

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

Sistema de Venta Alimenticia Personalizada

**Vicente Alejandro Retamal Aravena
Gustavo Lisandro Saavedra Saavedra**

Profesor Guía: **Rodrigo Alfaro Arancibia**

INFORME FINAL DE PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO EJECUCIÓN INFORMÁTICA

DICIEMBRE, 2016

Resumen

El proyecto presentado en este trabajo consiste en el diseño y construcción de un sistema que posibilite la venta de comida rápida en base a el peso y el valor de cada ingrediente en específico. Es así como el sistema construido permite conocer en todo momento la cantidad del ingrediente, el costo de este y el total de lo que el cliente va a pagar. Además la solución posee la capacidad de revisar el inventario en tiempo real, y conocer cuando se debe reponer cada ingrediente. Para generar la solución, fueron utilizados prototipos en base a *Arduino* y una serie de sensores, y plataformas como *Angular* y *Electron* para desarrollar la vista, así como *MongoDB* para el almacenamiento de información.

Palabras Claves. Alimentación, Arduino, Electron, Angular, MongoDB.

Índice

1. Introducción	1
2. Visión General del Proyecto	2
2.1. Propósito	2
2.2. Alcance	2
2.3. Objetivos	2
3. Situación en Estudio	3
3.1. Contexto Organizacional	3
3.1.1. Misión Organización	3
3.2. Descripción situación actual	4
3.3. Identificación de los problemas detectados	5
3.4. Estado del arte	5
4. Gestión del Proyecto	6
4.1. Estudio de factibilidad	6
4.1.1. Factibilidad Técnica	6
4.1.2. Factibilidad Económica	6
4.1.3. Factibilidad Operacional	7
4.2. Recursos requeridos para la solución	7
4.2.1. Recursos Tecnológicos	7
4.2.2. Recursos Humanos	8
4.2.3. Recursos Económicos	8
4.3. Carta Gantt	8
4.4. Gestión del Riesgo	8
4.4.1. Plan de mitigación de riesgos	10
5. Detalle de la Solución	11
5.1. Descripción general de la Solución	11
5.2. Paradigma y Metodología de desarrollo	12
5.3. Arquitectura lógica y física de la solución	13
5.4. Herramientas de desarrollo	14
5.4.1. Software	14
5.4.2. Hardware	14
6. Análisis de Requerimientos	15
6.1. Identificación de Actores involucrados	15
6.1.1. Administrador	15
6.1.2. Jefe de Local	15
6.1.3. Preparador de jugos	15
6.1.4. Preparador de sándwich	15
6.1.5. Cajero	15
6.1.6. Cocinero	15
6.1.7. Cliente	16
6.2. Requerimientos Funcionales	16
6.3. Requerimientos No Funcionales	17
6.4. Historias de usuario	17

7. Desarrollo del Proyecto	20
7.1. Componentes de Software	20
7.2. Base de datos	20
7.3. Interfaz	21
7.4. Implementación	25
7.4.1. Dispositivo Pesa	26
7.4.2. Broker	26
7.5. Servidor NodeJS	26
7.6. API REST	26
7.7. Base de datos	26
7.7.1. Aplicación Cocina	26
7.7.2. Aplicación Caja	27
7.7.3. Aplicación Administración	27
8. Plan de pruebas	28
8.1. Objetivos	28
8.1.1. Tareas	28
8.1.2. Alcance	28
8.1.3. Estrategia de Testing	28
8.1.4. Planificación de Testing	29
8.1.5. Procedimientos de Control	29
8.1.6. Características a ser testeadas	30
8.1.7. Características que no serán testeadas	31
8.1.8. Roles y responsabilidades	31
8.1.9. Riesgos	31
8.1.10. Criterios de aprobación	32
8.1.11. Pruebas de usabilidad	32
8.2. Pruebas	32
8.2.1. Resultados Pruebas	32
9. Conclusión	33
10. Anexos	34
10.1. Carta Gantt	34
10.2. Pruebas	35
10.2.1. Caja	35
10.2.2. Monitor Cocina	36
10.2.3. Administración	38

Glosario

Agil Envuelve un enfoque de desarrollo de software para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. 12

Angular Es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. 14

API Rest Corresponde a cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. 13

Arduino Consiste en una placa de circuito impreso con un microcontrolador, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (shields), que amplían las características de funcionamiento de la placa Arduino. Asimismo, posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación con el computador. Sirve para prototipar circuitos. 7

BDD Behavior-driven development o desarrollo guiado por pruebas es una metodología de desarrollo de software que esencialmente afirma que por cada unidad de software, un desarrollador de software debe: Primero definir un conjunto de pruebas para la unidad; después implementar la unidad; finalmente verificar que la implementación de la unidad haga exitosas las pruebas. . 13

Celda de carga Sensores que miden el peso de un objeto posicionado sobre ellos, a través de cambios en voltaje y/o resistencia. 7

Cucumber Es una herramienta de software usada para pruebas, especialmente de aceptación. Es comúnmente utilizada al aplicar estrategias de desarrollo como BDD. 14

Electron Es un framework de código abierto desarrollado por GitHub. Este permite el desarrollo de aplicaciones de escritorio como si de una aplicación web se tratase, utilizando node.js y chromium web browser. 14

Fair trade Es un principio organizacional, cuyo principal objetivo es asegurar la equitativa división de ganancias entre entes que se relacionan con un producto (cliente, proveedor, vendedor). 3

Gherkin Gherkin is el lenguaje utilizado por cucumber para definir los casos de prueba. esta diseñado para ser legible por humanos,y colectivamente describe los casos de uso relativos al sistema.El propósito detrás de esta sintáxis es el promover las prácticas de BDD transversalmente al equipo de desarrollo, sin importar su especialidad o si tiene conocimientos técnicos o no. 32

Grunt Es un lanzador automático de tareas para Javascript. 14

Jenkins Es un software de Integración continua open source escrito en Java. 14

Kanban En este enfoque de desarrollo de software, el proceso, desde la definición de una tarea hasta su entrega al cliente, se muestra para que los participantes lo vean y los miembros del equipo tomen el trabajo de una cola. Generalmente se ocupa una pizarra con las columnas por hacer, en desarrollo, y finalizado. 13

Lean Este paradigma de desarrollo de software ofrece todo un marco teórico sólido y basado en la experiencia, para las prácticas ágiles de gestión. Se basa en 7 principios: Eliminar los desperdicios, ampliar el aprendizaje, decidir lo más tarde posible, reaccionar tan rápido como sea posible, potenciar el equipo, integridad, ver el todo como conjunto. 12

MongoDB Es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto. 14

Mosquitto Es un servidor de código abierto del tipo broker. Este se encarga de recibir y enviar mensajes a las aplicaciones que se suscriban a él. 14

MQTT Es un protocolo de comunicación máquina a máquina, el cual sirve para el envío de mensajes de forma rápida y ligera. 14

Protoboard Es un tablero con orificios conectados eléctricamente entre sí, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos. 7

Protractor Es un conjunto de herramientas de prueba desarrollado para el framework angular. 14

RFID Es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio. 7

Selenium Es un entorno de pruebas de software para aplicaciones basadas en la web. 14

Stakeholder Se utiliza para referirse a quienes son afectados o pueden ser afectados por las actividades de una empresa. 5

Lista de Figuras

- 1. Diagrama de procesos: Venta 4
- 2. Arquitectura fisica 13
- 3. Esquema Base de datos 21
- 4. Vista #1: Inicio de sesión 22
- 5. Vista #2: Dashboard 22
- 6. Vista #3: Agregar nuevo ingrediente 23
- 7. Vista #4: Actualizar valores de ingredientes 23
- 8. Vista #5: Vista de cocina 24
- 9. Vista #6: Vista de interfaz de caja 24
- 10. Diagrama de Componentes 25
- 11. Modelo de proceso de negocio 41

Lista de Tablas

1.	Costos de Recursos Tecnológicos y Humanos	7
2.	Riesgos	9
3.	Historia de usuario #1: Selección de ingrediente	17
4.	Historia de usuario #2: Finalizar compra	17
5.	Historia de usuario #3: Estado de ingredientes en mesón e inventario	18
6.	Historia de usuario #4: Agregar y modificar ingredientes	18
7.	Historia de usuario #5: Ver información histórica	18
8.	Historia de usuario #6: Modificar inventario	19
9.	Carta Gantt	34

1. Introducción

Durante el siglo XXI las tecnologías de la información se han vuelto imprescindibles. La necesidad de integrar estas nuevas herramientas a todos los ámbitos ha llevado a referirse a la actualidad como “Revolución Industrial 4.0”. Esta revolución consiste en la utilización de forma eficiente de los recursos productivos, lo cual se posibilita gracias a la captura y procesamiento de la información obtenida en cada uno de los procesos productivos. Además, la interactividad brindada gracias a internet posibilita que estos datos sean enviados y compartidos entre sistemas, resultando en una nueva forma de concebir, administrar y mejorar la productividad y el negocio en sí.

En esta línea, la industria alimentaria es una de las principales beneficiadas por la tecnología. Por un lado, ya que una gran parte de los alimentos es perecible y cuentan con tiempos acotados de vida útil, se puede contar con información fidedigna y actualizada del estado y condiciones de estos. Al procesar estos datos, se mejora la toma de decisiones y se puede llegar a minimizar las pérdidas. Sin embargo, aún con las posibilidades que brindan los avances para este sector, las implementaciones de sistemas globales para el manejo de este mercado son pocas, y centrados en aspectos generales y especulativos, más que en sistemas completamente detallados, precisos y que respondan a las reales necesidades del cliente.

Dentro del sector alimenticio, la venta de comida rápida es un rubro especialmente delicado y con necesidades cambiantes. Aquí no solo es necesaria una atención rápida, sino que los estándares de calidad, cantidad, y eficiencia son fundamentales para los administradores. Por otra parte, el cliente es cada vez más exigente y siente la necesidad de saber que es lo que está comiendo y si lo que se le está entregando es realmente por lo que está pagando. O acaso ¿realmente el consumidor está siempre conforme con la ración servida?, ¿el consumidor sabe lo que está comiendo?, ¿Sabe el origen de los ingredientes que come?, ¿Sabe si lo que está recibiendo fue hecho hoy, ayer, o hace una semana?. Bajo esta disconformidad nace la idea de un local de comida rápida que entregue toda la información necesaria al cliente, con ingredientes de alta calidad, en donde paga solo por lo que está consumiendo, y está siempre informado de cuanto es lo que está pagando.

Para que la idea de un local de estas características se pueda hacer realidad, es necesario generar una solución a la medida del problema. Es necesaria una conexión directa entre el inventario y la venta, entre el proceso de venta y el cliente, entre la información que el proceso entrega y la que el cliente necesita. El propósito del actual trabajo es el de solucionar este problema, para permitir el correcto funcionamiento de este emprendimiento.

