



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E**  
**INDUSTRIAL**  
**CARRERA DE INGENIERÍA EN SISTEMAS**  
**COMPUTACIONALES E INFORMÁTICOS**

**Tema:**

---

**DESARROLLO DE UN SISTEMA DE MONITOREO DE TEMPERATURA**  
**Y HUMEDAD PARA CÁMARAS DE SECADO DE PASTAS EN LA**  
**EMPRESA FIDEOS VICTORIA**

---

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

**ÁREA:** Software

**LÍNEA DE INVESTIGACIÓN:** Desarrollo de Software

**AUTOR:** Carlos Israel Tixilema Merchan

**TUTOR:** Ing. David Omar Guevara Aulestia Mg.

Ambato - Ecuador

agosto - 2021

## APROBACIÓN DEL TUTOR

En calidad de Tutor del Trabajo de Titulación con el tema: DESARROLLO DE UN SISTEMA DE MONITOREO DE TEMPERATURA Y HUMEDAD PARA CÁMARAS DE SECADO DE PASTAS EN LA EMPRESA FIDEOS VICTORIA, desarrollado bajo la modalidad Proyecto de Investigación por el señor Carlos Israel Tixilema Merchan, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, agosto 2021

---

Ing. David Omar Guevara Aulestia Mg.

TUTOR

## AUTORÍA DEL TRABAJO

El presente Proyecto de Investigación titulado: DESARROLLO DE UN SISTEMA DE MONITOREO DE TEMPERATURA Y HUMEDAD PARA CÁMARAS DE SECADO DE PASTAS EN LA EMPRESA FIDEOS VICTORIA es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2021



---

Carlos Israel Tixilema Merchan

C.C. 1804582755

Autor

## APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Carlos Israel Tixilema Merchan, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, DESARROLLO DE UN SISTEMA DE MONITOREO DE TEMPERATURA Y HUMEDAD PARA CÁMARAS DE SECADO DE PASTAS EN LA EMPRESA FIDEOS VICTORIA, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, agosto 2021

---

Ing. Pilar Urrutia Mg.

PRESIDENTA DEL TRIBUNAL

---

Ing. Hernando Buenaño Mg.  
PROFESOR CALIFICADOR

---

Ing. Andrea Sánchez  
PROFESOR CALIFICADOR



## DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, agosto 2021



---

Carlos Israel Tixilema Merchan

C.C. 1804582755

Autor

## DEDICATORIA

A lo largo de este camino he tenido la ayuda de seres maravillosos que de una u otra manera contribuyeron para que este día llegara, para todos ellos va dedicado este trabajo haciendo un énfasis a mi familia sin la cuál nada de esto hubiera sido posible.

Gracias.

«En este mundo hay cosas que sólo puedes hacer sola y cosas que sólo puedes hacer con otra persona. Es importante ir combinando las unas con las otras» After Dark, Haruki Murakami

Carlos Israel Tixilema Merchan

## AGRADECIMIENTO

Quiero agradecer a la vida con todas sus imperfecciones por la oportunidad de haber nacido en esta época y lugar.

Gracias a mis padres por haber inculcado en mi los valores con los que han logrado superar las dificultades que se han presentado a lo largo de sus vidas, a mis hermanos por su compañía en buenos y malos momentos también al resto de mi familia, sin olvidarme de mis mascotas que han formado parte de la misma.

Gracias a mis amigos por su compañía y alivianar el peso de este viaje.

Finalmente agradezco a los maestros que me han infundido el conocimiento necesario para atravesar el largo camino hasta mi formación profesional, en especial a mi tutor Ing. David Guevara por su ayuda en este proyecto.

Carlos Israel Tixilema Merchan

## ÍNDICE

<b>APROBACIÓN DEL TUTOR</b>	<b>ii</b>
<b>AUTORÍA</b>	<b>iii</b>
<b>APROBACIÓN COMISIÓN CALIFICADORA</b>	<b>iv</b>
<b>DERECHOS DE AUTOR</b>	<b>v</b>
<b>Dedicatoria</b>	<b>vi</b>
<b>Agradecimiento</b>	<b>vii</b>
<b>INTRODUCCIÓN</b>	<b>xvii</b>
<b>CAPÍTULO I MARCO TEÓRICO</b>	<b>1</b>
1.1 Tema de Investigación . . . . .	1
1.2 Antecedentes Investigativos . . . . .	1
1.2.1 Contextualización del problema . . . . .	2
1.2.2 Fundamentación teórica . . . . .	4
1.2.2.1 Dashboard. . . . .	4
1.2.2.2 IOT . . . . .	5
1.2.2.3 Dominios del IOT. . . . .	5
1.2.2.4 Aplicaciones Web. . . . .	6
1.2.2.5 Análisis de metodologías de desarrollo de software	7
1.2.2.6 Análisis de tecnologías de desarrollo web . . . . .	10
1.3 Objetivos . . . . .	16
1.3.1 Objetivo General . . . . .	16
1.3.2 Objetivos Específicos . . . . .	16
<b>CAPÍTULO II METODOLOGÍA</b>	<b>17</b>
2.1 Materiales . . . . .	17
2.1.1 Humanos . . . . .	17



2.1.2	Institucionales . . . . .	17
2.1.3	Otros . . . . .	17
2.1.4	Económicos . . . . .	18
2.2	Métodos . . . . .	18
2.2.1	Modalidad de la Investigación . . . . .	18
2.2.2	Población y Muestra . . . . .	19
2.2.3	Recolección de Información . . . . .	19
2.2.4	Metodología seleccionada . . . . .	19
2.2.5	Lenguajes de programación utilizados . . . . .	20
<b>CAPÍTULO III RESULTADOS Y DISCUSIÓN</b>		<b>21</b>
3.1	Análisis y discusión de resultados . . . . .	21
3.1.1	Análisis del sistema de secado empleado en la empresa Fideos Victoria. . . . .	21
3.1.2	Descripción de la plataforma utilizada para la adquisición de temperatura y humedad . . . . .	22
3.1.2.1	Programación del dispositivo . . . . .	25
3.1.3	Estructura base del sistema de monitoreo . . . . .	26
3.1.4	Desarrollo de la metodología . . . . .	31
3.1.4.1	Roles del proyecto . . . . .	31
3.1.4.2	Planificación del Proyecto . . . . .	31
3.1.4.3	Diseño . . . . .	42
3.1.4.4	Codificación . . . . .	46
3.1.4.5	Pruebas . . . . .	76
<b>CAPÍTULO IV CONCLUSIONES Y RECOMENDACIONES</b>		<b>90</b>
4.1	Conclusiones . . . . .	90
4.2	Recomendaciones . . . . .	91
<b>Bibliografía</b>		<b>92</b>
<b>ANEXOS</b>		<b>96</b>

## ÍNDICE DE TABLAS

1.1	Diferencias entre metodologías tradicionales y ágiles . . . . .	8
2.1	Recursos Económicos . . . . .	18
2.2	Recolección de Información . . . . .	19
3.1	Comparación placas de desarrollo . . . . .	22
3.2	Comparación sensores de temperatura/humedad . . . . .	23
3.3	Comparación de lenguajes de programación . . . . .	25
3.4	Roles XP del proyecto . . . . .	31
3.5	Historia de usuario: Ingreso al sistema . . . . .	32
3.6	Historia de usuario: Registro de Usuarios . . . . .	32
3.7	Historia de usuario: Gestión de Cámaras . . . . .	32
3.8	Historia de usuario: Gestión de Productos . . . . .	33
3.9	Historia de usuario: Gestión de Sensores . . . . .	33
3.10	Historia de usuario: Gestión de Producciones . . . . .	34
3.11	Historia de usuario: Eliminación de Producciones . . . . .	34
3.12	Historia de usuario: Registro de Temperatura . . . . .	34
3.13	Historia de usuario: Visualización de la Temperatura . . . . .	35
3.14	Historia de usuario: Reportes de temperaturas . . . . .	35
3.15	Historia de usuario: Exportación de datos . . . . .	35
3.16	Diseño de interfaz de ingreso . . . . .	36
3.17	Diseño de interfaz de registro . . . . .	36
3.18	Implementación de confirmación vía email . . . . .	36
3.19	Implementación de manejo de roles de usuario . . . . .	37
3.20	Diseño de interfaz para la gestión de cámaras . . . . .	37
3.21	Diseño de interfaz para la gestión de productos . . . . .	37
3.22	Diseño de interfaz para la gestión de sensores . . . . .	38
3.23	Diseño de interfaz para la gestión de producciones . . . . .	38
3.24	Diseño Web API para el registro de temperatura . . . . .	38
3.25	Implementación de lógica de los dispositivos IOT . . . . .	39
3.26	Diseño de un dashboard . . . . .	39

3.27	Diseño de un medio para comunicación en tiempo real . . . . .	39
3.28	Diseño de reportes . . . . .	40
3.29	Exportación de datos para análisis . . . . .	40
3.30	Plan de entrega . . . . .	41
3.31	Tarjeta CRC - TemperaturaController . . . . .	42
3.32	Tarjeta CRC - AuthController . . . . .	42
3.33	Tarjeta CRC - CamaraController . . . . .	43
3.34	Tarjeta CRC - ProduccionController . . . . .	43
3.35	Tarjeta CRC - ProductoController . . . . .	43
3.36	Tarjeta CRC - RolesSeed . . . . .	43
3.37	Tarjeta CRC - EmailSender . . . . .	44
3.38	Tarjeta CRC - NavigationIndicatorHelper . . . . .	44
3.39	Tarjeta CRC - SistemaHub . . . . .	44
3.40	Iteración 1 . . . . .	57
3.41	Iteración 2 . . . . .	59
3.42	Iteración 3 . . . . .	62
3.43	Iteración 4 . . . . .	71
3.44	Casos de pruebas unitarias . . . . .	77
3.45	Prueba de aceptación: ingreso de credenciales . . . . .	79
3.46	Prueba de aceptación: registro por parte del usuario . . . . .	79
3.47	Prueba de aceptación: creación de cámaras . . . . .	79
3.48	Prueba de aceptación: edición de cámaras . . . . .	80
3.49	Prueba de aceptación: eliminación de cámaras . . . . .	80
3.50	Prueba de aceptación: detalles por cámara . . . . .	81
3.51	Prueba de aceptación: creación de productos . . . . .	81
3.52	Prueba de aceptación: edición de productos . . . . .	82
3.53	Prueba de aceptación: eliminación de productos . . . . .	82
3.54	Prueba de aceptación: detalles por producto . . . . .	83
3.55	Prueba de aceptación: creación de sensores . . . . .	83
3.56	Prueba de aceptación: edición de sensores . . . . .	84
3.57	Prueba de aceptación: eliminación de sensores . . . . .	84
3.58	Prueba de aceptación: detalles por sensor . . . . .	85
3.59	Prueba de aceptación: creación de producciones . . . . .	85
3.60	Prueba de aceptación: edición de producciones . . . . .	86
3.61	Prueba de aceptación: detalles por producción . . . . .	86
3.62	Prueba de aceptación: eliminación de producciones . . . . .	87
3.63	Prueba de aceptación: envío de temperaturas del dispositivo IOT . . . . .	87

3.64	Prueba de aceptación: visualización de datos en tiempo real . . .	88
3.65	Prueba de aceptación: visualización de datos por producción . . .	88
3.66	Prueba de aceptación: exportación de datos por producción . . .	89
6.1	45Comparación entre datos capturados . . . . .	116
6.2	Comparación de tiempos de secado . . . . .	117



## ÍNDICE DE FIGURAS

1.1	Participación de empresas por provincia. . . . .	3
1.2	Estructura Asp.Net Core . . . . .	13
1.3	Web frameworks Stackoverflow . . . . .	15
3.1	Proceso de secado . . . . .	21
3.2	Códigos de colores del dispositivo de captura . . . . .	24
3.3	Conexiones del dispositivo . . . . .	25
3.4	Algoritmo para toma de datos . . . . .	26
3.5	Estructura del proyecto . . . . .	27
3.6	Información dotnet . . . . .	28
3.7	Configuración Nginx . . . . .	30
3.8	Servicio del sistema . . . . .	30
3.9	Diagrama de Base de Datos(1) . . . . .	45
3.10	Diagrama de Base de Datos(2) . . . . .	46
3.11	Clases base para el manejo de la base de datos . . . . .	48
3.12	Archivos generados por Identity . . . . .	51
3.13	Políticas de acceso . . . . .	52
3.14	Configuración JWT . . . . .	52
3.15	Archivos AdminLTE . . . . .	54
3.16	Layout AdminLTE . . . . .	55
3.17	Ingreso al sistema . . . . .	57
3.18	Registro de usuarios . . . . .	58
3.19	Manejo de roles . . . . .	59
3.20	Listado de Cámaras . . . . .	60
3.21	Listado de Productos . . . . .	60
3.22	Listado de Sensores . . . . .	61
3.23	Eliminación de producciones . . . . .	61
3.24	Listado de Producciones . . . . .	63
3.25	Método pushProducto . . . . .	63
3.26	Creación de producciones . . . . .	64
3.27	Búsqueda de la producción activa . . . . .	65

3.28	Notificación vía SignalR . . . . .	66
3.29	Modificación de la hora en TemperaturaController . . . . .	66
3.30	Conexión a wifi NodeMCU . . . . .	67
3.31	Clase configuración . . . . .	67
3.32	Función asíncrona Micropython . . . . .	68
3.33	Dashboard(Producción inactiva) . . . . .	69
3.34	Dashboard . . . . .	70
3.35	Configuración cliente SignalR . . . . .	70
3.36	Búsqueda de producción activa . . . . .	71
3.37	Reporte de temperaturas . . . . .	73
3.38	Búsqueda de máximos y mínimos por sensor . . . . .	74
3.39	Exportación de datos . . . . .	75
3.40	Datos exportados . . . . .	76
3.41	Comprobación de pruebas unitarias . . . . .	78
6.1	Gráfica producción #56 . . . . .	117
6.2	Gráfica producción #76 . . . . .	118
6.3	Gráfica producción #82 . . . . .	118

## RESUMEN EJECUTIVO

El sistema de secado de pastas empleado en la empresa Fideos Victoria se basa en el aumento de temperatura y la disminución de la humedad presente en la cámara y por consiguiente del producto, siendo de vital importancia el conocer si la temperatura ha alcanzado el mínimo necesario para que el producto se encuentre en condiciones para su venta. Tomando como base la experiencia adquirida por el operario encargado del control del proceso, se evidenciaron algunas falencias en cuanto al hecho de realizar el monitoreo de las cámaras de secado de forma presencial, entre ellas el no contar con un registro detallado del proceso.

El presente proyecto se ha enfocado en mejorar el proceso de monitoreo de las cámaras de secado por parte del operario, por medio de una herramienta tecnológica que hace uso de técnicas para la visualización de datos provenientes de dichas cámaras, tanto en tiempo real como de datos almacenados.

Como parte de este trabajo se desarrolló una aplicación web en Asp Net Core 3.1 con el fin de almacenar y presentar de forma visual los datos correspondientes al sistema de monitoreo, además se hizo uso de Micropython para la incorporación de la lógica de captura y envío de datos por parte de los dispositivos de tipo NodeMCU instalados en cada cámara de secado.

Asp Net Core facilitó el desarrollo en el lado servidor por medio de Entity Framework, Identity y SignalR para el manejo de la base de datos, de usuarios y para la comunicación en tiempo real respectivamente, como en el lado cliente por medio del patrón Razor Pages con la generación de las páginas en las cuales se integró la interfaz de usuario siendo esta adaptable para diferentes tamaños de pantalla.

El desarrollo se realizó por medio de la metodología XP, permitiendo la entrega de avances paulatinos generando una mejor recepción y adopción del sistema por parte del cliente.

**Palabras clave:** Aplicaciones web, asp net core, signalr, razor pages, monitoreo

## ABSTRACT

The pasta drying system used in the company Fideos Victoria is based on increasing the temperature and decreasing the humidity in the drying chamber and therefore the product, being of vital importance to know if the temperature has reached the minimum necessary to ensure that the product is ready for sale. Based on the experience acquired by the operator in responsible for controlling the process, some shortcomings were found in monitoring the drying chambers on site, including the lack of a detailed record of the process.

This project focuses on facilitating temperature and humidity monitoring of the drying chambers by the operator as well as the storage of the data captured by the devices for later analysis. As part of this work, a web application was developed in Asp Net Core 3.1 and Micropython was used to incorporate the logic for capturing and sending data from the NodeMCU devices installed in each drying chamber. Asp Net Core facilitated the development both server-side through the Entity Framework for database management, Identity for user management and SignalR for real-time communication, and client-side through the Razor Pages pattern for page generation in which the adaptable user interface for different screen sizes was integrated.

The development was done using the XP methodology, allowing the delivery of gradual advances generating a better system reception and adoption by the client.

**Keywords:** Web apps, asp net core, signalr, razor pages, monitoring



## INTRODUCCIÓN

En los tiempos actuales en los que cada aspecto se encuentra regido por la tecnología, desde el monitoreo del sueño por medio de relojes inteligentes hasta los grandes sistemas que monitorean el clima alrededor del mundo, la cadena de producción no podía quedarse atrás y requerir del monitoreo de las variables que afectan sus procesos.

Partiendo con la premisa anteriormente mencionada se ha planteado el desarrollo de un sistema de monitoreo de temperatura y humedad para las cámaras de secado de pastas que posee la empresa Fideos Victoria, con el fin de ayudar al operario encargado de controlar el proceso de secado, además del almacenamiento de datos proporcionados por los sensores.

Para la recolección de datos se utilizarán dispositivos IOT en conjunto con sensores de temperatura y humedad relativa encargados del envío de dichos datos a un equipo que tendrá por función el exponer un servidor web con la capacidad para recibir datos y presentarlos de forma legible por medio de un dashboard.

El presente proyecto de investigación denominado DESARROLLO DE UN SISTEMA DE MONITOREO DE TEMPERATURA Y HUMEDAD PARA CÁMARAS DE SECADO DE PASTAS EN LA EMPRESA "FIDEOS VICTORIA" se encuentra dividido en los capítulos expuestos a continuación.

### CAPÍTULO I MARCO TEÓRICO.

En este capítulo se plantea el tema de investigación, sus antecedentes y los objetivos que determinarán el curso del desarrollo del proyecto

### CAPÍTULO II METODOLOGÍA.

En este capítulo se reúnen los materiales y métodos que se utilizarán en el desarrollo del proyecto.

### CAPÍTULO III RESULTADOS Y DISCUSIÓN.

En este capítulo se describe en detalle el desarrollo del proyecto tomando como base la metodología elegida.

### CAPÍTULO IV CONCLUSIONES Y RECOMENDACIONES.

En este capítulo se detallan las conclusiones y se ofrecen recomendaciones en aspectos hallados a lo largo del desarrollo del proyecto.

# CAPÍTULO I

## MARCO TEÓRICO

### 1.1. Tema de Investigación

DESARROLLO DE UN SISTEMA DE MONITOREO DE TEMPERATURA Y HUMEDAD PARA CÁMARAS DE SECADO DE PASTAS EN LA EMPRESA "FIDEOS VICTORIA"

### 1.2. Antecedentes Investigativos

La presente investigación se basa en los antecedentes investigativos detallados a continuación siendo el resultado de la investigación bibliográfica en una serie de repositorios.

De acuerdo con la tesis con el título “Implementación de un sistema de sensores, monitoreo y alertas de la temperatura y humedad de un centro de datos.” realizada por Marín Santana Carlos Julio en la Universidad de Guayaquil, evidencia que es más eficiente monitorear el ambiente de manera automática a que un operario lo realice cada cierto tiempo de forma presencial, concordando con la solución propuesta a la problemática actual de la empresa Fideos Victoria [1].

Continuando con la investigación, en el artículo titulado “Sistema de monitoreo automático de temperatura de bajo costo con alertas para unidades de cría de laboratorio” se recalca la importancia del monitoreo de la temperatura, ya que si ésta varía en unos pocos grados se evidencian problemas como el crecimiento y fecundidad de los insectos [2]. Para solventar dicha problemática el investigador creó un sistema de monitoreo haciendo uso de un Raspberry Pi, este artículo se consideró relevante ya que en el presente proyecto se utiliza un enfoque parecido en el cual la temperatura es la variable con mayor relevancia.

Al investigar en el repositorio de la Universidad de Cuenca se encontró una tesis con el título “Optimización del proceso de secado en pastas alimenticias” la cual hace referencia a que si durante el secado de pastas se presenta una temperatura excesiva en la pasta se provoca la aparición de marrón debido a un pardeamiento no enzimático extremo [3], además del problema mencionado anteriormente, la temperatura de secado influye directamente en la calidad de la pasta ya que si se aplican temperaturas altas al inicio del proceso se produce una desnaturalización

del gluten y la gelatinización del almidón [4].

### 1.2.1. Contextualización del problema

En la actualidad, el mundo se encuentra regido por la tecnología y en la base de cualquier avance tecnológico se encuentra el desarrollo de software. Una de las aplicaciones en las que el desarrollo de software se encuentra presente es el monitoreo de datos, el cual se fundamenta en la recolección, análisis y procesamiento de dichos datos para convertirlos en información útil pudiendo tener fines tan variados como los casos detallados a continuación.

DigiDhan Dashboard es un proyecto desarrollado en la India, que en el 2018 obtuvo el premio Technology Sabha en el segmento “Analytics/Big Data” en el cual han tomado datos sobre pagos, compras y tarjetas de crédito de instituciones financieras locales y extranjeras alimentando continuamente una base de datos con más 2 terabytes de información derivando en el monitoreo del crecimiento de pagos digitales y el desempeño de las instituciones asociadas por parte de los ministerios [5]. En China, por medio del trabajo de investigación titulado “Design and Realization of Covers Monitoring and Controlling System”(Diseño y realización de un sistema de supervisión y control de cubiertas) se manifiesta que haciendo uso del lenguaje C# se diseñó un sistema de monitoreo que captura información en tiempo real sobre el estado de las cubiertas de tuberías en las que se encuentran servicios públicos como agua o gas para almacenarla y en caso de detectarse una situación anormal generar alarmas [6].

En el Ecuador, uno de los tantos sistemas de monitoreo existentes, es el sistema de monitoreo de radiación UV (Ultravioleta) implementado por el Instituto Nacional de Meteorología e Hidrología «INAMHI» [hidrolog-a], el cual se encarga de mediante estaciones meteorológicas ubicadas en puntos estratégicos del país, mediante sensores monitorear la radiación UV y generar alertas en caso de que estos sean perjudiciales para la población.

En la ciudad de Ambato existe un sistema de monitoreo de la calidad del aire ubicado en la terraza del edificio del edificio Matriz GADMA (Gobierno Autónomo Descentralizado Municipio de Ambato) Sur, este sistema se encarga de tomar muestras de aire cada segundo para así comprobar la concentración de compuestos como el monóxido de carbono [7]. Este proyecto se da por una recomendación de la Organización Panamericana de la Salud, la cual estipula que en poblaciones de menos de un millón de habitantes debe existir al menos una estación de este tipo.

La pasta al ser un producto derivado del trigo ofrece los nutrientes que el posee,

siendo utilizada principalmente la variedad candeal que es mayormente producido en «Canadá y EE. UU. que respectivamente concentran un 45 % y 10 % de la participación del mercado mundial»[8].

Con la creciente demanda de alimento por parte la población, han surgido empresas de todo ámbito en cuanto a la generación de este, siendo la pasta un producto de consumo masivo.

En Argentina, por ejemplo, la elaboración de pastas es el segundo uso que se le da a la harina de trigo únicamente superada por la elaboración de pan industrial, así mismo con respecto a los cereales más consumidos las pastas secas se encuentran en una segunda posición junto con el pan envasado y únicamente superado por el pan de panadería [9].

En Ecuador, en el año 2019 se registran 23 empresas dedicadas a la elaboración de pastas generando 1985 plazas de trabajo y con una participación del 4.5 % del PIB (Producto interno bruto) del país, además, según previsiones del Banco Central del Ecuador para el año 2021 el sector crecería un 3 % [10].

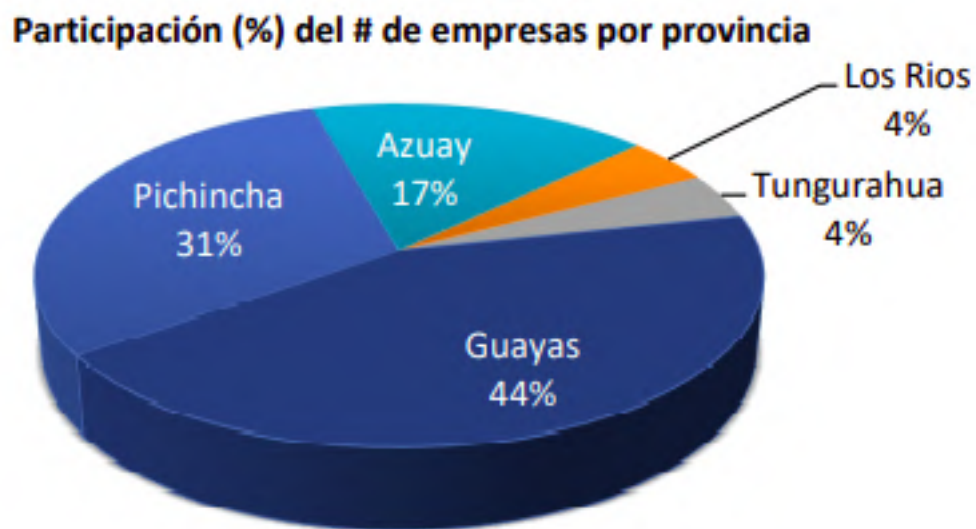


Figura 1.1: Participación de empresas por provincia.

Fuente: Corporación Financiera Nacional

En base a los datos expuestos anteriormente, se puede vislumbrar la importancia que tiene el consumo de pastas en cuanto a la alimentación y economía. Por lo tanto, para que este alimento sea apto para el consumo, debe cumplir con ciertos estándares detallados en la norma INEN 1375(Servicio Ecuatoriano de Normalización) en la cual se especifican los factores a tomar en cuenta en la producción de pastas, entre ellos los ingredientes a utilizarse y la humedad



porcentual del producto final [11], dicha humedad para alcanzar los valores establecidos debe completar un proceso de secado el cual debe ser constantemente monitoreado brindando así la oportunidad para que puedan surgir sistemas encargados de dicha tarea como es el caso del presente proyecto.

## **1.2.2. Fundamentación teórica**

### **1.2.2.1. Dashboard.**

Un dashboard es un panel de datos que consta de gráficos e indicadores interactivos interconectados, los cuales proveen una visión sobre las relaciones entre las variables presentes[12], generalmente no se requiere ninguna experticia para interpretar los datos presentados.

El conocimiento es un proceso determinista que permite tomar decisiones, pudiendo este ser transmitido de manera no verbal. La información visual puede ser una herramienta útil en la exploración de datos y la formulación de hipótesis, ya que al asociarla con la deducción matemática pueden extraerse patrones, aun en ambientes en los que la información cambia dinámicamente, siendo de gran utilidad la representación con gráficos estándar como son los gráficos de pastel o de barras, por mencionar algunos.

Los dashboards nacen a partir de la necesidad de una herramienta que cumpla con la función de presentar información de una forma accesible o darle forma a la información presente, generalmente están dirigidos a directores tecnológicos los cuales los utilizan para planear, diseñar, implementar y dirigir aplicaciones en tiempo real, así como para monitorear la información y tomar acciones en cuanto a la información presentada.

Dependiendo de su uso, los dashboards pueden ser clasificados como BI (Business Intelligence, inteligencia de negocios) dashboards, kpi(key performance indicator, indicador clave de desempeño) dashboards, website(sitios web) dashboards, open source(código libre) dashboards, free(gratis) dashboards, marketing dashboards, executive (ejecutivos) dashboards y real-time(en tiempo real) dashboards.

#### **Técnicas de representación.**

**Nube de palabras:** Es útil cuando se requiere visualizar palabras utilizadas en diferentes temas y su tamaño va acorde al número de veces que se repite dicha palabra.

**Diagramas de red:** Es usado principalmente para representar enlaces entre varias categorías, así como el flujo de información. Representa la información en forma de nodos y lazos siendo estos puntos y líneas.

**Matrices correlacionales:** Sirve para mostrar cuan fuertes son las relaciones entre las variables, esto se da por medio de una matriz en la que de acuerdo al valor de las celdas usarán una escala de colores para una mejor comprensión.

**Diagramas de Sankey:** Especialmente útil para identificar patrones de flujo que fallan en completar una acción designada al mostrar una serie de nodos enlazados donde el ancho del enlace indica su frecuencia o el valor de la medida.

#### 1.2.2.2. IOT

El IOT (Internet of Things - Internet de las cosas) es un paradigma emergente propuesto inicialmente por Kevin Ashton en 1990[13] como un concepto básico sobre la convergencia y enlace entre objetos físicos y virtuales para la creación de un mundo en el cual dichos objetos puedan atravesar fronteras sin importar el lugar, el tiempo y quien o que lo utilice.

La principal característica del IOT es el identificar, localizar, rastrear y manipular objetos para explotar los datos capturados por medio de técnicas de modelado y análisis. Esta tecnología consta de las cosas (objetos, vehículos y/o gente) que tienen equipados sensores, medios de conexión (radiofrecuencia, wifi, cableado), actuadores y procesadores.

En la actualidad coexisten millones de dispositivos conectados a internet que generan una cantidad abrumadora de datos los cuales deben ser tratados por algoritmos inteligentes, una de las soluciones para su tratamiento consiste en utilizar la potencia de la computación en la nube para obtener conocimiento relevante y útil. El auge de esta tecnología se ha dado por varias razones entre las cuales se podrían mencionar la reducción del costo, tamaño y consumo de dispositivos electrónicos, al igual que al desarrollo de un gran número de sensores y actuadores automáticos, sin restar importancia a la mejora en los medios de comunicación tanto cableadas como inalámbricas como es el caso de las redes móviles y de radiofrecuencia.

#### 1.2.2.3. Dominios del IOT.

El IOT comprende un sin número de aplicaciones, siendo algunas de las más importantes las siguientes:

**Localización y rastreo de información:** por medio de localización la información puede ser rastreada en tiempo real ayudando al seguimiento de objetos importantes.

