



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN SISTEMA DE CONTROL DEL PROCESO DE
MATERIA PRIMA PARA LA ELABORACIÓN DE EMBUTIDOS
MEDIANTE LISTAS DINÁMICAS

LEON AZANZA JEFFERSON LAUTARO
INGENIERO DE SISTEMAS

MACHALA
2019



UTMACH

FACULTAD DE INGENIERÍA CIVIL
CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN SISTEMA DE CONTROL DEL PROCESO DE
MATERIA PRIMA PARA LA ELABORACIÓN DE EMBUTIDOS
MEDIANTE LISTAS DINÁMICAS

LEON AZANZA JEFFERSON LAUTARO
INGENIERO DE SISTEMAS

MACHALA
2019



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

EXAMEN COMPLEXIVO

DESARROLLO DE UN SISTEMA DE CONTROL DEL PROCESO DE MATERIA
PRIMA PARA LA ELABORACIÓN DE EMBUTIDOS MEDIANTE LISTAS
DINÁMICAS

LEON AZANZA JEFFERSON LAUTARO
INGENIERO DE SISTEMAS

REDROVAN CASTILLO FAUSTO FABIAN

MACHALA, 21 DE AGOSTO DE 2019

MACHALA
21 de agosto de 2019

Nota de aceptación:

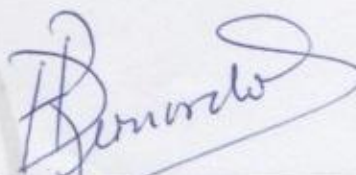
Quienes suscriben, en nuestra condición de evaluadores del trabajo de titulación denominado Desarrollo de un sistema de control del proceso de materia prima para la elaboración de embutidos mediante listas dinámicas, hacemos constar que luego de haber revisado el manuscrito del precitado trabajo, consideramos que reúne las condiciones académicas para continuar con la fase de evaluación correspondiente.



REDROVAN CASTILLO FAUSTO FABIAN
0702739228
TUTOR - ESPECIALISTA 1



JUMBO CASTILLO FREDDY ANIBAL
0704167949
ESPECIALISTA 2



HERNANDEZ ROJAS DIXYS LEONARDO
0923026298
ESPECIALISTA 3

Fecha de impresión: miércoles 21 de agosto de 2019 - 14:12

Urkund Analysis Result

Analysed Document: LEON AZANZA JEFFERSON LAUTARO.docx (D54787987)
Submitted: 8/12/2019 6:47:00 PM
Submitted By: fredrovan@utmachala.edu.ec
Significance: 1 %

Sources included in the report:

TrabajoComplejivo-Castillo Jimmy .docx (D40217812)

Instances where selected sources appear:

1

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, LEON AZANZA JEFFERSON LAUTARO, en calidad de autor del siguiente trabajo escrito titulado Desarrollo de un sistema de control del proceso de materia prima para la elaboración de embutidos mediante listas dinámicas, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

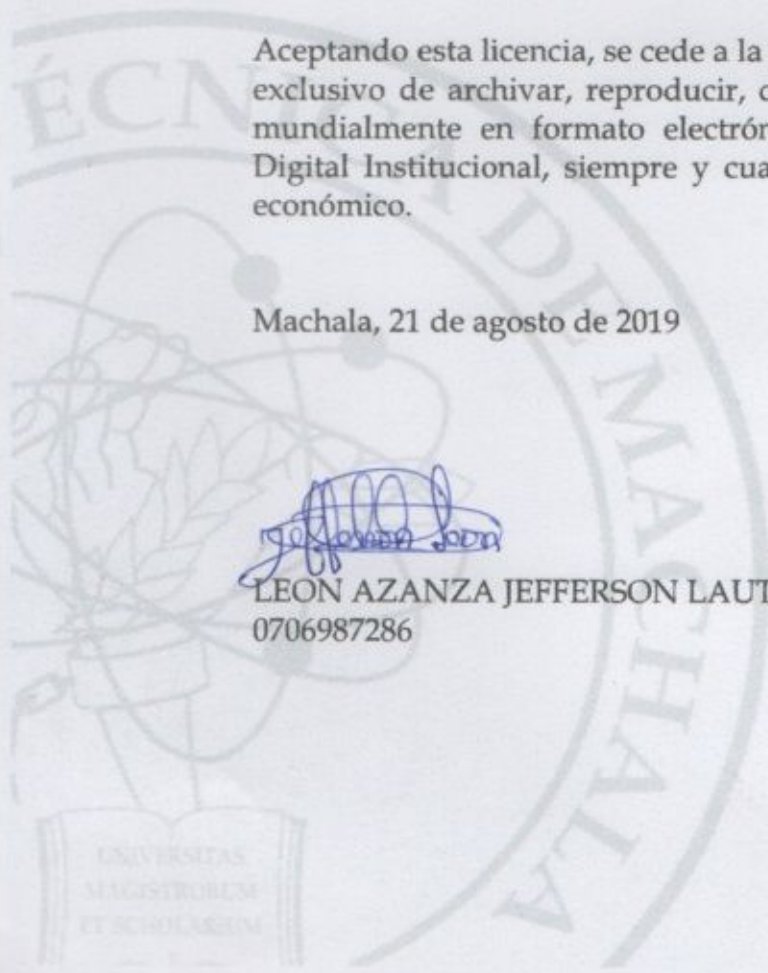
El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 21 de agosto de 2019



LEON AZANZA JEFFERSON LAUTARO
0706987286



DEDICATORIA

Dedico este trabajo principalmente a Dios, por darme salud, sabiduría, fortaleza y permitirme culminar esta etapa de mi vida profesional.

A mis padres, por su amor, sacrificio y apoyo incondicional en todos estos años, quienes me han motivado a seguir adelante y no rendirme ante los obstáculos que presenta la vida.

Sr. Jefferson Lautaro León Azanza

AGRADECIMIENTO

Agradezco a Dios, por haberme dado vida y permitirme culminar una etapa más en mi vida profesional.

A mis padres, por ser los principales motores de mis sueños y metas, por los consejos, valores y principios que me han inculcado.

Mi profundo agradecimiento a mi tutor Ing. Fausto Redrován, por su dedicación, conocimientos, enseñanza y colaboración lo que permitió desarrollar este trabajo.

Finalmente, a mis docentes que fueron parte de mi vida estudiantil, por todo el conocimiento adquirido durante la carrera.

Sr. Jefferson Lautaro León Azanza

RESUMEN

DESARROLLO DE UN SISTEMA DE CONTROL DE PROCESOS DE MATERIA PRIMA PARA LA ELABORACIÓN DE EMBUTIDOS MEDIANTE LISTAS DINÁMICAS

La tecnología digital se ha convertido en un imperativo para todas las empresas. Se necesita automatizar procesos manuales para mejorar su efectividad, promoviendo la interconectividad y mejorando la experiencia del usuario, por ejemplo en la adquisición de materia prima y gestión de la producción de embutidos. Actualmente, el desarrollo de aplicaciones de escritorio y móvil, se ha convertido en un tema empresarial clave, ya que ayudan a optimizar procesos, reducir costos, mejoran la forma de llegar a los clientes, entre otros beneficios. Por esta razón surge el presente proyecto que consiste en desarrollar aplicaciones de escritorio y móvil mediante ingeniería de software y listas dinámicas para la gestión y control de procesos de producción de embutidos. Para el desarrollo de la aplicación de escritorio se utilizó el lenguaje de programación C Sharp en la herramienta Visual Studio 2019, con un patrón de arquitectura de software modelo-vista-controlador. Para la administración de la base de datos se utilizó la herramienta SQLServer Management. Para el diseño de la aplicación móvil se utilizaron las herramientas API Rest en Node.js y React Native, que permiten escribir aplicaciones nativas en JavaScript para iOS y Android. Como metodología de desarrollo se implementó SNAIL, metodología híbrida para la creación de aplicaciones web.

PALABRAS CLAVE: Aplicación de escritorio, Aplicación móvil, Base de datos, Embutidos, Listas dinámicas, Metodología SNAIL

ABSTRACT

DEVELOPMENT OF A RAW MATERIAL PROCESS CONTROL SYSTEM FOR THE PREPARATION OF EMBUTIDES THROUGH DYNAMIC LISTS

Digital technology has become an imperative for all companies. It is necessary to automate manual processes to improve their effectiveness, promoting interconnectivity and improving the user experience, for example in the acquisition of raw material and management of sausage production. Currently, the development of desktop and mobile applications, has become a key business issue, as they help optimize processes, reduce costs, improve the way to reach customers, among other benefits. For this reason, the present project arises, which consists of developing desktop and mobile applications through software engineering and dynamic lists for the management and control of sausage production processes. For the development of the desktop application, the C Sharp programming language was used in the Visual Studio 2019 tool, with a model-view-controller software architecture pattern. The SQLServer Management tool was used for the administration of the database. The API Rest tools in Node.js and React Native were used to design the mobile application, which allow you to write native applications in JavaScript for iOS and Android. As a development methodology, SNAIL, a hybrid methodology for creating web applications, was implemented.