2. Visión General del Proyecto

2.1. Propósito

El propósito de este documento es describir de manera formal el comportamiento del sistema que se ha construido. A continuación se detallan los requerimientos funcionales, las restricciones y atributos de calidad que debe cumplir el sistema. El documento está dirigido a:

- El cliente.
- El equipo de desarrollo.
- El profesor guía.

2.2. Alcance

El alcance de este proyecto ha sido el construir un sistema integrado de 4 bandejas de ingredientes con balanzas que permita, mediante un software, calcular el precio de un sándwich en base a los ingredientes utilizados y la cantidad de estos. Además deberá permitir al administrador del local modificar el precio individual de cada ingrediente.

Para el cliente, la solución deberá tener la opción de ver el precio total que lleva en el pedido y el precio por cada ingrediente que va escogiendo en tiempo real.

En la sección cocina debe existir un monitor que indica la cantidad de cada ingrediente que se encuentra en el mesón de producción, así como también la cantidad que se encuentra en bodega del mismo ingrediente.

2.3. Objetivos

El objetivo principal de este proyecto es generar e implementar una solución a medida del cliente, con el cual este pueda realizar la venta de productos en forma personalizada, dependiendo del peso y valor de cada uno de los ingredientes. Además como objetivos específicos se cuenta con:

- Dar la libertad al cliente de armar su sándwich como el lo requiera.
- Dar al cliente la posibilidad de tener comida rápida a su gusto y saludable.
- Dar al administrador del local los datos necesarios para una buena toma de decisiones en su negocio.
- Mejorar la fluidez de trabajo en la cocina mediante datos reales del estado del mesón de producción.

3. Situación en Estudio

Para entender mejor el proyecto y todas sus aristas es que se presenta esta sección, con el fin de explicar desde varios puntos de vista la situación actual en que se encuentra nuestro cliente y el por qué se ha decidido realizar este proyecto.

3.1. Contexto Organizacional

Las empresas de comida rápida venden en su gran mayoría productos estándar, teniendo que elegir desde un catálogo un producto. Este ya tiene su precio e ingredientes o subproductos incluidos, en donde la mayor posibilidad de personalización es agrandar uno o dos por un costo ligeramente superior al original.

En otros casos, se da la opción al cliente de poder armar su pedido de manera un poco más libre que en los casos anteriores. Aquí, se puede ir agregando “porciones” de alimentos al pedido, cada una con su precio establecido. Esto le da más libertad al cliente pero sigue siendo una limitante en términos de costos y versatilidad.

Ambas alternativas generalmente cubren solo un tipo de comida, la menos saludable, olvidando a un público que busca lo contrario, cuidar de sí mismos y del medio ambiente.

3.1.1. Misión Organización

La misión de este emprendimiento es generar una opción para adquirir comida rápida saludable. Su público objetivo comprende la sociedad en general, pero se focaliza especialmente en aquellos que buscan comida saludable, orgánica, con un manejo de procesos sustentable y que promueve el reciclaje. En otras palabras, una empresa *verde*, que sea amigable con el medio ambiente. Sin embargo, se busca que estas prácticas no sean sinónimo de un gasto desmesurado para el cliente, sino cuya política sea la de precio justo o *Fair trade*.

3.2. Descripción situación actual

El proyecto, en la actualidad, se encuentra en etapa de planificación avanzada. Sin embargo, el siguiente diagrama de procesos da una idea del proceso principal de venta de la empresa:

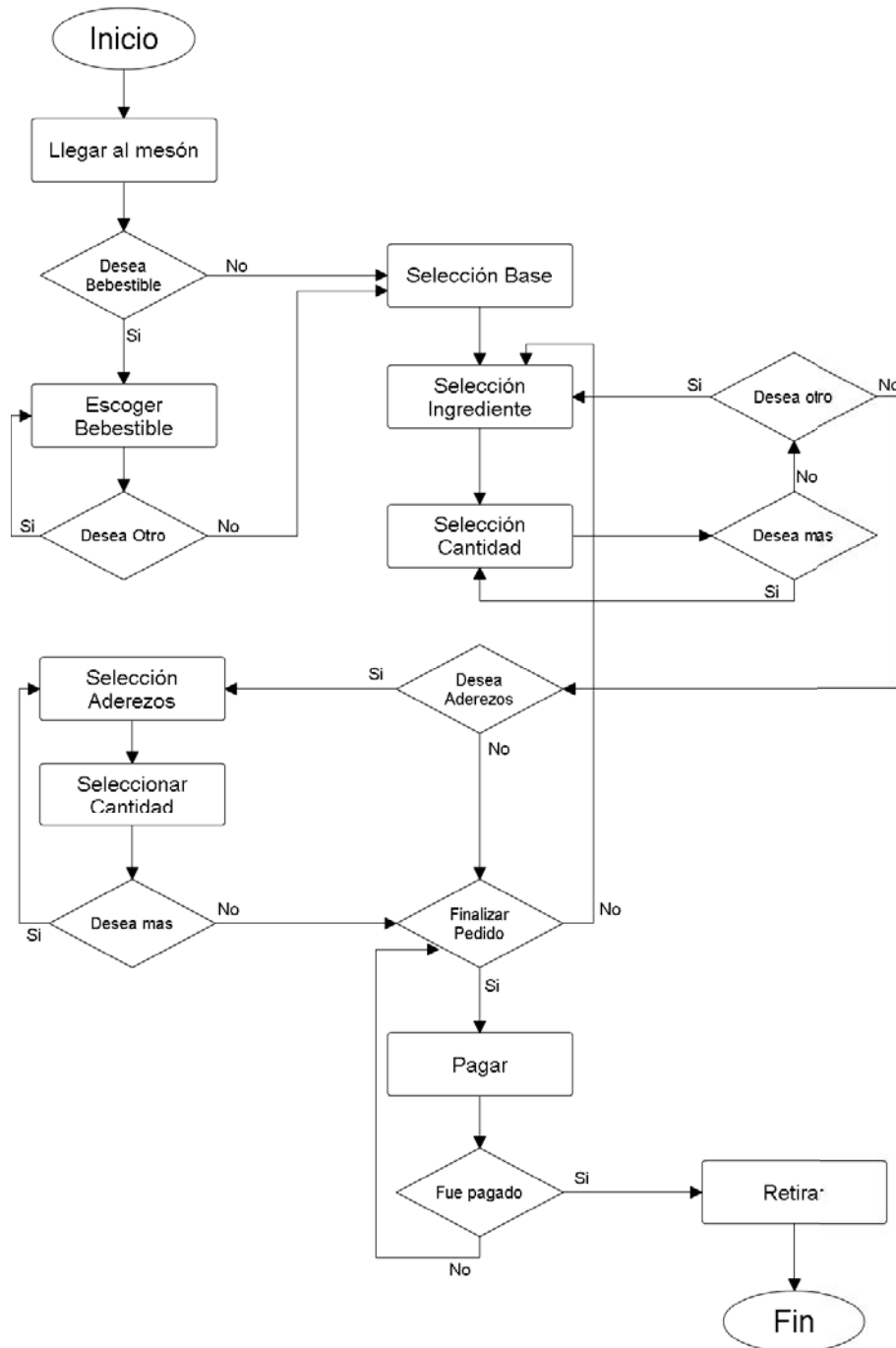


Figura 1: Diagrama de procesos: Venta

3.3. Identificación de los problemas detectados

El emprendimiento que se desea instaurar necesita solucionar un problema bastante específico: ¿Cómo poder vender un sándwich con un precio variable dependiendo de la cantidad de cada ingrediente?

Pero si bien este problema es el mayor, engloba varios problemas menores, los cuales corresponden a:

- Cómo mostrar al cliente la cantidad del ingrediente.
- Cómo mostrar al cliente cuanto está pagando por el ingrediente que está escogiendo.
- Cómo mostrar al cliente el subtotal actual a pagar.
- Cómo saber cuando reponer d un producto en específico y cuando no.
- Cómo conocer la cantidad de cada ingrediente disponible en el local.

Los encargados del emprendimiento no lograron encontrar una solución en el mercado que se adecuase a sus necesidades, es por ello que necesitan de un equipo que pueda ayudarlos a generar una, a descubrir requisitos que no han previsto y a implementar una solución que les permita comenzar a atender sus clientes.

3.4. Estado del arte

Al averiguar sobre posibles alternativas en el mercado que se adecuen a la necesidad de este negocio, no fue posible encontrar una que se adecuase realmente a sus requerimientos. Las alternativas presentan elementos muy dispersos, con pocas posibilidades de intercomunicación, de gran dificultad, y que no cuentan con el soporte necesario ni con la visualización deseada.

Una alternativa barajada era la utilización de una pesa al final del proceso de venta, pero esto impediría la venta diferenciada por cada ingrediente. Por otra parte podría contar con balanzas específicas para cada ingrediente, pero el proceso se haría más lento y engorroso.

En resumen, el modelo de negocio del emprendimiento es tan específico, que no existe una alternativa en el mercado que se pueda adecuar a este, y se vuelve fundamental el desarrollar una solución a la medida del *Stakeholder*.

4. Gestión del Proyecto

Para la correcta ejecución del proyecto ha sido necesario realizar una serie de trabajos en conjunto con el cliente, para crear una visión previa de las distintas áreas del negocio como lo son los recursos necesarios, estudios de factibilidad pertinentes y la planificación del mismo. Además fue necesario tener en cuenta otros puntos como son la gestión del riesgo.

4.1. Estudio de factibilidad

Además de analizar y describir los recursos que se han utilizado para la realización del proyecto, es necesario realizar estudios de factibilidad técnica, económica y operacional, para poder asegurar que el producto se pueda realizar con los recursos que hoy en día hay disponibles en el mercado.

4.1.1. Factibilidad Técnica

Podemos separar los elementos técnicos en dos secciones dependiendo del sitio en el que se encontrarán ubicados.

Por un lado se encuentran los elementos que serán utilizados dentro del local, es decir, las balanzas puestas en las bandejas (mesón de producción), un monitor ubicado en la cocina, y un dispositivo para la Caja. Los elementos que para esta parte del proyecto se encuentran disponibles en su mayoría en tiendas de electrónica, con un precio de insumos relativamente accesible, por lo que no supone un problema la confección de esta parte del sistema.

Por otro lado tenemos el servidor que será utilizado para el manejo de datos y su almacenamiento histórico. Este, debido a la masificación de servicios de datos que ha existido en el último tiempo, se encuentra un precio razonable. Además el manejo de datos será tradicional, por lo que es completamente factible, desde este punto de vista, el proyecto.