**Manejo de vehículos:** haciendo uso de una serie de sensores interconectados y algoritmos complejos un vehículo puede ser conducido, pero no es lo único en lo

que a IOT respecta ya que también se encuentra inmerso en el mantenimiento, manejo de combustible y seguridad por mencionar algunos aspectos.

**Manejo de logística:** no solo mejora el flujo de materiales, también ayuda en el posicionamiento global y la identificación automática de stock. Se espera que el IOT sea el propulsor de cambios profundos en la cadena de suministro al encargarse del rastreo en tiempo real de objetos.

**Monitoreo de tráfico:** forma parte de una ciudad inteligente ya que producen un control y manejo eficiente del tráfico usando sensores e información en red.

**Monitoreo de ambientes:** los sistemas IOT ayudan a monitorear condiciones como la temperatura, humedad, presión, ruido, radiación y polución, siendo clave para solucionar algunos retos globales como la provisión de energía, alimentación y limpieza generando ambientes libres de contaminación e incluso ayudando a la creación de una agricultura inteligente y la conservación del agua por medio del monitoreo inteligente del agua y el suelo.

**Monitoreo remoto:** en casas inteligentes, el IOT ayuda a la gente a tener control y monitorear dispositivos mediante comandos enviados a través de aplicaciones, estos pueden aplicarse en la detección de emergencias como robos o mediante un control de desastres como un incendio o un desastre natural fomentando así la prevención.

#### **Sistemas de recolección de datos.**

Los sistemas de recolección de datos constan de 4 etapas[14] relacionadas de la siguiente manera: «En la etapa transductora, el transductor o sensor, siendo sensible a esta variable, la transforma en una señal eléctrica que luego ingresa a la etapa de acondicionamiento de señal. En esta etapa la señal es amplificada o filtrada en frecuencias, para luego ser llevada al conversor análogo-digital donde es acoplada al sistema digital. Este sistema, que puede ser un microcontrolador como los presentes en las placas Arduino, puede realizar algún procesamiento de señal o solo enviarla al computador donde se realiza toda la etapa de procesamiento numérico correspondiente a cada experimento particular.» Siendo usados para el almacenamiento de las variables medidas para así poder usar dichos datos en distintas formas como tomar medidas en caso de que dichas variables afecten el normal comportamiento de un escenario.

#### **1.2.2.4. Aplicaciones Web.**

Las aplicaciones web son programas que están almacenados en un servidor, por medio del uso de tecnologías web (Javascript, Html y Css) y navegadores. Existe cierta confusión al momento de diferenciar un sitio web de una aplicación web,

pero la cualidad que los diferencia es que si esta realiza una tarea, por pequeña que sea, debe ser tratada como una aplicación web[15].

Hoy en día resulta difícil concebir la idea de una empresa que no haga uso de aplicaciones basadas en tecnologías web, ya que estas poseen las siguientes ventajas con respecto a las aplicaciones de escritorio:

- Son accesibles por red local o externa vía internet.
- No requieren instalación en el cliente.
- Accesibles por medio de navegadores web.
- El rendimiento esta limitado por el servidor, en su mayor parte.
- Puede adaptarse visualmente a distintos dispositivos, haciendo uso de un diseño responsivo.

#### **1.2.2.5. Análisis de metodologías de desarrollo de software**

Para alcanzar un objetivo se requiere utilizar las habilidades y conocimientos relacionados con dicho objetivo. Una metodología provee de los procedimientos base por medio de los cuales se puede alcanzar el objetivo planteado de forma ordenada y formalizada.

##### **Metodología tradicional**

Las metodologías tradicionales imponen una forma de trabajo haciendo énfasis en la planificación total del proyecto antes de su ejecución. Al crear una documentación extensa sobre todas las fases del proyecto en la fase inicial del desarrollo se consigue un producto eficiente. Estas metodologías no son adecuadas para proyectos en los que existan cambios impredecibles o requieran una alta flexibilidad en sus requerimientos además de poseer un alto costo de implementación.[16]

##### **Metodología ágil**

Las metodologías ágiles nacen para dar solución a los problemas generados por el uso de metodologías tradicionales ya que poseen la flexibilidad para tratar los proyectos como «subproyectos» más pequeños y fáciles de manejar, además al incluir una comunicación constante con los clientes puede reaccionar y adaptarse adecuadamente a los cambios.



## Diferencias entre metodologías tradicionales y ágiles

Tabla 1.1: Diferencias entre metodologías tradicionales y ágiles

Metodología tradicional	Metodología ágil
Documentación extensa	Documentación corta
Planificación en la fase inicial	Planificación en iteraciones
Un solo lanzamiento final	Lanzamiento de versiones preliminares
El cliente participa en la fase de toma de requerimientos	El cliente participa activamente a lo largo del proyecto

Fuente: Elaborado por el investigador

Tomando en cuenta las características presentadas en la tabla 1.1 como la extensión de la documentación además de la participación activa del cliente se consideró que las metodologías ágiles encajan con en el presente proyecto.

### Metodología XP (Extreme Programming - Programación extrema)

Esta metodología fue desarrollada por Kent Beck en la búsqueda por guiar equipos de trabajo pequeños o medianos, entre dos y diez programadores, en ambientes con requerimientos imprecisos o cambiantes [17].

Una característica de esta metodología es el uso de historias de usuario, las cuales son una especie de ficha técnica en la que se especifican los requisitos, características y funcionalidades, del sistema.

#### Proceso.

Xp se basa en iteraciones, las cuales constan de los siguientes pasos:

- **Planeación:** Se crean las historias de usuario con el cliente en base a la definición de requisitos.
- **Diseño:** Se diseña la solución en base a la historia de usuario correspondiente.
- **Codificación:** Se realiza la codificación validando las características solicitadas por medio de buenas practicas como es la programación en pares.
- **Pruebas:** Por medio de pruebas unitarias se asegura la calidad del software.
- **Lanzamiento:** Suma las nuevas características al software que se presenta al cliente.

## Ventajas

- Posee una buena capacidad de respuesta en entornos volátiles.
- Buena aceptación del cliente, al conocer de antemano las historias de usuario.
- La realimentación de parte de los usuarios brinda información muy útil.
- Permite realizar modificaciones rápida y fácilmente.

## Desventajas

- No es aconsejable para el desarrollo masivo de software.
- No es recomendable en proyectos a largo plazo.
- Genera muchas versiones del mismo software.

## Metodología Scrum

Esta metodología tiene como objetivo el maximizar el retorno de inversión para la empresa que la utiliza. Se basa en la construcción de la funcionalidad de mayor valor para el cliente además de la inspección continua, adaptación, autogestión e innovación, el equipo que colabora en la elaboración del producto al ser multidisciplinario genera un alto valor desde sus primeras versiones.

Los equipos Scrum trabajan por medio de sprints, ciclos de ejecución cortos, los cuales entregan productos con valor de forma rápida y continua. Para la organización del equipo Scrum se basa en los siguientes roles:

- **Product Owner:** Persona que sirve de interlocutor entre los stakeholders del proyecto.
- **Scrum Master:** Es el responsable que la metodología sea comprendida y aplicada correctamente, su trabajo principal es ayudar a la adopción de la metodología.
- **Equipo de Desarrollo:** Son aquellas personas encargadas de completar tareas, siendo principalmente multifuncional y autoorganizado.
- **Stakeholders:** Son personas relacionadas a las actividades y decisiones de la empresa, a pesar de no tener un rol formal su opinión debe ser tomada en cuenta.

## **Proceso Scrum**

El eje central de la metodología Scrum son los sprints con sus respectivas repeticiones, siendo los siguientes pasos a seguir para su aplicación:

1. Planificación
2. Sprint diario
3. Revisión del sprint
4. Retrospectiva del sprint

## **Ventajas**

- Buen manejo de comunicación
- Puede ejecutar proyectos de gran tamaño
- Por medio de los sprints se puede incluir cambios y ajustes fácilmente

## **Desventajas**

- Los cambios son añadidos al finalizar el sprint
- Adopción compleja por la determinación de roles
- Consumo de tiempo en reuniones que pueden alargarse demasiado

### **1.2.2.6. Análisis de tecnologías de desarrollo web**

El desarrollo web consta, generalmente, de dos ámbitos bien diferenciados entre sí conocidos como «BackEnd» y «FrontEnd» los cuales hacen referencia directamente al servidor y los clientes respectivamente.

#### **Patrones de diseño**

El enfoque tradicional de desarrollo en la que se crea una interfaz de usuario para posteriormente crear la lógica de programación genera problemas ya que da como resultado una fuerte dependencia entre la interfaz de usuario, los datos y la lógica de negocio, haciendo que el añadir o modificar funcionalidades se convierta en un dolor de cabeza para el desarrollador.

Una solución a los problemas que se generan al usar un desarrollo tradicional sin una estructura lista, es el uso de patrones de diseño ya que estos presentan diferentes enfoques para la organización y comunicación entre la interfaz grafica y las diferentes capas que propone cada patrón.

Los patrones de diseño tienen como objetivos principales:[18]

- Proporcionar catálogos de componentes reutilizables para el diseño de sistemas de software
- Estandarizar una terminología común entre diseñadores
- Estandarizar la manera en que se realiza el diseño
- Facilitar el aprendizaje de futuros diseñadores integrando un conocimiento ya existente

### **MVC (Model View Controller, Modelo Vista Controlador)**

Este patrón se basa en separar los componentes principales como la manipulación de datos (Modelo), interfaz (Vista) y el proceso (Controlador) para que el proyecto sea más ordenado, estructurado y fácil de desarrollar [19].

Debido a la capacidad de separar el proyecto en base a su funcionalidad, MVC puede escalar fácilmente sin perder funcionalidad e incluso pudiendo separar el trabajo en base a los módulos a desarrollar.

### **MVVM (Model View View-Model, Modelo Vista Modelo de Vista)**

El patrón MVVM ayuda a separar la lógica de la interfaz de usuario al dividir el proyecto en tres partes principales, Model, View-Model y View.

Model o modelo representa el modelo de datos y la lógica de negocios, siendo el lugar en el cual se alojan los atributos y clases base del proyecto.

View-Model o Modelo de Vista contiene la lógica de presentación que se transfiere a la vista además las clases alojadas en el View-Model contienen funciones y parámetros encargados de realizar la comunicación entre los modelos y las vistas.

View o Vista consta de la interfaz gráfica y tiene la función de presentar la información recibida del View-Model

### **Razor Pages**

Este patrón se define como un modelo basado en páginas, construido internamente en base a la arquitectura MVC, siendo ambas técnicas totalmente compatibles y extensibles.

Al estar basado en MVC, Razor Pages proporciona características similares pero con su sintaxis propia siendo esta algo más limpia. La principal diferencia que posee Razor Pages a MVC es que el primero posee un directorio llamado Pages, en el cual se encuentran las paginas que serán mostradas, así como sus respectivas clases auxiliares evitando tener que crear un controlador para cada página. Esta aproximación tiene algunas similitudes al patrón MVVM (Model View View-Model, Modelo Vista Modelo de Vista), utilizado ampliamente en el desarrollo de aplicaciones móviles, ya que la clase asociada a la pagina podría ser considerada una View-Model el cual evita totalmente del uso de controladores.

Ambas aproximaciones, MVC y Razor Pages, al ser compatibles la una con la otra, se procedió a utilizar lo mejor de cada una para así crear un proyecto ordenado y escalable con la facilidad de representar paginas además del uso de controladores para crear las Web API's que comunican a los sensores con la lógica interna del sistema.

### **Tecnología BackEnd**

En los tiempos actuales la forma más común para el desarrollo de aplicaciones se basa en servicios web que interactúan con diferentes plataformas y lenguajes.

Una de las características más importantes sobre un servicio web es la capacidad para manejar múltiples usuarios de forma eficiente, dicha eficiencia viene dada por diversos factores como son el hardware, el sistema operativo, la infraestructura o el framework en el que se desarrolla. Tomando como base el ultimo punto se procedió a realizar una comparación entre algunas alternativas para el desarrollo del presente proyecto.

#### **Laravel**

Laravel es un framework de código abierto basado en el lenguaje PHP descrito en su pagina como el «framework PHP para artesanos web» por la facilidad al momento de crear código sencillo y con una sintaxis elegante.

Una de sus características es el uso del patrón MVC(Modelo Vista Controlador), el cual facilita la estructuración del sistema y la delegación de responsabilidades entre sus componentes, además posee las siguientes características de alto nivel:

- ORM(Object Relational Mapping o Mapeo de Objeto Relacional) integrado completamente
- Sistema para el manejo de rutas
- Gestión de sesiones
- Sistema de autenticación

#### **Asp Net Core**

Asp Net Core es la plataforma de desarrollo web de Microsoft, la versión original fue liberada en 2002 la cual ha sufrido una serie de cambios, entre los que constan que Asp Net Core fuera hecho de cero, con una licencia abierta y con la capacidad de correr en distintas plataformas como es Linux o MacOS suponiendo un gran avance al eliminar la dependencia del sistema operativo Windows y sus costosas licencias.

Asp Net Core consiste en una plataforma que procesa llamadas HTTP, no esta limitada a esta pero es la más común, en una serie de frameworks para crear

aplicaciones así como en utilidades internas que proveen características extra como muestra la figura 1.1.

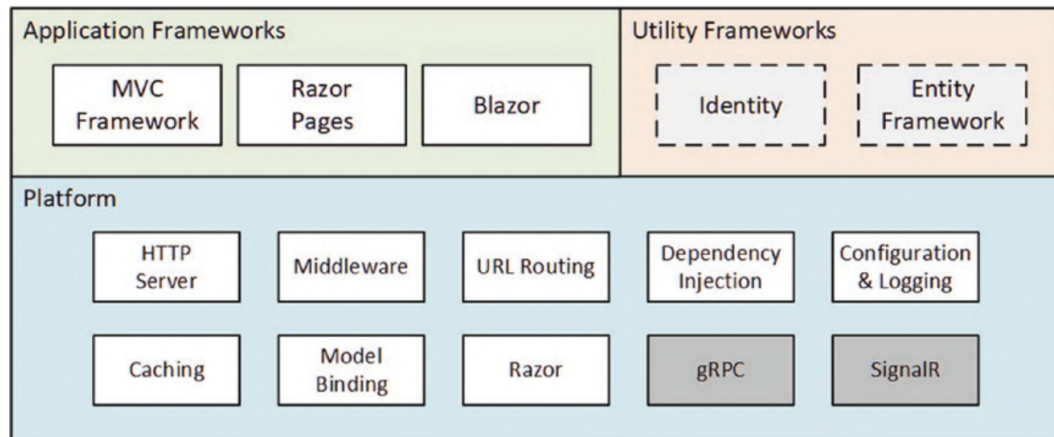


Figura 1.2: Estructura Asp.Net Core  
Fuente: Elaborado por el investigador

### **Entity Framework Core(EF Core)**

Esta es una tecnología ORM desarrollada por Microsoft para el manejo de datos en distintos motores de bases de datos. En el presente proyecto se ha usado el enfoque Code-First, que como su nombre lo indica primero se genera el código C# con sus relaciones para que después Entity Framework se encargue de generar la estructura en la base de datos elegida, en este caso MariaDb. Algunas de las ventajas que posee Entity Framework con relación a otros ORM es el apoyo de la comunidad de Microsoft y al ser de código abierto brinda la posibilidad de crear sus propias versiones con funciones extendidas, como es el caso del Nuget(Gestor de paquetes para el marco .Net) Pomelo.EntityFrameworkCore.Mysql el cual se encarga de generar las conexiones con la base de datos y asignar correctamente los tipos de datos utilizados en los modelos.

### **Asp Net Core Identity**

Asp Net Core Identity es una API de Microsoft usada para el manejo de usuarios en aplicaciones de Asp Net. Con Identity se pueden crear autorizaciones así como autenticación de usuarios por medio de credenciales, con doble factor de autenticación por mencionar los más comunes, pudiendo ser usado tanto en pequeños proyectos como en grandes ambientes corporativos.

## **SignalR**

SignalR es una librería creada por Microsoft para realizar comunicación en tiempo real de tipo cliente-servidor [20], sin limitarse al entorno Web ya que posee clientes para Javascript, .Net, Java, C++ y Swift.

En el presente proyecto, SignalR es usada para mostrar gráficos en tiempo real, usando el cliente Javascript, basados en los datos capturados por los sensores que estén en funcionamiento en dicho instante, brindando una experiencia detallada y que permite saber si el sistema de secado se encuentra en funcionamiento.

## **Tecnología «FrontEnd»**

Por sus capacidades, así como las del investigador, se han utilizado las siguientes tecnologías:

### **Vanilla Javascript**

Se denomina «Vanilla Javascript» al uso de Javascript sin ningún framework o librería, este es especialmente útil al combinarse con Razor Pages, ya que proporciona funcionalidad adicional al código generado.

### **Jquery**

Es una biblioteca de Javascript que simplifica el manejo del DOM(Document Object Model, Modelo de objetos del documento), es un requisito para trabajar con Bootstrap además de ser utilizada por el generador de código de Razor Pages para las validaciones en los campos que lo requieran.

### **Alpine.js**

Framework de tipo reactivo y de código libre inspirado en la sintaxis de Vue. Cuenta con un tamaño muy reducido, 6.4 kb en su versión reducida, y el poder de ser inicializado en un archivo común de Javascript o incluso en una etiqueta script de una pagina HTML, lo cual lo convierte en una forma rápida de crear funcionalidad extra que se combina fácilmente con tecnologías de tipo renderizado en el servidor como es el caso de Razor Pages.

### **Bootstrap**

Es un conjunto de herramientas para el diseño de aplicaciones web, por medio de sus plantillas y componentes brinda la posibilidad de que la aplicación adquiera la capacidad de adaptarse a distintos dispositivos, sin la necesidad de crear funciones Javascript propias o código CSS.

### **AdminLTE**

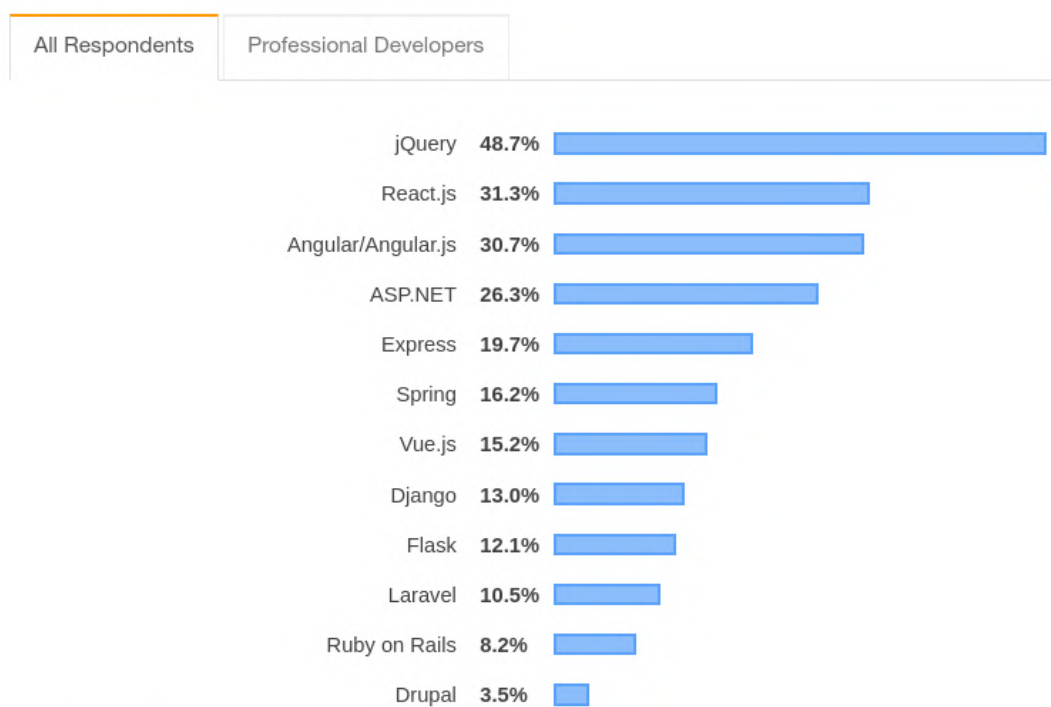
Plantilla de código libre creada en base a HTML5 y CSS3, contiene una serie de componentes responsivos y reutilizables, además de ser compatible con diversas librerías de Javascript como Chart.js.



## Comparación Web Frameworks

En el conocido foro StackOverflow, año a año realizan encuestas sobre distintos ámbitos del desarrollo de software. En la encuesta realizada en el año 2020 sobre los frameworks más utilizados se observa que ASP NET se encuentra en la cuarta posición solo superado por JQuery, React y Angular, los cuales son tecnología FrontEnd evidenciando la acogida que posee el Framework de Microsoft superando a tecnologías de renombre como Laravel(PHP) o Django(Python).

### Web Frameworks



63,585 responses; select all that apply

This year, we asked about frameworks for the web separately from other frameworks and libraries. jQuery is the most broadly used of these web frameworks, and this year more developers say they use React.js than Angular, a switch from last year.

Figura 1.3: Web frameworks Stackoverflow  
Fuente: Stackoverflow

### **1.3. Objetivos**

#### **1.3.1. Objetivo General**

Implementar un sistema de monitoreo de temperatura y humedad para cámaras de secado de pastas en la empresa Fideos Victoria.

#### **1.3.2. Objetivos Específicos**

- Describir el proceso de secado de pastas utilizado en la empresa Fideos Victoria.
- Determinar la plataforma a utilizar para la adquisición de temperatura y humedad.
- Analizar los datos obtenidos mediante los sensores de temperatura y humedad de las cámaras de secado de la empresa Fideos Victoria.
- Desarrollar un dashboard para el monitoreo de temperatura y humedad.

## **CAPÍTULO II**

### **METODOLOGÍA**

#### **2.1. Materiales**

En el presente proyecto de investigación se realizó una entrevista al encargado de los controles de automatización de las cámaras y al gerente/propietario para su correspondiente análisis.

##### **2.1.1. Humanos**

- Docente Tutor.
- Investigador.
- Encargado de los controles de las cámaras de secado de la empresa Fideos Victoria.

##### **2.1.2. Institucionales**

- Facultad de Ingeniería en Sistemas Electrónica e Industrial.
- Empresa Fideos Victoria.
- Biblioteca Virtual de la Universidad Técnica de Ambato.

##### **2.1.3. Otros**

- Suministros de oficina.
- Internet.
- Libros.
- Dispositivos de almacenamiento.
- Laptop.

#### 2.1.4. Económicos

Tabla 2.1: Recursos Económicos

N	Detalle	Unidad	Cantidad	Valor Unitario	Valor Total
1	Internet	horas	220	0.75	165.00
2	Gastos del proyecto	c/u	1	300.00	300.00
3	Resma de papel bond A4	c/u	1	5.00	5.00
4	Anillado	c/u	2	6.00	12.00
5	Carpeta	c/u	1	1.00	1.00
6	Lápiz	c/u	2	0.80	1.60
7	Copia	c/u	400	0.02	8.00
8	Impresión	c/u	150	0.05	7.50
9	Borrador	c/u	1	0.45	0.45
10	Laptop	c/u	1	800	800.00
				Subtotal	1300.55
				Imprevistos (10%)	130.06
				Total	1430.61

Fuente: Elaborado por el investigador

## 2.2. Métodos

### 2.2.1. Modalidad de la Investigación

Las modalidades de investigación utilizadas fueron, investigación bibliográfica y de campo.

#### **Investigación Bibliográfica**

Se considera Investigación Bibliográfica porque utiliza fuentes tales como libros, artículos, documentos entre otros para la construcción del marco teórico y el monitoreo de temperatura y humedad en distintos ámbitos, para así comprender el valor de un sistema de monitoreo.

#### **Investigación de Campo**

Se considera Investigación de Campo porque busca registrar los valores tomados por los sensores instalados en las cámaras de la empresa para así implementar el sistema de monitoreo y mejorar la perspectiva con respecto al comportamiento en general de las cámaras de secado. Ver anexo E.

### 2.2.2. Población y Muestra

Por la naturaleza de la investigación no se requiere de una investigación a una población objetivo, razón por la cual se aplicará una entrevista al encargado de los controles de automatización de las cámaras, así como al gerente/propietario ya que por su estructura organizacional ellos son los encargados del manejo del sistema de secado. Dichas entrevistas constan en los anexos 1 y 2.

Las entrevistas fueron formuladas de manera que se pueda evidenciar las falencias de la forma en la que se maneja actualmente el sistema de secado bajo la perspectiva de quienes se encuentran en contacto con el mismo.

### 2.2.3. Recolección de Información

Tabla 2.2: Recolección de Información

Preguntas Básicas	Explicación
1.¿Para qué?	Monitorear dos de las variables que más influyen en el proceso de secado de pastas
2.¿De qué personas u objetos?	Empresa Fideos Victoria, área de secado
3.¿Sobre qué aspectos?	El sistema de monitoreo de temperatura y humedad
4.¿Quién?	Investigador: Carlos Israel Tixilema Merchan
5.¿Cuándo?	La presente investigación se desarrollará en el periodo abril - septiembre 2020
6.¿Dónde?	Empresa Fideos Victoria
7.¿Cuántas veces?	Las veces que sea necesario
8.¿Qué técnicas de recolección?	Entrevista
9.¿Con que?	Observación de campo, bitácoras con información de las cámaras de secado
10.¿En qué situación?	En el proceso de secado de pastas

Fuente: Elaborado por el investigador

### 2.2.4. Metodología seleccionada

En respuesta al cambio constante que opera en el mundo empresarial, así como las oportunidades del mercado y teniendo en cuenta que el software participa en muchas de las operaciones empresariales, las soluciones informáticas deben ser desarrolladas de una forma tal que puedan entregar una respuesta acorde a dicho cambio pero sin perder calidad.

Tomando en cuenta las características presentadas en la tabla 1.1 como la extensión de la documentación además de la participación activa del cliente se

consideró el uso de una metodología ágil siendo finalmente elegida la metodología XP.

### 2.2.5. Lenguajes de programación utilizados

Para este proyecto se utilizaron tres lenguajes de programación diferentes, siendo estos:

1. **C#:** usado en el «BackEnd» con el framework Asp Net Core 3.1, alojando las entidades base así como las Web API (Application Programming Interface o Interfaz de Programación de Aplicaciones) y las «Razor Pages», paginas generadas en base a entidades, como el «Dashboard» que presenta la información capturada por los sensores o las interfaces de las entidades base.
2. **Javascript:** usado para la interacción del usuario en las paginas generadas por el «BackEnd». Aparte del uso de Vanilla Javascript también se utilizo el framework Alpine.js por su capacidad de generar componentes dinámicos.
3. **Python:** en concreto se usó la variante Micropython diseñada para sistemas embebidos. Su función es capturar datos de los sensores y enviarlos al servidor vía Http, todo esto por medio de la placa NodeMCU.

## CAPÍTULO III

### RESULTADOS Y DISCUSIÓN

#### 3.1. Análisis y discusión de resultados

##### 3.1.1. Análisis del sistema de secado empleado en la empresa Fideos Victoria.

El sistema de secado que utiliza la empresa Fideos Victoria se basa en el calentamiento de un elemento de tipo radiador por medio del cual circula aire impulsado por un ventilador, el aire caliente pasa a través del producto para finalmente ser expulsado de la cámara junto con la humedad captada permitiendo el secado del producto, este proceso puede ser observado con mayor claridad en la figura 3.1.

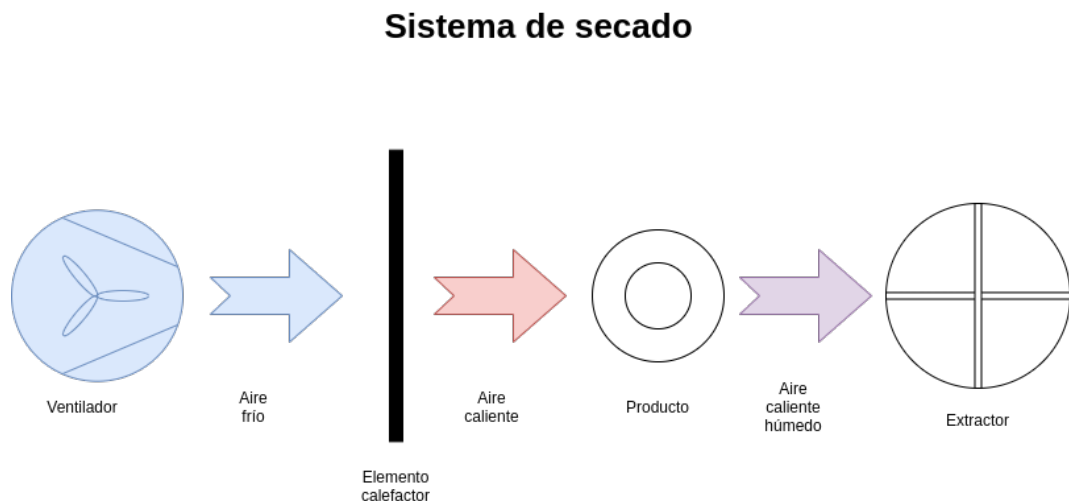


Figura 3.1: Proceso de secado  
Fuente: Elaborado por el investigador

El proceso de secado empieza en el momento en que la cámara se llena con el o los productos que van a ser secados, siendo controlado mediante un mando electrónico encargado de controlar el tiempo de entrada de calor, extracción de humedad y viento para que se produzca el secado.

Normalmente para conocer el estado actual del sistema, si la temperatura ha subido o bajado, se requiere la presencia física de un operario ya que los sensores



actualmente instalados indican la temperatura en paneles lcd integrados y cuentan con la limitante de unicamente medir la temperatura y no la humedad.

### 3.1.2. Descripción de la plataforma utilizada para la adquisición de temperatura y humedad

Para la toma de datos, con anterioridad, la empresa intentó hacer un proceso similar al presentado en este proyecto, pero por motivos internos solo se llevó a cabo la construcción de los dispositivos para la toma de datos, en vista de esto la empresa vio la necesidad de la elaboración del sistema de monitoreo por parte del investigador.

El proceso de recolección de datos, por parte de los dispositivos, se basa en peticiones web al servidor y la representación de errores por medio de códigos de color descritos en la figura 3.2.