KEY WORDS: Desktop application, Mobile application, Database, Sausages, Dynamic Lists, SNAIL Methodology

ÍNDICE DE CONTENIDO

DEDICATORIA	1
AGRADECIMIENTO	2
RESUMEN	3
ABSTRACT	4
ÍNDICE DE CONTENIDO	5
ÍNDICE DE TABLAS	7
ÍNDICE DE FIGURAS	8
1. INTRODUCCIÓN	9
1.1 Marco Contextual	10
1.2 Problema	10
1.3 Objetivo General	10
2. DESARROLLO	11
2. 1 Marco Teórico	11
2.1.1 Producción de alimentos	11
2.1.2 Inventarios de materias primas	11
2.1.3 Arquitectura cliente-servidor	11
2.1.4 Modelo Vista Controlador (MVC)	12
2.1.5 Base de datos relacional	12
2.1.5.1 SQLServer	12
2.1.6 Programación orientada a objetos	12
2.1.6.1 Lenguaje de programación C#	12
2.1.7 Microsoft Visual Studio	13
2.1.8 Estructuras de datos dinámicos.	13
2.1.9 React native	13
2.1.10 Node js	13
2.1.11 Framework Bootstrap	13
2.1.12 Metodología Híbrida SNAIL	14
2. 2 Solución del problema	15
2.2.1 Arquitectura Lógica del Sistema	15
2.2.2 Arquitectura Tecnológica del Sistema	15
2.2.3 Materiales	16
2.2.4 Metodología Híbrida de desarrollo SNAIL	17
2.2.4.1 Fase 1: Análisis de requerimientos	17

2.2.4.2 Fase 2: Planificación	17
2.2.4.3 Fase 3: Diseño	17
2.2.4.4 Fase 4: Programación	18
2.2.4.5 Fase 5: Pruebas	19
2.2.4.5.1 Aplicación de escritorio	19
2.2.4.5.1 Aplicación móvil	21
2.3 Resultados	22
3. CONCLUSIONES	23
BIBLIOGRAFÍA	24
ANEXOS	26

ÍNDICE DE TABLAS

Tabla 1: Herramientas empleadas	17
Tabla 2: RF01 – Gestión de inventarios	26
Tabla 3: RF02 – Reporte Kardex	26
Tabla 4: RF03 – Control de stock	27
Tabla 5: RF04 – Gestión de fórmulas	27
Tabla 6: RF05 – Gestión de producción	28
Tabla 7: RF06 – Gestión de órdenes de producción	28
Tabla 8: RF07 – Proceso de producción	29
Tabla 9: RF08 – Productos terminados	29
Tabla 10: RF09 – Monitoreo de producción	30
Tabla 11: RNF01 – Arquitectura MVC	30
Tabla 12: RNF02 – Control de usuarios	31
Tabla 13: RNF03 – Validaciones	31
Tabla 14: RNF03 – Aplicación móvil	31

ÍNDICE DE FIGURAS

Figura 1: Arquitectura Lógica del sistema	15
Figura 2: Arquitectura Tecnológica del sistema	16
Figura 3: Guardar orden de compra	18
Figura 4: Guardar orden de producción	18
Figura 5: Monitorear producción	19
Figura 6: Orden de compra	19
Figura 7: Monitoreo de producción	20
Figura 8: Producciones	20
Figura 9: Página principal del aplicativo móvil	21
Figura 10: Monitoreo de producción del aplicativo móvil	21
Figura 11: Prototipo Menú Principal	32
Figura 12: Prototipo Ingreso materia prima	32
Figura 13: Prototipo monitoreo producciones	33
Figura 14: Prototipo orden de producción	33
Figura 15: Prototipo Dashboard aplicación móvil	34
Figura 16: Prototipo Inicio de Sesión aplicación móvil	34
Figura 17: Diagrama de base de datos relacional	35
Figura 18: Login del Sistema	35
Figura 19: Menú principal	36
Figura 20: Registro de usuarios	36
Figura 21: Kardex Materia Prima	37
Figura 22: Orden de compra materia prima	37
Figura 23: Creación de fórmulas	38
Figura 24: Orden de producción	38
Figura 25: Página principal de la aplicación móvil	39
Figura 26: Inicio de Sesión de la aplicación móvil	39
Figura 27: Dashboard aplicación móvil	40
Figura 28: Órdenes de producción activas	40
Figura 29: Detalles de producción	41
Figura 30: Monitoreo de producción	41

1. INTRODUCCIÓN

En Ecuador existen empresas dedicadas a la producción de embutidos que poseen la tecnología adecuada que garantizan la calidad de los mismos. Se produce mortadelas, jamones, salchichas, etc. Siendo las más demandas las mortadelas y salchichas; representando el 75% de la producción nacional [1].

Actualmente, el desarrollo de sistemas informáticos se ha visto en auge debido a la necesidad de mejorar procesos productivos, científicos y tecnológicos. Software que permita automatizar estos procesos de manera ágil, organizada y estructurada [2]. Además la comunicación hoy en día se desarrolla en gran parte a través de dispositivos tecnológicos, ya sea celulares o computadoras; permitiendo que se pueda compartir información con personas en diferentes lugares, de manera oportuna y precisa.

Muchas empresas y organizaciones se ven impulsados en optar por servicios tecnológicos ya sea escritorio, web o móvil para facilitar el control de sus procesos en diferentes áreas. En el desarrollo de sistemas una arquitectura utilizada es la de cliente-servidor, la cual es una tecnología que permite compartir las tareas de procesamiento de datos entre un servidor y los clientes [3] Se ha convertido en una de las ideas centrales de la computación en red, es decir, el cliente establece una conexión con el servidor a través de una red de área local, donde la mayoría de servicios y aplicaciones tienen como gran necesidad ser consumidas por varios usuarios de manera simultánea, y aquí radica su importancia y el porqué muchas empresas implementan esta arquitectura.

El objetivo de esta investigación es desarrollar una aplicación de escritorio y una móvil mediante ingeniería de software y listas dinámicas para la gestión y control de procesos de producción de embutidos. En la aplicación de escritorio se controlará la entrada de la materia prima y su procesamiento hasta la finalización como un producto terminado; mientras que la aplicación móvil tiene como objetivo proporcionar un entorno agradable (dashboard) para monitorear los procesos de producción.

En el capítulo 1 del presente informe se detalla la introducción, el marco contextual, problema y objetivo del trabajo realizado. En el capítulo 2 se fundamenta la presente investigación, en el capítulo 3 se establece conclusiones producto del trabajo implementado.

1.1 Marco Contextual

El presente trabajo, como parte escrita de la fundamentación a un problema planteado como examen complejo, demuestra la solución implementada para gestión y control de los procesos de producción de embutidos, desde la adquisición de la materia prima, generación de órdenes de producción e inventario de productos terminados, todo esto mediante la automatización de procesos utilizando listas dinámicas, programación de escritorio y programación móvil.

1.2 Problema

Según el problema planteado, las empresas dedicadas a la producción de embutidos necesitan un sistema que controle sus operaciones, desde la entrada de materia prima hasta la fabricación de un producto terminado, controlar el inventario de la materia prima, registrar órdenes de producción, generar reportes que muestren las entradas y salidas de los productos, y también tener un control de los productos terminados. Además de proveer alertas que indiquen un mínimo y un máximo; esto para que no exista escasez de la misma y no se acumule más de lo necesario. De acuerdo a esto se genera también la necesidad de tener un mejor análisis de datos, tomar decisiones adecuadas prácticamente en tiempo real y planificar la producción.

1.3 Objetivo General

Desarrollar una aplicación de escritorio y móvil mediante ingeniería de software y listas dinámicas para la gestión y control de procesos de producción de embutidos.

1.3.1 Objetivos Específicos

- Investigar los controles de proceso de materia prima para la elaboración de embutidos.
- Emplear las herramientas y tecnologías correctas para el desarrollo del sistema.
- Aplicar una metodología de desarrollo de software adecuada.
- Aplicar listas dinámicas para la estructuración de datos involucrados en los procesos que se automaticen.

2. DESARROLLO

2. 1 Marco Teórico

2.1.1 Producción de alimentos

Las materias primas son importantes ya que de esto dependen las condiciones de elaboración y calidad de un producto final. Cada producto tiene características propias, debido a sus diferentes ingredientes para su elaboración como al proceso tecnológico utilizado, la elección está en función al producto a terminar. Por ejemplo, de acuerdo a González, Giraldo y Restrepo [4], la elaboración de salchichas estándar se realiza con pasta de pollo para determinar la mejor mezcla entre fibra, cuero de cerdo y carragenina. Además, según Nascimento et al [5] se debe preparar salchichas con 78,95% de carne magra, 15,68% de grasa de cerdo, 2,15% de cloruro de sodio, 0,1% de sacarosa, 0,2% de ajo, 0,05% de pimienta negra, 0,03% de chile en polvo, 6 ml de vino blanco / kg, 0,05% de nuez moscada, 2% de agua helada, 1,25 g / kg de polvo de Prague y 1,25 g / kg de eritorbato de sodio. Por último, en una investigación [6], se realizó un estudio de embutidos, lo que presentó excesos de sodio en un 88%, grasas totales 75% y el mismo porcentaje grasa saturada y 13% contenían excesos de azúcares.

2.1.2 Inventarios de materias primas

El costo de materias primas se reconoce en el inventario de adquisición, y con el tiempo se extrae materia prima del inventario para la producción de productos terminados. Realizar un proceso de inventario eficiente ayuda a mejorar el rendimiento de la cadena de suministro, el cual se determina en función de los costos de pedido y almacenamiento [7]. Para un buen manejo de inventario en bodegas de almacenamiento de materias primas, se considera volumen, localización y su contribución a la estrategia productiva en la industria [8].

2.1.3 Arquitectura cliente-servidor

Esta arquitectura combina cliente (front-end) que interactúa con el usuario mediante una interfaz, para solicitar servicios del servidor (back-end) y muestra los resultados que el servidor devuelve. Los sistemas que aplican esta arquitectura son desplegados en una red, lo que implica que son accesibles desde cualquier lugar y momento [9].

2.1.4 Modelo Vista Controlador (MVC)

Es un patrón de diseño que permite desarrollar una aplicación separándola en tres partes: lógica empresarial (Modelo), visualización de datos (Vistas) y el intermedio entre Modelo y Vista (Controlador) [10].

La estructura MVC proporciona la interacción entre el usuario final de una aplicación y cómo se comunica con el servidor, además facilita la navegación entre cada una de las capas de un proyecto [11].

2.1.5 Base de datos relacional

Son un conjunto de tablas desde las cuales se pueden acceder a datos de muchas maneras sin tener la necesidad de reorganizar las tablas de la base de datos. Es un modelo relacional, que representa los datos en las tablas de manera intuitiva y directa. Están organizados por tablas, registros y columnas. Las tablas se comunican y comparten información, lo que facilita la búsqueda de datos. Usa el lenguaje de consultas estructuradas (SQL) [12].

2.1.5.1 SQLServer

Admite una variedad de aplicaciones de procesamiento de transacciones, ya que es un sistema de gestión de base de datos relacional basado en el lenguaje sql, la cual trabaja bajo la plataforma cliente-servidor. Soporta transacciones, procedimientos almacenados, además incluye un entorno gráfico de administración que permiten el manejo de comando DDL y DML [13].

