se efectúa por la vía marítima, siendo solamente un 5% de los alimentos transportados por la vía aérea. Debido a esto, es de suma importancia velar y dar un gran enfoque en la seguridad y calidad de los productos [14].

El sistema desarrollado justamente cumple con este enfoque y se puede implementar en cualquiera de las vías descritas (terrestre, marítima y aérea) ya que solo se tiene que efectuar la instalación de los microcontroladores con sus respectivos sensores en cada área (refrigerada o congelada) que tenga la maquinaria de transporte a utilizar y posteriormente, cuando el transporte llegue a su lugar de destino, solo se tendría que realizar el registro de los datos anteriormente descritos.

Cabe destacar que, en cuanto a la vía aérea, hay muchos problemas porque el producto está desprotegido de refrigeración durante gran parte del viaje, aunque esté revestido con algún material aislante [16]. En cuanto a la vía marítima, la gran mayoría de los contenedores destinados al transporte de alimentos se caracterizan por ser aislados o refrigerados; los contenedores aislados utilizan unidades de refrigeración de tipo enchufe o pueden conectarse directamente a un sistema de manipulación de aire de una bodega o muelle. En cambio, los contenedores refrigerados incorporan aislamiento y tienen unidades de refrigeración incorporadas en su estructura [17]. Finalmente, en lo que respecta a la vía terrestre, la maquinaria utilizada puede variar desde pequeñas furgonetas no aisladas que suministran alimentos a los puntos de ventas y consumidores, hasta contenedores de 12 metros, refrigerados y diseñados para rutas de larga distancia.

En lo que respecta a la metodología utilizada para llevar a cabo el proyecto, en primera instancia se usó una metodología bibliográfica con el fin de seleccionar y explicitar toda la teoría necesaria para entender y especificar el área a la que está destinada el proyecto. Gracias a esta metodología pudimos obtener información sobre la calidad actual en el transporte de alimentos, el rol de la nanotecnología en la industria y transporte de alimentos, y también

podimos obtener información sobre las distintas plataformas y sistemas en los cuales podíamos desarrollar y llevar a cabo el proyecto.

En segunda instancia, luego de analizar distintas metodologías en lo que respecta al desarrollo de sistemas de información, se decidió, en conjunto con el profesor guía del proyecto, utilizar la metodología Design Thinking debido a que tiene dos principales características:

- Está muy centrada en el usuario, debido a que las necesidades de este se observan y se añaden constantemente a un proceso de diseño promovido por el conocimiento y la creatividad.
- Es iterativa, ya que se corrige y se vuelve a evaluar, en una constante retroalimentación.

La segunda característica fue fundamental debido a que, gracias a esta, el sistema desarrollado evolucionó considerablemente y de forma positiva. Tales evoluciones están detalladas y se pueden observar en el capítulo cuatro (4) del presente libro.

Luego de discutir y relacionar los resultados obtenidos con la teoría y metodología del proyecto, finalmente daremos respuesta a las preguntas que se plantearon al inicio del proyecto, las cuales fueron propuestas con la finalidad de ser respondidas una vez que el sistema ya realice las tareas correspondientes.

Las preguntas se describen a continuación:

- ¿Quién debe realizar el control y seguimiento de los alimentos transportados?

El sistema está diseñado para que un usuario previamente registrado, pueda acceder a los datos mediante el login. Por lo tanto, la persona encargada de realizar el control es un trabajador de la empresa, el cual podría ser el encargado de bodega, de inventario o cualquier otro empleado que sea designado para esa labor, en donde registra los datos de la temperatura y humedad, al igual que el cliente, la fecha y el área del producto, de manera manual en el sistema como se mencionó anteriormente en las desventajas.

- ¿Cuál es la causa de que los alimentos pierdan sus condiciones fitosanitarias durante la fase de transporte?

La fase de transporte es muy importante dentro de la cadena, debido a que en esta deben garantizar la llegada de un producto fresco a su comercialización con rapidez y eficiencia. A pesar de esto, hay pérdidas muy elevadas relacionadas en esta fase debido a que no se respetan las condiciones necesarias para el transporte, como por ejemplo que el producto esté bien embalado y que sea trasladado en un vehículo apropiado.