#### Comparación placas de desarrollo

Para la recolección de datos se requiere del uso de dispositivos que tienen como fin el ser un puente entre el entorno y el sistema que almacena los datos. En la tabla 3.1 se especifican las características mas relevantes de las placas tomadas a consideración, de las cuales se decidió por NodeMCU ya que existe un balance entre su precio y características útiles para el desarrollo del presente proyecto.

Tabla 3.1: Comparación placas de desarrollo

Parámetros	Arduino	Raspberry	NodeMCU
<b>Conexión incluida por defecto</b>	Serial	Serial, wifi, ethernet	<i>Serial, Wifi</i>
<b>Precio en dólares</b>	Desde 9 Hasta 25	Desde 60 hasta 180	<i>Desde 7 Hasta 20</i>
<b>Lenguajes de programación</b>	C++	Cualquiera compatible con ARM	<i>C++, Lua, Micropython, Javascript</i>

Fuente: Elaborado por el investigador

#### Comparación sensores de temperatura

Una parte fundamental del presente proyecto es la toma de datos de las cámaras, para dicho fin se procedió a comparar entre los sensores que se encuentran en el mercado por medio de la tabla 3.2 para finalmente elegir el sensor DHT22 por su fiabilidad.

Tabla 3.2: Comparación sensores de temperatura/humedad

Parámetros	DHT11	DHT22
Voltaje de operación	3.3/5.5	3.3/5.5
Rango de temperatura	0 ~ 50°C	-42 ~ 80°C
Precisión (temperatura)	± 2°C	± 0.5°C
Rango de humedad	0 ~ 100 %	0 ~ 100 %
Precisión (humedad)	± 5 %	± 2 %
Frecuencia de muestreo	1 Hz	2 Hz
Precio	\$4	\$8

Fuente: Elaborado por el investigador

Finalmente la captura de datos se basó en los componentes electrónicos descritos a continuación:

- **NodeMCU:** Es una placa de desarrollo basada en el chip esp8266 o alguna de sus variantes como es el caso de esp12-ex usado en la tercera versión de dicha placa. El porque del uso de dicha placa fue el contar con un dispositivo con conexión wifi integrado a un costo realmente bajo(en torno a los 10 dólares)[21] para, haciendo uso de sensores y programación contar con comunicación en tiempo real con el Dashboard desarrollado.
- **DHT22:** Es un sensor digital de temperatura y humedad capaz de medir al mismo tiempo ambas magnitudes ademas de ser relativamente fácil de conectar y programar. Este sensor puede medir rangos de temperatura de -40 a 80 grados centígrados con una variación de +- 0.5 grados centígrados y de humedad relativa de 0 a 100 % con una variación de +- 2%, con un intervalo mínimo para la captura de datos de 2 segundos [22] siendo perfecto para el proyecto.
- **Leds:** Son diodos emisores de luz con un consumo energético muy bajo, usado en este proyecto como medio indicador de actividad y/o errores en el dispositivo.

**Códigos de Colores.**




			
apagado	apagado	apagado	<i>Apagado</i>
prendido	apagado	apagado	<i>Conectado a la red</i>
parpadeante	apagado	apagado	<i>No se puede conectar a la red</i>
prendido	prendido	apagado	<i>No existe una producción activa</i>
prendido	parpadeante	apagado	<i>La cámara no se encuentra en la producción actual</i>
prendido	prendido	prendido	<i>Error general</i>
prendido	prendido	parpadeante	<i>Error al subir datos al servidor</i>
prendido	parpadeante	parpadeante	<i>El servidor no se encuentra disponible</i>
parpadeante	parpadeante	parpadeante	<i>Error general</i>

Figura 3.2: Códigos de colores del dispositivo de captura

Fuente: Elaborado por el investigador

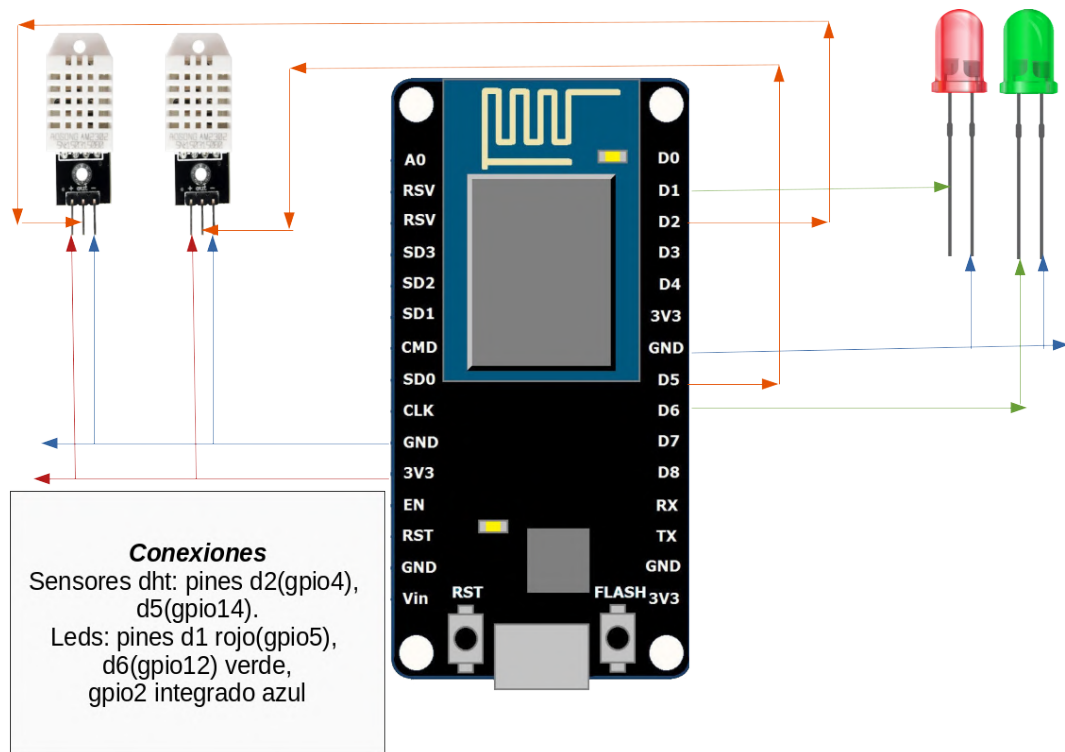


Figura 3.3: Conexiones del dispositivo  
 Fuente: Elaborado por el investigador

### 3.1.2.1. Programación del dispositivo

La placa de desarrollo NodeMCU, por su chip esp-12, posee la capacidad de ser programada en lenguajes de programación tan diversos como es C++, Javascript, Micropython o Lua(lenguaje por defecto [23]). Por medio de la tabla 3.3 se realizó una comparación entre los principales lenguajes que poseen soporte en la placa de desarrollo para finalmente, por los beneficios que ofrece, elegir Micropython.

Tabla 3.3: Comparación de lenguajes de programación

	C++	Javascript	<i>Micropython</i>	Lua
<b>Documentación</b>	Media	Baja	<b><i>Media</i></b>	Baja
<b>Velocidad de ejecución</b>	Alta	Media	<b><i>Alta</i></b>	Alta
<b>Tamaño de ficheros</b>	Baja	Baja	<b><i>Baja</i></b>	Baja
<b>Complejidad</b>	Alta	Media	<b><i>Baja</i></b>	Media
<b>Librerías externas</b>	Alta	Baja	<b><i>Alta</i></b>	Baja

Fuente: Elaborado por el investigador

El algoritmo por medio del cual el dispositivo procede a la captura de datos se

basa en primero inicializar la conexión con la red Wifi a usarse para después por medio de una serie de llamadas al servidor web comprobar si debe o no empezar el proceso de recolección de datos, entrando en un ciclo infinito hasta el momento en el que termina el proceso de secado como puede observarse en la figura 3.4. Cabe destacar que uno de los limitantes de la placa NodeMCU es el contar con un solo núcleo haciendo imposible el realizar tareas en paralelo, pero los desarrolladores de Micropython tomando como base crearon la librería uasyncio, esta librería realiza una serie de procesos internos logrando realizar una especie de programación en paralelo. Por esta característica, la toma de datos se realiza de forma asíncrona evitando el bloqueo del dispositivo al momento de mostrar códigos de errores por medio de leds.

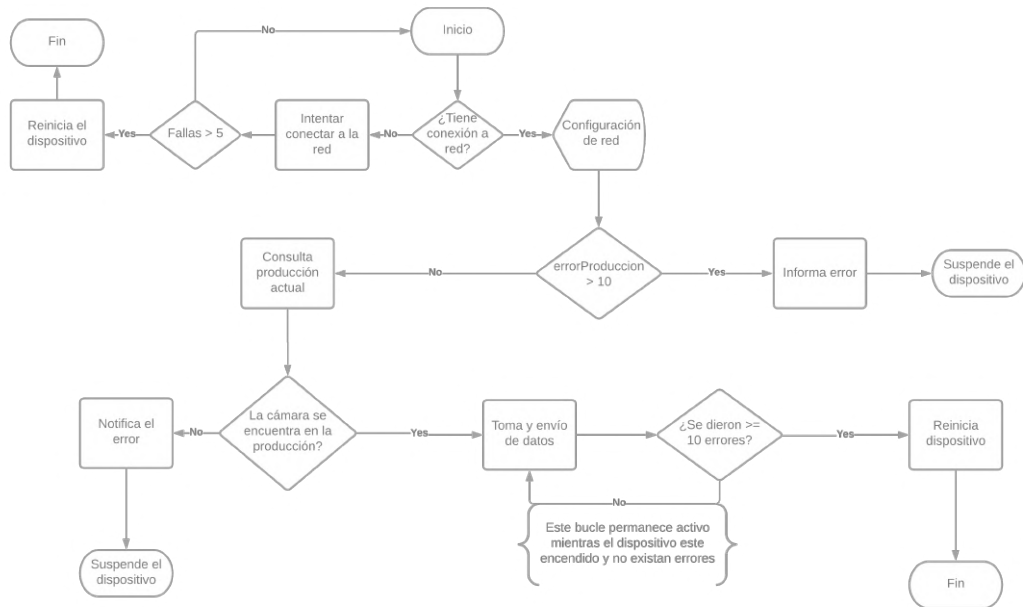


Figura 3.4: Algoritmo para toma de datos  
Fuente: Elaborado por el investigador

### 3.1.3. Estructura base del sistema de monitoreo

En este proyecto se han utilizado algunas tecnologías que por sus características no van a generar un impacto en la economía de la empresa Fideos Victoria. Esta estructura consta de una red LAN en la que se encuentran:

- Un equipo que actúa como servidor web y de base de datos.
- Un conjunto de dispositivos NodeMCU encargados del envío de datos al servidor.

- Cada dispositivo NodeMCU consta de leds para mostrar información del estado del sistema, así como de los sensores DHT22 encargados de la captura de temperatura y humedad.
- A modo de prueba, para la comunicación con dispositivos externos se ha utilizado el servicio Telebit el cual permite exponer a internet un servicio alojado localmente.

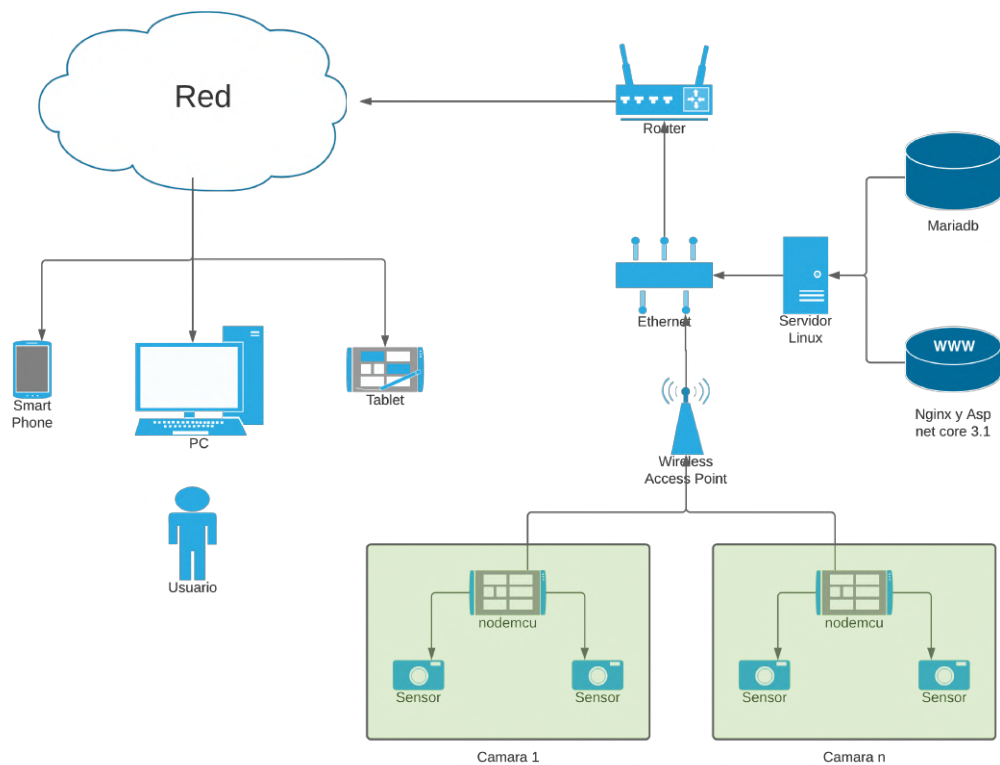


Figura 3.5: Estructura del proyecto  
Fuente: Elaborado por el investigador

### Instalación Asp Net Core 3.1

Asp Net Core es la evolución del .Net Framework de Microsoft, el cual posee algunas ventajas con respecto a su antecesor, siendo la principal el ser multiplataforma. Para el desarrollo de este proyecto se eligió la versión 3.1 ya que es la versión de largo soporte(LTS) que tendrá soporte oficial hasta el 3 de Diciembre del 2022.

Para instalar Asp Net Core en Fedora se realizan los siguientes pasos

1. Importación de la llave

```
sudo rpm --import
https://packages.microsoft.com/keys/microsoft.asc
```

## 2. Inicialización del repositorio

```
sudo wget -O /etc/yum.repos.d/microsoft-prod.repo\  
https://packages.microsoft.com/config/fedora/33/prod.repo
```

## 3. Instalación del sdk

```
sudo dnf install dotnet-sdk-3.1.3
```

## 4. Instalación del runtime

```
sudo dnf install aspnetcore-runtime-3.1.3
```

Para verificar la correcta instalación, en un terminal se digita el siguiente comando.

```
dotnet --info
```

Y como respuesta recibimos la información del sistema operativo así como del sdk y del runtime instalados. Figura 3.6.

```
[administrador@localhost ~]$ dotnet --info  
SDK de .NET Core (reflejando cualquier global.json):  
Version: 3.1.301  
Commit: 7feb845744  
  
Entorno de tiempo de ejecución:  
OS Name: fedora  
OS Version: 31  
OS Platform: Linux  
RID: fedora.31-x64  
Base Path: /usr/share/dotnet/sdk/3.1.301/  
  
Host (useful for support):  
Version: 3.1.5  
Commit: 65cd789777  
  
.NET Core SDKs installed:  
3.1.301 [/usr/share/dotnet/sdk]  
  
.NET Core runtimes installed:  
Microsoft.AspNetCore.App 3.1.5 [/usr/share/dotnet/shared/Microsoft.AspNetCore.App]  
Microsoft.NETCore.App 3.1.5 [/usr/share/dotnet/shared/Microsoft.NETCore.App]  
  
To install additional .NET Core runtimes or SDKs:  
https://aka.ms/dotnet-download
```

Figura 3.6: Información dotnet  
Fuente: Elaborado por el investigador

## Instalación y configuración de MariaDb

MariaDb es un motor de base de datos nacido como una bifurcación de Mysql con licencia GPL(código libre) utilizado por algunas empresas de la talla de Google. En este proyecto es utilizado para almacenar los datos necesarios para el funcionamiento del sistema.

Para instalar MariaDb en Fedora se siguen los siguientes pasos:

### 1. Instalación del servicio

```
sudo dnf -y install mariadb-server
```

## 2. Cambio de la configuración para soportar utf8

```
sudo vi /etc/my.cnf.d/mariadb-server.cnf
```

En la línea 21 especificamos lo siguiente

```
[mysqld]
character-set-server=utf8
```

## 3. Habilitación del servicio al inicio del sistema operativo

```
sudo systemctl enable --now mariadb
```

## 4. Revisión de la versión instalada

```
mysql --version
```

## 5. Configuración inicial

```
sudo mysql_secure_installation
```

## 6. Ingreso a la consola con las credenciales creadas en el paso anterior

```
mysql -u root -p
```

## 7. Habilitación del servicio en el firewall

```
sudo firewall-cmd --add-service=mysql --permanent
```

## 8. Creación de la base de datos a utilizar

```
CREATE DATABASE monitoreo_victoria;
```

## 9. Creación de las credenciales de usuario

```
CREATE USER 'monitor'@'%' IDENTIFIED BY '*****';
```

## 10. Asignación de permisos sobre la base de datos al usuario creado

```
GRANT ALL PRIVILEGES ON monitoreo_victoria.* TO 'monitor'@'%';
FLUSH PRIVILEGES;
```

## Instalación Nginx

Nginx es un servidor web de alto rendimiento que posee funciones de proxy inverso(intermediario entre peticiones de recursos del cliente al servidor). En este proyecto se ha usado para redirigir las peticiones web hacia el servidor kestrel(servidor web en el que se alojan las aplicaciones de Asp Net Core) y así aprovechar los beneficios que Nginx ofrece, como la capacidad de soportar hasta 10000 conexiones simultaneas y el bajo consumo de memoria RAM, comparado con el servidor web Apache.

Para su instalación se siguieron los siguientes pasos:

### 1. Instalación de los paquetes requeridos

```
sudo dnf install nginx
```



2. Habilitación del servicio http en el firewall.

```
firewall-cmd --add-service=http --permanent
```

### Configuración Nginx

Para hacer uso de Nginx como proxy inverso primero se configuraron los parámetros especificados en la figura 3.7 del archivo nginx.conf.

```
server {
    listen      80 default_server;
    server_name _;
    root        /var/www/monitoreovictoria;
    location / {
        proxy_pass          http://localhost:5000;
        proxy_http_version 1.1;
        proxy_set_header    Upgrade $http_upgrade;
        proxy_set_header    Connection keep-alive;
        proxy_set_header    Host $host;
        proxy_cache_bypass  $http_upgrade;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header    X-Forwarded-Proto $scheme;
    }
}
```

Figura 3.7: Configuración Nginx  
Fuente: Elaborado por el investigador

### Creación servicio de administración

Nginx puede administrar peticiones web pero no el inicio y la supervisión del sistema en sí, para solventar esta limitante se procedió a la creación de un servicio basado en systemd con los parámetros especificados en la figura 3.8.

```
[Unit]
Description=Servicio para el monitoreo de fideos victoria

[Service]
WorkingDirectory=/var/www/monitoreovictoria
ExecStart=/usr/bin/dotnet /var/www/monitoreovictoria/MonitoreoFideosVictoria.dll
Restart=always
# Restart service after 10 seconds if the dotnet service crashes:
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=dotnet-monitoreo
User=nginx
Environment=ASPNETCORE_ENVIRONMENT=Production
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false

[Install]
WantedBy=multi-user.target
```

Figura 3.8: Servicio del sistema  
Fuente: Elaborado por el investigador

### 3.1.4. Desarrollo de la metodología

El presente proyecto se llevará a cabo haciendo uso de la metodología XP la cual, por sus características, permite la realización del diseño, codificación y pruebas de manera rápida por medio de las fases mencionadas a continuación:

1. Planificación
2. Diseño
3. Codificación
4. Pruebas

#### 3.1.4.1. Roles del proyecto

Tabla 3.4: Roles XP del proyecto

<b>Rol</b>	<b>Responsable</b>	<b>Función</b>
Usuario	Gerente/propietario empresa Fideos Victoria	Determinar los procesos a implementar
Programador	Tixilema Merchan Carlos Israel	Desarrollar el sistema de monitoreo de temperatura y humedad para las cámaras de secado
Tester	Tixilema Merchan Carlos Israel	Realizar las pruebas de aceptación

Fuente: Elaborado por el investigador

#### 3.1.4.2. Planificación del Proyecto

En esta fase se define el alcance y las prioridades en base a historias de usuario hechas en conjunto con el usuario. Las historias de usuario representan los requisitos expresados en lenguaje natural y con una extensión corta, siendo realizadas en base a la recolección de información por medio de reuniones con el cliente y la observación del proceso de secado.

## Historias de Usuario

Tabla 3.5: Historia de usuario: Ingreso al sistema

<b>Historia de Usuario</b>	
<b>Código:</b> H1	<b>Usuarios:</b> Todos
<b>Nombre:</b> Ingreso al sistema	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Bajo
<b>Puntos estimados:</b> 6	<b>Iteración asignada:</b> 1
<b>Descripción:</b> Para ingresar al sistema se debe solicitar un usuario y una contraseña	
<b>Observación:</b> El usuario será un email único con un rol Administrador, Empleado o Sensor	
Fuente: Elaborado por el investigador	

Tabla 3.6: Historia de usuario: Registro de Usuarios

<b>Historia de Usuario</b>	
<b>Código:</b> H2	<b>Usuarios:</b> Todos
<b>Nombre:</b> Registro de Usuarios	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Alto
<b>Puntos estimados:</b> 6	<b>Iteración asignada:</b> 1
<b>Descripción:</b> Los usuarios se auto registrarán ingresando su email y una contraseña	
<b>Observación:</b> Por defecto debe asignarse el rol empleado los nuevos usuarios, excepto por los emails listados para ser administradores	
Fuente: Elaborado por el investigador	

Tabla 3.7: Historia de usuario: Gestión de Cámaras

<b>Historia de Usuario</b>	
<b>Código:</b> H3	<b>Usuarios:</b> Todos
<b>Nombre:</b> Gestión de Cámaras	
<b>Prioridad:</b> Media	<b>Riesgo:</b> Bajo
<b>Puntos estimados:</b> 8	<b>Iteración asignada:</b> 2
<b>Descripción:</b> El usuario podrá visualizar, crear, editar y eliminar cámaras	
<b>Observación:</b> La acción de eliminar debe poseer una confirmación para evitar errores además los usuarios que no sean administradores no pueden crear, modificar o eliminar	
Fuente: Elaborado por el investigador	

Tabla 3.8: Historia de usuario: Gestión de Productos

<b>Historia de Usuario</b>	
<b>Código:</b> H4	<b>Usuarios:</b> Todos
<b>Nombre:</b> Gestión de Productos	
<b>Prioridad:</b> Media	<b>Riesgo:</b> Bajo
<b>Puntos estimados:</b> 8	<b>Iteración asignada:</b> 2
<b>Descripción:</b> El usuario podrá visualizar, crear, editar y eliminar productos	
<b>Observación:</b> La acción de eliminar debe poseer una confirmación para evitar errores además los usuarios que no sean administradores no pueden crear, modificar o eliminar	

Fuente: Elaborado por el investigador

Tabla 3.9: Historia de usuario: Gestión de Sensores

<b>Historia de Usuario</b>	
<b>Código:</b> H5	<b>Usuarios:</b> Administrador
<b>Nombre:</b> Gestión de Sensores	
<b>Prioridad:</b> Media	<b>Riesgo:</b> Bajo
<b>Puntos estimados:</b> 8	<b>Iteración asignada:</b> 2
<b>Descripción:</b> El usuario podrá visualizar, crear, editar y eliminar sensores	
<b>Observación:</b> Estos deben coincidir con los sensores instalados en las cámaras además los usuarios que no sean administradores no pueden crear, modificar o eliminar	

Fuente: Elaborado por el investigador

Tabla 3.10: Historia de usuario: Gestión de Producciones

<b>Historia de Usuario</b>	
<b>Código:</b> H6	<b>Usuarios:</b> Todos
<b>Nombre:</b> Visualización, Creación y Modificación de Producciones	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Alto
<b>Puntos estimados:</b> 8	<b>Iteración asignada:</b> 3
<b>Descripción:</b> El usuario podrá crear una producción nueva en base a los productos y cámaras existentes, así como, visualizar y modificar las producciones	
<b>Observación:</b> Si existe una producción activa no debe permitir crear una nueva y en caso de existir varias producciones activas para las operaciones se usará la más reciente	

Fuente: Elaborado por el investigador

Tabla 3.11: Historia de usuario: Eliminación de Producciones

<b>Historia de Usuario</b>	
<b>Código:</b> H7	<b>Usuarios:</b> Administrador
<b>Nombre:</b> Eliminación Producciones	
<b>Prioridad:</b> Baja	<b>Riesgo:</b> Alto
<b>Puntos estimados:</b> 5	<b>Iteración asignada:</b> 2
<b>Descripción:</b> Eliminar una producción implica eliminar muchos elementos por lo que debe ser hecho con sumo cuidado	
<b>Observación:</b> Por la eliminación en cascada este proceso puede generar errores inesperados	

Fuente: Elaborado por el investigador

Tabla 3.12: Historia de usuario: Registro de Temperatura

<b>Historia de Usuario</b>	
<b>Código:</b> H8	<b>Usuarios:</b> Sensor
<b>Nombre:</b> Registro de Temperatura	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Medio
<b>Puntos estimados:</b> 10	<b>Iteración asignada:</b> 3
<b>Descripción:</b> El dispositivo enviará las temperaturas al servidor	
<b>Observación:</b> Debe controlarse que solo el usuario con rol Sensor ingrese los datos de tipo temperatura	

Fuente: Elaborado por el investigador

Tabla 3.13: Historia de usuario: Visualización de la Temperatura

<b>Historia de Usuario</b>	
<b>Código:</b> H9	<b>Usuarios:</b> Todos
<b>Nombre:</b> Visualización de la Temperatura	
<b>Prioridad:</b> Alta	<b>Riesgo:</b> Medio
<b>Puntos estimados:</b> 15	<b>Iteración asignada:</b> 3
<b>Descripción:</b> Por medio de gráficos debe presentarse la temperatura de la producción que se encuentra activa	
<b>Observación:</b> La comunicación con los sensores debe darse en tiempo real	

Fuente: Elaborado por el investigador

Tabla 3.14: Historia de usuario: Reportes de temperaturas

<b>Historia de Usuario</b>	
<b>Código:</b> H10	<b>Usuarios:</b> Administrador
<b>Nombre:</b> Reportes de temperaturas	
<b>Prioridad:</b> Media	<b>Riesgo:</b> Bajo
<b>Puntos estimados:</b> 10	<b>Iteración asignada:</b> 4
<b>Descripción:</b> Por medio de gráficos debe representarse la temperatura y/o humedad de la producción que el usuario elija	
<b>Observación:</b> El tipo de gráfico debe presentar los datos de la forma mas legible posible	

Fuente: Elaborado por el investigador

Tabla 3.15: Historia de usuario: Exportación de datos

<b>Historia de Usuario</b>	
<b>Código:</b> H11	<b>Usuarios:</b> Administrador
<b>Nombre:</b> Exportación de datos	
<b>Prioridad:</b> Media	<b>Riesgo:</b> Bajo
<b>Puntos estimados:</b> 6	<b>Iteración asignada:</b> 4
<b>Descripción:</b> El usuario puede exportar los datos que desee de la producción seleccionada.	
<b>Observación:</b> El archivo exportado debe poder ser abierto en Excel o Libreoffice calc para su análisis.	

Fuente: Elaborado por el investigador

## Tareas

Las Tareas son actividades con un inicio y un fin asignadas a un responsable el cual llevará a cabo dichas actividades en un tiempo determinado.

Tomando como base las historias de usuario y las iteraciones previamente definidas se procedió a la creación de las tareas mostradas a continuación.

**Historia de usuario:** Ingreso al sistema

Tabla 3.16: Diseño de interfaz de ingreso

Tarea	
<b>Número de Tarea:</b> T1	<b>Código de Historia:</b> H1
<b>Nombre de Tarea:</b> Diseño de interfaz de ingreso	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 6
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará el diseño de la interfaz para el ingreso de usuarios.	

Fuente: Elaborado por el investigador

**Historia de usuario:** Registro de Usuarios

Tabla 3.17: Diseño de interfaz de registro

Tarea	
<b>Número de Tarea:</b> T2	<b>Código de Historia:</b> H2
<b>Nombre de Tarea:</b> Diseño de interfaz de registro	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 3
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará el diseño de la interfaz para el registro de usuarios.	

Fuente: Elaborado por el investigador

Tabla 3.18: Implementación de confirmación vía email

Tarea	
<b>Número de Tarea:</b> T3	<b>Código de Historia:</b> H2
<b>Nombre de Tarea:</b> Implementación de confirmación vía email	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 2
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará el envío de un email para confirmar la cuenta del usuario.	

Fuente: Elaborado por el investigador

Tabla 3.19: Implementación de manejo de roles de usuario

<b>Tarea</b>	
<b>Número de Tarea:</b> T4	<b>Código de Historia:</b> H2
<b>Nombre de Tarea:</b> Implementación de manejo de roles de usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará la interfaz para que un usuario administrador modifique los roles de cada usuario	

Fuente: Elaborado por el investigador

#### Historia de usuario: Gestión de Cámaras

Tabla 3.20: Diseño de interfaz para la gestión de cámaras

<b>Tarea</b>	
<b>Número de Tarea:</b> T5	<b>Código de Historia:</b> H3
<b>Nombre de Tarea:</b> Diseño de interfaz para la gestión de cámaras	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 8
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará el diseño de la interfaz para la gestión de cámaras.	

Fuente: Elaborado por el investigador

#### Historia de usuario: Gestión de productos

Tabla 3.21: Diseño de interfaz para la gestión de productos

<b>Tarea</b>	
<b>Número de Tarea:</b> T6	<b>Código de Historia:</b> H4
<b>Nombre de Tarea:</b> Diseño de interfaz para la gestión de productos	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 8
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Descripción: Se realizará el diseño de la interfaz para la gestión de productos.	

Fuente: Elaborado por el investigador



**Historia de usuario:** Gestión de sensores

Tabla 3.22: Diseño de interfaz para la gestión de sensores

<b>Tarea</b>	
<b>Número de Tarea:</b> T7	<b>Código de Historia:</b> H5
<b>Nombre de Tarea:</b> Diseño de interfaz para la gestión de sensores	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 8
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará el diseño de la interfaz para la gestión de sensores.	