2.1.6 Programación orientada a objetos

La programación orientada a objetos se refiere a los lenguajes que usan objetos en la programación. Es un enfoque para el desarrollo de software que trabaja entorno a objetos del mundo real. Esta se relaciona con las tecnologías de desarrollo de software, las cuales utilizan bibliotecas y marcos especiales que reducen el tiempo de programación [14].

2.1.6.1 Lenguaje de programación C#

Es un lenguaje de programación moderno y orientado a objetos, fue desarrollado por Microsoft; es muy similar a Java sintácticamente y de fácil manejo para los usuarios. Sirve para crear aplicaciones cliente-servidor, servicios web XML, aplicaciones de

base de datos, entre otras. Admite los conceptos de encapsulación, herencia y polimorfismo [15].

2.1.7 Microsoft Visual Studio

Proporciona una interfaz de usuario, un depurador integrado y muchas otras herramientas que facilitan el desarrollo de aplicaciones basados en C#, Visual C++, Visual J# y .NET framework. Incluye compiladores, herramientas para completar código, diseñadores gráficos y muchas funciones que facilitan el desarrollo de software [16].

2.1.8 Estructuras de datos dinámicos.

Son una forma en la cual un computador puede organizar datos, utilizando diferentes tipos como son: listas, pilas, colas, árboles con el fin de mejorar la eficiencia en el manejo de los mismos. Son utilizadas para el almacenamiento de datos del mundo real que pueden cambiar constantemente. Las listas enlazadas, en particular, son estructuras de datos implementadas mediante un grupo de nodos que juntos representan una secuencia, donde cada nodo tiene un dato y una referencia. Pilas, implementadas mediante listas enlazadas también, son también estructuras de datos tipo LIFO: último en entrar, primero en salir. Otra implementación de las listas enlazadas son las colas como estructuras de datos tipo FIFO, el primer elemento agregado será el primero que saldrá [17] [18].

2.1.9 React native

Es un framework desarrollado para escribir apps nativas utilizando JavaScript, para iOS y Android. Se basa en React, la biblioteca de JavaScript de Facebook para crear interfaces de usuarios, pero en lugar de dirigirse al navegador, se dirige a plataformas móviles [19].

2.1.10 Node js

Es un entorno de ejecución de JavaScript construida en el Motor de JavaScript de Google Chrome (Motor V8). Ofrece un entorno multiplataforma para el desarrollo de aplicaciones web del lado del servidor [20].

2.1.11 Framework Bootstrap

Es un framework, que se destaca por adaptar la interfaz al tamaño del dispositivo en el que se visualice, esta técnica se conoce como Resposive Design o Diseño Adaptativo.

Es el más popular para el desarrollo de sitios web adaptables y móviles. Bootstrap es un diseño de cuadrículas de 12 columnas que incluye clases y estilos CSS pre-construidos, además de algunas funciones de JavaScript. También tiene plantillas para el diseño de botones, imágenes, formas, navegación, etc [21] [22].

2.1.12 Metodología Híbrida SNAIL

La combinación de las metodologías ágiles y tradicionales se denomina metodología híbrida. La metodología híbrida permite una mejor fluidez en los proyectos. Por ende, para el desarrollo de una aplicación web la metodología SNAIL es adecuada, ya que es una metodología creada para el desarrollo de aplicaciones web, aunque se puede aplicar en forma general para aplicaciones de escritorio. Se basa en la simplicidad, comunicación y planificación de código [23].

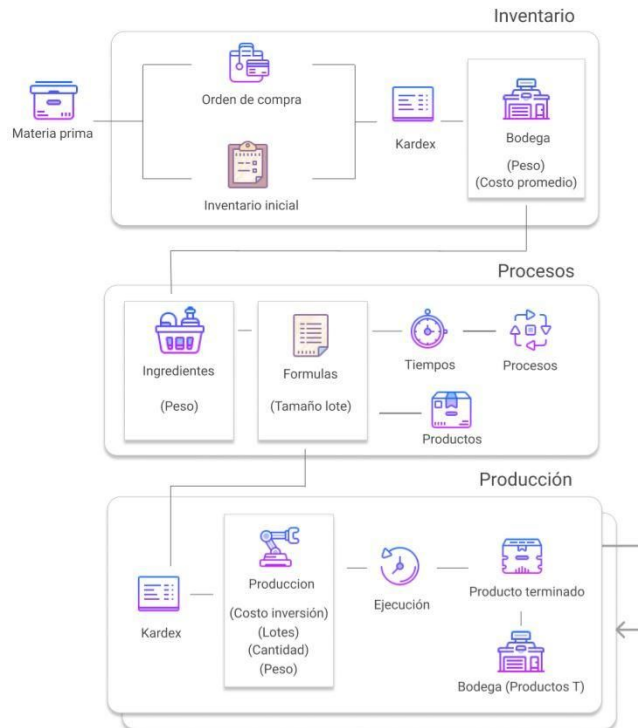
2.1.13 Métricas de calidad

Entre los aspectos más importantes en una aplicación se han encontrado: contenido, seguridad, navegabilidad, usabilidad y diseño. La navegabilidad ayuda al usuario a encontrar la información requerida mediante palabras claves. El contenido debe ser organizado, coherente, y relevante. El diseño es un aspecto importante, debe ser un diseño limpio, organizado y amigable. La usabilidad hace referencia a la facilidad de uso de una aplicación por parte del usuario. Seguridad se refiere a la protección de un sitio web contra accesos no autorizados y/o ataques cibernéticos [24].

2. 2 Solución del problema

2.2.1 Arquitectura Lógica del Sistema

Figura 1: Arquitectura Lógica del sistema



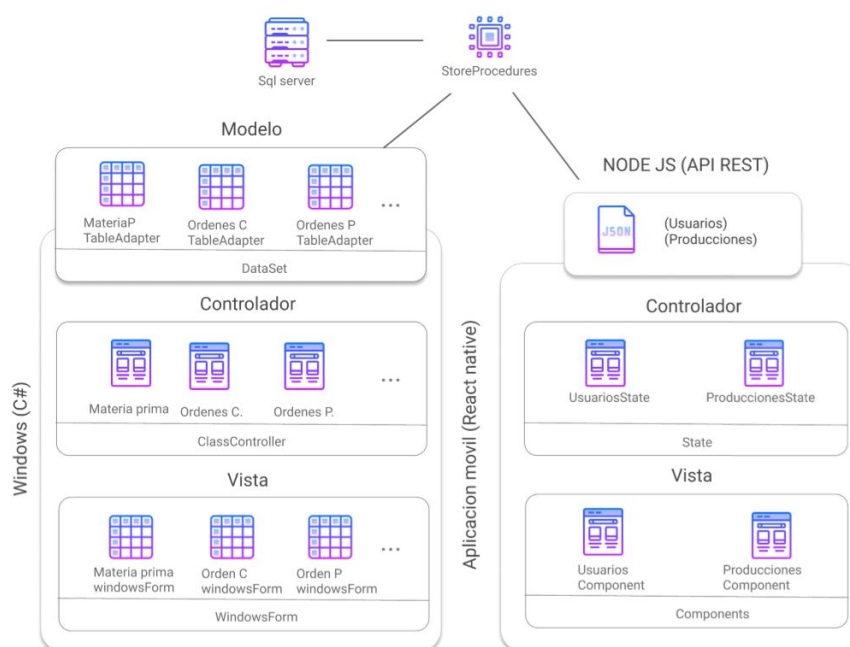
Fuente: Elaboración propia

Se puede observar, que en primer lugar tenemos la entrada de la materia prima; la cual puede ser por orden de compra o inventario inicial. Luego se registra la entrada en el kardex y pasa a bodega de materia prima. Posteriormente, a ingredientes donde se coloca el peso que se necesita de materia prima; estos ingredientes están dados por una fórmula, en la cual se establece el tamaño del lote a producir (en base a esta cantidad se producirá más o menos embutidos), cada fórmula se produce en un tiempo determinado y este tiempo comprende subprocesos (molienda, preparación de pasta, embutido, cocción, rebanado).

A continuación, se registra la salida en el kardex, luego pasa a producción donde se calcula los lotes totales entre fecha de inicio y fin de producción. Por último, se llega a la etapa de ejecución, donde una vez finalizada la producción, pasa a productos terminados y se guarda en bodega de productos terminados. (Ver Figura 1)

2.2.2 Arquitectura Tecnológica del Sistema

Figura 2: Arquitectura Tecnológica del sistema



Fuente: Elaboración propia

En esta arquitectura tenemos como primer punto la aplicación de escritorio, la cual se detalla de la siguiente manera: Como base de datos SQL Server, que contiene las tablas y procedimientos almacenados (para operaciones CRUD). Esto pasa a la aplicación de escritorio que entra al DataSet, el cual maneja la conexión y posee tableAdapter; si hay 10 tablas en la base de datos habrá 10 tableAdapter. Luego se conecta al controlador que usa clases que manejan la conexión del modelo con la vista, una clase para cada tableAdapter, pasa a la vista la cual posee formularios Windows que usan su controlador respectivo. En el segundo punto tenemos la parte de la aplicación móvil, primero se hace un APIrest en Node.js, la cual provee un JSON para usuarios y otro para producciones, luego pasa al controlador el cual usa las clases de React native para manejar esos JSON y proveerlos en formato State a la vista la cual maneja los componentes (partes de la interfaz), y los componentes usan el State para enviar o mostrar datos. (Ver Figura 2)

2.2.3 Materiales

En la Tabla 1 se muestra los materiales o recursos tecnológicos utilizados en el proyecto.

Tabla 1: Herramientas empleadas

Descripción	Herramientas
Lenguajes de desarrollo	C#
Marcos de trabajo (aplicativo móvil)	Bootstrap
Base de datos	SqlServer2014
Metodología	SNAIL
Estructuras de datos	Listas enlazadas (listas dinámicas, pilas y colas)
Aplicaciones	De escritorio y móvil

Fuente: Elaboración propia

2.2.4 Metodología Híbrida de desarrollo SNAIL

2.2.4.1 Fase 1: Análisis de requerimientos

En esta fase se determinó las expectativas del usuario en el desarrollo del nuevo producto, es decir, se realizó la recopilación de requisitos del sistema. Los requerimientos funcionales y no funcionales del sistema son importantes ya que son el punto de partida del desarrollo de un sistema [23]. (Ver Anexo A).