4.1.2. Factibilidad Económica

Desde el ámbito económico, tal como se mencionó anteriormente, son muchos los productos y servicios que se utilizarán para este proyecto que se pueden encontrar en tiendas especializadas a un costo bastante reducido para la magnitud del proyecto.

Tal como se muestra en la Tabla 1, los recursos económicos solicitados no son muy elevados, y estos han sido cubiertos por el cliente dentro de los costos de implementación, lo cual fue conversado y aprobado por ambas partes. Por lo que lleva al proyecto a ser factible económicamente.

Descripción	Cant.	Precio Unitario	Precio Total
Arduino Uno	4	\$ 7.000	\$ 28.000
Célula de carga	16	\$ 9.990	\$ 159.840
Protoboard	4	\$ 2.390	\$ 9.560
Pack de cables Arduino	3	\$ 3.890	\$ 11.670
Módulo RFID	4	\$ 4.390	\$ 17.560
Módulo LED 8 dígitos	8	\$ 5.990	\$ 47.920
Amplificador célula de carga HX711	4	\$ 8.590	\$ 34.360
Impresora térmica (POS) ¹	1	-	-
Horas Hombre	200	\$ 5.000	\$ 1.000.000
Total			\$ 1.308.910

Tabla 1: Costos de Recursos Tecnológicos y Humanos

4.1.3. Factibilidad Operacional

Si miramos la factibilidad operacional, por un lado tenemos el cómo será operado el sistema una vez esté finalizado. Este, en la parte de mayor interacción con el usuario, será operado por el “Preparador de Sándwich”, el cual será visto en mayor detalle en la sección 6.1. Este tendrá interacción directa con el sistema a través de tarjetas RFID, para las cuales no necesitará mayor capacitación.

Podemos apreciar también que este sistema, si bien no es necesario, si ayuda a que realmente sea comida rápida la que se estará vendiendo en el local, puesto que se puede realizar el mismo pedido con pesas manuales y un cálculo manual, sin embargo, este proceso se tornaría excesivamente lento y tedioso, perdiendo importante clientela.

Por lo nombrado anteriormente podemos concluir que el proyecto tendrá factibilidad operacional.

4.2. Recursos requeridos para la solución

Dentro de los recursos necesarios para satisfacer los requerimientos del cliente se encuentran los económicos, tecnológicos y humanos.

4.2.1. Recursos Tecnológicos

Para poder dar una solución de la mejor manera posible, son necesarios ciertos recursos tecnológicos que permitirán automatizar los procesos requeridos por el cliente. Estos son los siguientes:

- Arduino Uno.
- Celda de carga.

¹Debido a los requerimientos y especificaciones que tiene el SII para las impresoras POS, no ha sido posible cotizar la impresora adecuada y autorizada por el SII.

- Protoboard.
- Módulos RFID.
- Módulos LED.
- Amplificador de carga.
- Impresora POS.

Dichos elementos son necesarios para confeccionar el sistema de balanzas digitales que registrarán la variación de peso producida en las bandejas metálicas al retirar los ingredientes. Además los módulos RFID serán necesarios para identificar cada pedido como será explicado posteriormente.

Además estos elementos permitirán enviar los datos a un servidor remoto en el cual se recolectarán, para posteriores reportes y estadísticas.

4.2.2. Recursos Humanos

Para la confección de prototipos y el desarrollo de los servicios de los cuales consumirán estos mismos es necesario una fuerza de trabajo humana, la cual dispondrá de su tiempo (medible en Horas Hombre), además de su conocimiento para una correcta implementación. Los costos monetarios serán especificados en la sección 4.2.3.

4.2.3. Recursos Económicos

Para poder obtener y financiar, tanto los recursos humanos cómo tecnológicos, son necesarios los recursos económicos que serán otorgados en su totalidad por el cliente. Los cuales se encuentran especificados en la Tabla 1

4.3. Carta Gantt

Debido al tamaño de la Carta Gantt, esta puede ser vista en la sección 10 Anexos, en la Tabla 9.

4.4. Gestión del Riesgo

Para el proyecto presentado se han estipulado diversos riesgos que podrían complicar el normal flujo de trabajo e implementación del sistema. En primer lugar se listarán los riesgos detectados y posteriormente se generará un plan de mitigación a seguir en caso de que cada suceso ocurra durante el desarrollo de la solución y posterior implementación.

Riesgo	categoría	Probabilidad	Impacto
Aplicación no cumple con necesidades del cliente	Proyecto	40 %	Crítico
Diseño de dispositivo no agrada al usuario	Proyecto	50 %	Crítico
Problemas en coordinación de equipo	Equipo	40 %	Crítico
Escalabilidad de uso mayor a la esperada	Proyecto	35 %	Marginal

Poca experiencia con las tecnologías a utilizar	Desarrollo	50 %	Marginal
Cambios en plazos de entrega	Entorno	10 %	Marginal
Integración de plataformas falla	Proyecto	40 %	Crítico
Implementación no adecuada	Proyecto	60 %	Crítico

Tabla 2: Riesgos

4.4.1. Plan de mitigación de riesgos

Para poder bajar el porcentaje de ocurrencia de cada riesgo crítico, se han tomado ciertas medidas:

- Involucrar al cliente: es necesario que el cliente esté involucrado en el desarrollo, tanto para otorgar su visión del problema, como para validar lo realizado y que puedan ejecutar pruebas reales del uso de lo generado.
- Comunicación en el equipo: Con el fin de no realizar el mismo trabajo más de una vez, es necesario facilitar la comunicación directa del equipo en todo momento. También es útil para corregir errores, solucionar problemas de las herramientas utilizadas, entre otros factores.
- Estudio de alternativas a las ya escogidas: siempre se barajan posibilidades distintas a las ya escogidas, en caso de imposibilidad de uso de una en específico por restricciones de la solución.
- Implementación por partes: para ir probando como se comporta de manera real cada una de las partes de la solución, se irán implementando de a poco, y luego replicando. Con ello se minimizan las demoras para encontrar los problemas que se están dando y posibilita una rápida solución de los errores.

5. Detalle de la Solución

5.1. Descripción general de la Solución

Para poder brindar una solución útil y adecuada a los problemas planteados, se generará un dispositivo de medición específica para cada uno de los ingredientes que se desea comercializar. Este dispositivo contará con:

- Celda de Carga

Mantendrá una medición en todo momento del peso que contiene el recipiente en el mostrador de venta y en bodega.

- Display

Cada uno de los dispositivos de medición contarán con 2 displays, uno con la información de peso y precio del ingrediente actual que se está sirviendo, y por otra parte, uno que indique el total de lo que el cliente está consumiendo.

- Lector RFID

Este componente se utilizará para distinguir de forma única cada una de las ordenes de venta que se están realizando en el momento.

- Controlador

Será el encargado de manejar y sincronizar la información entregada por la celda de carga, para calcular la diferencia y con ello poder obtener el valor real, y de la forma que el cliente lo necesita.

Tal cual ya se mencionó, este dispositivo será incorporado a cada uno de los contenedores de ingredientes. Con ello, la solución funcionará de la siguiente manera:

- Al cliente, en el momento de comenzar el pedido, le es asignada una tarjeta RFID con una identificación única de su orden.
- Cada vez que el cliente desea añadir un ingrediente, se le identifica con su tarjeta mediante el lector RFID que corresponde a ese ingrediente.
- En el display se va mostrando la información de lo que se le está añadiendo a su pedido.
- Al finalizar el pedido, se utiliza la información recaudada en la orden para imprimir el detalle con cada uno de los ingredientes, sus cantidades y el total del pedido.

Sin embargo, lo descrito solo considera el proceso de venta. La solución también contempla un modulo de revisión de inventario, reposición y de administración.

Por una parte, la solución de inventario contempla la utilización de los dispositivos ya descritos, para mantener el registro de cada ingrediente en bodega.

Por otra parte, la solución de reposición se basa en la utilización de la información obtenida por los dispositivos en el punto de venta y en el inventario. Esta información se utiliza para mostrar de manera gráfica ambos niveles por cada ingrediente y se utilizan con el fin de alertar cuando un ingrediente se debe reponer o comprar más de este.

Finalmente, el módulo de administración consiste en una plataforma en la cual se puedan manejar y visualizar aspectos relevantes del negocio. Una de las funciones incluidas aquí, es el manejo de ingredientes, precios, y de registro de ingresos de estos a bodega. Además es posible revisar información histórica de ventas, ingredientes más vendidos, ingredientes menos vendidos, cantidad de merma, entre otros.

5.2. Paradigma y Metodología de desarrollo

En vista de la solución que se desea generar, se deben establecer ciertas pautas para el correcto desarrollo del sistema. Es así, que se ha tomado la decisión de seguir como guía el paradigma de tipo Ágil y “Lean”.

En primer lugar, el paradigma ágil fue escogido ya que se adecuaba al proyecto de mejor manera que otros. Un ejemplo de ello es que, si bien existen un número menor de requerimientos explícitos en este proyecto, no se tiene una visión completa del negocio, sino que se encuentra en desarrollo. Es por esto que al generar funcionalidades y mostrar el sistema a los stakeholders, estos podrán realizar retroalimentación acerca de la experiencia al utilizar el sistema, y con ello, implementar y corregir funcionalidades nuevas acordes a lo que realmente necesita. El paradigma ágil no solo es una opción viable por las razones anteriores, sino que además la cantidad de miembros en el equipo es pequeña, la necesidad de entregar un producto de manera rápida y funcional es apremiante.

En segundo lugar tenemos el paradigma “lean”, es un paradigma más reciente derivado del ágil, el cual se puede resumir en 7 principios:

- Eliminar los desperdicios: con ello se refiere a dejar de lado prácticas comunes en el rubro, de tal forma que se eliminen las funcionalidades innecesarias, la sobrecarga de los programadores, trabajos no completados, evitar el sobre-procesamiento, errores, esperas burocráticas, requisitos poco claros y comunicación interna lenta.
- Ampliar el aprendizaje: La idea es generar conocimiento que pueda ser compartido en el equipo o que pueda ayudar a mejorar la herramienta en desarrollo. Por ejemplo, pair programming permite que dos personas puedan compartir su conocimiento para generar un mejor resultado.
- Decidir lo más tarde posible: la idea es posponer las decisiones cruciales del proyecto hasta una etapa en que se debe tomar una decisión. Así en caso de error es más sencillo y menos costoso reparar el error que si esta decisión se hubiese tomado en un momento más temprano y hubiese involucrado a todo el proyecto o gran parte de este.
- Reaccionar tan rápido como sea posible: hacer una solución rápida y simple, que a su vez pueda ser escalable en el tiempo. No pensar demasiado ni invertir tiempos innecesarios en el diseño de soluciones complejas que no cumplirán con los requerimientos reales del cliente.
- Potenciar el equipo: generar una comunicación transversal en el equipo, potenciando la capacidad de escuchar atentamente todas las opiniones, de forma fluida y sin intermediarios,
- Crear la integridad: Referente a comprender el dominio del problema, para así desarrollar un sistema íntegro, flexible y mantenible en el tiempo, a través de refactorización.
- Véase todo el conjunto: si bien el sistema puede ser desarrollado de manera modular, el foco se centra en que el sistema en su conjunto funcione a la perfección, en que la integridad del sistema sea óptima, ya que al desarrollarse e incluirse en el sistema de forma aislada los errores tienden a acumularse y a generar problemas mayores, además de repeticiones innecesarias de código, símbolo inherente de un mal diseño de software.