Fuente: Elaborado por el investigador

**Historia de usuario:** Visualización, creación y modificación de producciones

Tabla 3.23: Diseño de interfaz para la gestión de producciones

<b>Tarea</b>	
<b>Número de Tarea:</b> T8	<b>Código de Historia:</b> H6
<b>Nombre de Tarea:</b> Diseño de interfaz para la gestión de producciones	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 8
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará el diseño de la interfaz para la gestión de producciones.	

Fuente: Elaborado por el investigador

**Historia de usuario:** Registro de temperatura

Tabla 3.24: Diseño Web API para el registro de temperatura

<b>Tarea</b>	
<b>Número de Tarea:</b> T9	<b>Código de Historia:</b> H8
<b>Nombre de Tarea:</b> Diseño Web API para el registro de temperatura	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 5
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará el diseño de una Web API para que los dispositivos IOT envíen los datos al servidor	

Fuente: Elaborado por el investigador

Tabla 3.25: Implementación de lógica de los dispositivos IOT

Tarea	
<b>Número de Tarea:</b> T10	<b>Código de Historia:</b> H8
<b>Nombre de Tarea:</b> Implementación de lógica de los dispositivos IOT	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 5
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará la implementación de lógica de captura de temperatura y humedad en los dispositivos IOT para su envío al servidor por medio de una Web API	

Fuente: Elaborado por el investigador

**Historia de usuario:** Visualización de la temperatura

Tabla 3.26: Diseño de un dashboard

Tarea	
<b>Número de Tarea:</b> T11	<b>Código de Historia:</b> H9
<b>Nombre de Tarea:</b> Diseño de un dashboard	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 8
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará el diseño de un dashboard que debe mostrar la información actual sobre la producción activa en el momento	

Fuente: Elaborado por el investigador

Tabla 3.27: Diseño de un medio para comunicación en tiempo real

Tarea	
<b>Número de Tarea:</b> T12	<b>Código de Historia:</b> H9
<b>Nombre de Tarea:</b> Diseño de un medio para comunicación en tiempo real	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 7
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará el diseño de un medio para la comunicación en tiempo real entre los dispositivos IOT y el dashboard para visualizar activamente los cambios con respecto a la temperatura y humedad en cada cámara	

Fuente: Elaborado por el investigador

**Historia de usuario:** Reportes de temperaturas

Tabla 3.28: Diseño de reportes

<b>Tarea</b>	
<b>Número de Tarea:</b> T13	<b>Código de Historia:</b> H10
<b>Nombre de Tarea:</b> Diseño de reportes	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 10
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará el diseño de reportes en base a los datos recabados y parámetros establecidos por el usuario	

Fuente: Elaborado por el investigador

**Historia de usuario:** Exportación de datos

Tabla 3.29: Exportación de datos para análisis

<b>Tarea</b>	
<b>Número de Tarea:</b> T14	<b>Código de Historia:</b> H11
<b>Nombre de Tarea:</b> Exportación de datos para análisis	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 6
<b>Programador Responsable:</b> Carlos Tixilema	
<b>Descripción:</b> Se realizará la lógica para que por medio de una interfaz el usuario pueda elegir los datos a exportar.	

Fuente: Elaborado por el investigador

### **Plan de entrega del proyecto**

Después de definir las historias de usuario y las tareas para el desarrollo del sistema de monitoreo para cámaras de secado de la empresa Fideos Victoria, se ha elaborado un plan de entrega en el que constan las diferentes actividades a desarrollar en un tiempo medio así como las historias de usuario que serán llevadas a cabo en cada iteración.

Tabla 3.30: Plan de entrega

Módulo	Historias	Nombre Historia	Iteraciones	Prioridad	Días estimados	Horas estimadas
Administración	Historia 5	Gestión de Sensores	2	Media	8	32
	Historia 7	Eliminación Producciones	2	Baja	5	20
	Historia 1	Ingreso al sistema	1	Alta	6	24
	Historia 2	Registro de Usuarios	1	Alta	6	24
Visualización	Historia 3	Gestión de Cámaras	2	Media	8	32
	Historia 4	Gestión de Productos	2	Media	8	32
Historia 6	Visualización, Creación y Modificación de Producciones	Visualización, Creación y Modificación de Producciones	3	Alta	8	32
			3	Alta	15	60
			3	Alta	10	40
IOT	Historia 8	Registro de Temperatura	4	Media	10	40
			4	Media	6	24
Reportes	Historia 10	Reportes de temperaturas	4	Media	6	24
			4	Media	6	24

Fuente: Elaborado por el investigador

### 3.1.4.3. Diseño

En la metodología XP el diseño debe ser simple, comprensible y aplicable para así generar un costo y esfuerzo bajo.

#### Tarjetas CRC (Clase-Responsabilidad-Colaboración)

Una de las técnicas usadas para el desarrollo ágil, incluyendo la metodología XP, es el uso de tarjetas CRC. Por su forma estas tarjetas permiten realizar un diseño orientado a objetos, permitiendo ver las clases de una forma dinámica basada en su comportamiento y no solo como un almacén de datos.

Una tarjeta CRC está dividida en tres zonas distintas, las cuales son:

- **Parte superior:** contiene el nombre de la clase.
- **Abajo Izquierda:** contiene las responsabilidades de la clase.
- **Abajo Derecha:** contiene las clases que colaboran para cumplir con las responsabilidades.

Tabla 3.31: Tarjeta CRC - TemperaturaController

Clase: TemperaturaController	
Responsabilidades	Colaboradores
Recibir las temperaturas capturadas	SistemaHub
Verificar que solo se puedan insertar las temperaturas de la producción activa	
Notificar al dashboard las temperaturas recibidas	

Fuente: Elaborado por el investigador

Tabla 3.32: Tarjeta CRC - AuthController

Clase: AuthController	
Responsabilidades	Colaboradores
Recibir las credenciales de usuario	
Generar un token de acceso	

Fuente: Elaborado por el investigador

Tabla 3.33: Tarjeta CRC - CamaraController

<b>Clase:</b> CamaraController	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Entregar las cámaras almacenadas en el sistema	

Fuente: Elaborado por el investigador

Tabla 3.34: Tarjeta CRC - ProduccionController

<b>Clase:</b> ProduccionController	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Entregar la producción que se encuentre activa	

Fuente: Elaborado por el investigador

Tabla 3.35: Tarjeta CRC - ProductoController

<b>Clase:</b> ProductoController	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Entregar los productos almacenados en el sistema	

Fuente: Elaborado por el investigador

Tabla 3.36: Tarjeta CRC - RolesSeed

<b>Clase:</b> RolesSeed	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Crear los roles por defecto en caso de no existir	IdentityRole

Fuente: Elaborado por el investigador

Tabla 3.37: Tarjeta CRC - EmailSender

<b>Clase:</b> EmailSender	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Proporcionar una forma de enviar emails	Registro de usuarios

Fuente: Elaborado por el investigador

Tabla 3.38: Tarjeta CRC - NavigationIndicatorHelper

<b>Clase:</b> NavigationIndicatorHelper	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Marcar la pagina actual como activa	Navegación

Fuente: Elaborado por el investigador

Tabla 3.39: Tarjeta CRC - SistemaHub

<b>Clase:</b> SistemaHub	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Proporcionar una forma de comunicación en tiempo real	

Fuente: Elaborado por el investigador

### Diseño de la Base de Datos

La base de datos fue desarrollada en MariaDB 10.3 usando los datos otorgados por la empresa. Tomando como partida el modelo Code-First de Entity Framework primero se crearon las clases, para por medio del proceso de Scaffolding se generaran las tablas con sus respectivas relaciones y atributos, siendo el modelo final el presentado en las figuras 3.9 y 3.10.

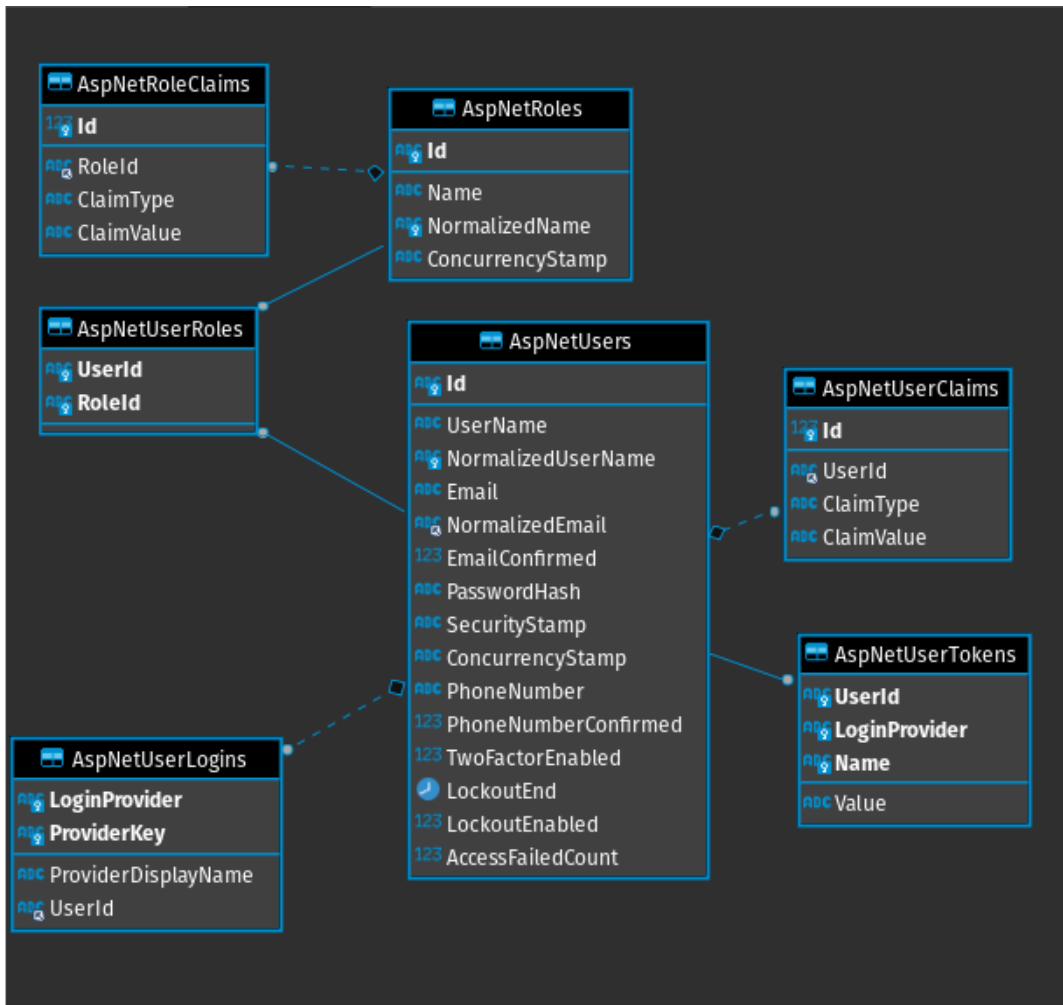


Figura 3.9: Diagrama de Base de Datos(1)  
 Fuente: Elaborado por el investigador



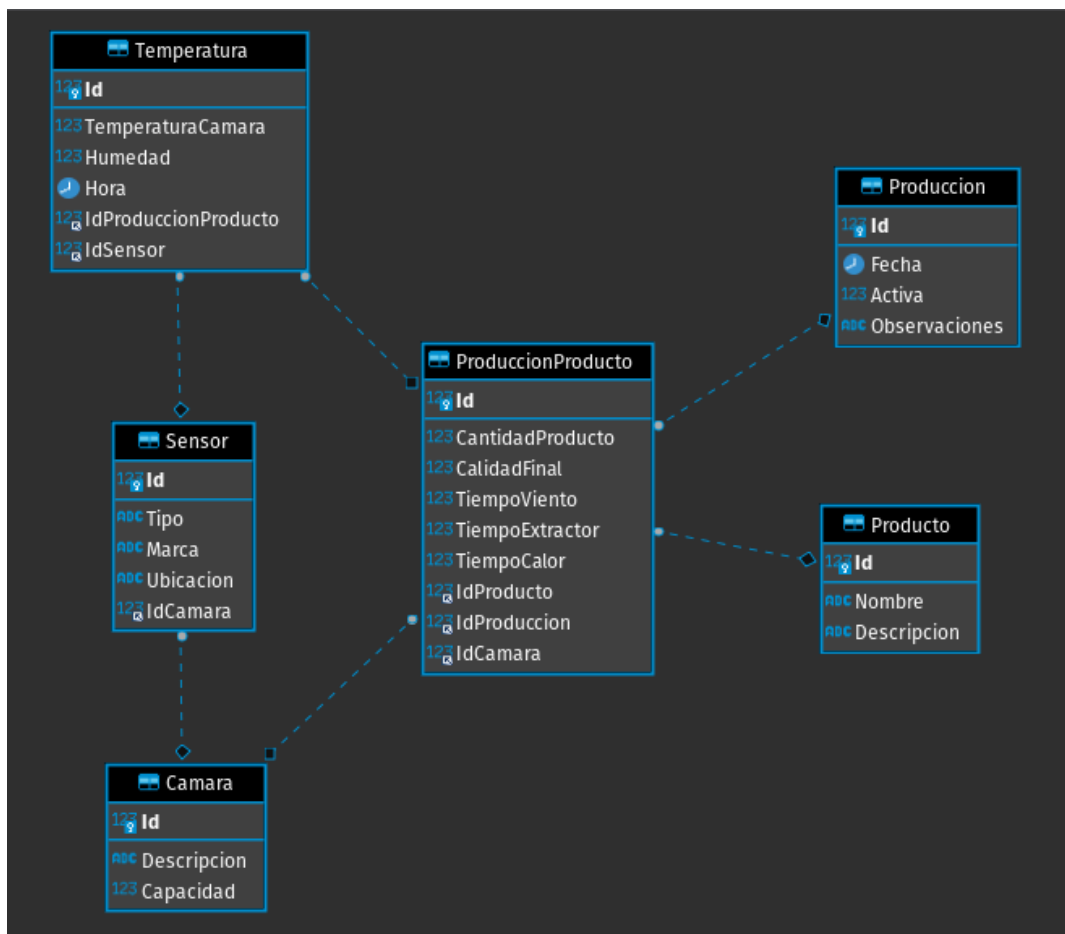


Figura 3.10: Diagrama de Base de Datos(2)

Fuente: Elaborado por el investigador

#### 3.1.4.4. Codificación

El sistema fue construido usando la tecnología Razor Pages, al estar basado en MVC, su estructura puede ser combinada aprovechando las mejores características de ambas. La estructura que se usó fue la siguiente:

- **Areas:** Contiene las clases utilizadas por el marco Identity para la gestión de usuarios.
- **Controller:** Contiene los controladores(puntos de acceso de tipo API) para el manejo de datos por parte de los dispositivos IOT y funcionalidad Javascript.
- **Data:** Contiene el contexto de la base de datos, encargado de abstraer las acciones que se realizan con los datos, además de una clase usada para la generación de roles por defecto.

- **Hubs:** Contiene el Hub o concentrador usado para la comunicación en tiempo real vía SignalR.
- **Micropython:** Contiene los scripts escritos en python para ejecutarse en los dispositivos IOT de tipo NodeMCU, estos datos no se compilan con el proyecto razón por la cual no permanecieron al momento de desplegar el servidor web.
- **Migrations:** Contiene las clases generadas por EF Core para crear la base de datos, cada una contiene un identificador basado en la fecha que se realizó e internamente los cambios a hacer en la base de datos.
- **Models:** Contiene las clases modelo, las cuales contienen la estructura base del sistema. Estas clases, por medio de EF Core, son reflejadas en la base de datos simplificando el despliegue de los cambios realizados.
- **Pages:** Contiene las vistas, estas son paginas de tipo cshtml que se renderizan en HTML para mostrarse al usuario. Cada subcarpeta representa la ruta que sigue el navegador.
- **Utils:** Contiene clases auxiliares, como son enumeraciones, interfaces o clases que agregan funcionalidad extra al sistema.
- **wwwroot:** Contiene archivos estáticos de tipo Javascript, CSS e imágenes. En esta sección se incluyen los scripts que ayudan a mitigar las limitaciones de Asp Net Framework Core, especialmente Razor Pages, como es la creación de interfaces que se modifiquen en base a parámetros dinámicos así como las librerías y plugins de Javascript usados en el sistema.

### Creación del proyecto

El proyecto fue creado utilizando la línea de comandos proporcionada Asp Net Core, esta herramienta proporciona una serie de comandos para configurar el proyecto a crearse, siendo el comando utilizado el siguiente:

```
dotnet new webapp -au Individual -o MonitoreoFideosVictoria
```

Los parámetros especificados son los siguientes:

- **dotnet new webapp:** crea un proyecto de Razor Pages por medio de la plantilla webapp.
- **-au Individual:** configura el uso de Identity para el manejo de usuarios con la plantilla de autenticación individual basada en el almacenamiento de los datos de ingreso en una base de datos.

- **-o MonitoreoFideosVictoria:** especifica el espacio de nombres que utiliza el sistema y por lo consiguiente el nombre de la carpeta que contendrá el proyecto.

### Desarrollo de la base de datos

Para el manejo de los datos se utilizó el enfoque Code First de EF Core, en este enfoque lo primero, y más importante, es la creación de las clases con todos sus atributos para por medio del proceso de Scaffold integrado en EF Core se reflejen dichas clases en forma de tablas en la base de datos.

Para hacer uso de EF Core y la herramienta de Scaffolding primero se instalaron las herramientas usando los siguientes comandos:

```
//herramientas
dotnet tool install --global dotnet-ef
dotnet tool install --global dotnet-aspnet-codegenerator
//nugets
dotnet add package Microsoft.EntityFrameworkCore.Sqlite
dotnet add package Microsoft.VisualStudio.Web.CodeGeneration.Design
dotnet add package Microsoft.EntityFrameworkCore.Design
dotnet add package Microsoft.EntityFrameworkCore.SqlServer
```

Las clases utilizadas fueron las mostradas en la figura 3.11, cada una de ellas posee atributos internos que especifican características como las relaciones con otras clases así como el formato que tendrá al momento del renderizado.

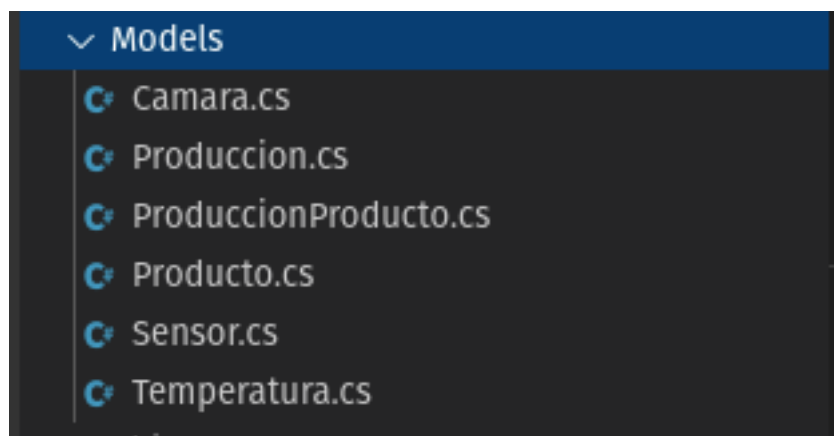


Figura 3.11: Clases base para el manejo de la base de datos  
Fuente: Elaborado por el investigador

Ademas de las clases, se requirió modificar el contexto con el que trabaja EF Core especificando las clases anteriormente mencionadas como propiedades del contexto, quedando de la siguiente forma:

```

public class ApplicationDbContext : IdentityDbContext
{
    public DbSet<Producto> Producto { get; set; }
    public DbSet<Camara> Camara { get; set; }
    public DbSet<Produccion> Produccion { get; set; }
    public DbSet<ProduccionProducto> ProduccionProducto { get; set; }
    public DbSet<Sensor> Sensor { get; set; }
    public DbSet<Temperatura> Temperatura { get; set; }
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext>
        options :: base(options)) {}
}

```

## Uso de Mysql

Por defecto la plantilla webapp usa Sqlite como base de datos, pero como se optó por Mysql se deben realizar algunos cambios. Primeramente, se debe instalar el nuget Pomelo.EntityFrameworkCore.MySql, el cual extiende la funcionalidad de EF Core para trabajar con Mysql o MariaDb, para eso se utilizó el siguiente comando:

```
dotnet add package Pomelo.EntityFrameworkCore.MySql
```

Además se requirió la modificación de la cadena de conexión en el archivo appsettings especificando las credenciales para el acceso a MariaDb.

```

"ConnectionStrings":
  {
    "DefaultConnection":
      "Server=192.168.1.**;
      Database=***;
      User=*****;
      Password=*****;"
  },

```

En el archivo startup se modificó el método ConfigureServices para especificar el uso de MariaDb tomando como base la cadena de conexión del archivo appsettings.

```

services.AddDbContext<ApplicationDbContext>(options =>
options.UseMySQL(
Configuration.GetConnectionString("DefaultConnection"),
mysqlOptions =>
    mysqlOptions.ServerVersion
    (new Version(10, 3, 30), ServerType.MariaDb)
)
);

```

Ya que no se hace uso de Sqlite se procedió a eliminar la base de datos creada por defecto así como la migración inicial con los siguientes comandos:

```
dotnet ef migrations remove
dotnet ef database drop
```

Una vez creadas las clases, por medio de las siguientes líneas de comandos se creó una migración y luego se ejecutó el proceso de scaffolding concluyendo con la generación de las tablas de la base de datos.

```
dotnet ef migrations add migracioninicial
dotnet ef database update
```

### **Manejo de Usuarios**

Para el manejo de usuarios se utilizó el marco Identity, al momento de generar el proyecto se especificó el tipo de autenticación individual, es decir una autenticación basada en el almacenamiento de los datos de usuario en una base proporcionada por el desarrollador. Dicho marco proporciona algunas características, como son el manejo de roles o el uso de tokens, que lo convierten en una opción robusta para el manejo de usuarios.

Todas las características mencionadas anteriormente se complementan con una interfaz de usuario configurada por defecto con todas las opciones que necesita un usuario como son el registro, ingreso, recuperación de contraseña por mencionar algunas. Estas interfaces son generadas en inglés y son parte de un archivo dll del sistema, por este motivo al inicializar el proyecto no aparecen los archivos que contienen su lógica y presentación. Como se requería modificarlas se usó el proceso de scaffolding con el siguiente comando:

```
dotnet aspnet-codegenerator identity
```

Al terminar su ejecución se generaron los archivos correspondientes a la interfaz utilizada por Identity como se observa en la figura 3.12.

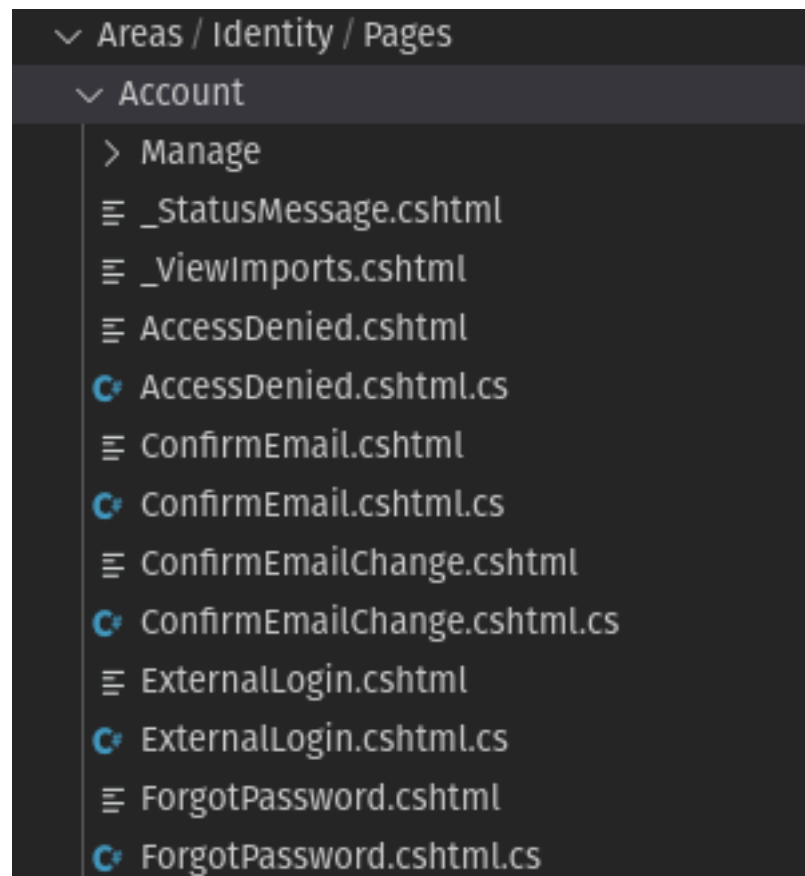


Figura 3.12: Archivos generados por Identity  
Fuente: Elaborado por el investigador

### Políticas de Usuario

Las políticas son un mecanismo por medio del cual se controla el acceso al sistema o parte del mismo tomando como base reglas establecidas en el método `ConfigureServices` del archivo `Startup.cs`.

Las políticas implementadas fueron las siguientes:

- **API:** Esta política se encargó del acceso de usuarios que poseen el rol «Sensor» y que únicamente se autenticuen por medio de un token JWT. Su importancia radica en evitar que cualquier usuario sin los privilegios necesarios realice acciones que le corresponden únicamente a aquellos que tengan un rol de Sensor, como es el caso del envío de datos por medio de los dispositivos IOT.
- **Administrador/Empleado:** Estas políticas se encargan de controlar y limitar el acceso de los usuarios tomando como base el rol que posea.

```

services.AddAuthorization(options =>
{
    options.AddPolicy("API", policy =>
    {
        policy.AuthenticationSchemes.Add(JwtBearerDefaults.AuthenticationScheme);
        policy.RequireAuthenticatedUser();
        policy.RequireRole("Sensor");
    });
    options.AddPolicy("Politica-Administrador", policy =>
    {
        policy.RequireRole("Administrador");
    });
    options.AddPolicy("Politica-Empleado", policy =>
    {
        policy.RequireRole("Empleado");
    });
});

```

Figura 3.13: Políticas de acceso  
Fuente: Elaborado por el investigador

### Uso de JWT(Json Web Token)

Uno de los puntos más importantes de todo sistema es la seguridad, para cumplir con este punto se creó un controlador que expone un método Http encargado de recibir las credenciales de acceso de un usuario y generar un token. Este token sirve para el ingreso al sistema sin la necesidad de usar las credenciales del usuario evitando que puedan ser vulnerados datos como el usuario o la contraseña. Dentro del sistema de monitoreo su función principal es asegurar las llamadas a métodos Http por parte de los dispositivos IOT al servidor.

Para que JWT pudiera ser utilizado primeramente se modificó el método ConfigureServices del archivo Startup.cs añadiendo los parámetros que pueden observarse en la figura 3.14.

```

services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
.AddRoles<IdentityRole>(C)
.AddEntityFrameworkStores<ApplicationDbContext>(C);
services.AddAuthentication(options => options.RequireAuthenticatedSignIn = true)
.AddJwtBearer(x =>
{
    x.RequireHttpsMetadata = false;
    x.SaveToken = true;
    x.TokenValidationParameters = new TokenValidationParameters
    {
        ValidateIssuerSigningKey = true,
        IssuerSigningKey = new SymmetricSecurityKey(key),
        ValidateIssuer = false,
        ValidateAudience = false
    };
});

```

Figura 3.14: Configuración JWT  
Fuente: Elaborado por el investigador

## Habilitación SignalR

SignalR es el componente principal del sistema de monitoreo, ya que por medio del uso de esta librería los dispositivos IOT comunican los valores medidos al servidor y este se encarga de notificar a todos aquellos clientes que se encuentran suscritos al Hub.

Entre las características más prominentes de SignalR constan, la administración de conexiones automáticamente, el envío de mensajes simultáneos entre clientes conectados, envío de mensajes a usuarios específicos y la elección automática del medio de comunicación en tiempo real adecuado a cada cliente y servidor, esta última característica convierte a SignalR en un medio de comunicación altamente robusto, ya que, si un medio de comunicación no se encuentra disponible la librería se encarga automáticamente de cambiar a otro que si lo esté. Los medios de comunicación que puede utilizar SignalR son:

- **WebSockets:** protocolo de red basado en TCP que establece conexiones entre puntos finales de comunicación llamados sockets para el intercambio de información en forma bidireccional.
- **Long Polling:** haciendo uso de peticiones Http, el servidor se encarga de mantener abierta una petición hasta que reciba una actualización para inmediatamente actualizar el cliente y generar una nueva petición.
- **Server-Sent Events:** En este protocolo la comunicación se da solo de servidor al cliente y acepta solo texto.