2.2.4.2 Fase 2: Planificación

En esta fase se realiza la estimación de tiempo del desarrollo del sistema, las estimaciones razonables de los entregables. Se establece las actividades del proyecto, recursos, tiempo, entre otros elementos que son necesarios establecer en esta etapa. La planificación debe ser organizada para el éxito de un proyecto [23].

2.2.4.3 Fase 3: Diseño

En esta fase se realizó la maquetación de las posibles interfaces gráficas del sistema a desarrollar. (Ver Anexo B). Además el diseño de la base de datos, la cual se administra mediante la herramienta SQL Server Management. Se realizó el diseño de las tablas relacionales de la base de datos (Ver Anexo C).

2.2.4.4 Fase 4: Programación

Figura 3: Guardar orden de compra

```
private void MetroTile1_Click(object sender, EventArgs e)
{
    OrdenCompraController oc = new OrdenCompraController();
    int idProveedor = Int32.Parse(orProveedor.SelectedValue.ToString());
    decimal subTotal = Decimal.Parse(orSubtotal.Text);
    decimal iva = Decimal.Parse(orIva.Text);
    decimal total = Decimal.Parse(orTotal.Text);
    //Se registra la orden de compra
    oc.registrarOrden(idProveedor, orFechaHoy.Text, orAplazo.Text, subTotal, iva, total);
    //Se registra la materia prima utilizada para la orden

    foreach (DataGridViewRow row in tablaDetalle.Rows)
    {
        if (row.Cells[0].Value != null)
        {
            decimal cant = Decimal.Parse(row.Cells[2].Value.ToString());
            decimal pre = Decimal.Parse(row.Cells[1].Value.ToString());
            int idb = Int32.Parse(row.Cells[3].Value.ToString());
            oc.generar(cant, "x", pre, oc.max_id_orden(), idb);
        }
    }
    //Se muestra la notificacion
    ordenCompraNoti.ShowBalloonTip(0);
}
}
```

Fuente: Elaboración propia

Este método fue creado para guardar órdenes de compra. Se registra la orden de compra y la materia prima utilizada en esa orden. Además se muestra la respectiva notificación.

Figura 4: Guardar orden de producción

```
private void MetroTile1_Click(object sender, EventArgs e)
{
    OrdenProduccionController opc = new OrdenProduccionController();
    DateTime inicio = DateTime.Parse(ordInicio.Text);
    string ini = inicio.ToString("d");
    DateTime fin = DateTime.Parse(ordFin.Text);
    string fi = fin.ToString("d");
    opc.insertar(id_formula_pro, ini, fi, Decimal.Parse(orCantidad.Text), Decimal.Parse(orCostoT.Text), Int32.Parse(proLote.Text), diasF, pesoFinal);
    registrarMateria();
    actualizarInventario(id_formula_pro);
}
}
```

Fuente: Elaboración propia

Este método fue creado para guardar las órdenes de producción. Orden de producción primero toma la fórmula, fecha de inicio y fin de la producción, cantidad de embutidos a producir y calcula los lotes necesarios para la producción de esos embutidos, el costo total de inversión, peso total, además el peso y costo de inversión de cada materia prima utilizada por la fórmula. Este método es importante porque da inicio a la producción del embutido a realizar.

Figura 5: Monitorear producción

```
private void MonitoreoProceso2_Load(object sender, EventArgs e)
{
    MonitoreoController mc = new MonitoreoController();
    OrdenProduccionController opc = new OrdenProduccionController();
    estadoA.Text = estado;
    decimal minutosActuales = cantMinutosActual();
    if (estado!="Terminado")
    {
        moProceso.Text = proceso(cantMinutosActual());
        moLote.Text = tiempoLoteAjustado(cantMinutosActual(), id_procesoActual(cantMinutosActual())).ToString();
        moPeso.Text = pesoLote(Decimal.Parse(moLote.Text)).ToString();
        moCantidad.Text = cantidadLote(Decimal.Parse(moPeso.Text)).ToString();
        cargarGraficoTabla();
        cargarGraficoLinea();
    }
    else
    {
        moProceso.Text = "...";
    }
    minutosActuales = 0;
}

//moLote.Text = mc.loteActual2(id_produccion).ToString();
if (minutosActuales!=0)
{
    decimal tempo = Math.Ceiling((minutosTotales() - minutosActuales))/60;
    TimeSpan timespan = TimeSpan.FromHours((double)tempo);
    moTiempo.Text = timespan.ToString("h\\:mm");
}
else
{
    moTiempo.Text = "0";
}
}
```

Fuente: Elaboración propia

Este método nos permite monitorear el estado de la orden creada en el método anterior.

2.2.4.5 Fase 5: Pruebas

2.2.4.5.1 Aplicación de escritorio

Figura 6: Orden de compra

Nombre	Costo	Peso (kg)	
Cabeza de lomo de cerdo	6	5000	1,44
Espaldilla de res	6	5000	13,44

Fuente: Elaboración propia

En la figura 6, se observa el proceso de generar orden de compra, con los siguientes datos: fecha de emisión, proveedor, fecha de plazo y materia prima. La materia prima se detalla en una tabla, por último se tiene los datos de IVA, Subtotal y Total.

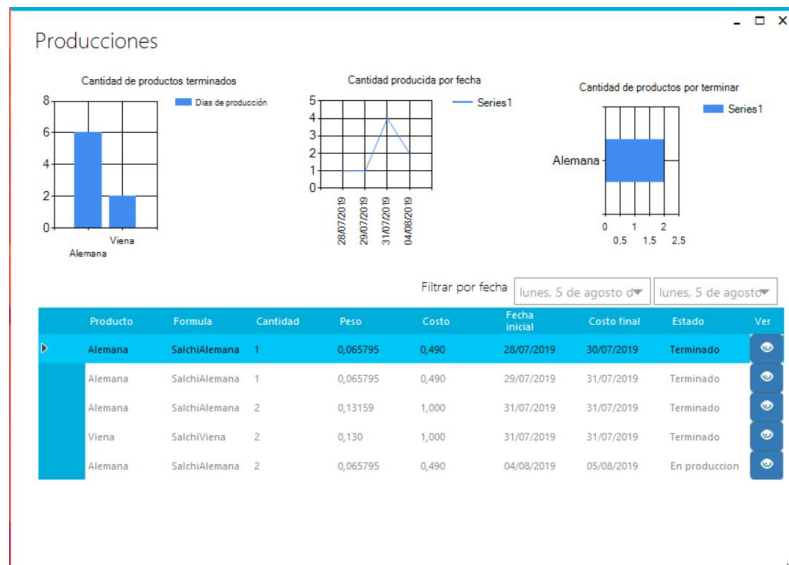
Figura 7: Monitoreo de producción



Fuente: Elaboración propia

En la figura 7, se observa el estado de la producción, el lote, cantidad, peso y tiempo restante de la producción. Además de gráficos estadísticos sobre el costo de inversión actual y en espera, y una tabla con el tiempo por procesos de producción.

Figura 8: Producciones



Fuente: Elaboración propia

En la figura 8, se visualiza una ventana de aplicación de escritorio con gráficos estadísticos: primero cantidad de productos terminados, segundo cantidad producida por fecha y tercero cantidad de productos por terminar. Además, se puede visualizar órdenes de producción por fechas.

2.2.4.5.1 Aplicación móvil

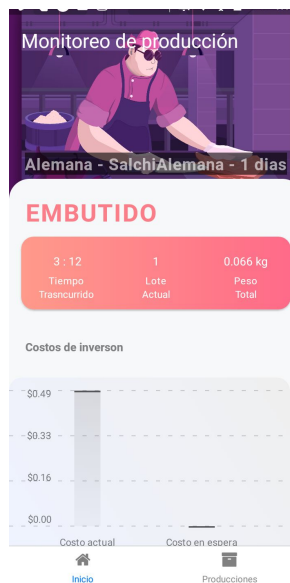
Figura 9: Página principal del aplicativo móvil



Fuente: Elaboración propia

En la figura 9, se puede observar la ventana principal del aplicativo móvil. Un dashboard que permite visualizar la producción y el estado de los productos.

Figura 10: Monitoreo de producción del aplicativo móvil



Fuente: Elaboración propia

En la figura 10, se puede observar el estado de la producción, el lote, cantidad, peso y tiempo restante. Además de gráficos estadísticos sobre el costo de inversión actual y en espera.

2.3 Resultados

En primer lugar se encuentra la realización de la aplicación de escritorio, la cual se desarrolló en el lenguaje C Sharp mediante la herramienta Visual Studio 2019, utilizando la arquitectura modelo-vista-controlador. La administración de la base de datos se realizó mediante la herramienta SQL Server Management. Además se empleó las estructuras de datos de listas enlazadas en la creación de órdenes de compra y kardex; colas en órdenes de producción, monitoreo de estado de producción; y finalmente pilas para el empaquetado.

El sistema cuenta con un control de ingreso en donde deberá ingresar usuario y contraseña. Dentro de las funciones principales de la aplicación de escritorio, se encuentran las siguientes (Ver Anexo D):

- Gestión de usuarios.
- Gestión de roles.
- Gestión Materia Prima.
- Gestión de Procesos.
- Gestión de Fórmulas.
- Gestión de Ingredientes.
- Gestión Órdenes de compra.
- Gestión Órdenes de producción.
- Monitoreo de producción.
- Gestión de Inventarios.

La aplicación móvil se realizó mediante las herramientas React Native y Node.js, la cual tiene las siguientes funciones (Ver Anexo E):

- Órdenes de Producción.
- Monitoreo de órdenes pasadas: ver el historial de las órdenes realizadas.
- Monitoreo de órdenes en producción: producciones que se están realizando.
- Control de usuario: donde el usuario puede iniciar sesión y puede editar sus datos.