Queda más claro que cada uno de estos 7 puntos están en directa concordancia con la forma de trabajo que se desea plantear para el desarrollo de la solución, ya que nuestro desarrollo debe ser rápido pero cada una de las piezas que integrarán el sistema deben estar en perfecta armonía.

Por otra parte se debe también escoger una metodología de desarrollo del proyecto, es por eso que se escogerán 2: Kanban y BDD. Como ya se ha mencionado, el proyecto y su solución necesitan de una forma bastante libre y “liviana” de resolución, que permite una gran flexibilidad para introducir cambios o perfeccionar lo que ya se ha producido. Kanban es una de las metodologías ágiles “ultra ligeras” que cumplen con esta especificación, otorgando la flexibilidad que necesitamos, que además cuenta con una ayuda visual para poder manejar el estado del proyecto actual y la cantidad de trabajo restante hasta la finalización del sistema.

Finalmente tenemos como metodología BDD, “behaviour driven development”, un tipo de pruebas que se enfoca en lo que el usuario final debería poder hacer con el sistema, dejando de lado todo aquello que el usuario no puede ver. Esta forma de desarrollo nos indica primero que la etapa de pruebas es una parte fundamental para la correcta generación de la solución, es por ello que se deben generar los test de forma anticipada y con ello ir controlando la posterior creación de las funcionalidades. Además el proceso de testeo se lleva a cabo durante todo el proceso de desarrollo, evitando problemas tanto de cada módulo, como de integridad del sistema.

5.3. Arquitectura lógica y física de la solución

Para poder llevar a cabo la solución de buena forma, se debe estructurar su arquitectura lógica y física. Para el caso de la distribución lógica se utilizará una arquitectura Rest basada en capas. Con esta distribución se desea tener una capa de datos completamente separada a la cual la aplicación pueda hacer consultas y obtener la información que se desea, mientras que la capa de negocio toma y procesa la información y la lógica de negocio directa. Por otra parte la vista se encuentra en una capa completamente separada, lo cual permitirá cambios en esta capa sin afectar la funcionalidad del sistema, y la generación de interfaces distintas para cada dispositivo en el cual se desee utilizar la aplicación.

En cuanto a la arquitectura física, se pretende contar con un equipo de servidor de datos, que contenga la API Rest, mientras que las capas de negocio y vista se encontrarán en otro servidor completamente aparte. Esta división se estructura teniendo en cuenta la importancia de los datos de la aplicación, su seguridad, mantención y respaldo.

Para visualizar de mejor forma la idea de la estructura, queda plasmada en la siguiente ilustración:

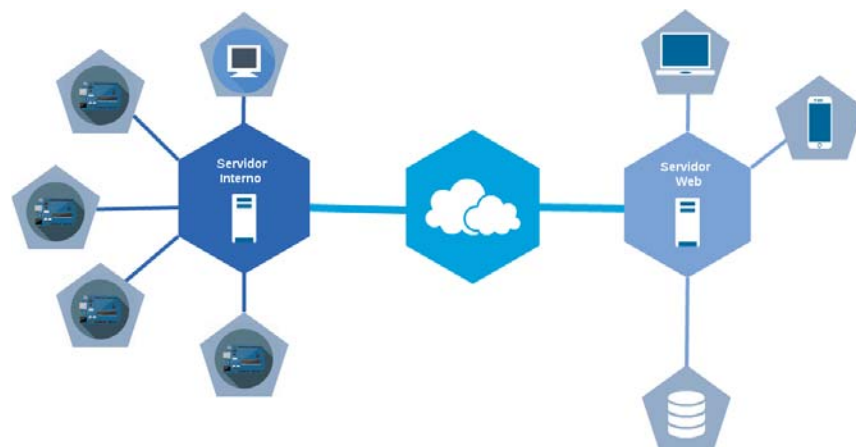


Figura 2: Arquitectura física

5.4. Herramientas de desarrollo

Dentro del proyecto se requieren una serie de herramientas para el desarrollo tanto del software como del hardware

5.4.1. Software

El software necesario para el desarrollo del sistema utilizado se detalla a continuación:

- AngularJS 2 (Front-End).
- Electron.
- Mosquitto MQTT.
- Cucumber.js.
- Protractor.
- Grunt.
- Jenkins.
- Webstorm (IDE y Deployment).
- Selenium webdriver.
- Xbfv.
- MongoDB.

5.4.2. Hardware

El hardware es una de las herramientas que más pueden ser susceptibles a cambios posteriores y a reevaluación. Para una etapa inicial se utilizarán:

- Arduino Uno.
- Celda de Carga (HX711).
- Amplificador celda de carga (HX711).
- Lector tarjetas RFID (RC522).
- Tarjeta RFID (S50).
- Display LCD (HD44780).
- Protoboard.
- Servidor de alojamiento aplicación.

6. Análisis de Requerimientos

Para hacer un análisis acabado de los requerimientos, se ha desarrollado el proceso de levantamiento de requerimientos, análisis y validación, en conjunto con los clientes. Para ello, cada semana se tiene una reunión, en la cual se discuten los requerimientos encontrados, lo que permite corregir y perfeccionar el modelamiento de la solución. Es así, que gracias a este proceso se ha logrado identificar a los actores involucrados, los requerimientos funcionales y no funcionales de la solución a desarrollar.

6.1. Identificación de Actores involucrados

Para tener un correcto entendimiento del funcionamiento del sistema y de todos los requerimientos que este posee, es necesario definir los principales actores que tendrán participación dentro de él, con sus tareas.

6.1.1. Administrador

Es quien toma las decisiones estratégicas del negocio a largo plazo. Este actor puede verificar cada una de las transacciones históricas del local, filtrando por los parámetros que estime conveniente, además del flujo de clientes, cantidad de ingredientes utilizados, mermas, entre otros. Además es quien agrega y quita ingredientes, define sus precios y en base a la información obtenida decide cuanto y que es lo que debe comprar.

6.1.2. Jefe de Local

Es quien vela por que todo se cumpla dentro del local. Tiene la atribución de suprimir ingredientes por un tiempo determinado, además de resolver conflictos dentro del establecimiento.

6.1.3. Preparador de jugos

Se encarga de preparar los jugos en base a lo solicitado por el cliente.

6.1.4. Preparador de sándwich

Es el encargado de realizar el sándwich del cliente en base a los ingredientes que va solicitando, es quien más contacto con el cliente tiene.

6.1.5. Cajero

Es el ente que recibe el dinero cancelado por concepto del producto por parte del cliente, y quien lo entrega a este. Además es quien realiza la cuadratura de caja.

6.1.6. Cocinero

El cocinero se encarga de llevar la cocina del local, de preparar los ingredientes y reponerlos en caso de que se estén acabando en el mesón de servicio. Además debe informar sobre el estado del inventario.

6.1.7. Cliente

Es el cliente final del producto que la cadena de alimentos vende. El cliente tiene la capacidad de decidir que tipo de producto desea, ingredientes y cantidad de estos, que productos complementarios desea y finalmente es quien debe cancelar el valor de la orden.

6.2. Requerimientos Funcionales

- RF-1 El cliente debe poder escoger cada uno de los ingredientes de su orden y su cantidad.
- RF-2 El cliente debe ser capaz de visualizar el precio actual de su pedido.
- RF-3 El cliente debe ser capaz de visualizar el precio del ingrediente actual en función de la cantidad utilizada en su pedido.
- RF-4 El cliente debe ser capaz de regresar a una etapa anterior para aumentar la cantidad de un ingrediente.
- RF-5 El cliente debe poder recibir un resumen de los componentes de su orden al momento de finalizarla.
- RF-6 El cocinero debe poder revisar el estado actual de los ingredientes en el mesón de servicio.
- RF-7 El cocinero debe ser alertado cuando la cantidad de un ingrediente sea crítica.
- RF-8 El cocinero debe poder agregar stock al inventario.
- RF-9 El cocinero debe poder conocer la cantidad de cada ingrediente en el inventario.
- RF-10 El administrador debe poder modificar la forma de cobro por ingrediente, ya sea por unidad o por peso.
- RF-11 El administrador debe poder modificar los precios de cada ingrediente.
- RF-12 El administrador debe poder agregar o quitar ingredientes definitivos.
- RF-13 El jefe de local debe poder quitar ingredientes de manera temporal.
- RF-14 El administrador debe poder ver información histórica sobre los ingredientes, ordenes y ventas.
- RF-15 El jefe de local debe poder ver información histórica sobre los ingredientes, ordenes y ventas.
- RF-16 El jefe de local debe poder realizar ingresos de inventario por cada producto.
- RF-17 El sistema debe tener capacidad mínima para 2 tipos de envase y 23 ingredientes.
- RF-18 El sistema debe ser capaz de relacionar los ingredientes con un pedido en particular.

6.3. Requerimientos No Funcionales

- RNF-1 El sistema debe permitir alta demanda de transacciones a la vez (mayor a 10).
- RNF-2 El sistema debe tener una fiabilidad mayor al 90 %.
- RNF-3 El sistema debe responder de manera rápida, con tiempos de actualización menores a 3 segundos.
- RNF-4 El sistema debe responder de manera rápida, con tiempos de actualización menores a 20 segundos, para el estado de ingredientes en inventario y mesón.
- RNF-5 El sistema debe permitir más de 4 clientes a la vez.