Para habilitar la comunicación en tiempo real por medio de Signalr, primero se configuró el Hub con el método por medio del cual se comunican los clientes y el servidor. En el método `ConfigureServices` de la clase `Startup` se habilitó el uso de SignalR con el código indicado a continuación:

```
services.AddSignalR();
```

Por su parte el Hub se configuró creando una carpeta llamada `Hubs`, la cual contiene una clase llamada `SistemaHub`, que a su vez contiene el método descrito a continuación:

```
public async Task EnvioProductoProduccion(ProduccionProducto
    produccionProducto)
{
    await Clients.All
        .EnvioProductoProduccion(produccionProducto);
}
```



## Configuración de envío de emails

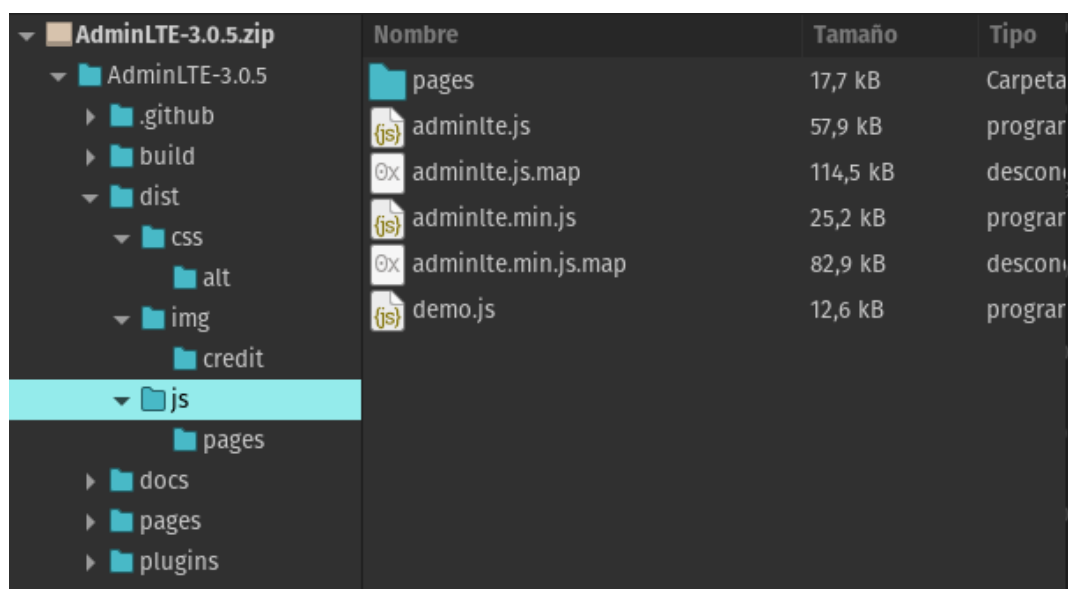
Para el ingreso al sistema de monitoreo se requiere de una cuenta validada por el administrador, por este motivo se requirió el uso de un servicio de envío de emails. El servicio elegido fue Sendgrid ya que posee un servicio gratuito en el que puede enviar 100 emails al día, además de estar completamente integrado con .Net Core.

Su configuración está dada por la clase EmailSender siendo habilitada en la clase Startup con el código indicado a continuación:

```
services.AddTransient<IEmailSender, EmailSender>();  
services.Configure<AuthMessageSenderOptions>(Configuration);
```

## Integración de AdminLTE

Para hacer uso de esta plantilla, pudiendo ser cualquier otra basada en HTML y CSS, se debieron colocar los archivos proporcionados en lugares específicos del proyecto, más concretamente el contenido de la carpeta dist, como se observa en la figura 3.15, del archivo descargado de su sitio oficial, a la carpeta wwwroot del proyecto.



Nombre	Tamaño	Tipo
pages	17,7 kB	Carpeta
adminlte.js	57,9 kB	programa
adminlte.js.map	114,5 kB	desconocido
adminlte.min.js	25,2 kB	programa
adminlte.min.js.map	82,9 kB	desconocido
demo.js	12,6 kB	programa

Figura 3.15: Archivos AdminLTE  
Fuente: Elaborado por el investigador

Una vez que las librerías, archivos de estilo e imágenes estuvieron colocadas en su correcta ubicación se procedió a la creación del Layout que contiene los enlaces para la navegación dentro del sistema y sirve como base común para todas las paginas. Además del Layout, se crearon vistas parciales que se encargan de tareas extra como la instanciación de los scripts que se utilizarán en el Layout,

su estructura final fue la mostrada en la figura 3.16.

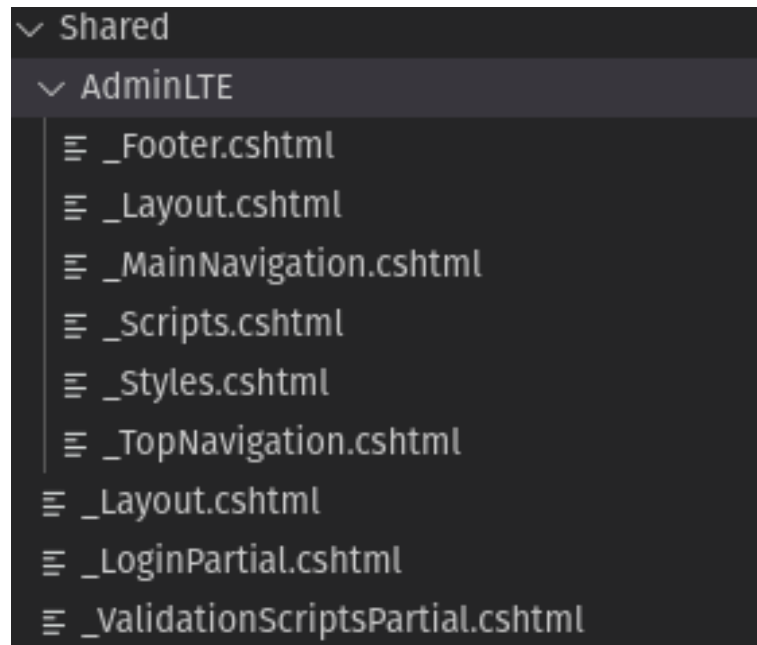


Figura 3.16: Layout AdminLTE  
Fuente: Elaborado por el investigador

A continuación se muestra el código utilizado en el archivo `_Layout` en el cual constan las llamadas a las vistas parciales que se encargan de diversas tareas, además de la etiqueta `@RenderBody()` encargada de especificar el lugar en el cual las paginas que ocupan dicho Layout puedan renderizar su contenido.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-
        scale=1"/>
    <meta http-equiv="x-ua-compatible" content="ie=edge" />
    <title>Fideos Victoria</title>
    <partial name="AdminLTE/_Styles" />
</head>
<body class="hold-transition sidebar-mini">
    <div class="wrapper">
    <partial name="AdminLTE/_TopNavigation" />
    <partial name="AdminLTE/_MainNavigation" />
    <div class="content-wrapper">
        <div class="content-header">
        </div>
        <div class="content">
```

```

        <div class="container-fluid">@RenderBody()</div>
    </div>
</div>
<aside class="control-sidebar control-sidebar-dark">
    <div class="p-3">
        <h5>
            @RenderSection("Titulo_Side", required: false)
        </h5>
        <p>
            @RenderSection("Contenido_Side", required: false)
        </p>
    </div>
</aside>
<partial name="AdminLTE/_Footer" />
</div> <partial name="AdminLTE/_Scripts" />
    @RenderSection("Scripts", required: false)
</body>
</html>

```

## Uso de Libman

Libman es un gestor multiplataforma de librerías Javascript, para hacer uso de esta herramienta primero tuvo que ser instalada usando el siguiente comando:

```
dotnet tool install -g Microsoft.Web.LibraryManager.Cli
```

Una vez instalado se inicializó el archivo libman.json y se procedió a instanciar las librerías que se utilizaron, todo esto se realizó por medio de los siguientes comandos:

```

libman init
libman install @microsoft/signalr@latest -p unpkg
    -d wwwroot/js/signalr --files dist/browser/signalr.js
    --files dist/browser/signalr.min.js

```

## Desarrollo de Iteraciones

El desarrollo del sistema de monitoreo esta dividido en 4 iteraciones en las cuales se implementan las historias de usuario comprendidas en cada iteración. A continuación se detalla el desarrollo de cada iteración junto con sus historias de usuario más relevantes.

### Iteración 1

Las historias de usuario fueron desarrolladas en base a los diseños y arquitectura descrita a continuación.

Tabla 3.40: Iteración 1

Iteración	Numero	Historia de usuario
1	1	Ingreso al sistema
1	2	Registro de usuarios

Fuente: Elaborado por el investigador

**Ingreso al sistema:** El ingreso al sistema consta del inicio de sesión por parte de un usuario cuya cuenta haya sido verificada. La interfaz utilizada para esta tarea se basa en la plantilla AdminLTE y consta de:

- Ingreso de email y contraseña
- Opción de registro
- Opción de recuperación de contraseña
- Opción de recordar la sesión abierta

Inicio de sesión

email

contraseña

Recuérdame

Ingresar

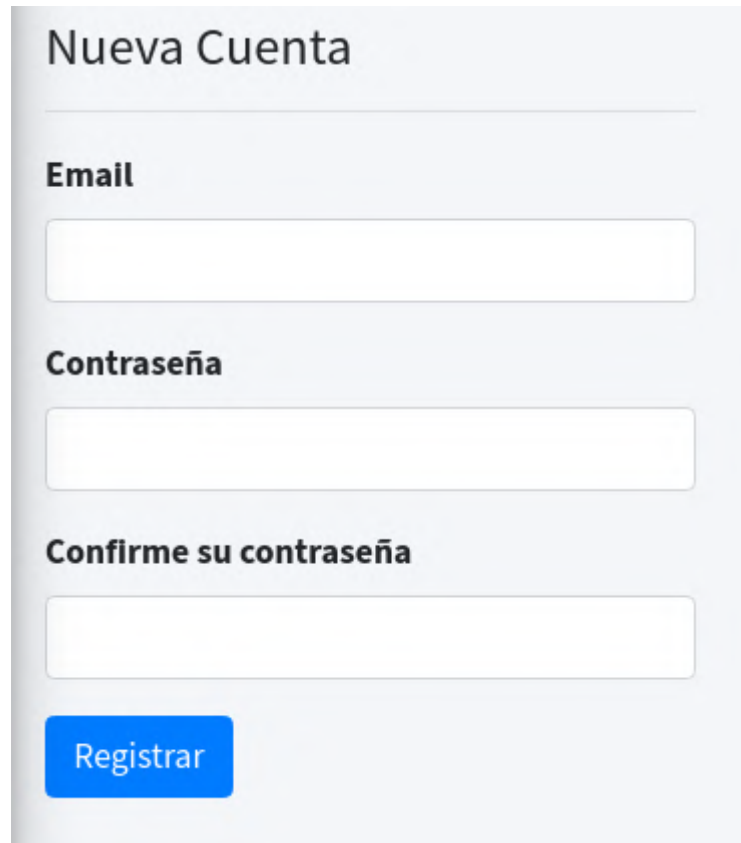
¿No tiene una cuenta?

Regístrate

>>Olvidé mi contraseña

Figura 3.17: Ingreso al sistema  
Fuente: Elaborado por el investigador

**Registro de usuarios:** El registro se basa en tomar un email y una contraseña que se almacena en la base de datos, su interfaz se observa en la figura 3.18. El email ingresado es único y usando Sendgrid se envía un código para la verificación de la cuenta por parte del administrador así evitando el registro de usuarios desconocidos.



El formulario de registro de usuarios, titulado "Nueva Cuenta", contiene tres campos de entrada de texto y un botón de acción. El primer campo está etiquetado como "Email", el segundo como "Contraseña" y el tercero como "Confirme su contraseña". El botón, que es azul y contiene el texto "Registrar", se encuentra debajo de los campos de entrada.

Figura 3.18: Registro de usuarios  
Fuente: Elaborado por el investigador

Los usuarios al momento de registrarse adquieren por defecto el rol de empleado, excepto un email que consta como administrador, por este motivo surgió la necesidad de implementar una interfaz por medio de la cual el administrador, en primera instancia existe solo uno, tenga la capacidad de modificar los roles de los usuarios registrados. Dicha interfaz consta de los roles que pueden ser seleccionados de acuerdo a lo que el administrador pretenda cambiar.

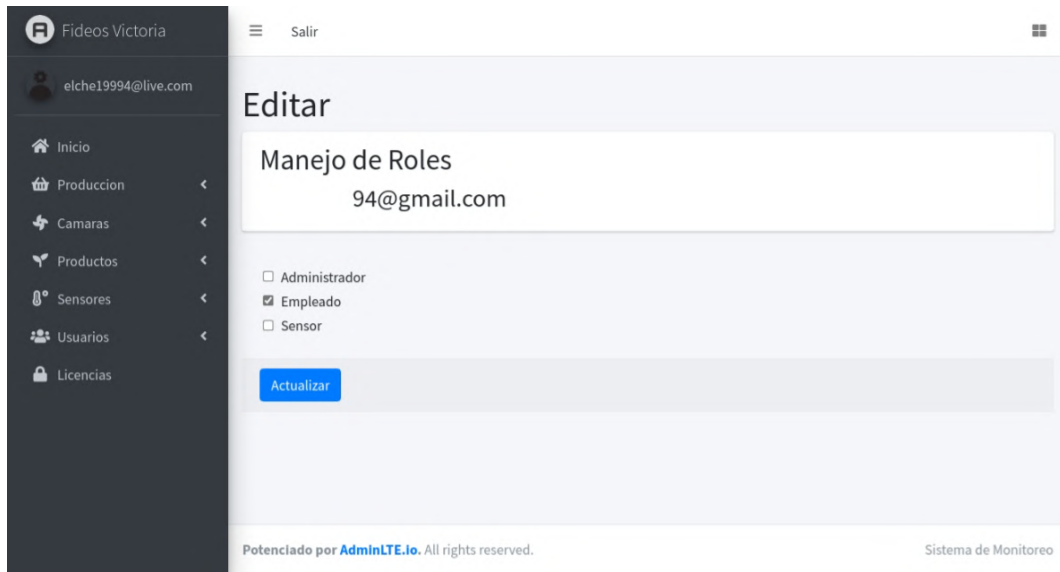


Figura 3.19: Manejo de roles  
Fuente: Elaborado por el investigador

## Iteración 2

La iteración 2 consta de la gestión de los módulos encargados del ingreso de datos, como son cámaras o sensores. Dependiendo de la acción, se procedió a usar el acceso por roles siendo el rol Administrador el que conlleva más responsabilidad. Estos datos son primordiales para el correcto funcionamiento del sistema, razón por la cual debieron ser creados con mucho cuidado y previas a que los dispositivos IOT empezaran la toma de datos, un ejemplo es el número de sensores que posee una cámara y afectan directamente al comportamiento del dashboard.

Tabla 3.41: Iteración 2

Iteración	Numero	Historia de usuario
2	3	Gestión de Cámaras
2	4	Gestión de Productos
2	5	Gestión de Sensores
2	7	Eliminación Producciones

Fuente: Elaborado por el investigador

**Gestión de Cámaras:** Una cámara representa, valga la redundancia, una cámara de secado. Un objeto de tipo cámara consta de una descripción así como de la capacidad media de quintales de producto que puede contener en su interior. Para el ingreso de datos por parte de dispositivos IOT se requiere conocer el id de la cámara, que puede o no coincidir con la descripción, en la que va a operar

por este motivo se permitió la visualización del id de las cámaras en el listado general de la figura 3.20.

Id	Descripción	Capacidad	
1	Cámara 1	36	<a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>
2	Cámara 2	40	<a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>

Figura 3.20: Listado de Cámaras  
Fuente: Elaborado por el investigador

**Gestión de Productos:** Los productos son entidades que funcionan como referencia para hacer comparaciones entre los datos recabados.

Nombre	Descripción	
p1	p1 prueba	<a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>
Regin Blanco	Regin enroscado blanco	<a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>
Regin Amarillo	Regin enroscado de color amarillo yemo	<a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>
Regin maquinado especial	Regin maquinado con huevo para enfundar	<a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>
Lazo gigante	Lazo gigante blanco	<a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>
Regin varios	Regin blanco y amarillo roscado	<a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>

Figura 3.21: Listado de Productos  
Fuente: Elaborado por el investigador

**Gestión de Sensores:** Por medio de interfaces se realizan las acciones de lectura, escritura y modificación de las entidades. En el listado de sensores de la figura 3.22 consta el id del sensor ya que dicho id consta en la configuración de los dispositivos IOT.

Id	Tipo	Marca	Ubicacion	Camara
1	dht22	asair	Sensor ubicado al lado de la puerta izquierda	Cámara 1 <a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>
2	dht22	asair	Sensor ubicado al lado de la puerta derecha	Cámara 1 <a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>
3	dht22	asair	Centro de la pared con puertas a una altura de 1.20 metros	Cámara 2 <a href="#">Editar</a>   <a href="#">Detalles</a>   <a href="#">Eliminar</a>

Figura 3.22: Listado de Sensores  
Fuente: Elaborado por el investigador

**Eliminación Producciones:** La eliminación de producciones corresponde un proceso que elimina una serie de datos relacionados, razón por la cual el proceso está unicamente habilitado para usuarios con rol Administrador.



Figura 3.23: Eliminación de producciones  
Fuente: Elaborado por el investigador



### Iteración 3

La iteración 3 consta de las historias de usuario más complejas y extensas abarcando el manejo de producciones, el registro de temperatura y la visualización de los datos capturados por los dispositivos IOT.

Tabla 3.42: Iteración 3

Iteración	Numero	Historia de usuario
3	6	Visualización, Creación y Modificación de Producciones
3	8	Registro de Temperatura
3	9	Visualización de la Temperatura

Fuente: Elaborado por el investigador

**Visualización, Creación y Modificación de Producciones:** En esta historia se procedió a la creación de las interfaces para como su nombre indica, visualizar, crear y modificar las producciones alojadas en el sistema.

La visualización de producciones consta de los puntos descritos a continuación y están visualmente dispuestos como se puede observar en la figura 3.24.

- **Nueva producción:** enlace a la pagina usada para crear producciones.
- **Fecha:** especificada al momento de crear la producción.
- **Activa:** indicador de actividad de la producción.
- **Observaciones:** información extra ingresada al crear o modificar la producción.
- **Acciones:** enlaces disponibles para editar o eliminar la producción además de la generación de reportes y visualización de los detalles de la producción.

Fecha	Activa	Observaciones	
17/3/2021	<input type="checkbox"/>	Produccion incompleta, fue generada para controlar como se comporta cuando se bloquea el caldero	<a href="#">Detalles</a>   <a href="#">Reporte</a>   <a href="#">Editar</a>   <a href="#">Eliminar</a>
16/3/2021	<input type="checkbox"/>	Cambiar los tiempos y cantidades. Lluvia desde las 3 de la tarde	<a href="#">Detalles</a>   <a href="#">Reporte</a>   <a href="#">Editar</a>   <a href="#">Eliminar</a>
15/3/2021	<input type="checkbox"/>	Día muy frío y la cámara 2 posee un retraso de 2 horas, debe modificarse la cantidad y los tiempos. El dispositivo de la cámara 2 se ha detenido 2 veces y se enciende el led rojo mientras titila el verde	<a href="#">Detalles</a>   <a href="#">Reporte</a>   <a href="#">Editar</a>   <a href="#">Eliminar</a>
8/3/2021	<input type="checkbox"/>	prueba con cambios finales, pero retrasada la camara 1 por 1 hora y la 2 por 3 horas	<a href="#">Detalles</a>   <a href="#">Reporte</a>   <a href="#">Editar</a>   <a href="#">Eliminar</a>
4/2/2021	<input type="checkbox"/>	Producción de prueba con 1 sola cámara y 2 horas de retraso Tuvo un retraso de 20 minutos desde que inició la toma de datos y en un momento cambio la hora a 0:00:00	<a href="#">Detalles</a>   <a href="#">Reporte</a>   <a href="#">Editar</a>   <a href="#">Eliminar</a>

Figura 3.24: Listado de Producciones  
Fuente: Elaborado por el investigador

Una de las limitantes de Razor Pages, es que, solo renderiza código HTML estático, en la pagina de creación de producciones en especifico se necesitó de un componente dinámico en el cual el usuario pueda agregar y eliminar el detalle en dependencia de si la producción utiliza una o más cámaras de secado. Para solventar esta limitante se utilizó el framework Alpine.js, que haciendo uso de llamados asíncronos cargan los datos necesarios para generar la interfaz en base a los productos y las cámaras que se encuentran ingresadas en el sistema. Una de las condiciones que se deben cumplir es que las cámaras no se repitan, para este fin se utilizó el método `pushProducto` detallado en la figura 3.25.

```

pushProducto() {
  let index = parseInt(this.productosLista[this.productosLista.length-1].camara);
  let _producto = { id: this.productosLista.length, camara: (index+1)+" " };
  //verifica que se puedan generar unicamente una produccionproducto por cada camara cada vez
  if (this.productosLista.length < this.camaras.length) {
    if(this.productosLista.some(
      producto => {return parseInt(producto.camara)==parseInt(_producto.camara);}
    ))
    {
      _producto.camara=_producto.camara-1;
    }
    this.productosLista.push(_producto);
  }
}

```

Figura 3.25: Método `pushProducto`  
Fuente: Elaborado por el investigador

Visualmente se organizó de la siguiente forma:

- **Fecha:** componente de tipo date que muestra por defecto la fecha actual.

- **Activa:** checkbox siempre marcada para que la producción tenga inicialmente el estado activo.
- **Observaciones:** espacio para que el usuario pueda añadir observaciones a la producción a crearse.
- **Nuevo:** botón utilizado para asignar un producto más al listado que se va a ingresar.
- **Cámara:** selector de la cámara que estará activa.
- **Producto:** selector del producto que contendrá la cámara.
- **Cantidad:** cantidad de producto en quintales de harina que contendrá la producción.
- **Tiempos:** configuración de tiempos para el extractor de humedad, calor y viento.

Figura 3.26: Creación de producciones  
Fuente: Elaborado por el investigador

Para mantener un diseño acorde y evitar confusiones, se decidió reutilizar parte de la interfaz para la creación de producciones para la modificación de las mismas, salvo las siguientes excepciones:

- El botón nuevo fue eliminado.
- Puede cambiarse el estado de activo a inactivo.

- La fecha no puede modificarse.

**Registro de Temperatura:** para el registro de temperatura se usaron dos dispositivos IOT de tipo NodeMCU junto a sensores de temperatura y humedad Dht22. Gracias a la capacidad de dichos dispositivos para conectarse en red, se procedió a la creación de una serie de controladores para que, haciendo uso de métodos Http, el dispositivo conozca si existe una producción activa y si dicho dispositivo debe o no enviar datos al servidor.

Los controladores que se usaron para la comunicación con los dispositivos IOT fueron `ProduccionController` y `TemperaturaController`, cuya funcionalidad se detalla a continuación.

### **ProduccionController**

Al momento de arrancar el proceso de captura de datos, el dispositivo IOT verifica si el id de la cámara al que pertenece se encuentra en la respuesta que obtiene del servidor, este proceso es manejado por el controlador `ProduccionController` mediante el código mostrado en la figura 3.27

```
public async Task<ActionResult<Produccion>> GetProduccionActiva()
{
    var produccion = _context.Produccion
        .Include(x => x.ProduccionProductos)
        .Where(x => x.Activa == true)
        .OrderBy(y => y.Fecha);
    if (!produccion.Any())
    {
        //el cliente debe capturar el erro 404 y tratarlo
        return NotFound();
    }
    return await produccion.LastAsync();
}
```

Figura 3.27: Búsqueda de la producción activa  
Fuente: Elaborado por el investigador

### **TemperaturaController**

El proceso de subida de datos además de registrar los datos enviados por el dispositivo, realiza tareas para verificar la validez de los datos para finalmente, haciendo uso de SignalR, notificar a todos los clientes que se encuentren conectados de los nuevos valores recibidos. Ver figura 3.28.

```

_context.Temperatura.AddRange(temperaturas);
var save = await _context.SaveChangesAsync();

var produccionProducto = await _context.ProduccionProducto.AsNoTracking()
.Where(x => x.Id == temperaturas.FirstOrDefault().IdProduccionProducto)
.FirstOrDefaultAsync();

produccionProducto.Temperaturas = temperaturas.OrderBy(x => x.IdSensor).ToList();
if (save > 0)
{
    await _hub.Clients.All.SendAsync("EnvioProductoProduccion", produccionProducto);
}

```

Figura 3.28: Notificación vía SignalR  
Fuente: Elaborado por el investigador

Inicialmente se planteo que los dispositivos NodeMCU manejaran la hora para el envío de temperaturas, pero debido a su limitada capacidad, 64KB de SRAM y un núcleo que funciona a 80MHz, provocó en pruebas iniciales que su reloj interno tuviera un desfase de varios minutos y llegado a un tiempo de uso colapsaba e indicaba la hora por defecto 0:00:00. Para solventar dicho problema se recurrió a la modificación de un segmento de código en el controlador TemperaturaController, el cual modifica la hora en los datos de tipo temperatura recibidos por la hora del servidor. Ver figura 3.29.

```

temperaturas.ForEach(
    temperatura =>
    {
        DateTime timeUtc = DateTime.UtcNow;
        TimeZoneInfo cstZone = TimeZoneInfo.FindSystemTimeZoneById("America/Guayaquil");
        DateTime cstTime = TimeZoneInfo.ConvertTimeFromUtc(timeUtc, cstZone);
        temperatura.Hora = cstTime.TimeOfDay;
    }
);

```

Figura 3.29: Modificación de la hora en TemperaturaController  
Fuente: Elaborado por el investigador

Una vez los controladores estuvieron terminados se procedió a la programación de los dispositivos IOT NodeMCU. Haciendo uso del lenguaje de programación Micropython, se crearon las clases y funciones necesarias para el correcto funcionamiento de los dispositivos.

### ConnectWifi

Este módulo provee la funcionalidad para que el dispositivo se conecte a la red local haciendo uso de las credenciales necesarias, SSID y contraseña, así como la asignación de una dirección de red estática y la muestra del estado por medio del led integrado en el dispositivo.

```

def connect():
    import network
    from machine import Pin
    from time import sleep
    from Objetos import Configuracion
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.scan()
    if not wlan.isconnected():
        print('conectando ... ')
        wlan.connect('***', '***')
        wlan.ifconfig(('192.168.1.*', '255.255.255.192', '192.168.1.1', '192.168.1.1'))
        while not wlan.isconnected():
            Configuracion.LEDS.get('red').off()
            sleep(5)
            Configuracion.LEDS.get('red').on()
            sleep(5)
            pass
    print('Ya esta conectado: '+ wlan.ifconfig()[0])
    Configuracion.LEDS.get('red').off()

```

Figura 3.30: Conexión a wifi NodeMCU  
Fuente: Elaborado por el investigador

## Objetos

Este módulo contiene las clases que utiliza el dispositivo para manipular los datos con la estructura correcta, además posee la clase Configuración en la cual se especifican los sensores que maneja, los leds que indican el estado y el id de la cámara en la que funcionará el dispositivo

```

class Configuracion():
    from machine import Pin
    SENSORES = [Sensor(1,14), Sensor(2,4)]
    LEDES = {'error1':Pin(5,Pin.OUT), 'toma':Pin(12,Pin.OUT), 'red':Pin(2,Pin.OUT)}
    ID_CAMARA = 1 #este id debe coincidir con el id de la base de datos

```

Figura 3.31: Clase configuración  
Fuente: Elaborado por el investigador

## Main

Este modulo forma parte de los módulos del sistema y siempre es ejecutado después del arranque del dispositivo. Su programación consta de los siguientes pasos:

1. Importación de las librerías a utilizar.
2. Conexión a la red Wifi.
3. Revisión de la producción activa.
4. Verificación del id de cámara.

5. Envío de datos al servidor.
6. Muestra de estado mediante leds.
7. Reinicio del dispositivo en caso superar el numero de errores configurado.

Los dispositivos NodeMCU funcionan internamente con un chip esp8266 el cual posee un único núcleo haciendo imposible que varias tareas se ejecuten al mismo tiempo, pero los desarrolladores de Micropython, tomando como base la librería asincio de python, crearon la librería uasyncio la cual permite crear funciones asíncronas que, aún sin poseer varios hilos de ejecución, permite ejecutar varias acciones al mismo tiempo. Haciendo uso de la librería uasyncio se crearon las funciones para el envío de datos y la muestra del estado del dispositivo usando leds. Ver figura 3.32.