Las principales causas que provocan que los alimentos pierdan sus condiciones fitosanitarias son las lesiones físicas y el recalentamiento, esta última relacionada directamente con los transportes no refrigerados, donde sus principales causas son:

- Utilizar un vehículo que no tenga ventilación.
- Abusar del espacio del vehículo para poder llevar más productos, impidiendo que circule el aire.
- Un mal embalaje del producto, esto debido a que no están suficientemente ventilados.
- Exponer los embalajes al sol antes de ser trasladados.

Mientras que las lesiones físicas se pueden producir por:

- Mal embalaje del producto.
- Una mala manipulación de la carga al apilarlos de forma incorrecta, produciendo derrumbes y una mala descarga.
- Por golpes que se producen principalmente en calles o carreteras que están en mal estado.
- Mala conducción del transportista o que el vehículo esté en mal estado.

## Capítulo 7: Conclusión

A modo de conclusión, podemos dar cuenta de que, en primera instancia, la investigación sobre los riesgos fitosanitarios en la logística de transporte fue de suma importancia para llevar a cabo este proyecto. Y es que, al cumplir este objetivo específico, se pudo tener conocimiento de las distintas vías y condiciones de transporte que experimentan los productos alimenticios hoy en día. Gracias a esto, y como segundo objetivo general, se propuso un sistema que contenga sensores, esto con la finalidad de rescatar información correspondiente a las máximas y mínimas de tanto la humedad como la temperatura. Este sistema se desarrolló a través de la plataforma de creación electrónica Arduino, cumpliendo perfectamente con la función esperada.

El desarrollo de un sistema de información fue de suma importancia, y es que, gracias a este se pudieron registrar los datos obtenidos desde la plataforma desarrollada con Arduino. Gracias a esto se pudieron llevar a cabo tres pruebas de concepto con la finalidad de implementar el sistema completo (plataforma desarrollada con Arduino y el sistema de información desarrollado) y verificar que la idea puede ser desarrollada y/o explotada de manera más detallada en un futuro.

Como es de conocimiento, el objetivo principal de este proyecto fue crear un prototipo de sistema que pueda garantizar las condiciones fitosanitarias de los alimentos a la hora de ser transportados, debido a que hoy en día no existe tal herramienta que pueda certificar de manera rápida y eficiente las condiciones e inocuidad de los alimentos. Tal prototipo fue desarrollado paso a paso mediante el cumplimiento de los objetivos específicos anteriormente mencionados.

A través de una solución que involucra a la nanociencia que poco a poco va tomando más fuerza en los últimos años, serían muchos los beneficios que traería un sistema como el desarrollado a la industria. Este cambio no está solamente en la etapa del transporte, como lo es en este caso, sino que en la agroindustria en general, donde se han creado embalajes con componentes especiales para conservar los alimentos; nanopartículas o biosensores que ayudarían a detectar la degradación o residuos de plaguicidas en los alimentos; entre otras nano soluciones que se han ido estudiando.