3. CONCLUSIONES

Como resultado del sistema para la gestión y control de procesos de producción de embutidos se concluye que:

- Se logró implementar exitosamente una aplicación de escritorio y otra móvil mediante ingeniería de software y listas dinámicas para la gestión y control de procesos de producción de embutidos
- A través de artículos científicos se adquirió los conocimientos relacionados a controles de procesos de materia prima para la elaboración de embutidos.
- Basado en los requisitos del sistema, se utilizó el lenguaje C# con gestor de base de datos SQL Server para la aplicación de escritorio. Para la aplicación móvil se utilizó las herramientas React Native y Node.js
- Se utilizó la metodología híbrida SNAIL para el desarrollo de los sistemas.
- Se aplicó correctamente listas dinámicas (listas enlazadas, pilas y colas), para la estructuración de los datos en el desarrollo de la aplicación de escritorio.

BIBLIOGRAFÍA

- [1] INEC, «Perfil de mercado embutidos. ProChile Guayaquil,» 2007. [En línea]. Available: http://www.exportapymes.com/documentos/productos/Pe1434_ecuador_embutidos.pdf.
- [2] A. Hernández P., K. Pérez T. y O. Correa M., «Engineering framework for the videogame development process,» *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software (RACCIS)*, vol. 1, n° 7, pp. 13-26, 2017.
- [3] J. L. Harrington, *Relational Database Design and Implementation*, Elsevier, 2016.
- [4] D. M. Gonzáles Rodríguez, E. Giraldo Lopera y D. A. Restrepo Molina, «Evaluación sensorial e instrumental de textura de salchichones tipo estándar que contienen un extensor cárnico de pasta de pollo,» *Vitae*, vol. 22, n° 2, pp. 101-110, 2015.
- [5] R. Nascimento, A. Fonseca, R. Franco y Z. Miranda, «Bacteriological Characteristics of Fresh Ostrich Sausage (Linguiça),» *Revista Brasileira de Ciência Avícola*, vol. 17, n° 2, pp. 151-157, 2015.
- [6] A. Hernández y A. B. T. O. A. Di Iorio, «Contenido de azúcar, grasa y sodio en alimentos comercializados en Honduras, según el etiquetado nutricional,» *Revista española de nutrición humana y dietética*, vol. 22, n° 2, pp. 108-116, 2018.
- [7] A. Nugraha, S. Sukardi y A. Rifin, «EFFICIENCY OF RAW MATERIAL INVENTORIES IN IMPROVING SUPPLY CHAIN PERFORMANCE of CV. FIVA FOOD,» *Indonesian Journal of Business and Entrepreneurship*, vol. 1, n° 1, pp. 23-32, 2016.
- [8] J. L. Cardona Tunubala, J. P. Orejuela Cabrera y C. A. Rojas Trejos, «Gestión de inventario y almacenamiento de materias primas en el sector de alimentos concentrados,» *Revista EIA*, vol. 15, n° 30, pp. 195-208, 2018.
- [9] D. Mercado Polo, L. Pedraza Caballero y E. Martínez Gómez, «Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo,» *Prospectiva*, vol. 13, n° 2, pp. 88-95, 2015.
- [10] A. Pinandito, F. B. Pristiawan Permana y R. Setya Perdana, «Framework Design for Modular Web-based Application Using Model-CollectionService-Controller-Presenter (MCCP) Pattern,» *JITeCS (Journal of Information Technology and Computer Science)*, vol. 2, n° 1, pp. 41-65, 2017.
- [11] J. M. Suárez y L. E. Gutiérrez, «Tipificación de Dominios de Requerimientos para la Aplicación de Patrones Arquitectónicos,» *Información tecnológica*, vol. 27, n° 4, pp. 193-202, 2016.
- [12] R. Čerešňák y M. Kvet, «Comparison of query performance in relational a non-relation databases,» *Transportation Research Procedia*, vol. 40, pp. 170-177, 2019.

- [13] J. Gabillaud, SQL Server 2014: Administración de una base de datos transaccional con SQL, Barcelona: Eni Ediciones, 2015.
- [14] T. A. Dmitrieva, A. V. Prutzkow y A. N. Pylkin, «TWO-LEVEL STUDY OF OBJECT-ORIENTED PROGRAMMING BY UNIVERSITY STUDENTS,» *Sovremennye Informacionnye Tehnologii i IT-obrazovanie.*, vol. 15, nº 1, pp. 200-206, 2019.
- [15] C. García Eberth, «Sistema integrado web para la atención de pacientes del policlínico Juan Pablo II - Chimbote,» 2018. [En línea]. Available: http://repositorio.usanpedro.edu.pe/bitstream/handle/USANPEDRO/6020/Tesis_57538.pdf?sequence=1&isAllowed=y.
- [16] S. Mukherjee, Aprende Microsoft Visual Studio App Center with Xamarin Forms, Tampa: Apress, 2019, p. 330.
- [17] L. Joyanes Aguilar, Fundamentos de programación Algoritmos, estructura de datos y objetos, Madrid: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, 2008.
- [18] P. A. Sznajdleder, Algoritmos a fondo con implementaciones en C y Java, Buenos Aires: Alfaomega, 2017.
- [19] B. Eisenman, Learning React Native: Building Native Mobile Apps with JavaScript, O. Media, Ed., 2015, p. 272.
- [20] D. M. O'Halloran, «phylo-node: A molecular phylogenetic toolkit using Node.js.,» *PLoS ONE*, vol. 12, nº 4, pp. 1-8, 2017.
- [21] Y. Manso Guerra, R. Cañizares González y J. P. Febles, «Diseño web adaptativo para la plataforma educativa ZERA,» *Revista Cubana de Ciencias Informáticas*, vol. 10, nº 2, pp. 100-115, 2016.
- [22] F. Luna, Programación Web Full Stack 12 - Sitios multiplataforma con Bootstrap, Argentina, 2018.
- [23] J. R. Molina Ríos, M. P. Zea Ordóñez, F. F. Redrován Castillo, N. M. Loja Mora, M. R. Valarezo Pardo y J. A. Honores Tapia, SNAIL, una metodología híbrida para el desarrollo de aplicaciones web, 3ciencias, 2018.
- [24] F. F. Redrován Castillo, N. M. Loja Mora, K. D. Correa Elizaldes y J. I. Piña Orozco, «Estado del Arte: Métricas de calidad,» *3C Tecnología: glosas de innovación aplicadas a la pyme*, vol. 6, nº 4, pp. 1-12, 2017.
- [25] A. N. L. Fernando, «Desarrollo De Aplicación Web Para Gestión De Empleados Usando Metodología Snail Y Lenguaje De Programación Java.Net,» Machala, 2019.

ANEXOS

Anexo A. Requisitos funcionales y no funcionales del sistema (formato tomado de [25])

Tabla 2: RF01 – Gestión de inventarios

Id del requisito	RF01
Nombre de requisito	Gestión de inventarios
Tipo	Funcional
Descripción	Se debe controlar el inventario de la materia prima que se almacena en las bodegas o centros de acopios con sus respectivos proveedores, productos, categorías, órdenes de compra, etc.
Actores	Usuario
Prioridad	Alta
Condición de éxito	Se pudo controlar el inventario de materia prima
Condición de fracaso	Incorrecto control del inventario

Fuente: Elaboración propia

Tabla 3: RF02 – Reporte Kardex

Id del requisito	RF02
Nombre de requisito	Reporte Kardex
Tipo	Funcional
Descripción	Se requiere de un reporte (Kardex) que muestre las entradas y las salidas de los productos, así como la cantidad y su valor.
Actores	Sistema
Prioridad	Alta
Condición de éxito	Implementación exitosa del Kardex
Condición de fracaso	Datos incoherentes

Fuente: Elaboración propia

Tabla 4: RF03 – Control de stock

Id del requisito	RF03
Nombre de requisito	Control de stock
Tipo	Funcional
Descripción	El software debe proveer alertas que indiquen en base a un mínimo y un máximo, cuanto falta de materia prima o si está por encima del límite máximo.
Actores	Sistema
Prioridad	Alta
Condición de éxito	Correcto control del stock
Condición de fracaso	No presenta las alarmas de control de stock

Tabla 5: RF04 – Gestión de fórmulas

Id del requisito	RF04
Nombre de requisito	Gestión de fórmulas
Tipo	Funcional
Descripción	Se debe tener un registro de las fórmulas en la mínima unidad de peso o litro, para cada tipo de embutido que se va a crear. Ejemplo: 1 Libra de carne pura, 1 fundas, entre otros ingredientes ya parametrizados.
Actores	Sistema
Prioridad	Alta
Condición de éxito	Registro correcto de fórmulas para la elaboración de embutidos
Condición de fracaso	Fórmulas inconsistentes

Fuente: Elaboración propia

Tabla 6: RF05 – Gestión de producción

Id del requisito	RF05
Nombre de requisito	Gestión de producción
Tipo	Funcional
Descripción	En cuando la etapa de producción de los embutidos, se debe registrar la orden de producción la cual tomará una fórmula, y dependiendo de la cantidad de embutidos a producir, se calculara la cantidad y total de materia prima a utilizar. Esto debe permitir ver los costos de inversión para tal producción.
Actores	Usuario
Prioridad	Alta
Condición de éxito	Registro correcto de órdenes de producción
Condición de fracaso	No se registran las órdenes de producción

Fuente: Elaboración propia

Tabla 7: RF06 – Gestión de órdenes de producción

Id del requisito	RF06
Nombre de requisito	Gestión de órdenes de producción
Tipo	Funcional
Descripción	Las diferentes orden producción transitan por diferentes etapas o estaciones de producción, las cual son: Molienda, Preparación de pasta, Embutido, Cocción, Rebanado y Empaque.
Actores	Usuario
Prioridad	Alta
Condición de éxito	Se registró correctamente las órdenes de producción
Condición de fracaso	No se registran las órdenes de producción

Fuente: Elaboración propia

Tabla 8: RF07 – Proceso de producción

Id del requisito	RF07
Nombre de requisito	Proceso de producción
Tipo	Funcional
Descripción	Es necesario planificar los tiempos de producción, dicho de esta manera una orden de producción tendrá una fecha de inicio y de fin. Así también sus etapas estarán sujetas a estas fechas generales.
Actores	Usuario
Prioridad	Alta
Condición de éxito	Se estableció fechas de inicio y fin de las órdenes de producción
Condición de fracaso	No se establece las fechas de las órdenes de producción