```
async def errorPersistente(led):
    while True:
        Configuracion.LEDS.get(led).on()
        await uasyncio.sleep(1)
        Configuracion.LEDS.get(led).off()
        await uasyncio.sleep(1)
```

Figura 3.32: Función asíncrona Micropython  
Fuente: Elaborado por el investigador

**Visualización de la Temperatura:** Esta historia de usuario comprende el diseño del dashboard además de la comunicación entre el servidor y los clientes para la muestra de los datos recibidos en tiempo real.

El dashboard consta de tres secciones claramente definidas y que representan información sobre la producción que se encuentra activa. La primera sección se basa en un componente de tipo box en el que consta el número de producción, la fecha y un enlace directo a la pagina de detalles de la producción actual si existe una producción activa, caso contrario el componente solo contiene una alerta para notificar al usuario y un enlace al listado de producciones. Ver figura 3.33.

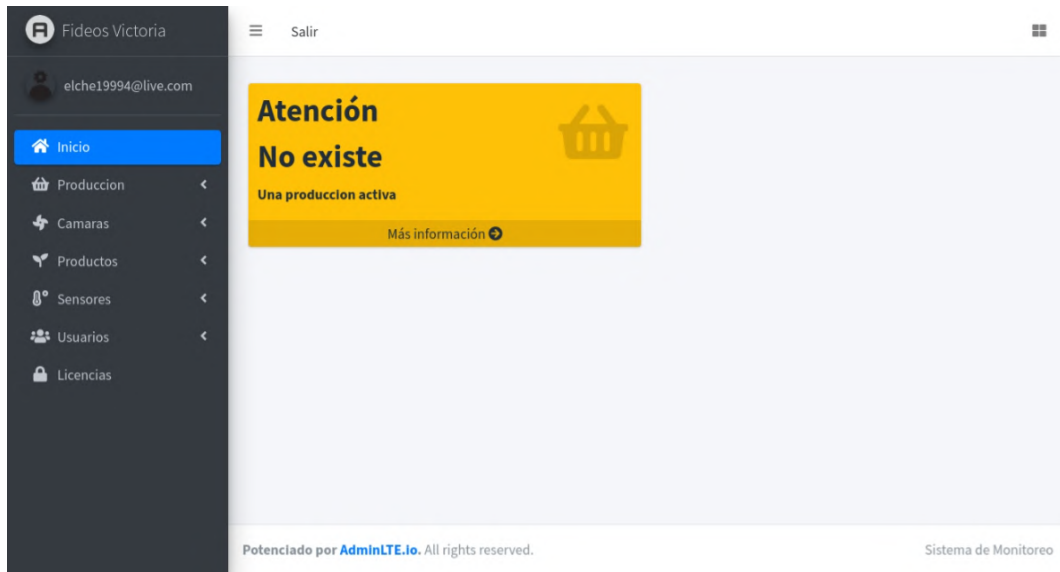


Figura 3.33: Dashboard(Producción inactiva)  
Fuente: Elaborado por el investigador

La segunda sección se basa en un elemento de tipo card combinado con un tab-pane para aprovechar el diseño y funcionalidad de ambos componentes. El componente tab-pane posee la capacidad de almacenar en su interior diferentes elementos y navegar entre ellos por medio de enlaces, en este caso utiliza botones con el identificador de la cámara que contiene. Internamente cuenta con un elemento info-box por cada sensor, estos elementos cuentan con un identificador único para que al momento de recibir datos en tiempo real, estos actualicen su contenido brindando así una perspectiva detallada del estado actual de la cámara. Cada cámara cuenta con dos gráficos, cada uno detalla los últimos 10 datos registrados por cada sensor, dichos gráficos fueron construidos en base a la librería Chart.js ya que ofrece características afines al sistema de monitoreo, como es el caso de adaptarse a distintos tamaños de pantalla, ser actualizables bajo demanda y altamente personalizables. Finalmente, esta sección, posee un enlace directo para finalizar la producción actual sin la necesidad de abrir otra página. Ver figura 3.34.



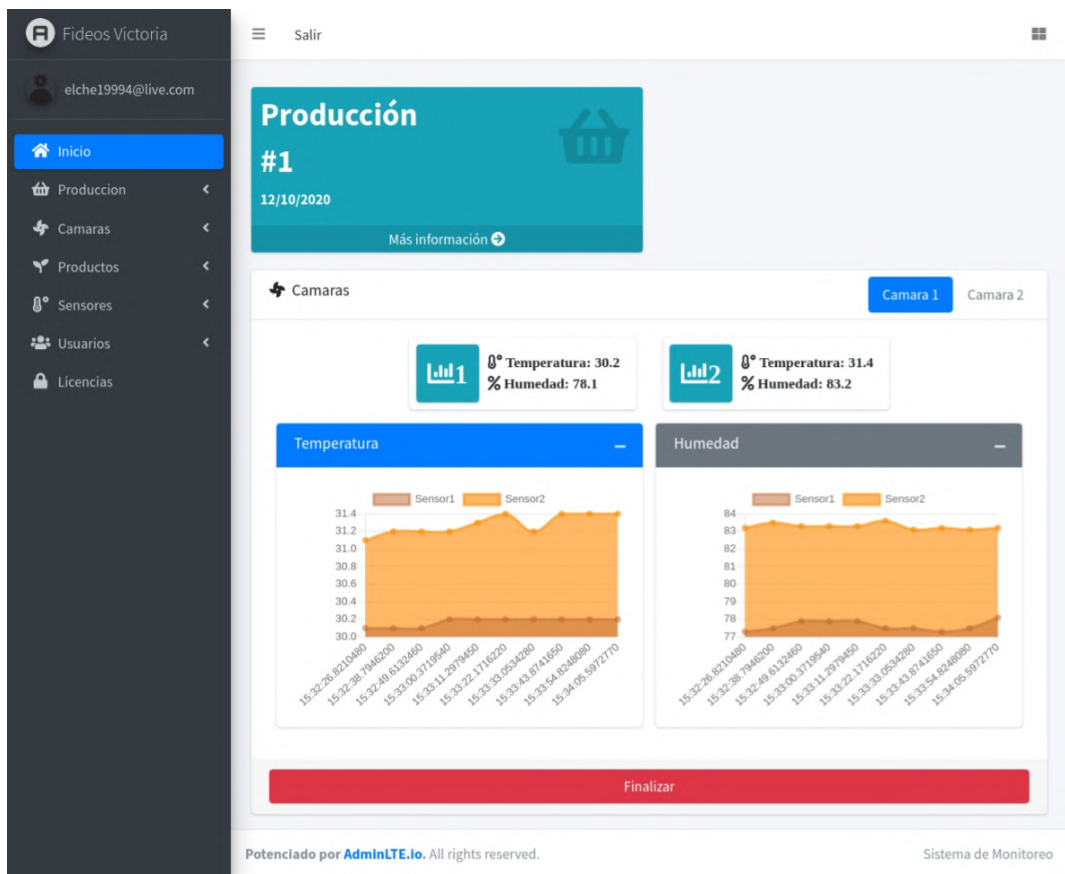


Figura 3.34: Dashboard  
Fuente: Elaborado por el investigador

La tercera sección funciona en base al botón situado en la esquina superior derecha, dicho botón despliega un componente oculto que contiene información extra sobre la producción actual. Este componente no es exclusivo del dashboard ya que funciona por medio de una vista parcial en la cual cualquier pagina puede generar su propia funcionalidad extra.

SignalR puede ser configurado en base a las necesidades del proyecto, en este caso fue configurado especificando la url del hub y además se estableció el parámetro withAutomaticReconnect el cual permite que si por algún motivo la conexión se cierra, SignalR intentará volver a conectarse ya sea usando el mismo medio de comunicación o cambiando a otro sin que el usuario se deba preocupar en hacer nada.

```
var connection = new signalR.HubConnectionBuilder().withUrl("/hub")
    .withAutomaticReconnect().build();
```

Figura 3.35: Configuración cliente SignalR  
Fuente: Elaborado por el investigador

Para la carga inicial de datos en los gráficos se realiza una consulta a un método alojado en la misma pagina, pero expuesto para que Javascript pueda hacer uso del mismo. Este método realiza la búsqueda de una producción activa para luego de validar si existe y generar un objeto Json con los datos encontrados o enviar un código de error 404. En caso de existir datos de tipo temperatura, el método se encarga de tomar los 10 últimos de cada sensor evitando así que los gráficos se vean saturados de información. Ver figura 3.36.

```

public async Task<JsonResult> OnGetJsonRAsync()
{
    var produccionActiva = await _context.Produccion
        .Include(x => x.ProduccionProductos)
        .ThenInclude(x => x.Temperaturas)
        .Include(x => x.ProduccionProductos)
        .Where(x => x.Activa).OrderByDescending(x => x.Fecha).FirstOrDefaultAsync();
    if (produccionActiva == null)
    {
        return new JsonResult(StatusCode(404));
    }
    else if (!produccionActiva.Activa)
    {
        return new JsonResult(StatusCode(404));
    }
    else
    {
        int valorEstandar = 10;
        produccionActiva.ProduccionProductos.ForEach(x =>
        {
            int numeroSensores = 1;
            numeroSensores = _context.Sensor.AsNoTracking().Where(y => y.IdCamara == x.IdCamara).Count();
            if (x.Temperaturas.Count >= numeroSensores * valorEstandar)
            {
                x.Temperaturas = x.Temperaturas.TakeLast(numeroSensores * valorEstandar).ToList();
            }
        });
        return new JsonResult(produccionActiva);
    }
}

```

Figura 3.36: Búsqueda de producción activa  
Fuente: Elaborado por el investigador

#### Iteración 4

La iteración 4 consta de la generación de reportes sobre las producciones almacenadas en el sistema así como la exportación de datos para su análisis con herramientas externas como Excel.

Tabla 3.43: Iteración 4

Iteración	Numero	Historia de usuario
4	10	Reportes de temperaturas
4	11	Exportación de datos

Fuente: Elaborado por el investigador

**Reportes de temperaturas:** Los reportes son de especial utilidad en la toma de decisiones ya que permiten visualizar la forma en la cual se comportó el proceso de secado.

Por medio de una interfaz gráfica, que posee una serie de elementos para la selección de datos, el usuario puede observar las mediciones a través de los gráficos específicos de cada cámara.

Para dicho fin la interfaz consta de las siguientes secciones:

- **Cabecera:** Consta del id de la producción, fecha y observaciones cargadas desde el servidor.
- **Navegación:** Por medio de botones se permite la navegación entre cámaras para así tener una interfaz mejor organizada y fácil de observar.
- **Selector:** Haciendo uso de elementos de tipo checkbox y un botón, se muestran los datos que el usuario puede elegir para modificar el gráfico y obtener una visualización acorde a sus necesidades.
- **Gráfica:** Muestra los datos elegidos por el usuario como una gráfica de tipo lineal, además de un botón por medio del cual el usuario puede guardar como imagen el gráfico actual.
- **Datos del sensor:** Indica los datos mayores y menores, tanto en temperatura como humedad, capturados por los sensores.
- **Exportar datos:** Despliega una ventana para que el usuario elija los datos a exportar. Ver figura 3.39.

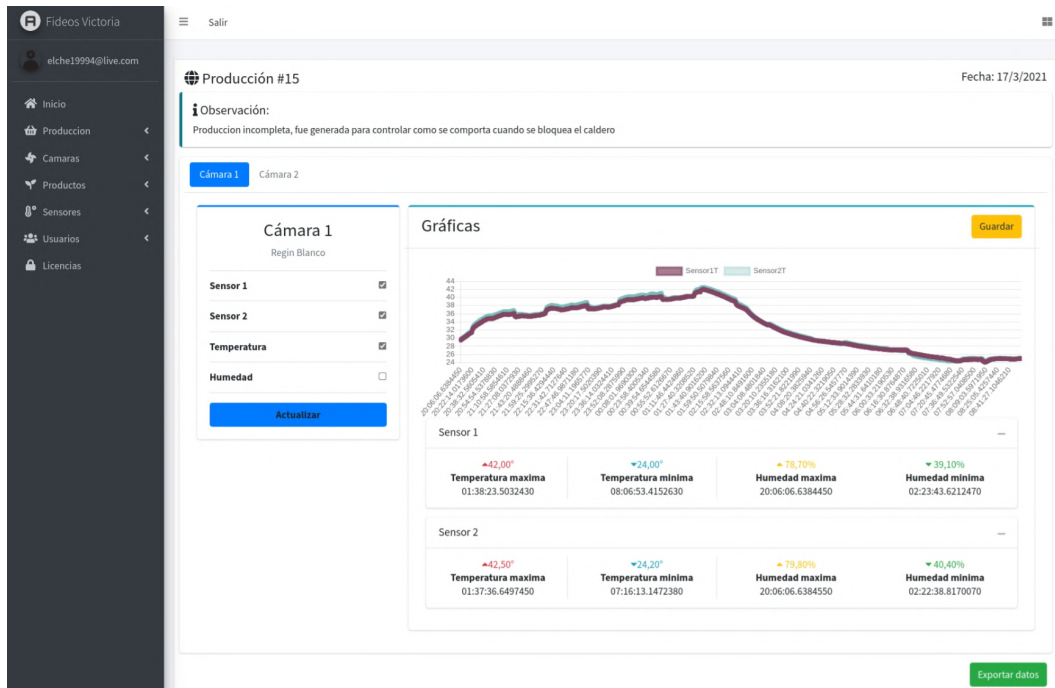


Figura 3.37: Reporte de temperaturas  
Fuente: Elaborado por el investigador

Los datos sobre temperatura y humedad presentadas se basaron en una clase alojada dentro de la misma página, en la cual haciendo uso de consultas a la base de datos se obtuvieron los valores buscados. El proceso de filtrado consta de la búsqueda de la producción actual incluyendo sus datos relacionados para, después de validar que exista la producción, recorrer los datos por medio de ciclos anidados. Finalmente, con los datos de cada ciclo se procede a la búsqueda, por medio del ordenamiento de los datos de tipo temperatura, de los máximos y mínimos por cada iteración. Ver Figura 3.38.

```

async Task<Estadisticas> CalcularValores(int _idSensor, int _idProduccionProducto)
{
    Models.Temperatura maxima = await _context.Temperatura
        .Where(x => (x.IdSensor == _idSensor && x.IdProduccionProducto == _idProduccionProducto))
        .OrderByDescending(x => x.TemperaturaCamara).FirstOrDefaultAsync();
    Models.Temperatura minima = await _context.Temperatura
        .Where(x => (x.IdSensor == _idSensor && x.IdProduccionProducto == _idProduccionProducto))
        .OrderBy(x => x.TemperaturaCamara).FirstOrDefaultAsync();
    Models.Temperatura humedad = await _context.Temperatura
        .Where(x => (x.IdSensor == _idSensor && x.IdProduccionProducto == _idProduccionProducto))
        .OrderBy(x => x.Humedad).FirstOrDefaultAsync();
    Models.Temperatura humedadMaxima = await _context.Temperatura
        .Where(x => (x.IdSensor == _idSensor && x.IdProduccionProducto == _idProduccionProducto))
        .OrderByDescending(x => x.Humedad).FirstOrDefaultAsync();
    var Estadisticas = new Estadisticas()
    {
        humedadMinima = humedad,
        temperaturaMaxima = maxima,
        humedadMaxima = humedadMaxima,
        idSensor = _idSensor,
        temperaturaMinima = minima
    };
    return Estadisticas;
}

```

Figura 3.38: Búsqueda de máximos y mínimos por sensor  
Fuente: Elaborado por el investigador

**Exportación de datos:** En la página de reportes en la esquina inferior derecha aparece el botón utilizado para exportar los datos que el usuario requiera, sobre la producción en la que se encuentra. Para este fin se diseñó una interfaz que funciona como una ventana modal para así evitar que el usuario tenga que salir de la página de reportes si desea exportar los datos que se encuentra visualizando. La interfaz consta de las siguientes secciones:

- **Cabecera:** muestra el título del modal.
- **Cuerpo:** consta de las cámaras que se encuentran en la producción además de los checkbox para seleccionar temperatura y/o humedad.
- **Pie de página:** consta de los botones para seleccionar o deseleccionar todos los checkbox, cerrar el modal y finalmente para exportar los datos elegidos.

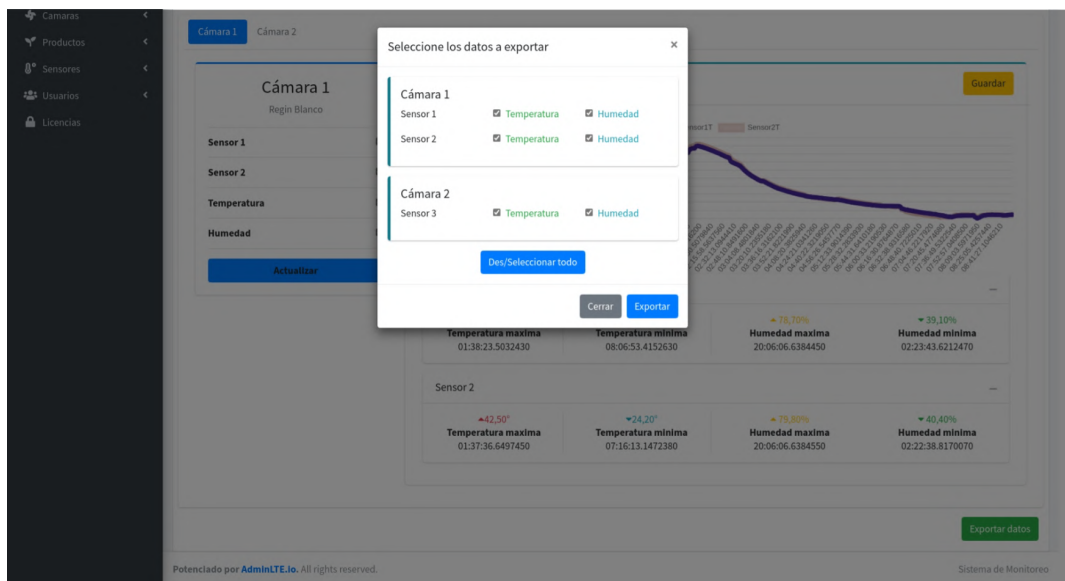


Figura 3.39: Exportación de datos  
Fuente: Elaborado por el investigador

El proceso de exportación de datos consta de selectores por tipo de elemento y una búsqueda intensiva en los datos proporcionados por el servidor que da como resultado la generación de un documento de tipo CSV. Se optó por el formato CSV ya que al no requerir el uso de archivos especiales, como XSLX de Excel, puede generarse de manera sencilla y eficiente. Otra ventaja del uso de CSV es el tamaño del archivo generado, esto es especialmente útil al exportar una cantidad elevada de datos como es el caso del sistema de monitoreo, en el cual tomando una media de cinco mil datos por sensor genera un archivo de apenas 72.3 Kb.

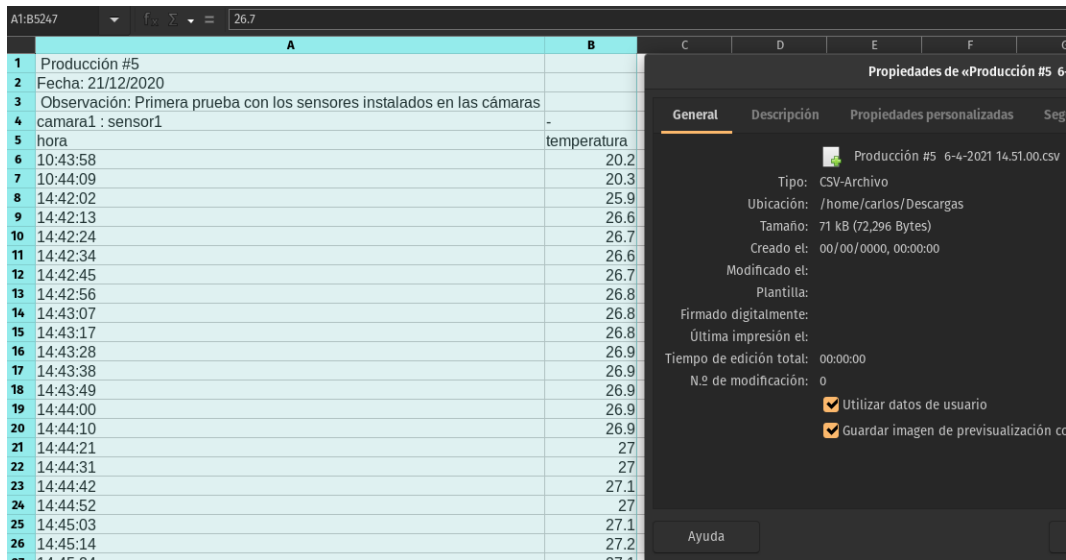


Figura 3.40: Datos exportados  
Fuente: Elaborado por el investigador

### 3.1.4.5. Pruebas

Esta fase de la metodología XP tiene como objetivo comprobar el correcto funcionamiento de la aplicación, siendo dividida en dos tipos de pruebas: unitarias y de aceptación.

#### Pruebas Unitarias

Las pruebas unitarias son parte fundamental de la metodología XP ya que sirven para comprobar el correcto funcionamiento del sistema.

Las pruebas unitarias se realizaron tomando como base el marco xUnit, ya que su aplicación no requiere de conocimientos demasiado específicos del marco además de su extensa documentación y comunidad.

Primeramente se creó un proyecto con el nombre `MonitoreoVictoria.Tests` por medio de la plantilla predefinida para xUnit definida en la herramienta de línea de comandos.