## Capítulo 8: Bibliografía

- [1] Royal Society, The (2004) *Nanoscience and Nanotechnologies: Opportunities and Uncertainties*. Reino Unido.
- [2] Delgado, G. C. (2008) *Guerra por lo invisible: negocio, implicaciones y riesgos de la nanotecnología*. México, Ceich.
- [3] Ojeda, G. A., Arias Gorman, A. M., & Sgroppo, S. C. (2019). La nanotecnología y su aplicación en alimentos. *Mundo Nano Revista Interdisciplinaria en Nanociencia y Nanotecnología*, 12(23).
- Chellaram, C., Murugaboopathi, G., John, A. A., Sivakumar, R., Ganesan, S., Krithika, S., & Priya, G. (2014). Significance of nanotechnology in food industry. *APCBEE procedia*, 8, 109-113.
- [4] Abbas KA, Saleh, A.M, Mohamed, A, MohdAzhan N. The recent advances in the nanotechnology and its applications in food processing: A review- Journal of Food, Agriculture & Environment, 2009; 7 (3&4), 14 - 17 .
- [5] Valdés, F., & Areny, R. P. (2007). *Microcontroladores fundamentos y aplicaciones con PIC* (Vol. 1149). Marcombo.
- [6] *Plataformas de desarrollo electrónico*. (s/f). Smelpro.com. Recuperado el 19 de agosto de 2021, de <https://smelpro.com/blog/plataformas-de-desarrollo-electronico/>
- [7] Vargas González, I. D., & Álvarez Villamizar, L. C. Sistema de información Web para la administracion y solicitud de actores por catálogo desde acceso móvil para Mainevent Producciones.
- [8] Luján-Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Editorial Club Universitario.
- [9] Lezoche, M., Hernandez, J. E., Díaz, M. D. M. E. A., Panetto, H., & Kacprzyk, J. (2020). Agri-food 4.0: a survey of the supply chains and technologies for the future agriculture. *Computers in Industry*, 117, 103187.

- [10] Mostafavi, H. A., Mirmajlessi, S. M., & Fathollahi, H. (2012). The potential of food irradiation: benefits and limitations. *Trends in vital Food and control engineering*, 5, 43-68.
- [11] Disdier, A. C., Fontagné, L., & Mimouni, M. (2008). The impact of regulations on agricultural trade: evidence from the SPS and TBT agreements. *American Journal of Agricultural Economics*, 90(2), 336-350.
- [12] Henson, S., y Humphrey, J. (2010). Comprender las complejidades de las normas privadas en las cadenas agroalimentarias mundiales a medida que afectan a los países en desarrollo. *The journal of development studies*, 46(9), 1628-1646.
- [13] Fulponi, L. (2007). The globalization of private standards and the agri-food system. *Global supply chains, standards and the poor: How the Globalization of Food Systems and Standards Affects Rural Development and Poverty*, 5-18.
- [14] Akkerman, R., Farahani, P., & Grunow, M. (2010). Quality, safety and sustainability in food distribution: a review of quantitative operations management approaches and challenges. *Or Spectrum*, 32(4), 863-904.
- [15] Trienekens, J., & Zuurbier, P. (2008). Quality and safety standards in the food industry, developments and challenges. *International journal of production economics*, 113(1), 107-122.
- [16] Sharp, A. K. (1988). Air freight of perishable produce.
- [17] James, S. J., James, C., & Evans, J. A. (2006). Modelling of food transportation systems—a review. *International Journal of Refrigeration*, 29(6), 947-957
- [18] Ackerley, N., Sertkaya, A., & Lange, R. (2010). Food transportation safety: characterizing risks and controls by use of expert. *Food protection trends*, 30(4), 212-222.

- [19] Figueroa, R. G., Solís, C. J., & Cabrera, A. A. (2008). Metodologías tradicionales vs. metodologías ágiles. Universidad Técnica Particular de Loja, Escuela de Ciencias de la Computación, 9.
- [20] Awad, M. A. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia*, 30.
- [21] Pelta Resano, R. (2013). Design Thinking. Tendencias en la teoría y la metodología del diseño, septiembre 2013.
- [22] Brown, T., & Wyatt, J. (2010). Design thinking for social innovation. *Development Outreach*, 12(1), 29-43.
- [23] Aguayo, P. (2019, January 22). Inicio. Retrieved May 10, 2021, from Arduino.cl website: <https://arduino.cl/>
- [24] Arduino - Home. (n.d.). Retrieved May 10, 2021, from Arduino.cc website: <https://www.arduino.cc/>
- Cobo, J. G. (2018, July 3). Las pantallas LCD y Arduino. Retrieved May 10, 2021, from Hardware libre website: <https://www.hwlibre.com/pantallas-lcd-arduino/>
- Fernández, Y. (2018, July 21). Qué es Arduino, cómo funciona y qué puedes hacer con uno. Retrieved May 10, 2021, from Xataka.com website: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- Getting Started with Arduino UNO. (n.d.). Retrieved May 10, 2021, from Arduino.cc website: <https://www.arduino.cc/en/Guide/ArduinoUno>
- ijorquera. (2014, November 20). ¿Qué es Arduino? Retrieved May 10, 2021, from Arduino.cl website: <https://arduino.cl/que-es-arduino/>
- Ramírez, L. G. C., Jiménez, G. S. A., & Carreño, J. M. (2014). Sensores y Actuadores. Azcapotzalco, Mexico: Grupo Editorial Patria.