6.4. Historias de usuario

Historia: #1	Usuario(s): Preparador de sándwich, Cliente
Nombre de historia:	Selección de ingrediente
Prioridad:	Alta
Requerimientos:	RF-1, RF-2, RF-3, RF-4 RF-18
Riesgo de desarrollo:	Media
Puntos estimados:	3
Responsable:	Vicente Retamal
Descripción:	El cliente selecciona un ingrediente. El preparador selecciona el ingrediente mediante el identificador. El sistema muestra el precio en base al ingrediente y gramaje del mismo. Se dirige al siguiente ingrediente
Validación:	

Tabla 3: Historia de usuario #1: Selección de ingrediente

Historia: #2	Usuario(s): Preparador de sándwich, Cajero, Cliente
Nombre de historia:	Finalizar compra
Prioridad:	Alta
Requerimientos:	RF-5, RF-18
Riesgo de desarrollo:	Media
Puntos estimados:	3
Responsable:	Gustavo Saavedra
Descripción:	El cliente al finalizar la compra se dirige a la Caja. Ahí es donde el Cajero debe finalizar la compra, el sistema imprimirá un <i>voucher</i> con el detalle de la compra, especificando cantidad y valor de todos los ingredientes utilizados
Validación:	

Tabla 4: Historia de usuario #2: Finalizar compra

Historia: #3	Usuario(s): Cocinero
---------------------	-----------------------------

Nombre de historia:	Estado de ingredientes en mesón e inventario
Prioridad:	Media
Requerimientos:	RF-6, RF-7, RF-9
Riesgo de desarrollo:	Media
Puntos estimados:	2
Responsable:	Vicente Retamal
Descripción:	Como cocinero quiero ver en una pantalla el estado actual de los ingredientes en el mesón y el inventario. Además recibir una alerta cuando algún ingrediente esté en un estado crítico
Validación:	

Tabla 5: Historia de usuario #3: Estado de ingredientes en mesón e inventario

Historia: #4	Usuario(s): Administrador
---------------------	----------------------------------

Nombre de historia:	Agregar y modificar ingredientes
Prioridad:	Media
Requerimientos:	RF-10, RF-11, RF-12
Riesgo de desarrollo:	Media
Puntos estimados:	2
Responsable:	Gustavo Saavedra
Descripción:	Como Administrador deseo agregar y modificar ingredientes, tanto en su forma de venta como en su precio en función de la forma de venta
Validación:	

Tabla 6: Historia de usuario #4: Agregar y modificar ingredientes

Historia: #5	Usuario(s): Administrador, Jefe de Local
---------------------	---

Nombre de historia:	Ver información histórica
Prioridad:	Media
Requerimientos:	RF-14, RF-15
Riesgo de desarrollo:	Media
Puntos estimados:	2
Responsable:	Vicente Retamal
Descripción:	Como Administrador y jefe de local deseo ver y revisar la información histórica de ventas y transacciones, logrando visualizar los datos separados por ingredientes.
Validación:	

Tabla 7: Historia de usuario #5: Ver información histórica

Historia: #6	Usuario(s): Jefe de Local, Cocinero
---------------------	--

Nombre de historia:	Agregar inventario
Prioridad:	Media
Requerimientos:	RF-8, RF-16
Riesgo de desarrollo:	Media
Puntos estimados:	2
Responsable:	Gustavo Saavedra
Descripción:	Como jefe de local deseo agregar productos al inventario, indicando el producto y la cantidad ingresada. A su vez como cocinero deseo quitar elementos del inventario para poder preparar dichos ingredientes.
Validación:	

Tabla 8: Historia de usuario #6: Modificar inventario

7. Desarrollo del Proyecto

Para poder dar una solución de manera óptima es necesario atacar el problema desde distintas perspectivas. Una de ellas es la parte física y por otro lado el diseño e implementación de Software.

7.1. Componentes de Software

Los componentes de software de la solución serán las siguientes:

- Software de caja: Se realizara un software encargado de registrar los pedidos que finalicen en la línea de producción. Posteriormente se procederá a pagar dicho pedido y el software queda disponible para recibir otro pedido.
- Inventario y alerta en cocina: Otro módulo será el disponible en cocina, en el cual se mostrarán los ingredientes en el mesón de producción, su estado, cantidad y lo que aún queda en inventario. Además este enviará alertas cuando un ingrediente se encuentre en estado crítico, ya sea por la cantidad en el mesón o el tiempo que lleva en él.
- Sitio de administración general: Se dispondrá de una aplicación web que permitirá al administrador poder modificar los precios, agregar ingredientes, ver el estado del inventario y por último ver estadísticas sobre ventas y uso de material.
- Sitio de inventario: Este permitirá al cocinero y jefe de local ir agregando o quitando elemento del inventario a medida que se vayan comprando o consumiendo.

7.2. Base de datos

Para la base de datos serán necesarias la siguientes entidades:

- Local: representa a cada local en que el sistema opera. De este, es necesario conocer quien es el jefe de local, además de los productos que ofrece.
- Ingrediente: este corresponde a cada uno de los ítems que se ofrecen por local. Contiene todos los detalles relevantes de este.
- Sándwich: es el producto en específico que se está vendiendo. Contiene uno o varios ingredientes con cantidades específicas del pedido.
- Pedido: Contiene el resumen de la información de un pedido, el cual puede contener uno o muchos sándwich, con un total global y una fecha de venta.
- Ingreso: corresponde a un ingreso de ingredientes al inventario. En este, se añade una cierta cantidad del ingrediente, con una fecha específica.
- Egreso: corresponde a la extracción de un ingrediente desde el inventario para ser puesto en el mesón. Esta acción reduce la cantidad del inventario y puede corresponder a un ingrediente que proviene de uno o mas ingresos.

El siguiente esquema muestra las tablas necesarias para la base de datos y las relaciones entre estos:

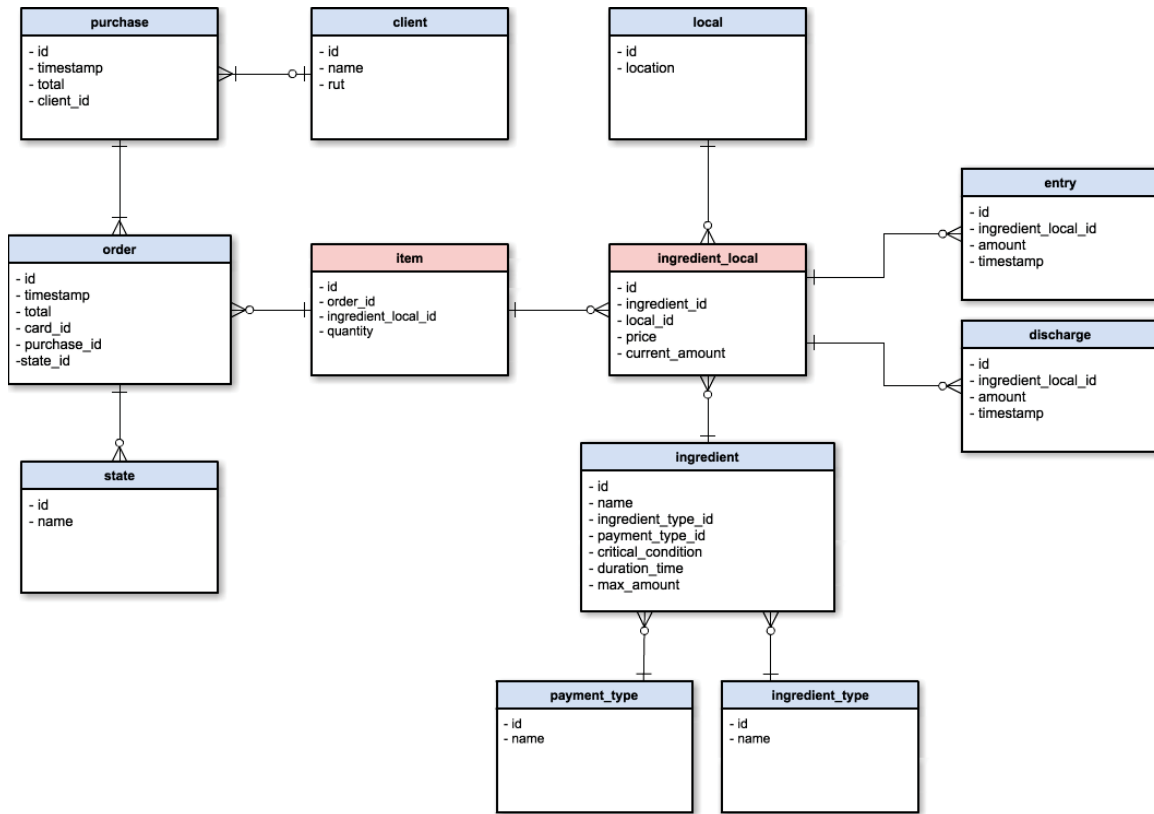


Figura 3: Esquema Base de datos

7.3. Interfaz

La interfaz que tendrá el software desarrollado será principalmente las mostradas en las siguientes Vistas:

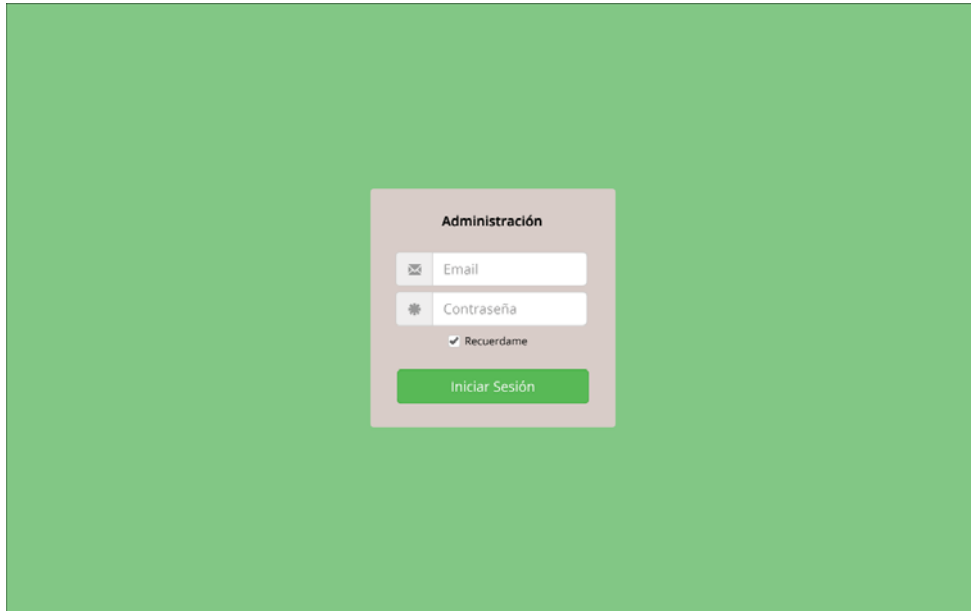


Figura 4: Vista #1: Inicio de sesión

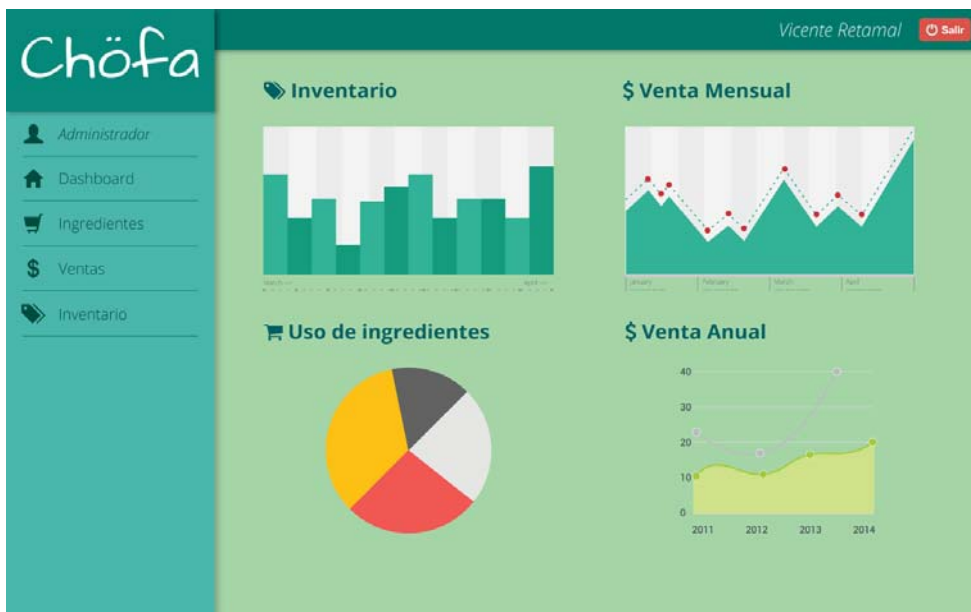


Figura 5: Vista #2: Dashboard

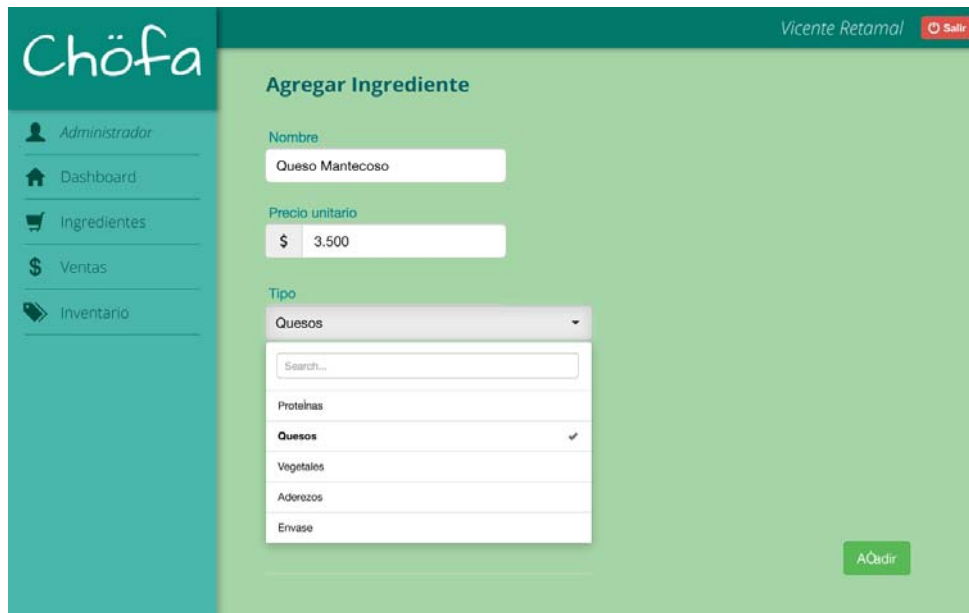


Figura 6: Vista #3: Agregar nuevo ingrediente

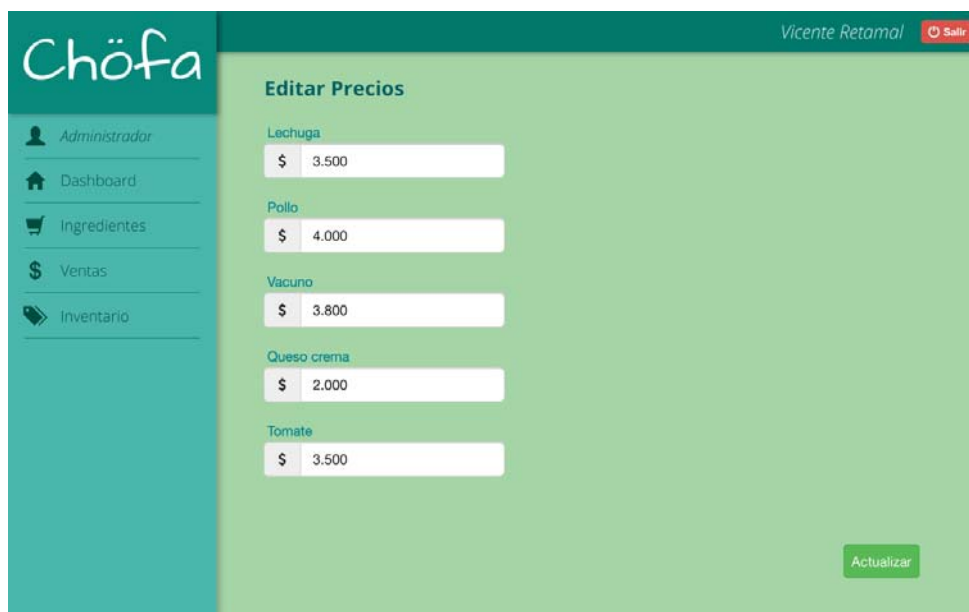


Figura 7: Vista #4: Actualizar valores de ingredientes

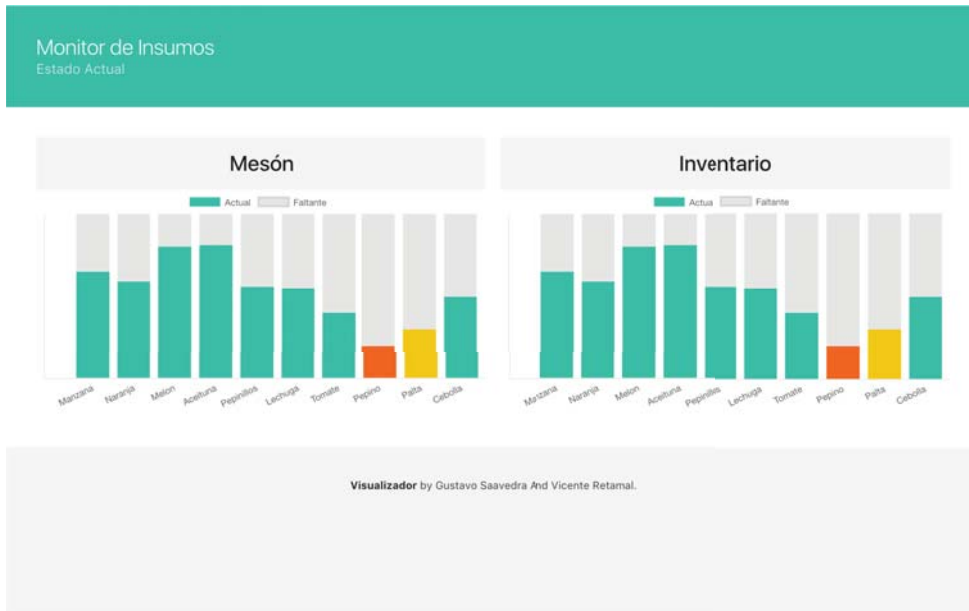


Figura 8: Vista #5: Vista de cocina

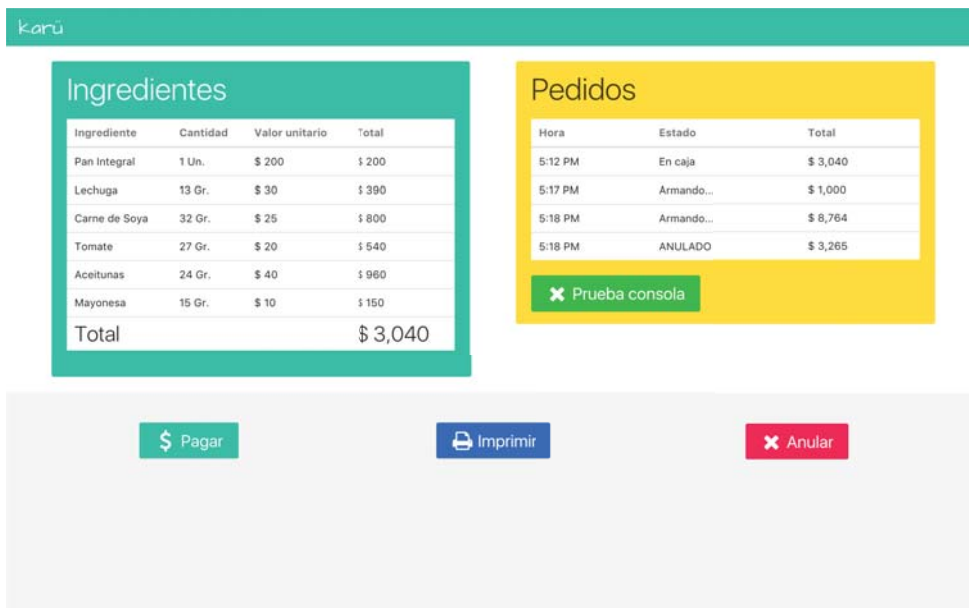


Figura 9: Vista #6: Vista de interfaz de caja

7.4. Implementación

Con el fin de llevar a cabo la solución propuesta ha sido necesario desarrollar, implementar e interconectar una serie de componente. Para comprender de buena forma la relación de estos, a continuación se presentará el diagrama correspondiente y posteriormente serán detallados individualmente:

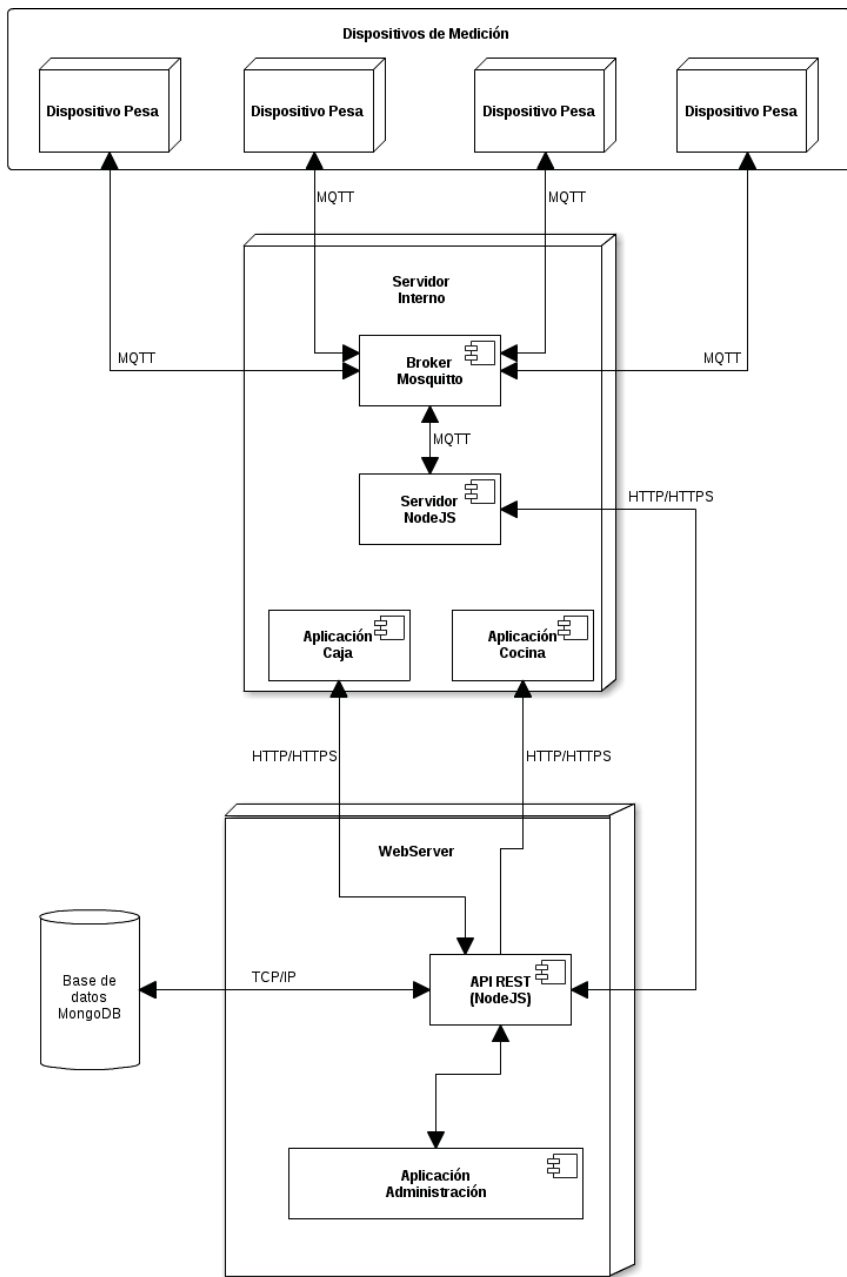


Figura 10: Diagrama de Componentes

7.4.1. Dispositivo Pesa

Este dispositivo está formado por un controlador (arduino), un amplificador de carga (encargado de traducir las señales), 4 celdas de carga (corresponden a los sensores), un lector RFID (encargado de leer tarjetas RFID), un conector Ethernet, y un Display LED. Este dispositivo lee las variaciones de peso en los sensores cada cierto intervalo de tiempo. Esta modificación en el peso es enviada a el Broker, por medio del protocolo MQTT. Además este dispositivo es capaz de identificar una tarjeta RFID, enviar el autentificador de esta, y obtener el valor parcial de la cantidad de peso asociada a esa balanza, además del total de la orden de esa tarjeta.

7.4.2. Broker

Como broker se utiliza Mosquitto. Este maneja la información de todos los agentes que publican en el y los envía a todos aquellos suscritos a este. Se utiliza el protocolo MQTT para recibir o enviar datos (con el fin de facilitar la conexión con el dispositivo pesa). Actúa como punto intermedio de conexión, ya que no almacena ni procesa información, solo la recibe y emite. Su utilización fue pensada para dar flexibilidad y escalabilidad al sistema.

7.5. Servidor NodeJS

Este representa a un servidor dentro de la red local. Este servidor se conecta directamente al broker, a través de MQTT, y procesa la información que este emite. Selecciona la información dependiendo del tema y con ello decide como procesarla y que hacer a continuación. Además publica en el broker el subtotal y total dependiendo del identificador. A su vez, si la información necesita que recibe necesita ser guardada, se conecta a un servidor externo (en la nube), donde guarda la información a través de una API REST. También extrae información desde la API, como el precio de un ingrediente.

7.6. API REST

Esta API ha sido desarrollada en javascript, específicamente en NODE JS. La API se encarga de hacer la conexión entre los datos solicitados y la base de datos. Gracias a esta capa de abstracción existe una mayor facilidad para obtener los datos y para evitar errores comunes en las conexiones directas al motor de base de datos.

7.7. Base de datos

La base de datos se encarga de la permanencia en el tiempo de los datos, de almacenarlos. En este caso el motor de base de datos elegido ha sido mongoDB, gracias a su facilidad de uso, versatilidad y gran cantidad de documentación. La única conexión que posee es directamente a la API Rest.

7.7.1. Aplicación Cocina

Esta aplicación se encarga de mostrar de forma gráfica el estado del mesón y del inventario, en tiempo real. Se extrae la información de la API Rest sobre cada uno de los ingredientes, se procesa y organiza de forma visual para comprender el nivel de cada ingrediente. La representación se realiza en porcentajes, con una variación de color cuando este se encuentra muy bajo o crítico.

7.7.2. Aplicación Caja

La aplicación de caja esta pensada para finalizar el proceso de compra y ayudar al cajero en su quehacer. La aplicación permite revisar todas las ordenes que se encuentran abiertas, escoger la que se desee, comprobar cada el listado de ingredientes que incluye, revisar el total de la orden, y escoger entre pagar , anular o imprimir el recibo del pedido.

7.7.3. Aplicación Administración

Para organizar el negocio y facilitar la administración de este es que ha sido pensada esta aplicación. Desde ella, se pueden modificar los ingredientes, su precio, agregar nuevos, revisar ventas globales, entre otros.

8. Plan de pruebas

8.1. Objetivos

Las pruebas tendrán como objetivo principal guiar el desarrollo de la aplicación y poder obtener retroalimentación directa sobre el cumplimiento de requisitos.

Un objetivo secundario de las pruebas es mantener “documentación viva”, la cual estará siempre actualizada y en directa concordancia con la versión más actual del software desarrollado. Otro objetivo en esta línea, es el mantener la integridad de la aplicación frente a cambios no planificados, a correcciones de código o a pérdida de funcionalidades.

Por su parte, este plan de pruebas tiene como funcionalidad, además de definir el que de las pruebas, metodología y alineamientos, la asignación de tareas y responsabilidades en el equipo.

8.1.1. Tareas

Las tareas a realizar serán:

- Generación de ambiente de testing.
- Especificación de requerimientos.
- Redacción de historias de usuario.
- Interpretación de las historias de usuario en formato Dado/Cuando/Entonces, y divididas por escenarios positivos y erróneos.
- Se realiza la implementación de las pruebas.
- Las pruebas son ejecutadas en su conjunto.
- Si las pruebas no fueron exitosas en su conjunto, se repara el error y luego se vuelven a ejecutar las pruebas.

8.1.2. Alcance

El alcance de las pruebas será simplemente la interacción de la aplicación con el usuario final. Todo lo que el usuario pueda y deba ver será testeado. La idea de la generación de pruebas en base al comportamiento del usuario tiene como finalidad el evitar programar código innecesario para la aplicación, por lo que las pruebas serán de gran ayuda para disminuir los tiempos totales de desarrollo y posterior mantención.

8.1.3. Estrategia de Testing

Como ya se mencionó, el testing se desarrollará en la capa de vista, es por ello que se aplicarán test de tipo e2e (“end to end”). Esta metodología de testing comprueba el como se desempeña la aplicación desde un inicio hasta un final, en como una entrada debe generar una salida. La principal ventaja que presenta es el poder ver de forma clara si cierta información enviada entre varios componentes o sistemas retorna la información correcta.

Para poder verificar la completitud de la aplicación, las pruebas se dividirán por características, las que a su vez tendrán un escenario positivo y diversos escenarios erróneos.

Con el fin de testear las funcionalidades, se revisará contenido de información inicial. También se verificará que existan los campos respectivos con sus etiquetas, y de que la información que se

pueda agregar por parte del usuario sea la correcta, del tipo adecuado, y por supuesto que estos datos puedan ser ingresados por el usuario. Luego, se espera a que el sistema responda por la acción y se verifica si lo obtenido es lo esperado.

Para cumplir con lo especificado, serán testeados elementos visuales como:

- Textos.
- Botones.
- Títulos.
- Campos de texto.
- Selectores.
- Imágenes.
- Enlaces.
- Menú.

Además será testeada su integridad y coherencia

8.1.4. Planificación de Testing

Ya que la estrategia a utilizar es sobre el comportamiento de usuario e integridad de la aplicación, cada vez que se desee testear la aplicación se deberá correr el set completo de pruebas. Por ello se estima que el máximo tiempo de testing no debe pasar los 20 minutos por ejecución.

Por otra parte, como el desarrollo será guiado por pruebas, la ejecución de los test se realizará durante todo el proceso de desarrollo, validación, puesta en marcha, mantención, soporte.

Sin embargo, ya que el código se mantendrá en un repositorio remoto, cada cambio que sea subido a este repositorio generará automáticamente una ejecución completa de las pruebas. Esto será realizado por un ambiente de testing, que será coordinado por un cliente de integración continua. Este cliente se encarga de forma revisar cambios en el repositorio, y al encontrarlos, ejecuta todas las acciones programadas, revisa el estado final de las pruebas, y envía un informe sobre el estado de la ejecución.

8.1.5. Procedimientos de Control

8.1.5.1. Reporte de problemas

Ya que se cuenta con un cliente de integración continua, cada vez que una de sus ejecuciones de resultados fallidos se reportará el error. Este error será analizado y posteriormente se le dará prioridad alta para ser solucionado. Para ello se establece:

- Encargado de reportar el error: Cliente integración continua.
- Tiempo máximo de respuesta frente a error reportado: 1 día.
- Prioridad solución: Muy alta.
- Análisis y planteo de solución: máximo 2 días.
- Encargado solución: Vicente Retamal.

8.1.5.2. Solicitud De Cambio

Si se desea realizar un cambio de funcionalidad en la aplicación durante o después de su desarrollo, esta deberá ser discutida para luego revisar como afecta a la estructura actual del sistema, y como se abordará. Se debe intentar seguir el siguiente esquema:

- Petición de cambio: será generada por todo el equipo, generando tareas en el software de manejo del proyecto.
- Análisis del cambio: el equipo analizará el estado actual de la aplicación, la estrategia de desarrollo a seguir, la integración de la nueva funcionalidad.(tiempo máximo: 5 días).
- Modificación y generación de nueva característica y escenarios en el set de pruebas: se deben generar las pruebas antes de realizar el desarrollo de las nuevas funcionalidades, luego serán subidas al cliente de integración continua. Tiempo máximo: 3 días.
- Desarrollo de la característica: se desarrollará la nueva característica. tiempo: de 1 a 10 días.
- Encargado actualización pruebas: Gustavo Saavedra.
- Encargado desarrollo: Vicente Retamal, Gustavo Saavedra.