```
dotnet new xunit --o MonitoreoFideosVictoria.Tests
```

Ya que las pruebas unitarias no deben interferir con los datos almacenados en la base de datos, se utilizaron dos técnicas diferentes para simular el acceso a datos y llamadas a otros servicios que originalmente utilizan el patrón Dependency Injection (inyección de dependencias o DI) implementado por defecto en .Net para manejar el acceso a Entity Framework Core por ejemplo, dichas técnicas fueron:

- **Base de datos en memoria:** Entity Framework Core puede usar como proveedor de base de datos la memoria del sistema brindando la capacidad

de probar la aplicación sin la necesidad de crear una base de datos real únicamente para pruebas.

- **Mocking:** Dicha técnica se basa en la «imitación» de entidades y sus comportamientos simulando así el llamado a un servicio externo por ejemplo, para esto se utilizó el marco Moq provisto como un paquete nuget.

### Casos de pruebas

Las pruebas unitarias se orientaron a la comprobación, principalmente, de los puntos de conexión expuestos como una API, siendo detallados a continuación.

Tabla 3.44: Casos de pruebas unitarias

Método	Caso de prueba
Envío de email	Validar que el método SendEmailAsync se ejecute correctamente
Búsqueda de cámaras	Verificar el correcto funcionamiento de los métodos para la búsqueda de cámaras
Devolución de la producción activa	Comprobar que el método devuelve los valores correctos con y sin una producción activa
Búsqueda de productos	Verificar el correcto funcionamiento de los métodos para la búsqueda de productos
Envío de temperaturas	Comprobar que no se permita guardar si no existe una producción activa
	Comprobar que se guardan correctamente un listado de temperaturas

Fuente: Elaborado por el investigador

En los casos descritos en la tabla 3.44 se comprobó su correcta ejecución por medio de la extensión .NET Core Test Explorer, la cual brinda una alternativa Open Source a la herramienta instalada por defecto en Visual Studio y que funciona correctamente en Visual Studio Code para Linux. Además de permitir la ejecución de los test creados, .NET Core Test Explorer genera una salida visual del resultado de la ejecución de cada prueba como se puede ver en la figura 3.41.



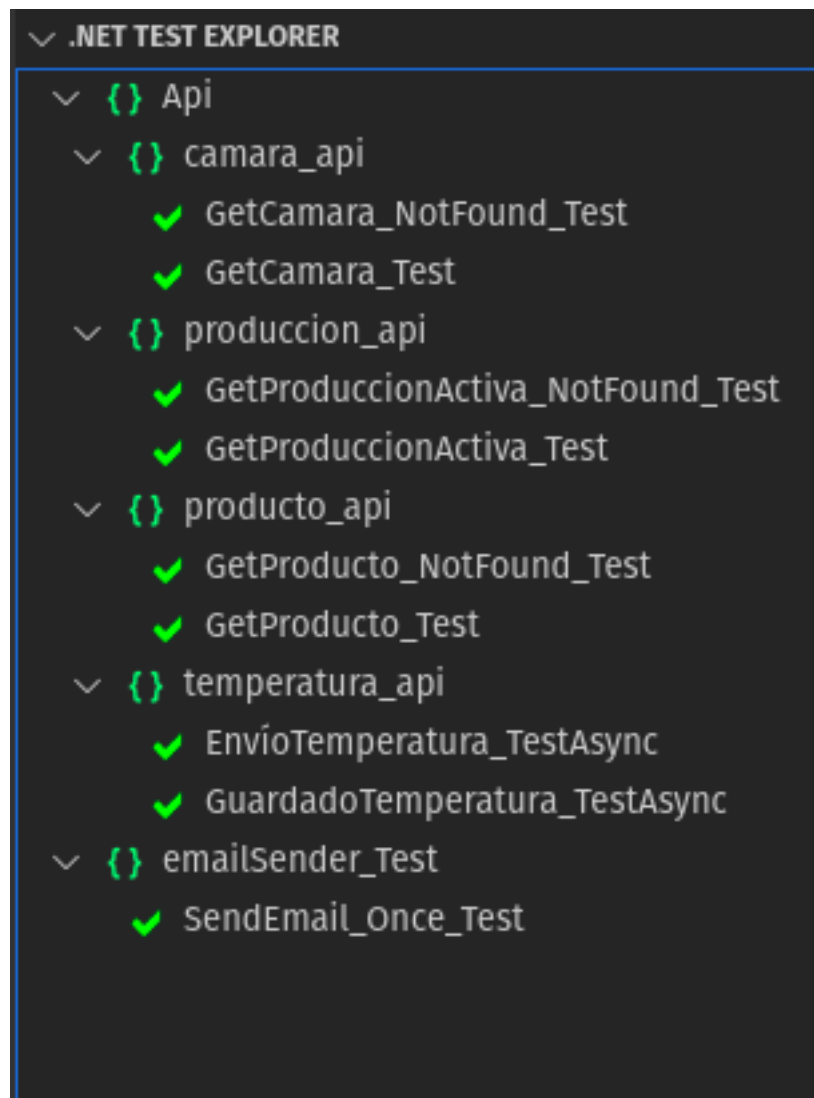


Figura 3.41: Comprobación de pruebas unitarias  
Fuente: Elaborado por el investigador

### Pruebas de aceptación

Las pruebas de aceptación comprueban que la funcionalidad especificada cumpla con los requerimientos que se recabaron en la fase de toma de requerimientos. Para la realización de las pruebas se usaron tarjetas de caso de prueba de aceptación.

Tabla 3.45: Prueba de aceptación: ingreso de credenciales

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC1	<b>Código historia:</b> H1 Ingreso al sistema
<b>Nombre:</b> Ingreso de credenciales	
<b>Condición de ejecución:</b> -Sesión de usuario no iniciada	
<b>Proceso de ejecución:</b> -El usuario ingresa sus credenciales	
<b>Resultado:</b> El usuario es redirigido a la pagina principal	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.46: Prueba de aceptación: registro por parte del usuario

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC2	<b>Código historia:</b> H2 Registro de usuarios
<b>Nombre:</b> Registro por parte del usuario	
<b>Condición de ejecución:</b> -Usuario sin registro previo	
<b>Proceso de ejecución:</b> -El nuevo usuario ingresa sus datos en el formulario	
<b>Resultado:</b> Un enlace de verificación es enviado al administrador, el nuevo usuario es redirigido a la pagina de ingreso	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.47: Prueba de aceptación: creación de cámaras

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC3	<b>Código historia:</b> H3 Gestión de cámaras
<b>Nombre:</b> Creación de cámaras	
<b>Condición de ejecución:</b> -Un usuario administrador debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción Nueva del menú cámaras -El usuario ingresa los datos correspondientes -El usuario selecciona guardar	
<b>Resultado:</b> El usuario es redirigido al listado de cámaras mostrando la cámara registrada anteriormente.	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.48: Prueba de aceptación: edición de cámaras

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC4	<b>Código historia:</b> H3 Gestión de cámaras
<b>Nombre:</b> Edición de cámaras	
<b>Condición de ejecución:</b> -Un usuario administrador debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú cámaras -El usuario elije la opción editar de la cámara a editarse -El usuario es redirigido a una ventana para ingreso de datos -El usuario ingresa los datos correspondientes -El usuario selecciona guardar	
<b>Resultado:</b> El usuario es redirigido al listado de cámaras mostrando la cámara editada anteriormente.	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.49: Prueba de aceptación: eliminación de cámaras

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC5	<b>Código historia:</b> H3 Gestión de cámaras
<b>Nombre:</b> Eliminación de cámaras	
<b>Condición de ejecución:</b> -Un usuario administrador debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú cámaras -El usuario elije la opción eliminar de la cámara a eliminarse -El usuario es redirigido a una ventana de confirmación -El usuario selecciona eliminar	
<b>Resultado:</b> El usuario es redirigido al listado de cámaras constatando la eliminación de la cámara	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.50: Prueba de aceptación: detalles por cámara

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC6	<b>Código historia:</b> H3 Gestión de cámaras
<b>Nombre:</b> Detalles por cámara	
<b>Condición de ejecución:</b> -Un usuario debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú cámaras -El usuario elije la opción detalles de una cámara	
<b>Resultado:</b> El usuario es redirigido a una ventana con la información sobre la cámara elegida	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.51: Prueba de aceptación: creación de productos

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC7	<b>Código historia:</b> H4 Gestión de productos
<b>Nombre:</b> Creación de productos	
<b>Condición de ejecución:</b> -Un usuario administrador debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción Nueva del menú productos -El usuario ingresa los datos correspondientes -El usuario selecciona guardar	
<b>Resultado:</b> El usuario es redirigido al listado de productos mostrando el producto registrado anteriormente.	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.52: Prueba de aceptación: edición de productos

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC8	<b>Código historia:</b> H4 Gestión de productos
<b>Nombre:</b> Edición de productos	
<b>Condición de ejecución:</b> -Un usuario administrador debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú productos -El usuario elije la opción editar del producto a editarse -El usuario es redirigido a una ventana para ingreso de datos -El usuario ingresa los datos correspondientes -El usuario selecciona guardar	
<b>Resultado:</b> El usuario es redirigido al listado de productos mostrando el producto editado anteriormente.	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.53: Prueba de aceptación: eliminación de productos

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC9	<b>Código historia:</b> H4 Gestión de productos
<b>Nombre:</b> Eliminación de productos	
<b>Condición de ejecución:</b> -Un usuario administrador debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú productos -El usuario elije la opción eliminar del producto a eliminarse -El usuario es redirigido a una ventana de confirmación -El usuario selecciona eliminar	
<b>Resultado:</b> El usuario es redirigido al listado de productos constatando la eliminación del producto	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.54: Prueba de aceptación: detalles por producto

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC10	<b>Código historia:</b> H4 Gestión de productos
<b>Nombre:</b> Detalles por producto	
<b>Condición de ejecución:</b> -Un usuario debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú productos -El usuario elije la opción detalles de un producto	
<b>Resultado:</b> El usuario es redirigido a una ventana con la información sobre el producto elegido	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.55: Prueba de aceptación: creación de sensores

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC11	<b>Código historia:</b> H5 Gestión de sensores
<b>Nombre:</b> Creación de sensores	
<b>Condición de ejecución:</b> -Un usuario administrador debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción Nueva del menú sensores -El usuario ingresa los datos correspondientes -El usuario selecciona guardar	
<b>Resultado:</b> El usuario es redirigido al listado de sensores mostrando el sensor registrado anteriormente.	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.56: Prueba de aceptación: edición de sensores

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC12	<b>Código historia:</b> H5 Gestión de sensores
<b>Nombre:</b> Edición de sensores	
<b>Condición de ejecución:</b> -Un usuario administrador debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú sensores -El usuario elije la opción editar del sensor a editarse -El usuario es redirigido a una ventana para ingreso de datos -El usuario ingresa los datos correspondientes -El usuario selecciona guardar	
<b>Resultado:</b> El usuario es redirigido al listado de sensores mostrando el sensor editado anteriormente.	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.57: Prueba de aceptación: eliminación de sensores

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC13	<b>Código historia:</b> H5 Gestión de sensores
<b>Nombre:</b> Eliminación de sensores	
<b>Condición de ejecución:</b> -Un usuario administrador debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú sensores -El usuario elije la opción eliminar del sensor a eliminarse -El usuario es redirigido a una ventana de confirmación -El usuario selecciona eliminar	
<b>Resultado:</b> El usuario es redirigido al listado de sensores constatando la eliminación del sensor	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.58: Prueba de aceptación: detalles por sensor

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC14	<b>Código historia:</b> H5 Gestión de sensores
<b>Nombre:</b> Detalles por sensor	
<b>Condición de ejecución:</b> -Un usuario debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú sensores -El usuario elije la opción detalles de un sensor	
<b>Resultado:</b> El usuario es redirigido a una ventana con la información sobre el sensor elegido	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.59: Prueba de aceptación: creación de producciones

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC15	<b>Código historia:</b> H6 Gestión de producciones
<b>Nombre:</b> Creación de producciones	
<b>Condición de ejecución:</b> -Un usuario debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción Nueva del menú producción -El usuario crea un registro nuevo si se va a utilizar más de una cámara -El usuario ingresa los datos correspondientes -El usuario selecciona crear	
<b>Resultado:</b> El usuario es redirigido al listado de producciones mostrando la producción registrada	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador



Tabla 3.60: Prueba de aceptación: edición de producciones

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC16	<b>Código historia:</b> H6 Gestión de producciones
<b>Nombre:</b> Edición de producciones	
<b>Condición de ejecución:</b> -Un usuario debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú producción -El usuario elije la opción editar de la producción a editarse -El usuario es redirigido a una ventana para ingreso de datos -El usuario modifica los datos correspondientes -El usuario selecciona guardar	
<b>Resultado:</b> El usuario es redirigido al listado de producciones mostrando la producción editada anteriormente	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.61: Prueba de aceptación: detalles por producción

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC17	<b>Código historia:</b> H6 Gestión de producciones
<b>Nombre:</b> Detalles por producción	
<b>Condición de ejecución:</b> -Un usuario debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú producción -El usuario elije la opción detalles de una producción	
<b>Resultado:</b> El usuario es redirigido a una ventana con la información sobre la producción elegida	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.62: Prueba de aceptación: eliminación de producciones

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC18	<b>Código historia:</b> H7 Eliminación de producciones
<b>Nombre:</b> Eliminación de producciones	
<b>Condición de ejecución:</b> -Un usuario administrador debe haber iniciado sesión	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú producción -El usuario elije la opción eliminar de la producción a eliminarse -El usuario es redirigido a una ventana de confirmación -El usuario selecciona eliminar	
<b>Resultado:</b> El usuario es redirigido al listado de producciones constatando la eliminación de la producción	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.63: Prueba de aceptación: envío de temperaturas del dispositivo IOT

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC19	<b>Código historia:</b> H8 Registro de temperatura
<b>Nombre:</b> Envío de temperaturas del dispositivo IOT	
<b>Condición de ejecución:</b> -El dispositivo IOT debe estar instalado y configurado correctamente	
<b>Proceso de ejecución:</b> -El dispositivo es encendido -Se enciende el led azul -El led verde se enciende mientras captura los datos requeridos -El led verde se apaga al enviar los datos al servidor -El proceso se repite mientras el proceso de secado esté activo	
<b>Resultado:</b> Los datos se ven reflejados en el dashboard	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.64: Prueba de aceptación: visualización de datos en tiempo real

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC20	<b>Código historia:</b> H9 Visualización de la temperatura
<b>Nombre:</b> Visualización de datos en tiempo real	
<b>Condición de ejecución:</b> -Un usuario debe haber iniciado sesión -Una producción debe estar activa -El dispositivo IOT debe estar encendido	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción inicio	
<b>Resultado:</b> El usuario visualiza como los gráficos de temperatura y humedad se actualizan automáticamente mientras llegan datos	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.65: Prueba de aceptación: visualización de datos por producción

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC21	<b>Código historia:</b> H10 Reportes de temperaturas
<b>Nombre:</b> Visualización de datos por producción	
<b>Condición de ejecución:</b> -Un usuario debe haber iniciado sesión -La producción debe haber capturado datos, caso contrario los gráficos aparecerán en blanco	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú producción -El usuario elije la opción reporte de la producción a visualizar -El usuario es redirigido a una ventana en la que constan los datos capturados en dicha producción	
<b>Resultado:</b> El usuario modifica los parámetros para visualizar la información requerida en los gráficos generados	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

Tabla 3.66: Prueba de aceptación: exportación de datos por producción

<b>Prueba de Aceptación</b>	
<b>Código:</b> PC22	<b>Código historia:</b> H10 Reportes de temperaturas
<b>Nombre:</b> Exportación de datos por producción	
<b>Condición de ejecución:</b> -Un usuario debe haber iniciado sesión -La producción debe haber capturado datos, caso contrario los gráficos aparecerán en blanco	
<b>Proceso de ejecución:</b> -En la barra de navegación el usuario elije la opción listado del menú producción -El usuario elije la opción reporte de la producción a visualizar -El usuario es redirigido a una ventana en la que constan los datos capturados en dicha producción -El usuario pulsa el botón exportar datos -El usuario elije los datos que requiere en la ventana generada -El usuario pulsa el botón exportar	
<b>Resultado:</b> El usuario constata que se genera un archivo con los datos que seleccionó al abrirlo en una aplicación tipo Excel	
<b>Evaluación:</b> La prueba finalizó con éxito	

Fuente: Elaborado por el investigador

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1. Conclusiones

- La implementación del sistema de monitoreo de temperatura y humedad brindó una perspectiva completa sobre el comportamiento interno de las cámaras de secado, como el aumento progresivo de temperatura con ligeras variaciones dependiendo de la ubicación del sensor.
- Para la toma de datos, los dispositivos NodeMCU conforman el aspecto más difícil de superar en el desarrollo de un sistema IOT ya que si bien existen alternativas como los dispositivos Arduino o Raspberry ninguno ofrece la capacidad de conexión a internet vía Wifi a un precio tan reducido, además los sensores DHT22 constan de la capacidad para funcionar en ambientes con una alta presencia de temperatura y humedad para la captura de dichos valores con un rango muy bajo de falla.
- Micropython fue de gran utilidad al momento de implementar la lógica en los dispositivos NodeMCU ya que su sintaxis además de clara posee capacidades de alto nivel como es el uso de funciones asíncronas por medio de la librería uasyncio aún si el dispositivo posee únicamente un hilo de ejecución.
- El desarrollo del dashboard para el monitoreo de temperatura y humedad facilitó el proceso de monitoreo de dichos parámetros por parte del operario ya que no se requería su presencia en el lugar.
- Se pudo tomar acciones correctivas después de registrar el proceso de la primera producción de un producto nuevo y detectar que la temperatura afectó la calidad final. Ver anexo F.
- El uso de un entorno Linux tanto para el desarrollo como el despliegue de una aplicación web con el Framework Asp Net Core resulta viable haciendo uso del IDE Visual Studio Code, concluyendo así que el uso de dicho Framework puede evitar el uso de tecnología propietaria como

el sistema operativo Windows y el IDE Visual Studio reduciendo los costos por licencias.

#### **4.2. Recomendaciones**

- Se recomienda a la empresa Fideos Victoria continuar con el desarrollo e incorporación de módulos para el monitoreo del área de almacenamiento de materia prima y del área de producción con el fin de tener un control preciso de una mayor cantidad de variables que influyen en el proceso de fabricación de pastas.
- Se sugiere que una vez se tenga recabado un gran conjunto de datos aplicar técnicas de análisis de datos como Business Intelligence (inteligencia de negocios) a las distintas variables y así buscar una mejora continua en el proceso de secado.
- Se recomienda la actualización del dispositivo NodeMCU de la cámara 2 a la versión 3 como el usado en la cámara 1 ya se constató que continuamente sufre desbordes de memoria ocasionando la pérdida de conexión y por lo consiguiente de datos.
- Se recomienda que antes de llegar al límite de soporte proporcionado al framework por Microsoft en 2022, realizar una migración del sistema al framework .Net 6 anunciado para Noviembre del 2021 y que contará con soporte hasta 2024.

## Bibliografía

- [1] M. S. C. Julio, “Implementación de un sistema de sensores, monitoreo y alertas de la temperatura y humedad de un centro de datos,” *Universidad de Guayaquil Facultad de Ingeniería Industrial Departamento Académico de Titulación*, 2017.
- [2] F. Rebaudo and R. Benoist, “Low-cost automatic temperature monitoring system with alerts for laboratory rearing units,” *MethodsX*, vol. 6, pp. 2127–2133, Jan. 2019.
- [3] I. Guncay, “Optimización del proceso de secado en pastas alimenticias.”
- [4] A. Baiano, A. Conte, and M. A. Del Nobile, “Influence of drying temperature on the spaghetti cooking quality,” *Journal of Food Engineering*, vol. 76, pp. 341–347, Oct. 2006.
- [5] D. C. Misra, I. P. S. Sethi, O. P. Gupta, M. Kapoor, and R. K. Dwivedi, “Digidhan Dashboard—Monitoring and Analysis of Digital Payments,” in *ICT Systems and Sustainability* (M. Tuba, S. Akashe, and A. Joshi, eds.), *Advances in Intelligent Systems and Computing*, (Singapore), pp. 91–104, Springer, 2020.
- [6] W. Lijuan and L. Siqi, “Design and Realization of Covers Monitoring and Controlling System,” in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pp. 2721–2725, Dec. 2018.
- [7] Comunicacion, “Ambato.gob.ec,” *GADMA Ambato*.
- [8] D. Galarce, *Situacion actual y perspectivas de mercado para el trigo candeal en Chile*. Santiago de Chile: Fundacion Chile.
- [9] L. Britos, *Análisis del consumo de pastas en la Argentina*. Buenos Aires: CEPEA, 2013.
- [10] C. Subgerencia de Análisis de Productos y Servicios, “Ficha sectorial: Fideos y pastas,” tech. rep., Corporación Financiera Nacional, 2021.
- [11] S. E. de Normalización INEN, “Pastas alimenticias o fideos. requisitos,” tech. rep., INEN, Baquerizo Moreno E8-29 y Av. 6 de Diciembre, 2014.

- [12] G. McArdle and R. Kitchin, “the dublin dashboard: Design and development of a real-time,” *ISPRS Annals of the Photogrammetry*, p. 7, 2016.
- [13] S. Muralidharan, A. Roy, and N. Saxena, *An Exhaustive Review on Internet of Things*, vol. 1. Springer Science+Business Media New York.
- [14] M. E. Cardona and S. López, *Una revisión de literatura sobre el uso de sistemas de adquisición de datos para la enseñanza de la física en la educación básica, media y en la formación de profesores*. Medellín: Universidad de Antioquia, 2017.
- [15] R. Sturm, C. Pollard, and J. Craig, “Chapter 7 - Managing Web-Based Applications,” in *Application Performance Management (APM) in the Digital Enterprise* (R. Sturm, C. Pollard, and J. Craig, eds.), pp. 83–93, Boston: Morgan Kaufmann, Jan. 2017.
- [16] A. Anwar, “A review of rup (rational unified process),” *International Journal of Software Engineering (IJSE)*, vol. 5, no. 2, pp. 12–19, 2014.
- [17] B. M. Montero, H. V. Cevallos, and J. D. Cuesta, “Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” *Espiraes Revista Multidisciplinaria de investigación*, vol. 2, June 2018. Number: 17.
- [18] G. Arcos-Medina, J. Menéndez, and J. Vallejo, “Comparative study of performance and productivity of mvc and mvvm design patterns,” *KnE Engineering*, pp. 241–252, 2018.
- [19] A. Sunardi and Suharjito, “MVC Architecture: A Comparative Study Between Laravel Framework and Slim Framework in Freelancer Project Monitoring System Web Based,” *Procedia Computer Science*, vol. 157, pp. 134–141, Jan. 2019.
- [20] K. Nayyeri and D. White, *Pro ASP.NET SignalR: Real-Time Communication in .NET with SignalR 2.1*. Apress, 2014.
- [21] M. Kashyap, V. Sharma, and N. Gupta, “Taking MQTT and NodeMcu to IOT: Communication in Internet of Things,” *Procedia Computer Science*, vol. 132, pp. 1611–1618, Jan. 2018.
- [22] A. Martín-Garín, J. A. Millán-García, A. Baïri, J. Millán-Medel, and J. M. Sala-Lizarraga, “Environmental monitoring system based on an Open Source Platform and the Internet of Things for a building energy retrofit,” *Automation in Construction*, vol. 87, pp. 201–214, Mar. 2018.



- [23] S. Sachdev, J. Macwan, C. Patel, and N. Doshi, “Voice-Controlled Autonomous Vehicle Using IoT,” *Procedia Computer Science*, vol. 160, pp. 712–717, Jan. 2019.

## Anexos

**Anexo A**

**Manual de usuario**

**«DESARROLLO DE UN SISTEMA DE MONITOREO DE  
TEMPERATURA Y HUMEDAD PARA CÁMARAS DE SECADO  
DE PASTAS EN LA EMPRESA FIDEOS VICTORIA»**

Versión: 001

Fecha: 05/04/2021

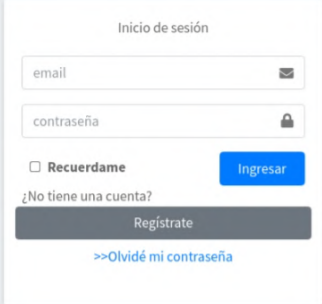
## **Introducción**

El presente sistema de monitoreo de temperatura y humedad fue desarrollado como proyecto de titulación de la carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato

El presente manual de usuario tiene como fin servir como una guía para el manejo del sistema por parte de los usuarios de la empresa Fideos Victoria.

## 1.1. Ingreso al sistema

Antes de acceder al sistema en sí primero el usuario debe ingresar sus credenciales en los recuadros email y contraseña para después presionar en el botón ingresar teniendo como opción que el sistema recuerde su sesión con la casilla recuerdame.



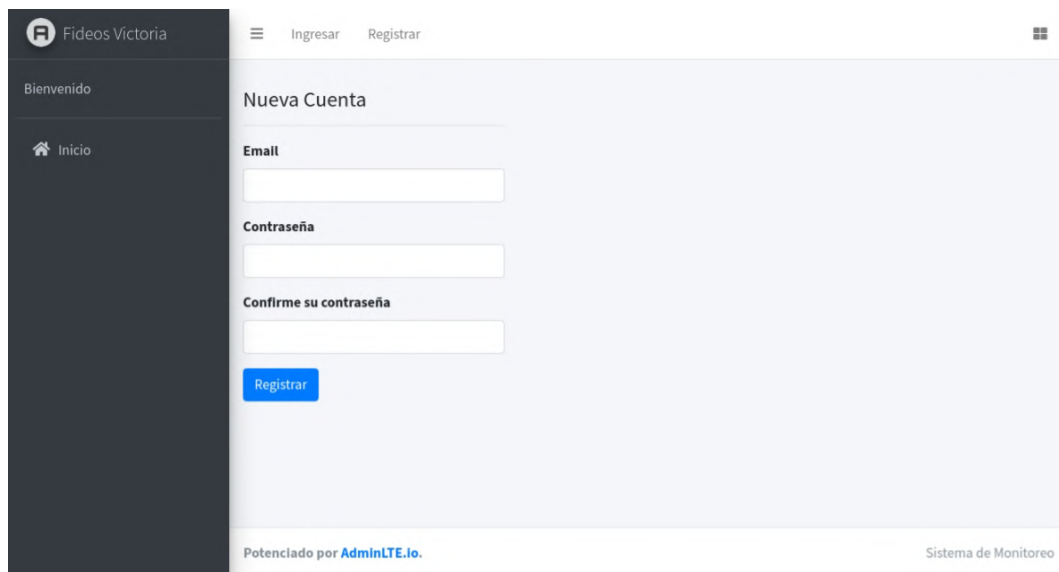
The screenshot shows a login form titled "Inicio de sesión". It contains two input fields: "email" with an envelope icon and "contraseña" with a lock icon. Below the fields is a checkbox labeled "Recuerdame" and a blue button labeled "Ingresar". Underneath, there is a link that says "¿No tiene una cuenta? Regístrate" and another link that says ">>Olvidé mi contraseña".

Con la opción olvidé mi contraseña el usuario accede a la interfaz para la recuperación de la contraseña por medio de la interfaz que se observa a continuación.




The screenshot shows a web interface for password recovery. On the left is a dark sidebar with the logo "Fideos Victoria" and a "Bienvenido" message. The main content area has a title "¿Olvidó su contraseña?" and a subtitle "Ingrese su email.". Below this is an "Email" input field and a blue "Enviar" button. At the bottom, it says "Potenciado por AdminLTE.io." and "Sistema de Monitoreo".

Con la opción regístrate el usuario se redirigirá a una ventana en la que se ingresan los datos correspondientes para que el administrador confirme la cuenta y el usuario pueda hacer uso de la misma.

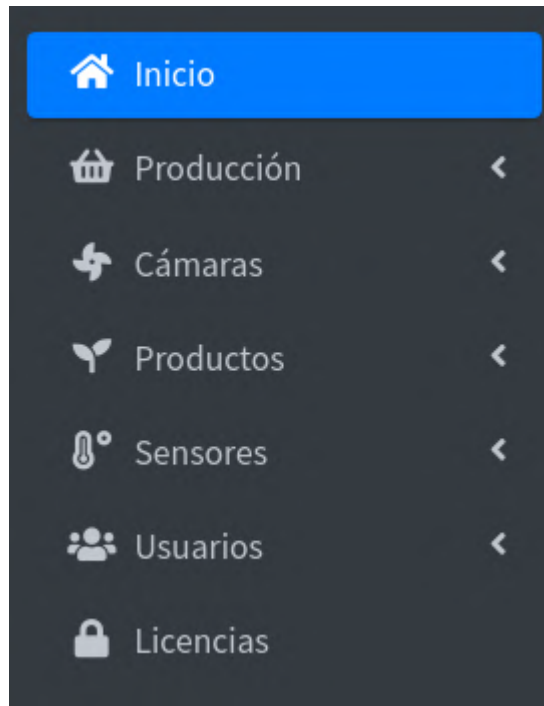


## 1.2. Navegación

Al lado izquierdo se encuentra un menú con las páginas a las que el usuario puede acceder y en caso que el dispositivo tenga una pantalla de pequeñas dimensiones como un smartphone el menú se encontrará oculto y puede ser desplegado con el botón 

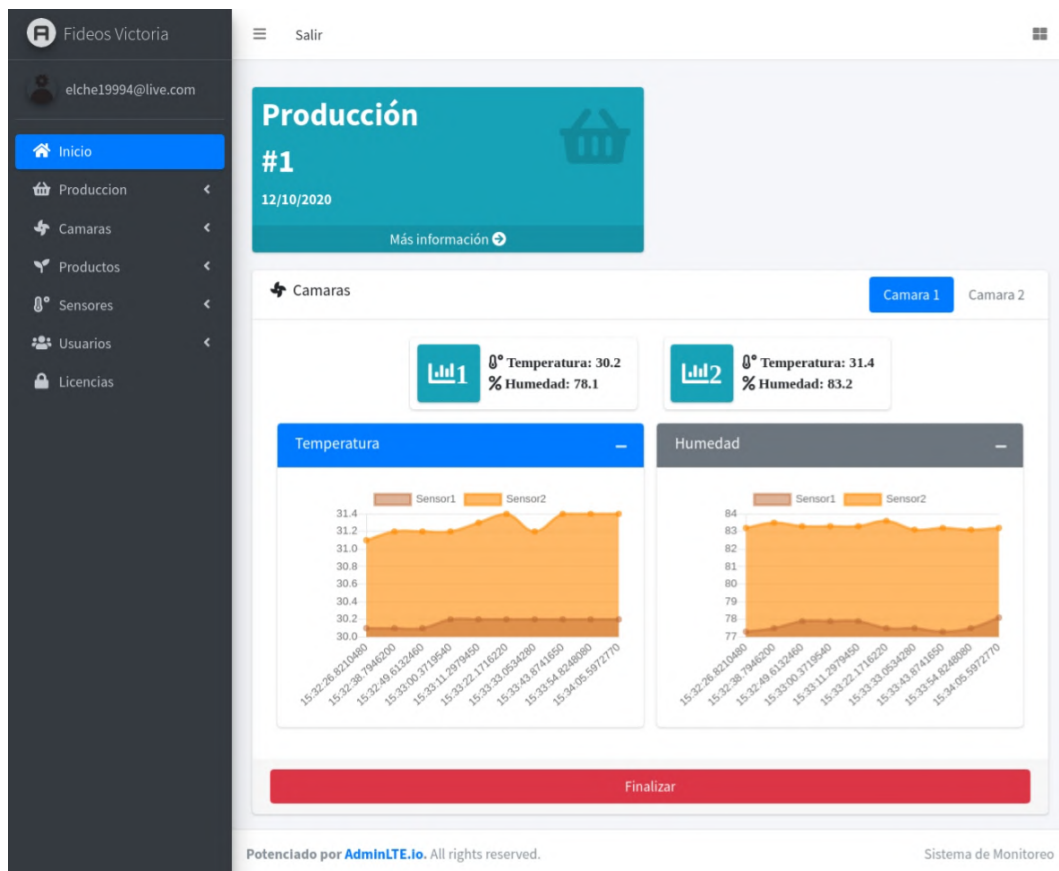
El menú contiene los siguientes accesos:

- **Inicio:** redirige al usuario a la página inicial en la que se encuentra el dashboard.
- **Producción:** contiene un submenú con acceso al listado de producciones y a crear una nueva producción.
- **Cámaras:** contiene un submenú con acceso al listado de cámaras y a crear una nueva cámara.
- **Productos:** contiene un submenú con acceso al listado de productos y a crear un nuevo producto.
- **Sensores:** contiene un submenú con acceso al listado de sensores y a crear un nuevo sensor.
- **Usuario:** contiene un enlace al listado de usuarios en el que el administrador puede modificar los roles que posee cada usuario o eliminar usuarios.
- **Licencias:** redirige al usuario a la página de atribuciones y agradecimientos a las tecnologías utilizadas en el sistema.




### 1.3. Página principal(Dashboard)

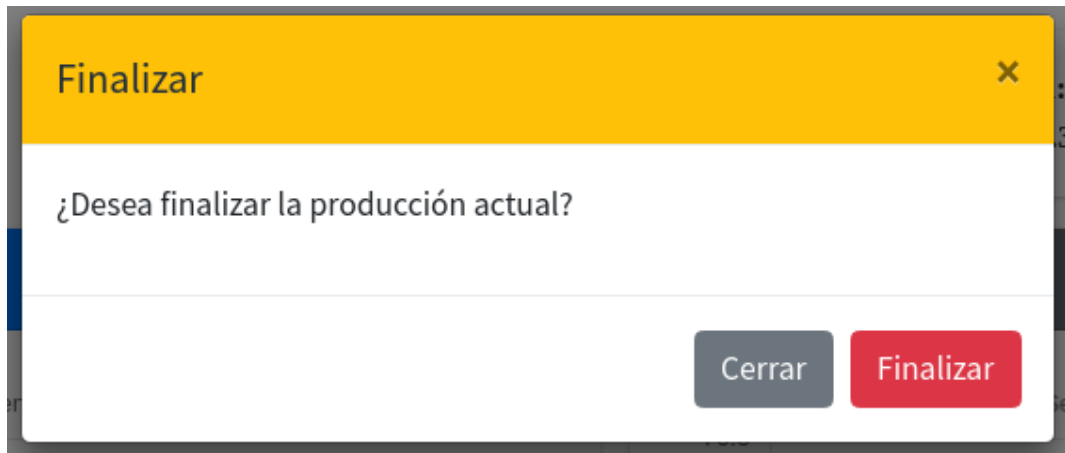
Una vez se haya ingresado correctamente el usuario y la contraseña el sistema redirigirá al usuario a la pagina principal en la que consta el dashboard con el número de producción, las cámaras que se encuentran activas y los datos mostrados en forma de un gráfico como se observa a continuación.



La interfaz presentada consta de la siguiente funcionalidad:

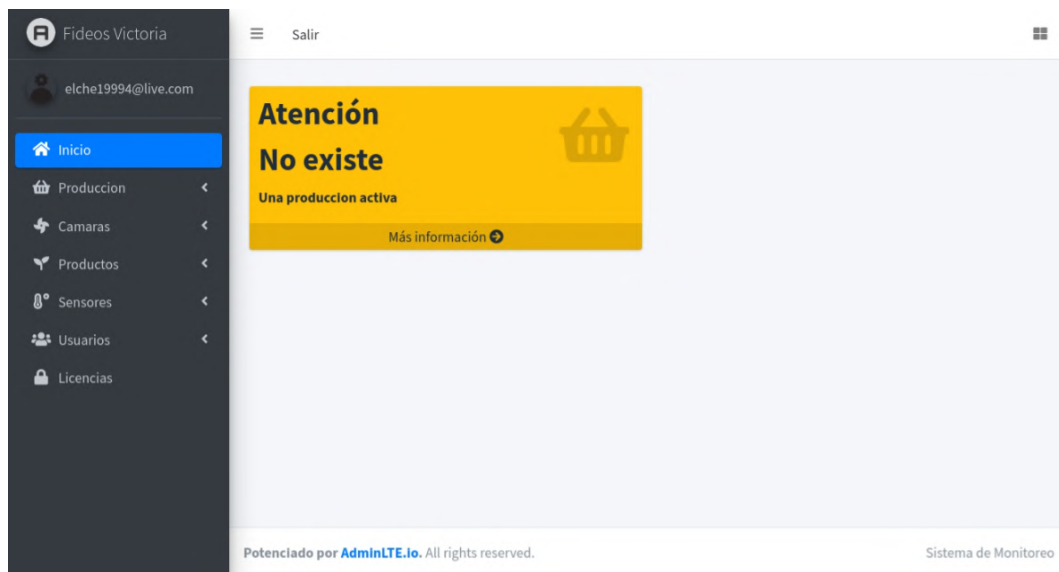
- **Botón salir:** ubicado en la parte superior izquierdo junto al botón para minimizar o maximizar el menú sirve para cerrar la sesión actual y redirige al usuario a una nueva pagina indicando que su sesión fue cerrada.
- **Botones cámara 1 o 2:** estos botones sirven para que el usuario pueda navegar entre los gráficos de las cámaras que se encuentran en la producción.
- **Finalizar:** por medio de este botón el usuario elije si desea finalizar la producción actual al desplegar un cuadro modal con las opciones cerrar y finalizar
- **Botón superior derecho:** por medio de este botón  el usuario despliega una ventana lateral con información extra de la producción para evitar salir de la página principal para observar dicha información.





Información extra	
Cámara 1	
<b>Producto</b>	Regin enroscado blanco
<b>Cantidad</b>	15,00
<b>Viento</b>	15,00 horas
<b>Extractor</b>	15,00 horas
<b>Calor</b>	15,00 horas
<b>Viento</b>	15,00 horas

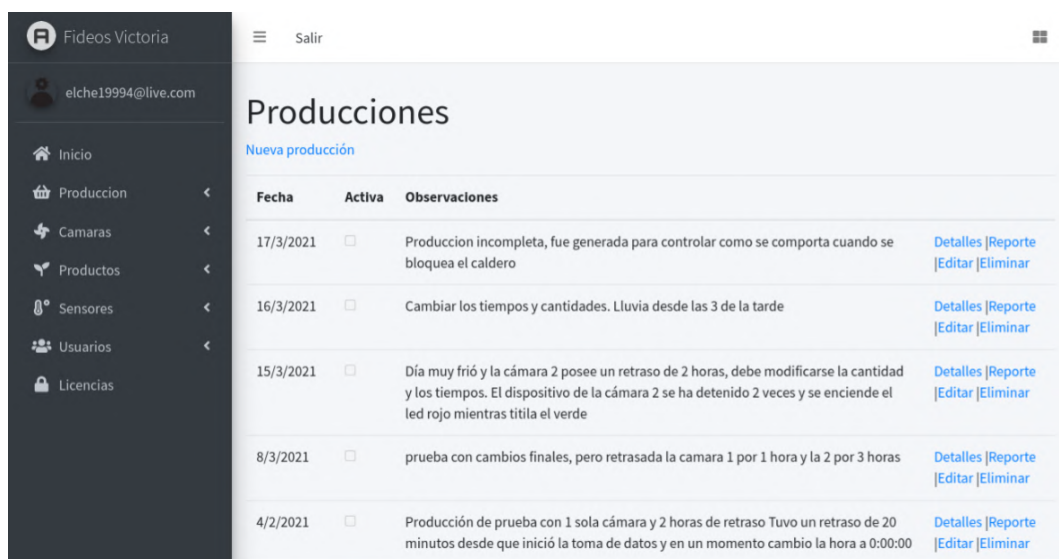
En caso de no existir una producción activa el dashboard estará deshabilitado y unicamente se muestra una advertencia con un enlace al listado de producciones.



## 1.4. Producciones

Para acceder a todas las opciones para el manejo de producción es necesario que el usuario acceda a través del enlace listado del menú producción. La ventana a la que el usuario es redirigido consta de las siguientes opciones:

- **Nueva producción:** enlace a la pagina de creación de producciones.
- **Listado de las producciones:** consta de todas las producciones existentes y opciones para el manejo de las mismas.
- **Botón superior derecho:** presenta una ventana con consejos



Cada producción presente consta de las opciones que se detallan a continuación.

### 1.4.1. Detalles

Presenta los detalles de la producción elegida en la pagina mostrada a continuación. Destacando el hecho de mostrar en forma de tablas los datos más relevantes como son el producto, la cantidad y la calidad que tuvo la producción elegida.

Fideos Victoria | elche19994@live.com | Salir

#### Produccion #15

Fecha: 17/3/2021  
Activa:   
Observaciones: Produccion incompleta, fue generada para controlar como se comporta cuando se bloquea el caldero

Cámara 1

Producto	Cantidad	Calidad
Regin Blanco	15,00	Alta

Cámara 2

Producto	Cantidad	Calidad
Lazo gigante	10,00	Alta

[Editar](#) | [Producciones](#)

Potenciado por [AdminLTE.io](#) | Sistema de Monitoreo

### 1.4.2. Reporte

Presenta la información de la producción elegida de una forma legible y ordenada. Para la modificación del gráfico, el usuario puede elegir las variables a visualizar en el recuadro izquierdo para luego presionar el botón actualizar.

Fideos Victoria | elche19994@live.com | Salir

#### Producción #15

Fecha: 17/3/2021

Observación: Produccion incompleta, fue generada para controlar como se comporta cuando se bloquea el caldero

Cámara 1 | Cámara 2

Cámara 1: Regin Blanco

Sensor 1:   
Sensor 2:   
Temperatura:   
Humedad:

[Actualizar](#)

#### Gráficas

Guardar

Sensor 1

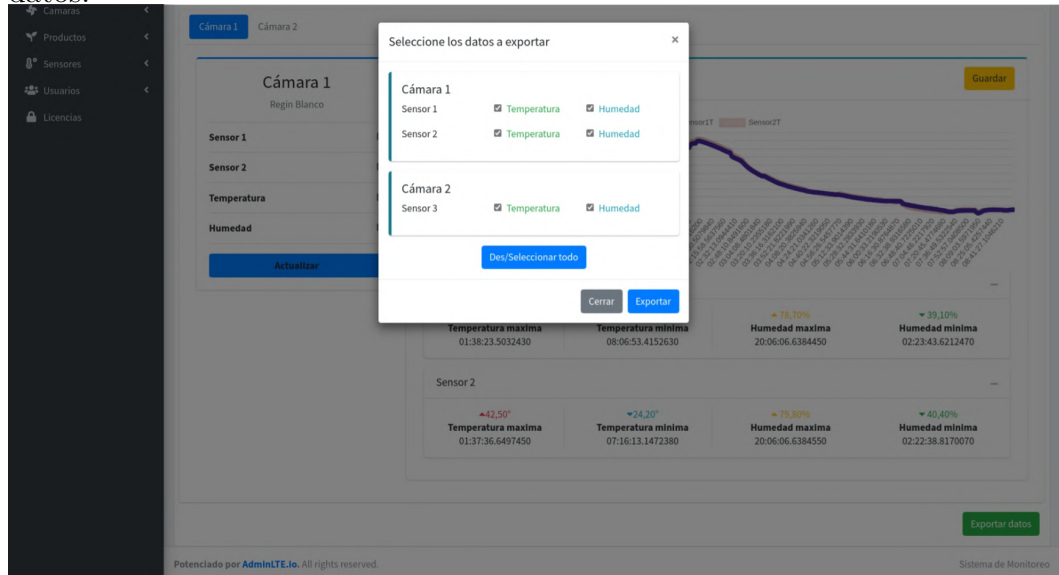
Temperatura maxima: +42,00° (01:38:23.5032430)	Temperatura minima: +24,00° (08:06:53.4152630)	Humedad maxima: +78,70% (20:06:06.6384450)	Humedad minima: -39,10% (02:23:43.6212470)
------------------------------------------------	------------------------------------------------	--------------------------------------------	--------------------------------------------

Sensor 2

Temperatura maxima: +42,50° (01:37:36.6497450)	Temperatura minima: +24,20° (07:16:13.1472380)	Humedad maxima: +79,80% (20:06:06.6384550)	Humedad minima: -40,40% (02:22:38.8170070)
------------------------------------------------	------------------------------------------------	--------------------------------------------	--------------------------------------------

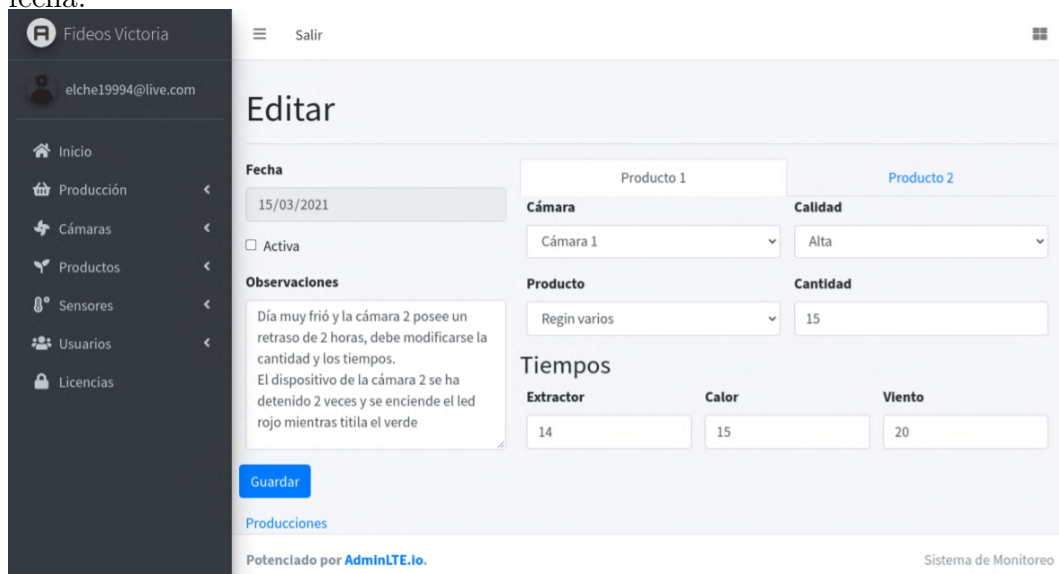
[Exportar datos](#)

Ademas de la visualización de la información, esta página posee la capacidad para guardar el gráfico en un formato de imagen y también puede exportar los datos en un formato csv que puede ser abierto por cualquier programa de análisis de datos.



### 1.4.3. Editar

Presenta la información de la producción para ser modificada a excepción de la fecha.



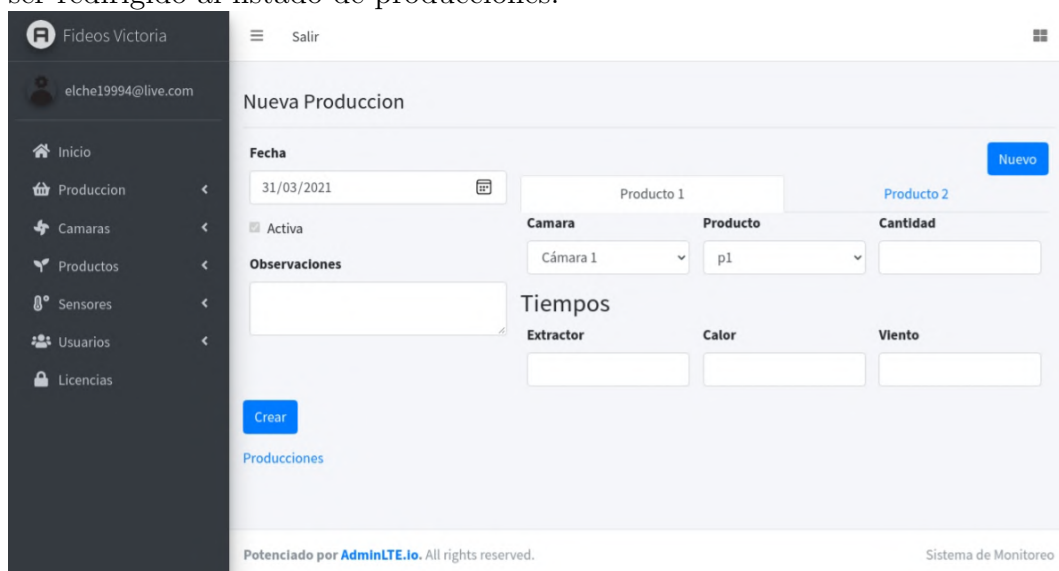
### 1.4.4. Eliminar

Presenta una ventana en la que se pide una confirmación para eliminar la producción elegida.



### 1.4.5. Nueva producción

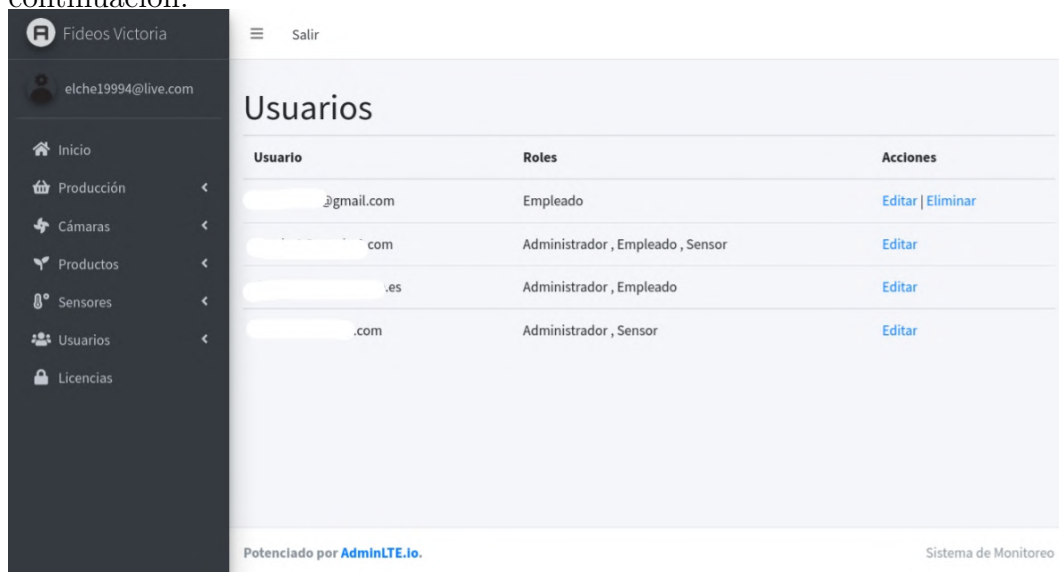
Para crear una nueva producción el usuario debe rellenar los campos presentes en la interfaz, además posee la capacidad para asignar más de un producto a la producción por medio del botón Nuevo para finalmente pulsar el botón crear y ser redirigido al listado de producciones.



Cámaras, productos, y sensores siguen el mismo patrón de uso contando con una lista con opciones como editar, detalles y eliminar siendo el proceso intuitivo además de tener un uso menos frecuente que con las producciones.

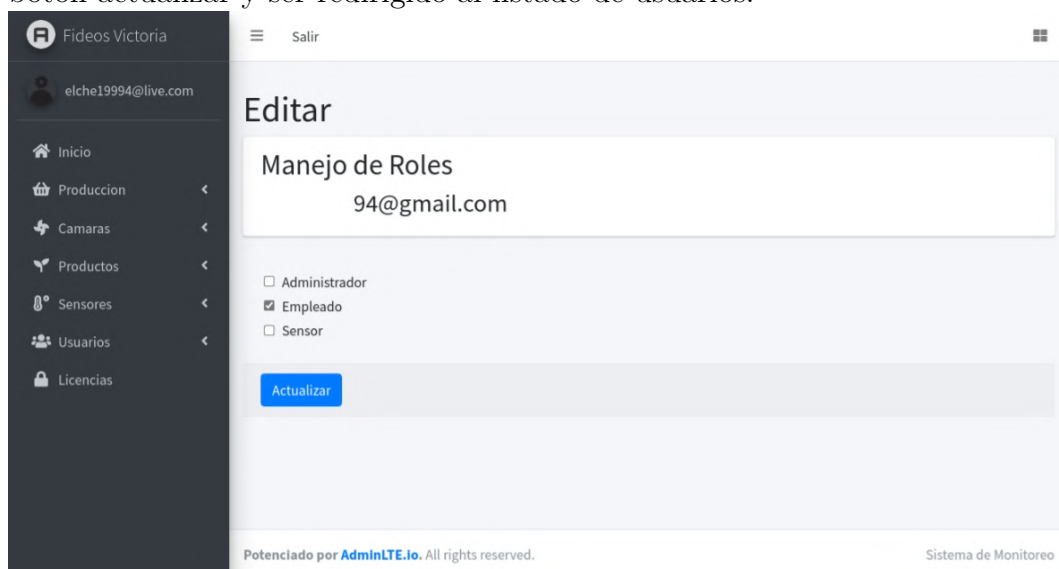
## 1.5. Usuarios

Una de las características del sistema de monitoreo es que al registrar un nuevo usuario este obtiene por defecto el rol empleado, pero para que los dispositivos IOT puedan acceder al sistema y enviar datos se requiere que el usuario tenga un rol Sensor. Por este motivo si el usuario es administrador posee la capacidad para modificar los roles de los usuarios y eliminarlos usando la interfaz presentada a continuación.



Usuario	Roles	Acciones
[redacted]@gmail.com	Empleado	<a href="#">Editar</a>   <a href="#">Eliminar</a>
[redacted].com	Administrador , Empleado , Sensor	<a href="#">Editar</a>
[redacted].es	Administrador , Empleado	<a href="#">Editar</a>
[redacted].com	Administrador , Sensor	<a href="#">Editar</a>

En caso de elegir la opción editar, el usuario será redirigido a la siguiente ventana en la que podrá elegir los roles que le corresponden para finalmente pulsar el botón actualizar y ser redirigido al listado de usuarios.

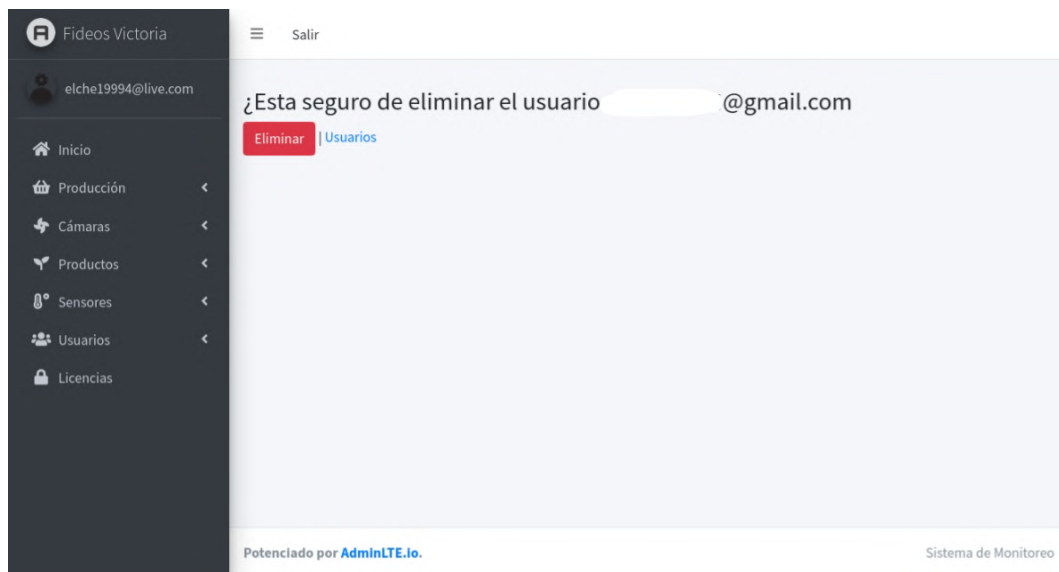


Manejo de Roles  
94@gmail.com

Administrador  
 Empleado  
 Sensor

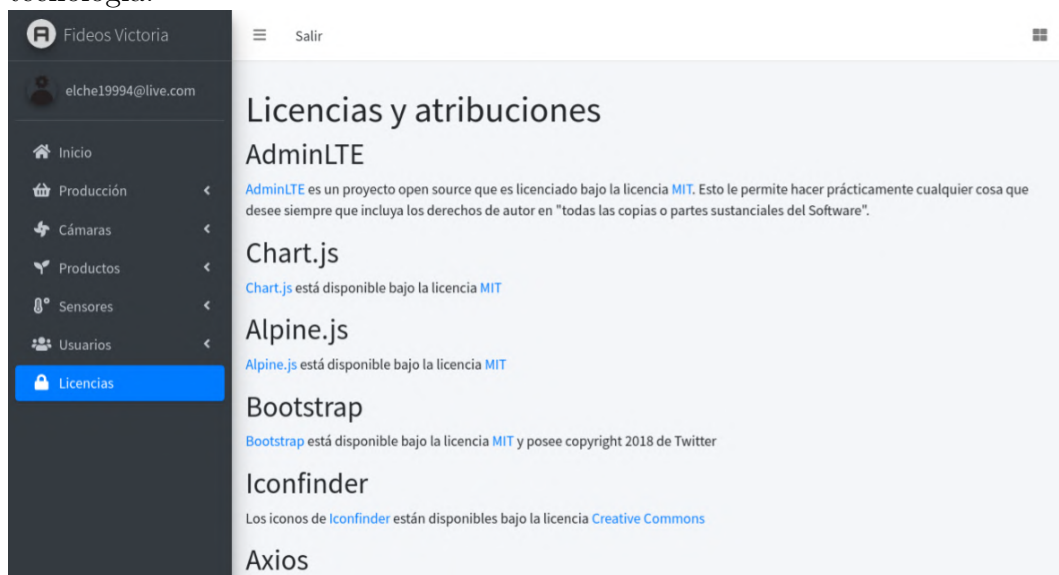
[Actualizar](#)

La opción eliminar está limitada a los usuarios que no sean administradores evitando así que por error se elimine un usuario con el rol con más privilegios.



## 1.6. Licencias

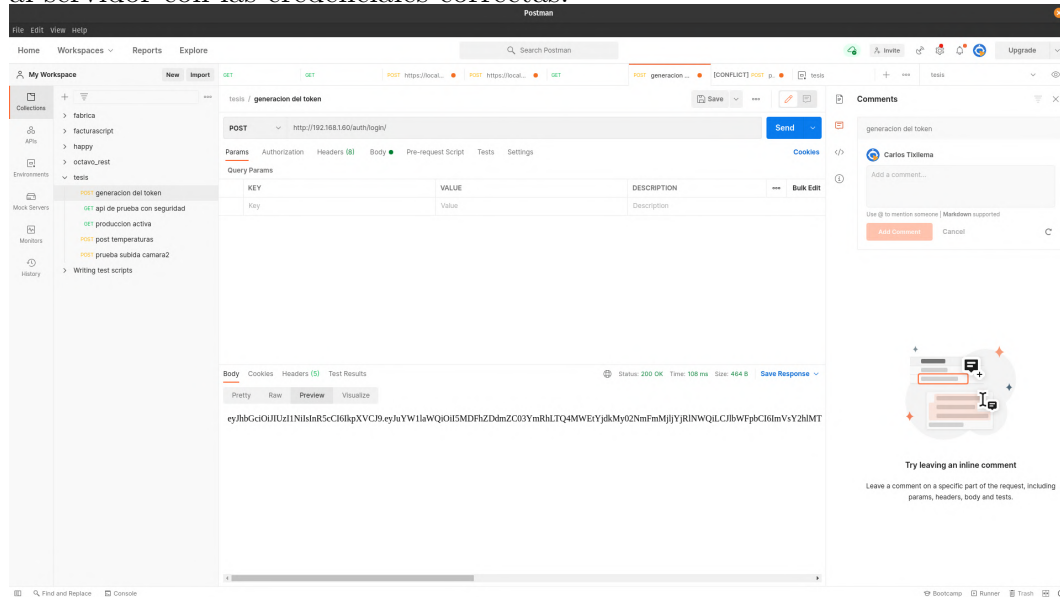
Presenta las licencias y atribuciones sobre las tecnologías utilizadas para el desarrollo del sistema de monitoreo conteniendo enlaces a las licencias Open Source que posee cada tecnología y a la página web con información sobre dicha tecnología.



## 1.7. Generación token para dispositivos IOT

La generación de tokens para que los dispositivos IOT puedan acceder al sistema y enviar los datos capturados debe hacerlo el encargado de la implementación de la lógica en los dispositivos.

Para esto se debe utilizar algún programa como Postman y realizar una petición al servidor con las credenciales correctas.



El token recibido debe colocarse en el archivo main.py incluido en cada dispositivo en la línea 21 que contiene los headers que se utilizan para las consultas al servidor