Software. (n.d.). Retrieved May 10, 2021, from Arduino.cc website:  
<https://www.arduino.cc/en/software>

Vidal-Silva, C., Lineros, M. I., Uribe, G. E., & Olmos, C. J. (2019). Electrónica para Todos con el Uso de Arduino: Experiencias Positivas en la Implementación de Soluciones Hardware-Software. *CIT Informacion Tecnologica*, 30(6), 377–386

Gonen, E. (2019). Tim brown, change by design: How design thinking transforms organizations and inspires innovation (2009). *Markets Globalization & Development Review*, 04(02). doi:10.23860/mgdr-2019-04-02-08

## Capítulo 9: Anexos

```
#include <Wire.h>
#include "Adafruit_SHT31.h"
#include <LiquidCrystal_I2C.h>

Adafruit_SHT31 sht31 = Adafruit_SHT31();
LiquidCrystal_I2C lcd(0x27,20,4);

void setup() {

  lcd.init();

  lcd.backlight();

  Serial.begin(9600);
  if (! sht31.begin(0x44))
  {
    Serial.println("Couldn't find SHT31");
    while (1) delay(1);
  }
}
float t_maxi =0;
float t_mini=999;
float h_maxi =0;
float h_mini=999;
```

Figura 14. Vista del primer código del Sistema Arduino - Fuente (creación propia- 2021)

```

void loop()
{
  float t = sht31.readTemperature();
  float h = sht31.readHumidity();

  if(t>t_maxi){
    t_maxi=t;
  }

  if(t<t_mini){
    t_mini=t;
  }

  if(h>h_maxi){
    h_maxi=h;
  }

  if(h<h_mini){
    h_mini=h;
  }

  lcd.setCursor(3,0);
  lcd.print("Temp");|
  lcd.setCursor(0,1);
  lcd.print ("Act:");
  lcd.print(t,1);
  lcd.print(" C");
}

```

Figura 15. Vista del primer código del Sistema Arduino - Fuente (creación propia- 2021)

```

  lcd.print(t,1);
  lcd.print(" C");|
  lcd.setCursor(0,2);
  lcd.print ("Max:");
  lcd.print(t_maxi,1);
  lcd.print(" C");
  lcd.setCursor(0,3);
  lcd.print ("Min:");
  lcd.print(t_mini,1);
  lcd.print(" C");
  lcd.setCursor(14,0);
  lcd.print("Hum");
  lcd.setCursor(11,1);
  lcd.print ("Act:");
  lcd.print(h,1);
  lcd.print("%");
}

```

Figura 16. Vista del primer código del Sistema Arduino - Fuente (creación propia- 2021)

```

# -- coding: utf-8 --

from odoo import models, fields, api

#Modelo de la clase de control

class Control(models.Model):
    _name = 'seguimiento.control'

    producto= fields.Char(string="Nombre del producto", required=True)
    fecha= fields.Date(string = "Fecha de llegada")
    temp_max= fields.Float(String="Temperatura máxima")
    temp_min= fields.Float(String="Temperatura mínima")
    hum_max= fields.Float(String="Humedad máxima %")
    hum_min= fields.Float(String="Humedad mínima %")

```

Figura 17. Vista de la clase de control - Fuente (creación propia - 2021)

```

<odoo>
  <data>
    <!-- Listado -->
    <record model="ir.ui.view" id="seguimiento.control_list">
      <field name="name">Seguimiento</field>
      <field name="model">seguimiento.control</field>
      <field name="arch" type="xml">
        <tree>
          <field name="producto"/>
          <field name="fecha"/>
          <field name="temp_max"/>
          <field name="temp_min"/>
          <field name="hum_max"/>
          <field name="hum_min"/>
        </tree>
      </field>
    </record>