Fuente: Elaboración propia

Tabla 9: RF08 – Productos terminados

Id del requisito	RF08
Nombre de requisito	Productos terminados
Tipo	Funcional
Descripción	Cuando la producción termina, todo pasa a un control de productos terminados, para ello es necesario supervisar su inventario
Actores	Usuario
Prioridad	Alta
Condición de éxito	Se estableció un inventario de productos terminados
Condición de fracaso	No se lleva un inventario de productos terminados

Fuente: Elaboración propia

Tabla 10: RF09 – Monitoreo de producción

Id del requisito	RF09
Nombre de requisito	Productos terminados
Tipo	Funcional
Descripción	Para la parte gerencial solicita llevar un control de observación de la planificación de la producción de embutidos con sus respectivos estados.
Actores	Usuario
Prioridad	Alta
Condición de éxito	Se estableció un monitoreo del estado de producción
Condición de fracaso	No se realiza el monitoreo del estado de producción

Fuente: Elaboración propia

Tabla 11: RNF01 – Arquitectura MVC

Id del requisito	RNF01
Nombre de requisito	Diseñar la solución utilizando una arquitectura cliente-servidor
Tipo	No Funcional
Descripción	El sistema será desarrollado mediante la arquitectura Modelo-Vista-Controlador
Prioridad	Alta/Esencial

Fuente: Elaboración propia

Tabla 12: RNF02 – Control de usuarios

Id del requisito	RNF02
Nombre de requisito	Control de usuarios
Tipo	No Funcional
Descripción	El sistema permitirá acceder solamente a los usuarios registrados a las funcionalidades asignadas.
Prioridad	Alta/Esencial

Fuente: Elaboración propia

Tabla 13: RNF03 – Validaciones

Id del requisito	RNF03
Nombre de requisito	Validaciones
Tipo	No Funcional
Descripción	El sistema debe realizar las respectivas validaciones de ingreso y salida de datos
Prioridad	Media

Fuente: Elaboración propia

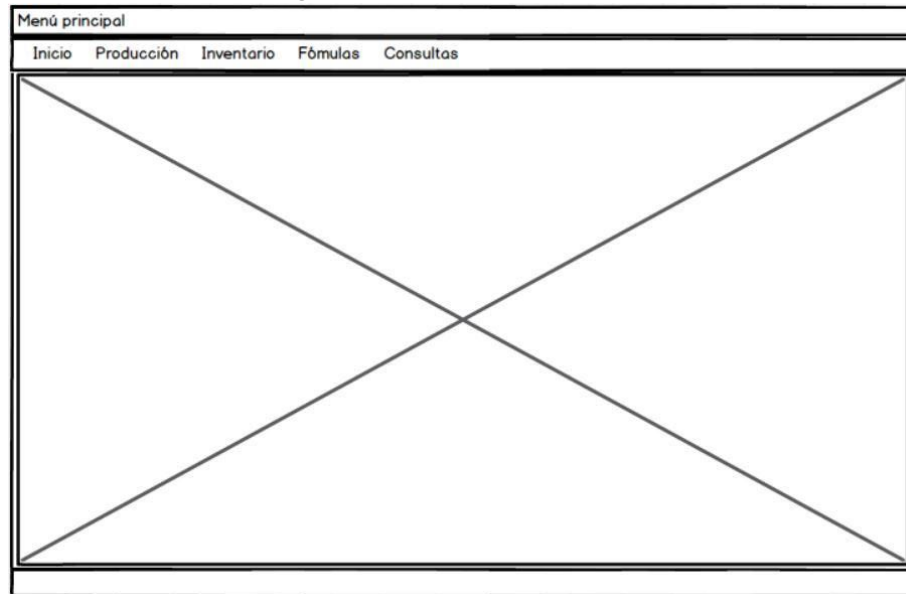
Tabla 14: RNF03 – Aplicación móvil

Id del requisito	RNF04
Nombre de requisito	Validaciones
Tipo	No Funcional
Descripción	Se debe implementar un módulo móvil que garantice alta usabilidad en diferentes dispositivos móviles (Smart phones o tablets).
Prioridad	Alta/Esencial

Fuente: Elaboración propia

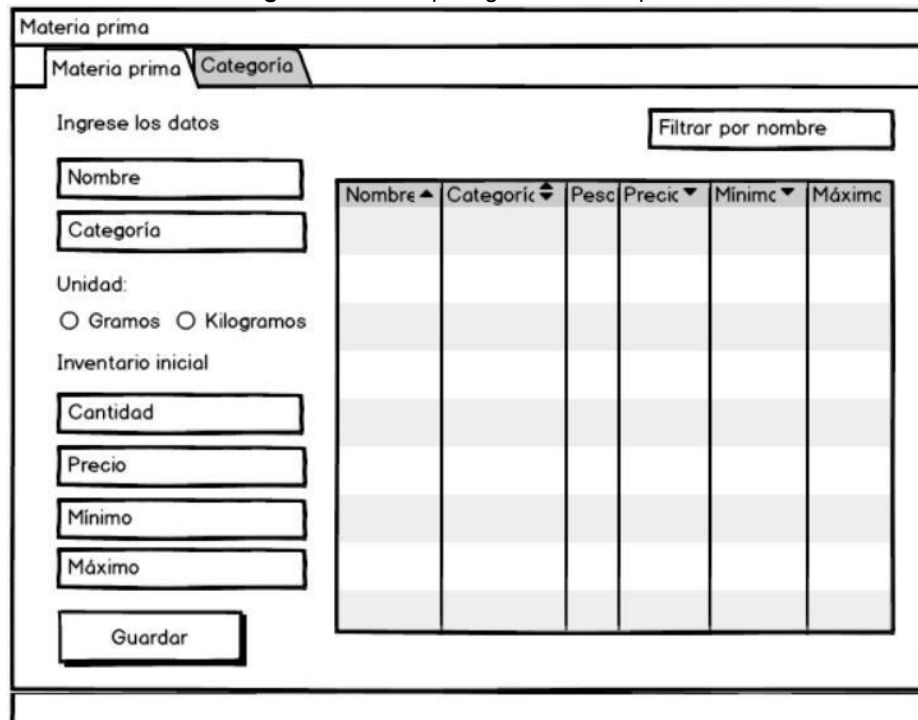
Anexo B. Prototipos de interfaces

Figura 11: Prototipo Menú Principal



Fuente: Elaboración propia

Figura 12: Prototipo Ingreso materia prima



Materia prima

Materia prima Categoría

Ingrese los datos Filtrar por nombre

Nombre

Categoría

Unidad:
 Gramos Kilogramos

Inventario inicial

Cantidad

Precio

Mínimo

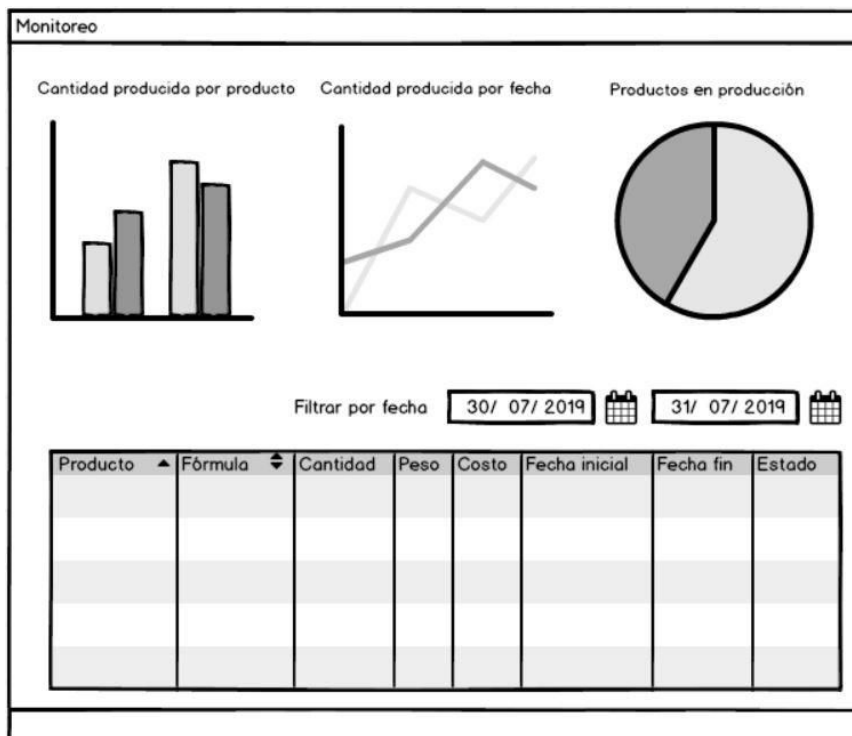
Máximo

Nombre ▲	Categoría ▼	Peso	Precio ▼	Mínimo ▼	Máximo

The image shows a window titled "Materia prima" with a tabbed interface. The "Materia prima" tab is selected. The form contains several input fields: "Nombre", "Categoría", "Unidad" (with radio buttons for "Gramos" and "Kilogramos"), "Inventario inicial" (with sub-fields for "Cantidad", "Precio", "Mínimo", and "Máximo"), and a "Guardar" button. To the right of the form is a table with 6 columns: "Nombre", "Categoría", "Peso", "Precio", "Mínimo", and "Máximo". Each column has a small arrow icon indicating it is sortable. The table has 10 rows, with the first row being the header and the remaining 9 rows being empty.