```
headers =
{'content-type': 'application/json', 'Authorization': '***token***'}
```



## Anexo B

### Entrevista dirigida al encargado de los controles de automatización de las cámaras

A través de la entrevista aplicada al encargado de los controles, se presenta cada una de las preguntas con su respectiva respuesta.

**Nombre del entrevistado:** Franklin Lozada

**Fecha:** 23/01/2021

**Lugar:** Empresa Fideos Victoria

1. ¿Actualmente tiene alguna forma de medir la humedad presente en las cámaras de secado?

**Responde:** No, los sensores instalados solo dan la temperatura.

2. ¿Cuánto tiempo considera que emplea en revisar la temperatura de las cámaras de secado?

**Responde:** Un par de horas más o menos.

3. ¿Actualmente tiene alguna forma de conocer si el secado sobrepasó el límite de temperatura?

**Responde:** Cuando reviso a la hora en la que se apaga el caldero.

4. ¿Qué aspecto del proceso de secado considera que podría mejorarse al implementarse un sistema de monitoreo?

**Response:** Pudiera servir si falla el caldero y la temperatura baja antes de hora.

5. ¿Considera que el monitoreo presencial del funcionamiento de las cámaras de secado ha afectado al correcto secado del producto?

**Responde:** Un par de veces no nos hemos dado cuenta que el caldero se bloqueó y el fideo no se secó bien.

## Anexo C

### Entrevista dirigida al gerente/propietario de la empresa Fideos Victoria

#### Entrevista dirigida

A través de la entrevista aplicada al gerente/propietario de la empresa Fideos Victoria, se presenta cada una de las preguntas con su respectiva respuesta.

**Nombre del entrevistado:** Carlos Cristóbal Tixilema Yanchaliquin

**Fecha:** 23/01/2021

**Lugar:** Empresa Fideos Victoria

1. ¿Actualmente tiene alguna forma de medir la humedad presente en las cámaras de secado?

**Responde:** No, siempre hemos medido solo la temperatura.

2. ¿Cuánto tiempo considera que emplea en revisar la temperatura de las cámaras de secado?

**Responde:** Algo más de una hora si todo va bien.

3. ¿Actualmente tiene alguna forma de conocer si el secado sobrepasó el límite de temperatura?

**Responde:** Cuando bajo a revisar la temperatura antes de que se apague el caldero

4. ¿Que aspecto del proceso de secado considera que podría mejorarse al implementarse un sistema de monitoreo?

**Response:** Podría ayudarme a ver como está el calor y saber si necesitará más tiempo.

5. ¿Considera que el monitoreo presencial del funcionamiento de las cámaras de secado ha afectado al correcto secado del producto?

**Responde:** Tal vez cuando se bloquea el caldero y se demora en darse cuenta.

## Anexo D

### Dispositivos y sensores

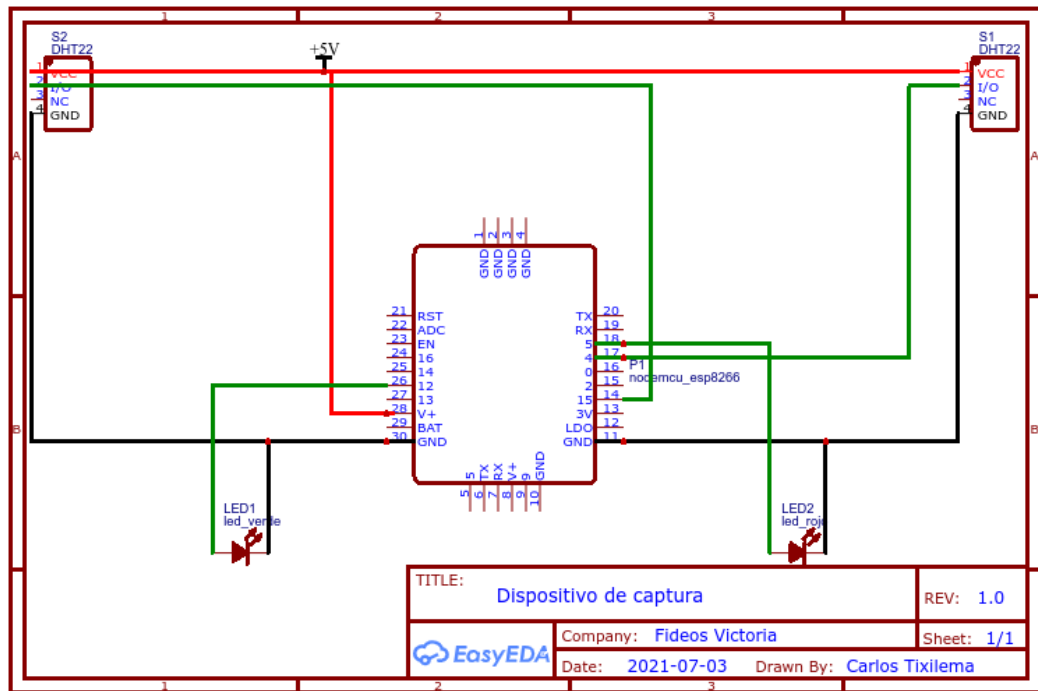






## Anexo E

### Esquema conexiones dispositivo de captura



## Anexo F

### Análisis de datos capturados

En base a los datos recabados se procede a realizar la comparación entre la producción #56 en la que se inició la producción de un nuevo tipo de producto, cabello entrefino prensado, y por exceso de calor al inicio de la producción la calidad del producto fue baja por lo consiguiente hizo imposible su salida al mercado, representando una perdida para la empresa.

Tabla 6.1: 45Comparación entre datos capturados

<b>Producción</b>	<b>Temperatura máxima</b>	<b>Temperatura mínima</b>	<b>Humedad máxima</b>	<b>Humedad mínima</b>
#56 05/06/2021	42.2	24.1	79.1	34.9
#76 26/06/2021	42.6	24.0	73.4	35.0
#82 02/07/2021	38.5	23.8	85.4	36.1

Fuente: Elaborado por el investigador

Los correctivos planteados para mejorar la calidad del secado, además de la disminución del calor, fue el establecer un aumento progresivo y lento de la temperatura para así evitar el sobre calentamiento del producto y utilizar unicamente la cámara 2 ya que al tener un tamaño mayor que la cámara 1 el calor se disipa rápidamente teniendo una mayor eficiencia en cuanto al secado de pastas sensibles al calor.

Tabla 6.2: Comparación de tiempos de secado

Producción	Tiempo de extractor	Tiempo de calor	Tiempo de viento
#56 05/06/2021	5 horas	6 horas	12 horas
#76 26/06/2021	9 horas	10 horas	18 horas
#82 02/07/2021	9 horas	10 horas	18 horas

Fuente: Elaborado por el investigador

Para una mejor comprensión del comportamiento de la producción se revisó las graficas generadas en la sección de reportes tomando en consideración la temperatura y humedad al mismo tiempo y pueden observarse en las figuras 6.1, 6.2 Y 6.3

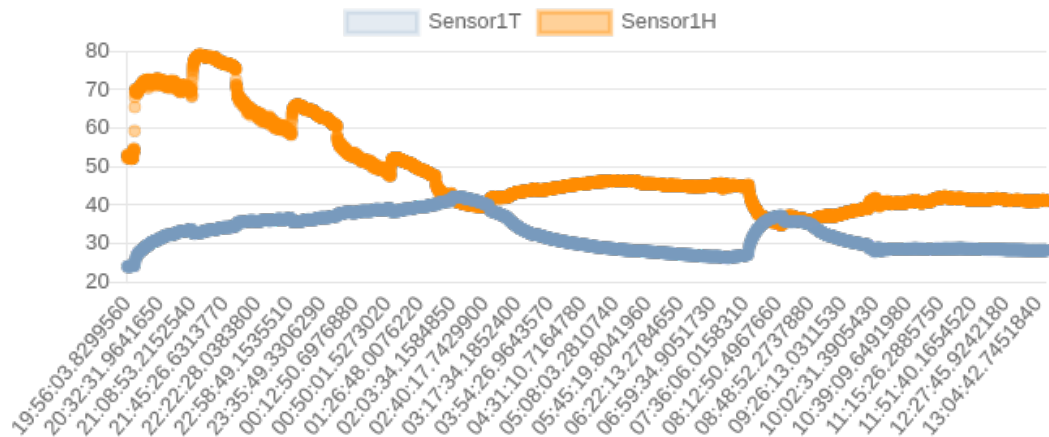


Figura 6.1: Gráfica producción #56  
Fuente: Elaborado por el investigador



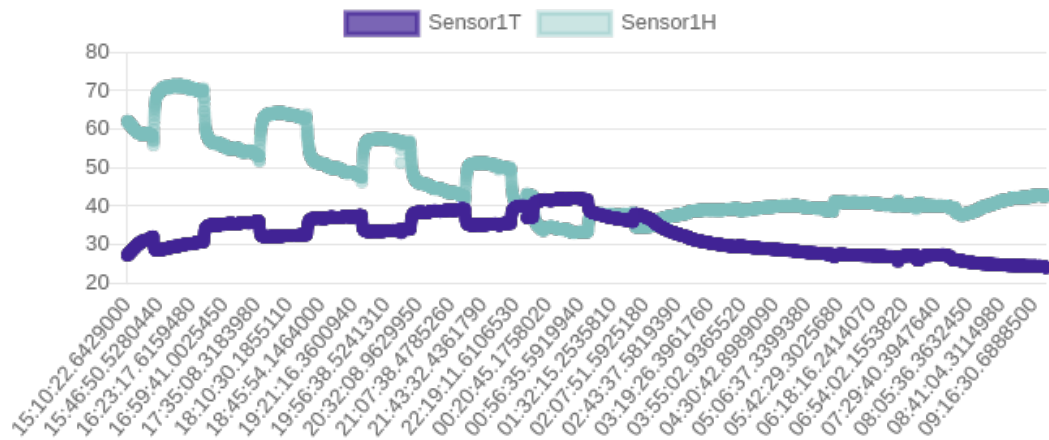


Figura 6.2: Gráfica producción #76  
Fuente: Elaborado por el investigador

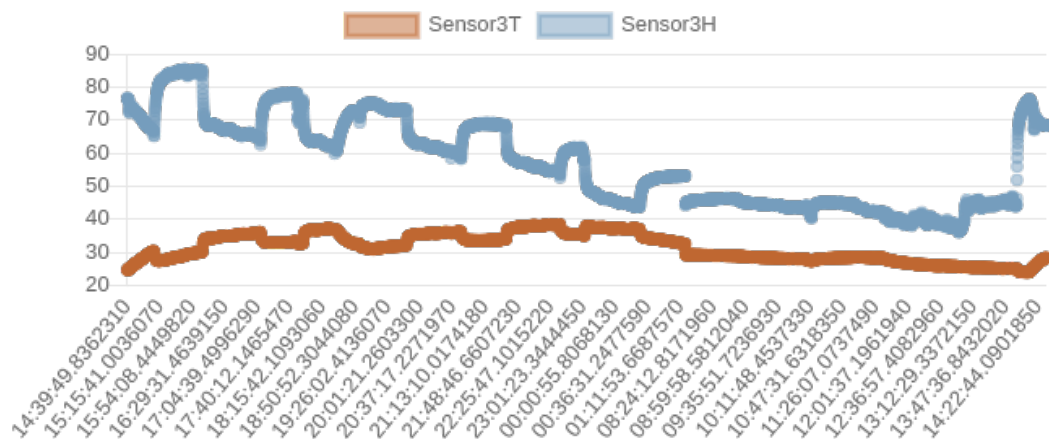


Figura 6.3: Gráfica producción #82  
Fuente: Elaborado por el investigador