```

Figura 18. Código del Archivo XML correspondiente vista tipo Lista de la clase Control - Fuente (creación propia- 2021)

```

<!-- Formulario -->
<record id="seguimiento.control_view" model="ir.ui.view">
  <field name="name">Seguimiento</field>
  <field name="model">seguimiento.control</field>
  <field name="arch" type="xml">
    <form string="Seguimiento">
      <sheet>
        <Separator string="Control de producto"/>
        <group>
          <field name="producto"/>
          <field name="fecha"/>
          <field name="temp_max"/>
          <field name="temp_min"/>
          <field name="hum_max"/>
          <field name="hum_min"/>
        </group>
      </sheet>
    </form>
  </field>
</record>

```

Figura 19. Código del Archivo XML correspondiente vista tipo Formulario de la clase Control - Fuente (creación propia- 2021)

```

<record model="ir.actions.act_window" id="seguimiento.control_action_window">
  <field name="name">Seguimiento</field>
  <field name="res_model">seguimiento.control</field>
  <field name="view_mode">tree,form</field>
</record>

<record model="ir.actions.server" id="seguimiento.control_action_server">
  <field name="name">Seguimiento server</field>
  <field name="model_id" ref="model_seguimiento_control"/>
  <field name="state">code</field>
  <field name="code">
    <action = {
      "type": "ir.actions.act_window",
      "view_mode": "tree,form",
      "res_model": self._name,
    }
  </field>
</record>

<menuitem name="Seguimiento" id="seguimiento.menu_root"/>
<menuitem name="Seguimientos" id="seguimiento.menu_control" parent="seguimiento.menu_root"
  action="seguimiento.control_action_window"/>

</data>
/odoo>

```

Figura 20. Código del Archivo XML correspondiente vista tipo Formulario de la clase Control parte final - Fuente (creación propia- 2021)

```

# -*- coding: utf-8 -*-

from odoo import models, fields, api

#Modelo de los productos

class Producto(models.Model):
    _name = 'inventario.producto'

    name = fields.Char(string="Nombre", required=True)
    date_elaboration = fields.Date(string = "Fecha de elaboración")
    date_expiration = fields.Date(string = "Fecha de vencimiento")
    categorias_id = fields.Many2one('inventario.categoria_producto', string="Categoría")

class CategoriaProducto(models.Model):
    _name = 'inventario.categoria_producto'
    name = fields.Char(string="Nombre", required=True)
    producto_ids = fields.One2many(
        'inventario.producto',
        'categorias_id',
        string = 'Productos')

    total_productos = fields.Integer(string = "Total Productos", compute = "_total_productos")

@api.one
def _total_productos(self):
    self.total_productos = len(self.producto_ids)

```

Figura 21. Vista del código de la clase de inventario- Fuente (creación propia - 2021)

```

<odoo>
  <data>
    <!-- <Listado de productos> -->
    <record model="ir.ui.view" id="inventario.producto_list">
      <field name="name">Productos</field>
      <field name="model">inventario.producto</field>
      <field name="arch" type="xml">
        <tree>
          <field name="name"/>
          <field name="date_elaboration"/>
          <field name="date_expiration"/>
          <field name="categorias_id"/>
        </tree>
      </field>
    </record>
  </data>
</odoo>

```

Figura 22. Código del Archivo XML correspondiente vista tipo Listado de productos de la clase Control - Fuente (creación propia- 2021)

```

<!-- <Form> -->
<record id="Inventario" model="ir.ui.view">
  <field name="name">Productos</field>
  <field name="model">inventario.producto</field>
  <field name="arch" type="xml">
    <form string="Productos">
      <sheet>
        <group col = "4">
          <field name="name" />
        </group>
        <group>
          <field name="date_elaboration" />
          <field name="date_expiration" />
          <field name="categorias_id"/>
        </group>
      </sheet>
    </form>
  </field>
</record>