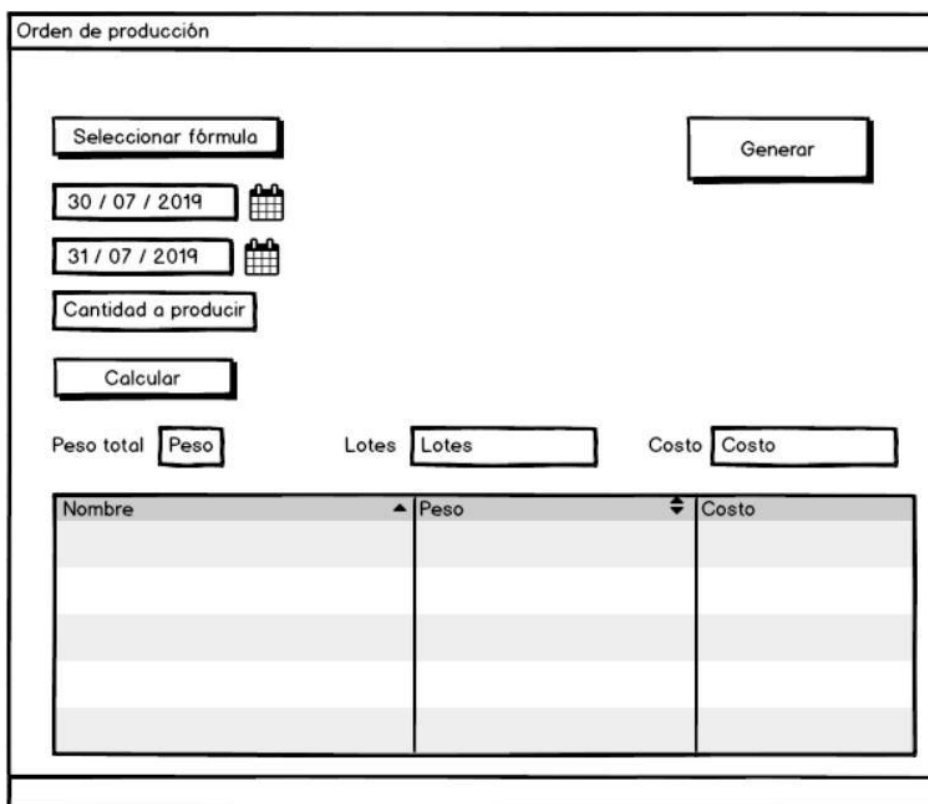
Fuente: Elaboración propia

Figura 13: Prototipo monitoreo producciones



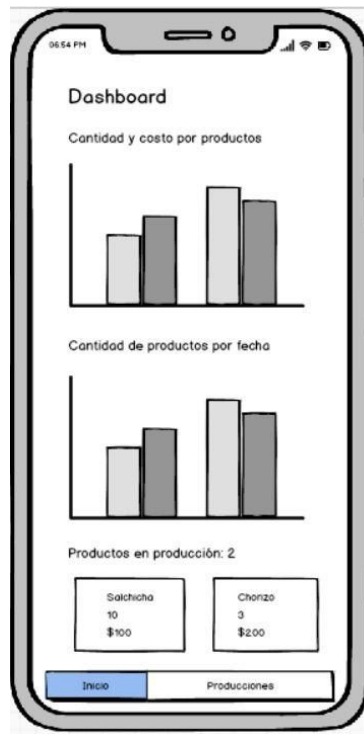
Fuente: Elaboración propia

Figura 14: Prototipo orden de producción



Fuente: Elaboración propia

Figura 15: Prototipo Dashboard aplicación móvil



Fuente: Elaboración propia

Figura 16: Prototipo Inicio de Sesión aplicación móvil

Inicio de Sesión

Usuario

Clave

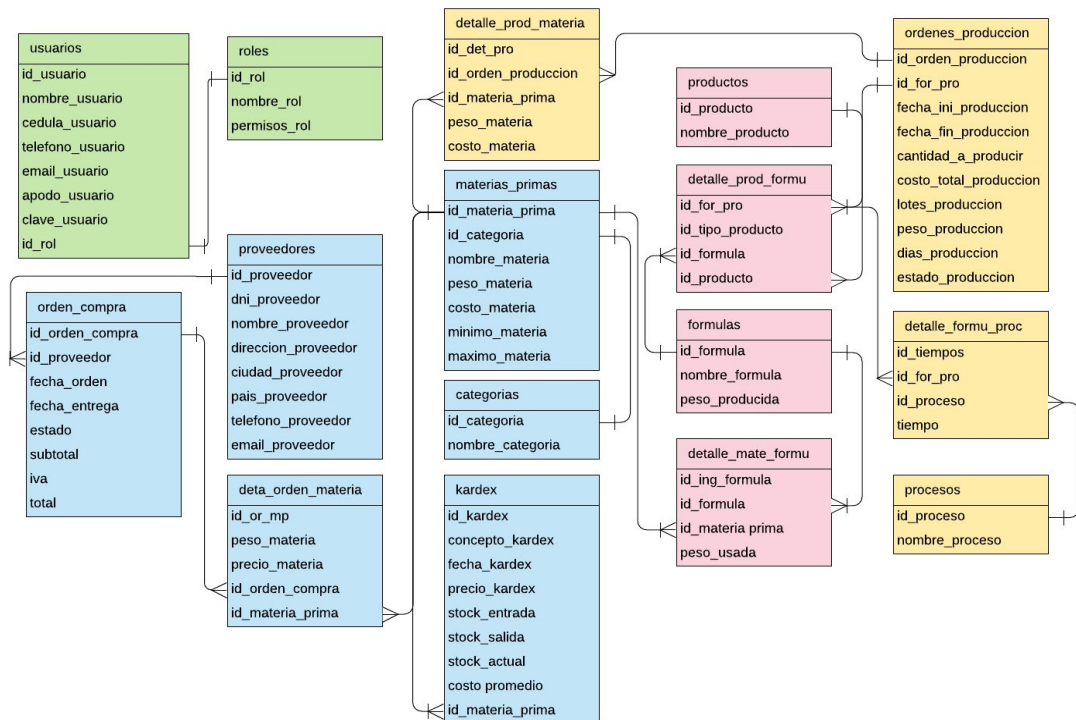
INICIAR SESIÓN

Monitoreo gerencial de producciones

Fuente: Elaboración propia

Anexo C. Diagrama de base de datos relacional

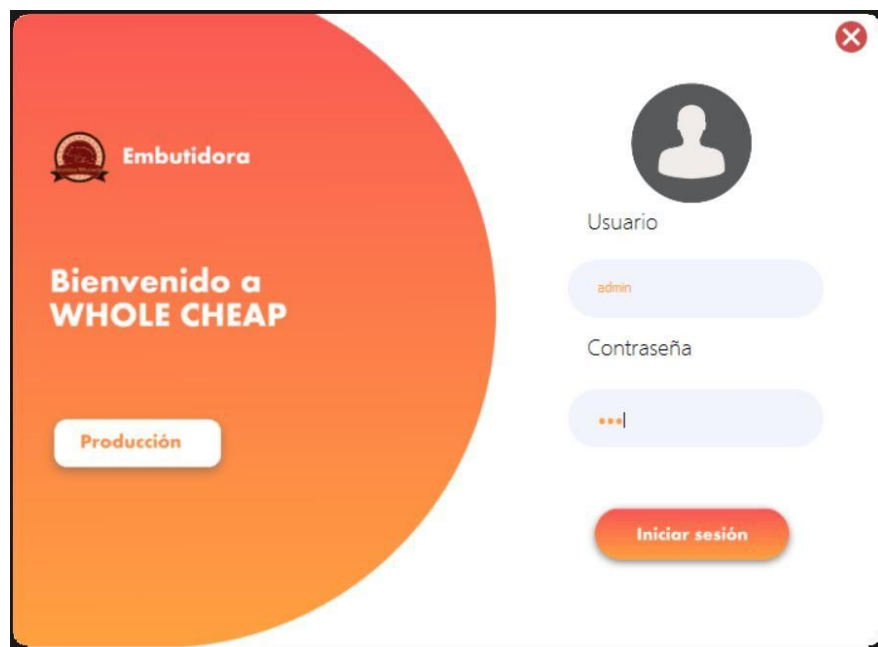
Figura 17: Diagrama de base de datos relacional