```

Figura 23. Código del Archivo XML correspondiente vista tipo Formulario de la clase Control corregido- Fuente (creación propia- 2021)

```

<!-- <Acciones de categorias> -->
<record model="ir.actions.act_window" id="inventario.categoria_producto_action_window">
  <field name="name">Categorias</field>
  <field name="res_model">inventario.categoria_producto</field>
  <field name="view_mode">tree,form</field>
</record>
<record model="ir.actions.server" id="inventario.producto_action_server">
  <field name="name">categorias server</field>
  <field name="model_id" ref="model_inventario_categoria_producto"/>
  <field name="state">code</field>
  <field name="code">
    action = {
      "type": "ir.actions.act_window",
      "view_mode": "tree,form",
      "res_model": self._name,
    }
  </field>
</record>

```

*Figura 24. Código del Archivo XML correspondiente vista tipo Acciones de Categorías de la clase Control - Fuente (creación propia- 2021)*

```

<!-- <Listado de categorías> -->
<record model="ir.ui.view" id="inventario.categoria_producto_list">
  <field name="name">Categorías</field>
  <field name="model">inventario.categoria_producto</field>
  <field name="arch" type="xml">
    <tree>
      <field name="name"/>
      <field name="total_productos"/>
    </tree>
  </field>
</record>

<record model="ir.actions.act_window" id="inventario.producto_action_window">
  <field name="name">Productos</field>
  <field name="res_model">inventario.producto</field>
  <field name="view_mode">tree,form</field>
</record>

<record model="ir.actions.server" id="inventario.producto_action_server">
  <field name="name">producto server</field>
  <field name="model_id" ref="model_inventario_producto"/>
  <field name="state">code</field>
  <field name="code">
    action = {
      "type": "ir.actions.act_window",
      "view_mode": "tree,form",
      "res_model": self._name,
    }
  </field>
</record>

```

Figura 25. Código del Archivo XML correspondiente vista tipo Listado de Categorías de la clase Control - Fuente (creación propia- 2021)

```
<!-- Menú de la barra superior -->
<menuitem name="Inventario" id="inventario.menu_root"/>
<!-- Categorías del menú -->
<menuitem name="Productos" id="inventario.menu_productos" parent="inventario.menu_root"
|   action="inventario.producto_action_window"/>
<menuitem name="Categorías" id="inventario.menu_categorias" parent="inventario.menu_root"
|   action="inventario.categoria_producto_action_window"/>

</data>
</odoo>
```

Figura 26. Código del Archivo XML correspondiente al menú de la clase Control - Fuente (creación propia- 2021)

```
# -*- coding: utf-8 -*-

from odoo import models, fields, api

#Modelo de la clase de control

class Control(models.Model):
    _name = 'seguimiento.control'

    cliente= fields.Char(string="Cliente", required=True)
    fecha= fields.Date(string="Fecha de transporte")

    detalle_control_ids= fields.One2many(
        'seguimiento.detalle_control', 'control_id', string="Detalle del seguimiento")

class DetalleControl(models.Model):
    _name= 'seguimiento.detalle_control'

    producto_id= fields.Many2one('inventario.producto', String="Producto")
    cantidad = fields.Integer(default=1)
    temp_max= fields.Float(String="Temperatura máxima")
    temp_min= fields.Float(String="Temperatura mínima")
    hum_max= fields.Float(String="Humedad máxima")
    hum_min= fields.Float(String="Humedad mínima")
    control_id= fields.Many2one('seguimiento.control', String="Control")
```

Figura 27. Código del archivo PY de la vista de la clase de control corregida - Fuente (creación propia - 2021)

```

<odoo>
  <data>
    <!-- Listado -->
    <record model="ir.ui.view" id="seguimiento.control_list">
      <field name="name">Seguimiento</field>
      <field name="model">seguimiento.control</field>
      <field name="arch" type="xml">
        <tree>
          <field name="cliente"/>
          <field name="fecha"/>
        </tree>
      </field>
    </record>
  </data>
</odoo>

```

Figura 28. Código del Archivo XML correspondiente vista tipo Listado de productos de la clase Control corregido - Fuente (creación propia- 2021)

```

<!-- Formulario -->
<record id="seguimiento.control_view" model="ir.ui.view">
  <field name="name">Seguimiento</field>
  <field name="model">seguimiento.control</field>
  <field name="arch" type="xml">
    <form string="Seguimiento">
      <sheet>
        <Separator string="Control de producto"/>
        <group>
          <field name="cliente"/>
          <field name="fecha"/>
        </group>
      </sheet>
    </form>
  </field>
</record>

```

Figura 29. Código del archivo XML correspondiente a la vista tipo Formulario de la clase de control corregida - Fuente (creación propia - 2021)

```
<!-- Lista de detalle de venta-->
<record model="ir.ui.view" id="seguimiento.detalle_control_list">
  <field name="name">Detalle de seguimiento</field>
  <field name="model">seguimiento.detalle_control</field>
  <field name="arch" type="xml">
    <tree>
      <field name="producto_id"/>
      <field name="cantidad"/>
      <field name="temp_max"/>
      <field name="temp_min"/>
      <field name="hum_max"/>
      <field name="hum_min"/>
    </tree>
  </field>
</record>
```

Figura 30. Código del archivo XML correspondiente a la vista tipo Lista de detalle de venta la clase de control corregida - Fuente (creación propia - 2021)

```
<!-- Formulario de detalle de venta -->
<record id="seguimiento.control_view" model="ir.ui.view">
  <field name="name">Detalle de seguimiento</field>
  <field name="model">seguimiento.detalle_control</field>
  <field name="arch" type="xml">
    <form string= "Seguimiento">
      <sheet>
        <Separator string= "Control de producto"/>
        <group>
          <field name="producto_id"/>
          <field name="cantidad"/>
          <field name="temp_max"/>
          <field name="temp_min"/>
          <field name="hum_max"/>
          <field name="hum_min"/>
        </group>
      </sheet>
    </form>
  </field>
</record>
```

Figura 31. Código del archivo XML correspondiente a la vista tipo Formulario de detalle de venta de la clase de control corregida - Fuente (creación propia - 2021)

```

class Control(models.Model):
    _name = 'seguimiento.control'

    cliente= fields.Char(string="Cliente", required=True)
    area= fields.Char(string="Area de transporte", required=True)
    fecha= fields.Date(string="Fecha de transporte")
    temp_max= fields.Float(String="Temperatura máxima")
    temp_min= fields.Float(String="Temperatura mínima")
    hum_max= fields.Float(String="Humedad máxima")
    hum_min= fields.Float(String="Humedad mínima")

    detalle_control_ids= fields.One2many(
        'seguimiento.detalle_control', 'control_id', string="Detalle del seguimiento")

```

*Figura 32. Vista de la clase Control corregida - Fuente (creación propia- 2021)*

```

class DetalleControl(models.Model):
    _name= 'seguimiento.detalle_control'

    producto_id= fields.Many2one('inventario.producto', String="Producto")
    cantidad = fields.Integer(default=1)
    control_id= fields.Many2one('seguimiento.control', String="Control")

```

*Figura 33. Vista de la clase DetalleControl - Fuente (creación propia- 2021)*

```

<!-- Listado -->
<record model="ir.ui.view" id="seguimiento.control_list">
  <field name="name">Seguimiento</field>
  <field name="model">seguimiento.control</field>
  <field name="arch" type="xml">
    <tree>
      <field name="cliente"/>
      <field name="area"/>
      <field name="fecha"/>
      <field name="temp_max"/>
      <field name="temp_min"/>
      <field name="hum_max"/>
      <field name="hum_min"/>
    </tree>
  </field>
</record>