Fuente: Elaboración propia

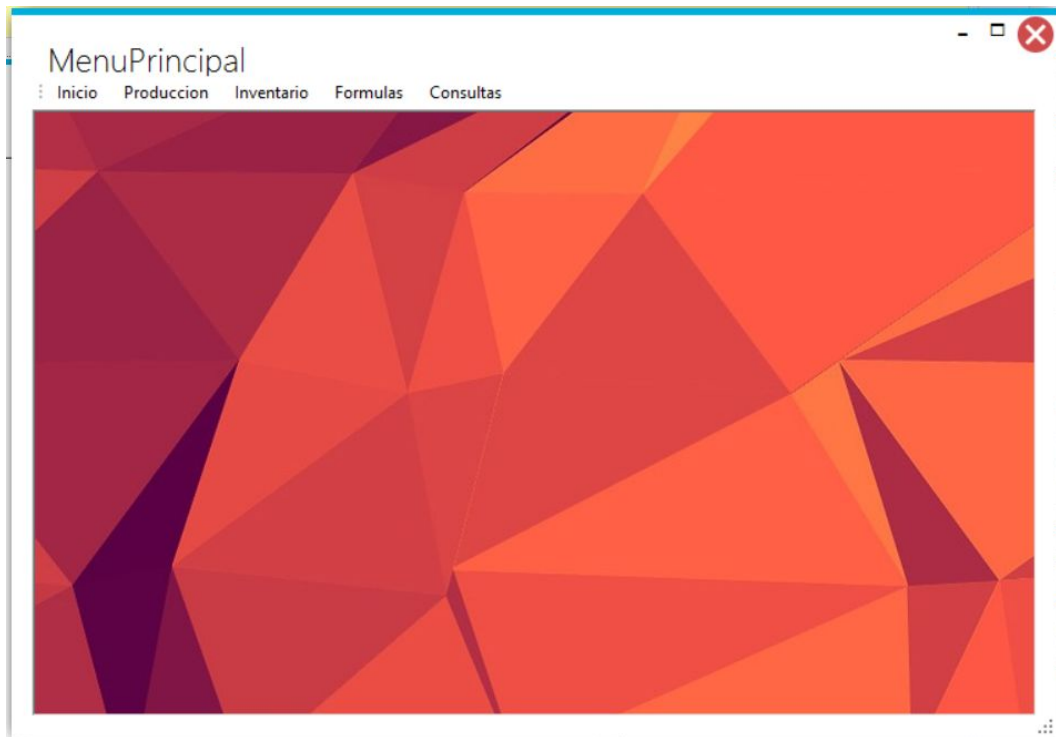
Anexo D. Aplicación de escritorio

Figura 18: Login del Sistema



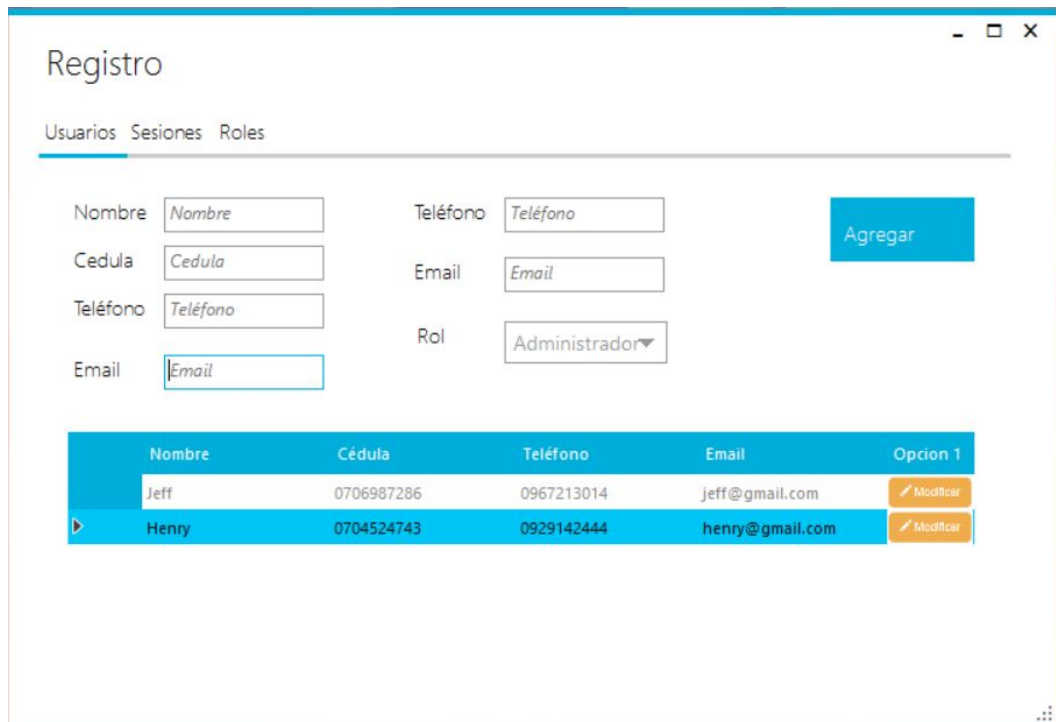
Fuente: Elaboración propia

Figura 19: Menú principal



Fuente: Elaboración propia

Figura 20: Registro de usuarios



Fuente: Elaboración propia

Figura 21: Kardex Materia Prima

Concepto	Fecha	Costo	Entrada	Salida	Stock actual	Materia p
Entrada	24/07/2019	235,500	50000,000	0,000	50000,000	Cabeza de
Entrada	24/07/2019	300,000	50000,000	0,000	50000,000	Espaldilla
Entrada	24/07/2019	100,000	50000,000	0,000	50000,000	Grasa de r
Entrada	24/07/2019	24,500	50000,000	0,000	50000,000	Sal
Entrada	24/07/2019	125,000	50000,000	0,000	50000,000	Dientes d
Entrada	24/07/2019	400,000	50000,000	0,000	50000,000	Mejorana
Entrada	24/07/2019	300,000	50000,000	0,000	50000,000	Anis seco
Entrada	24/07/2019	750,000	50000,000	0,000	50000,000	Nuez mos

Fuente: Elaboración propia

Figura 22: Orden de compra materia prima

Seleccione la materia prima Filtrar por nombre

Materia prima	Categoría	Peso	Costo	Mínimo
Cabeza de lomo de cerdo	Cerdo	26030,610	235,057	10,000
Espaldilla de res	Res	43059,390	300,000	10,000
Grasa de espalda de cerdo	Cerdo	16191,250	100,000	10,000
Sal	Condimentos	75667,500	24,500	10,000

Unidad:
 Gramos Kilogramos

Cabeza de lomo de cerd

Peso

Costo unitario (\$)

CONFIRMAR

Fuente: Elaboración propia

Figura 23: Creación de fórmulas

The 'Formulas' window contains the following elements:

- Input fields for 'Nombre' (empty), 'Producto' (dropdown), '63', and 'Peso por lote (gr)' (empty).
- Buttons: 'Guardar', 'Agregar ingredie...', and 'Asignar tiempos'.
- Table of ingredients:

Nombre	Peso (gr)
Cabeza de lomo de cerdo	50
Sal	5
Mejorana seca	8

Fuente: Elaboración propia

Figura 24: Orden de producción

The 'Orden de producción' window contains the following elements:

- Buttons: 'Seleccionar formula' and 'Generar orden'.
- Date pickers: 'lunes, 29 de julio de'.
- Input field: '1'.
- Button: 'Calcular'.
- Summary fields: 'Peso total' (0,065795), 'Lote' (1), 'Costo total' (0,49).
- Table of ingredients:

Nombre	Peso (kg)	Costo
Cabeza de lomo de ce...	0,01788	0,16141
Espaldilla de res	0,01612	0,11234
Grasa de espalda de c...	0,01375	0,08492
Dientes de ajo	0,001	0,00138
Pimienta blanca	0,00312	0,03368
Mejorana seca	0,00075	0,00324
Anís seco	0,00112	0,00379
Nuez moscada	0,00088	0,00717

Fuente: Elaboración propia

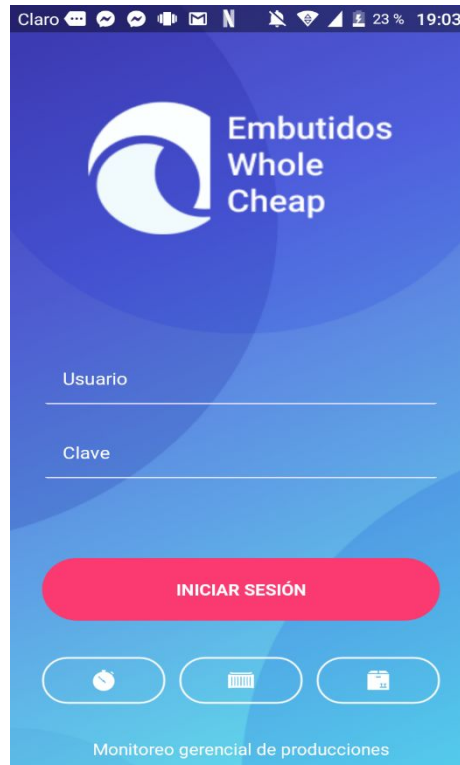
Anexo E. Aplicación móvil

Figura 25: Página principal de la aplicación móvil



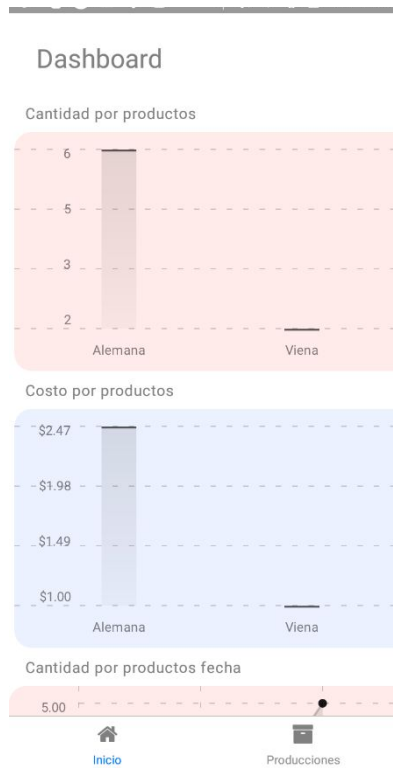
Fuente: Elaboración propia

Figura 26: Inicio de Sesión de la aplicación móvil



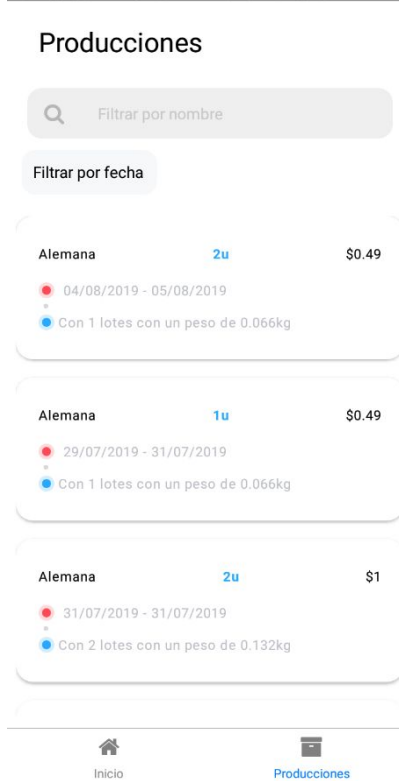
Fuente: Elaboración propia

Figura 27: Dashboard aplicación móvil



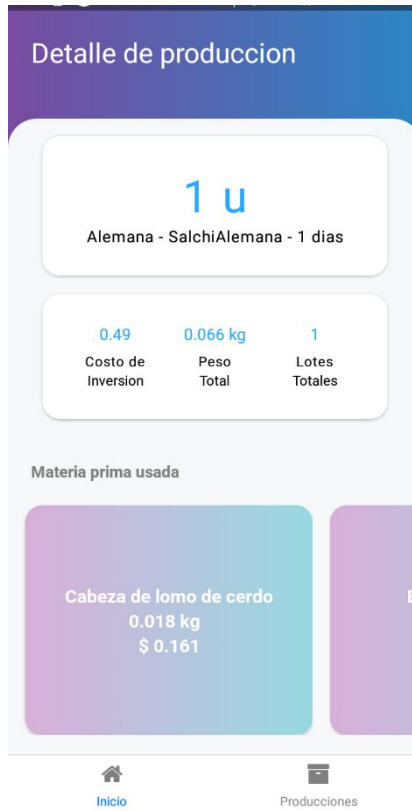
Fuente: Elaboración propia

Figura 28: Órdenes de producción activas



Fuente: Elaboración propia

Figura 29: Detalles de producción



Fuente: Elaboración propia

Figura 30: Monitoreo de producción



Fuente: Elaboración propia