```

*Figura 34. Código del Archivo XML correspondiente vista tipo Lista de la clase Control - Fuente (creación propia- 2021)*

```

<!-- Formulario -->
<record id="seguimiento.control_view" model="ir.ui.view">
  <field name="name">Seguimiento</field>
  <field name="model">seguimiento.control</field>
  <field name="arch" type="xml">
    <form string= "Seguimiento">
      <sheet>
        <Separator string= "Control de producto"/>
        <group>
          <field name="cliente"/>
          <field name="area"/>
          <field name="fecha"/>
          <field name="temp_max"/>
          <field name="temp_min"/>
          <field name="hum_max"/>
          <field name="hum_min"/>
        </group>
      </sheet>
    </form>
  </field>
</record>

```

*Figura 35. Código del Archivo XML correspondiente vista tipo Formulario de la clase Control - Fuente (creación propia- 2021)*

```

<!-- Lista de detalle de seguimiento-->
<record model="ir.ui.view" id="seguimiento.detalle_control_list">
  <field name="name">Detalle de seguimiento</field>
  <field name="model">seguimiento.detalle_control</field>
  <field name="arch" type="xml">
    <tree>
      <field name="producto_id"/>
      <field name="cantidad"/>
    </tree>
  </field>
</record>

```

*Figura 36. Código del Archivo XML correspondiente vista tipo Lista de la clase DetalleControl - Fuente (creación propia- 2021)*

```

<!-- Formulario de detalle de seguimiento -->
<record id="seguimiento.control_view" model="ir.ui.view">
  <field name="name">Detalle de seguimiento</field>
  <field name="model">seguimiento.detalle_control</field>
  <field name="arch" type="xml">
    <form string="Seguimiento">
      <sheet>
        <Separator string="Control de producto"/>
        <group>
          <field name="producto_id"/>
          <field name="cantidad"/>
        </group>
      </sheet>
    </form>
  </field>
</record>

```

Figura 37. Código del Archivo XML correspondiente vista tipo Formulario de la clase DetalleControl - Fuente (creación propia- 2021)

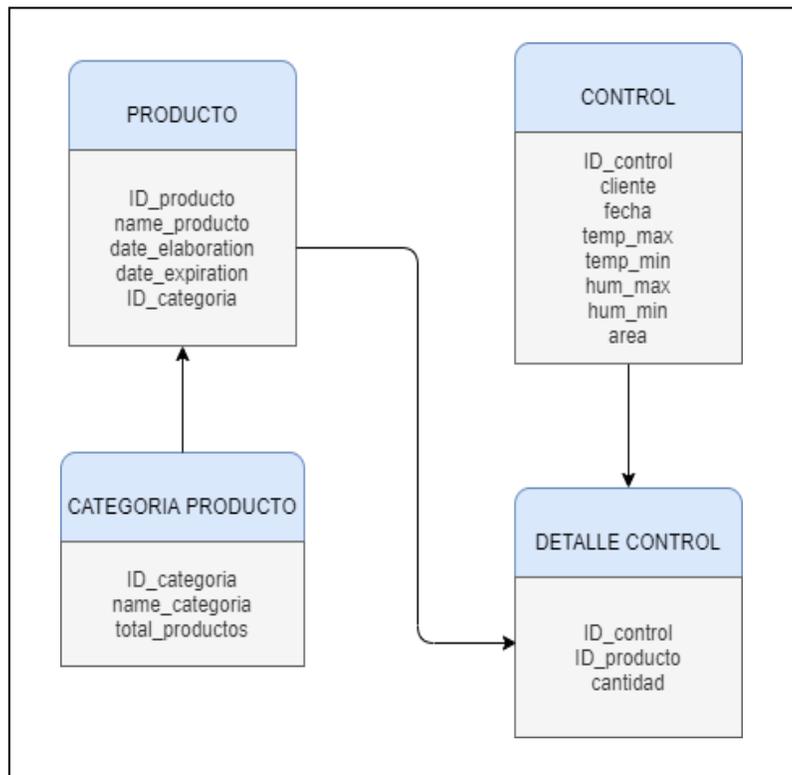


Figura 38. Modelo Relacional del sistema de información – Fuente (creación propia- 2021)