8.1.6. Características a ser testeadas

Las características que serán testeadas se pueden agrupar en:

- Ingreso a la aplicación.
- Identificación de usuario.
- Recuperación y cambio de contraseña.
- Registro de un nuevo Ingrediente.
- Eliminación de un ingrediente.
- Actualización de datos de un ingrediente.
- Actualización de datos del usuario.
- Menú y funcionamiento de este.
- Visualización de dashboard de información de datos históricos de ventas.
- Actualización de información en el dashboard.
- Visualización en detalle de información histórica de un ingrediente.
- Ingreso de productos al inventario.
- Cierre de sesión.
- Cierre de cuenta.

8.1.7. Características que no serán testeadas

No serán testeadas ninguna de las siguientes opciones:

- API generada.
- Comunicación de API con cliente (forma explícita).
- Request de la aplicación.
- Modelo de datos.
- Back-end.
- Base de datos.

Ninguno de estos aspectos será testado directamente, ya que si el usuario recibe correctamente el resultado de su acción, esto indica que los aspectos anteriores deben estar funcionando de manera correcta (o al menos, de forma aceptable). Por ello se considera un gasto de recursos innecesarios en tiempo y esfuerzo el probar cada uno de estos aspectos directamente.

8.1.8. Roles y responsabilidades

Los roles en el proceso de testing serán:

- Generación y mantención de ambiente: se debe generar una ambiente de testing y configurar un cliente de interacción continua. Además se debe revisar periódicamente que su funcionamiento sea el ideal Encargado: Gustavo Saavedra.
- Historias de usuario: diseñar historias de usuario y verificar que se adecuen a lo que el usuario necesita . Encargado: Vicente Retamal.
- Reestructuración test para formato testing: se deben escribir las historias en formato Dado/Cuando/Entonces, y generar los diversos escenarios de cada característica. Encargado: Gustavo Saavedra.
- Implementar pasos: se debe implementar y programar como cada uno de los pasos se ejecutará: Gustavo Saavedra
- Ejecución test: Vicente Retamal, Gustavo Saavedra.

8.1.9. Riesgos

Los principales riesgos de el actual plan de pruebas es:

- Pruebas no efectivas: que las pruebas realizadas no abarquen todos los aspectos visibles para el usuario de la aplicación.
- Pruebas insuficientes: que no logren tomar todos los casos de error posibles que el usuario puede sufrir.
- Pruebas no actualizadas: que las pruebas fallen ya que no son las adecuadas para testear el real estado de la aplicación en ese momento.
- Pruebas mal diseñadas: que la interacción programada para las pruebas tenga errores.
- Retraso en el desarrollo: que el desarrollo de la aplicación sea retardado por culpa de que las pruebas no han terminado de construirse.

8.1.10. Criterios de aprobación

Para que el proceso de pruebas sea considerado un éxito todos los escenarios y características deben ser ejecutadas con éxito. Sin embargo, si hay características que no se han implementado aún, todas las demás ya implementadas si deben ser finalizar de forma exitosa. En caso contrario, se debe aplicar el protocolo de control correspondiente.

8.1.11. Pruebas de usabilidad

Para el caso de usabilidad se tomará el sistema de “pasillo de pruebas”, con el fin de verificar:

- Disposición gráfica.
- Facilidad de uso.
- Flujo de navegación.
- Esquema de colores.
- Mejoras.

Al realizar estas pruebas, cada usuario tester contestará una mini encuesta sobre cada uno de estos puntos. Los resultados serán analizados y se sacarán conclusiones de estos. Este proceso se repetirá hasta que los resultados sean adecuados.

8.2. Pruebas

Cada una de las pruebas realizadas se encuentran en la sección de 10 Anexos, en el punto 10.2. Cabe destacar que estas se han realizado bajo el estándar Gherkin, el cual es de gran utilidad para los proyectos realizados con BDD.

8.2.1. Resultados Pruebas

Los resultados para las pruebas fueron los siguientes:

- Monitor Cocina:
 - Ensayo N° 1: Insatisfactorio - bajo 30 %
 - Ensayo N° 2: Incompleto - bajo 60 % (Falta de implementación de opciones por color)
 - Ensayo N° 3: Satisfactorio - 90 % (errores menores en css)
 - Ensayo N° 4: Exitoso - 100 % (Todos los errores corregidos) (01 / 12 /2016)
- Caja:
 - Ensayo N° 1: Insatisfactorio - bajo 40 % (No se despliega Data)
 - Ensayo N° 2: Insatisfactorio - bajo 20 % (Errores de visualización y despliegue)
 - Ensayo N° 3: Satisfactorio - 70 % (Se muestra datos parcialmente)
 - Ensayo N° 4: Satisfactorio - 75 % (Botones deshabilitados)
 - Ensayo N° 4: Satisfactorio - 85 % (Botón anular no actualiza visualización)
 - Ensayo N° 5: Éxito - 100 % (02 / 12 /2016)

9. Conclusión

Se debe recordar que el proyecto actual trata de resolver el problema que el cliente presenta, ¿Cómo poder vender un sándwich con un precio específico dependiendo de la cantidad de cada ingrediente?. Si bien esto es lo que se desea resolver, no por ello es el único problema o el de mayor importancia. Es así como se torna necesario un trabajo conjunto, para poder guiar a este stakeholder a descubrir lo que realmente necesita. Es un trabajo constante y cíclico que puede demorar bastante tiempo. Para este caso, además, quien solicita los servicios no ha concretado su emprendimiento, por lo cual es aún más indispensable el brindar apoyo a su labor.

Para poder realizar este trabajo, se utilizaron paradigmas y metodologías adecuadas al caso, ágil y Kanban-BDD. Por una parte, el paradigma ágil permite una mayor libertad en el proceso creativo, en incentivar el trabajo conjunto, optimizar el tiempo, reaccionar a los cambios y mejorar la comunicación del equipo, generando instancias para compartir conocimiento y superar problemas que otros miembros del equipo ya han sufrido. Por otra parte, kanban entrega las pautas para organizar el trabajo de forma fácil y BDD posibilita la seguridad durante el proceso de desarrollo y evita codificación innecesaria.

La solución ideada no solo ha contemplado la generación de un software, sino que también de un dispositivo de medición. Para desarrollarlo ha sido necesaria la utilización de prototipos funcionales, con la que se pueda testear la viabilidad del proyecto, pero que además entregue una visión realista al cliente. Es así, como la utilización de herramientas de prototipado rápido, como arduino, es fundamental, pero añade dificultad para el equipo y para quienes no conozcan de estos dispositivos. Sin embargo, la documentación para su uso es abundante, y, al invertir y planificar tiempo para su correcta utilización, ha posibilitado el correcto diseño del proyecto.

En resumen, con el desarrollo ha sido posible comprender la importancia de arriesgarse y no sucumbir ante el miedo a innovar, a indagar en temas de los cuales no se tiene conocimiento. El foco siempre debe estar en generar una solución adecuada para el cliente y responder de mejor forma a sus intereses. Escoger las herramientas adecuadas es fundamental para llevar a cabo un trabajo expedito y que culmine en un éxito. Así, se puede conseguir a futuro, el propósito final de este proyecto: generar una opción grata y justa para adquirir comida rápida, de manera sencilla y sustentable.

10. Anexos

10.1. Carta Gantt

Tarea	Duración	Inicio	Fin
Generar Dispositivo	16 días	lun 05-09-16	lun 26-09-16
Identificación del dolor	1 día	lun 05-09-16	lun 05-09-16
Identificación de requerimientos	1 día	mar 06-09-16	mar 06-09-16
Análisis de requerimientos	2 días	mié 07-09-16	jue 08-09-16
Validación de requerimientos	1 día	vie 09-09-16	vie 09-09-16
Investigación de sensores	1 día	lun 12-09-16	lun 12-09-16
Prototipado	3 días	mar 13-09-16	jue 15-09-16
Pruebas de conectividad	2 días	vie 16-09-16	lun 19-09-16
Cambios de prototipo	3 días	mar 20-09-16	jue 22-09-16
Pruebas de funcionamiento	2 días	vie 23-09-16	lun 26-09-16
Generar Plataforma	24 días	mar 27-09-16	jue 13-11-16
Identificación de requerimientos	1 día	mar 27-09-16	mar 27-09-16
Análisis de requerimientos	1 día	mié 28-09-16	mié 28-09-16
Validación de requerimientos	1 día	jue 29-09-16	jue 29-09-16
Investigación de Tecnologías	2 días	vie 30-09-16	lun 03-10-16
Diseño de plataforma	3 días	mar 04-10-16	jue 06-10-16
Codificación Plataforma	24 días	vie 07-10-16	mar 11-11-16
Pruebas de plataforma	2 días	mié 12-11-16	jue 13-11-16
Comunicación del dispositivo	10 días	vie 14-11-16	jue 27-11-16
Análisis de requerimientos	1 día	vie 14-11-16	vie 14-11-16
Investigación de alternativas	1 día	lun 17-11-16	lun 17-11-16
Identificar problemas	1 día	mar 18-11-16	mar 18-11-16
Elección de comunicación	1 día	mié 19-11-16	mié 19-11-16
Conexión con servidor	3 días	jue 20-11-16	lun 24-11-16
Pruebas de conexión	3 días	mar 25-11-16	jue 27-11-16
Análisis de información	6 días	vie 28-11-16	lun 05-12-16
Identificación de datos	2 días	vie 28-10-16	lun 31-10-16
Jerarquización de datos	2 días	mar 01-11-16	mié 02-11-16
Diseño algoritmo de análisis de datos	3 días	jue 03-11-16	lun 07-11-16
Codificación de análisis de datos	3 días	mar 08-11-16	jue 10-11-16
Almacenamiento información refinada	2 días	vie 11-11-16	lun 14-11-16

Tabla 9: Carta Gantt

10.2. Pruebas

10.2.1. Caja

```
1 # language: es
2 Característica: Ingresar al sistema de cajas
3   Como usuario
4   Quiero poder ingresar al sistema de cajas
5   Y revisar las ordenes activas
6
7   Escenario: No hay ordenes Activas
8     Dado que el cajero ingresa a la aplicación
9     Y que hay "0" ordenes activas
10    Entonces puede ver la pantalla "principal" de la aplicación
11    Y en el listado de ordenes activas hay "0" ordenes
12
13   Escenario: Hay ordenes activas
14     Dado que el cajero ingresa a la aplicación
15     Y que hay "1 o mas" ordenes activas
16     Entonces puede ver la pantalla "principal" de la aplicación
17     Y en el listado de ordenes activas hay "1 o mas" ordenes
18
19   Escenario: Hay ordenes activas sin ingredientes
20     Dado que el cajero ingresa a la aplicación
21     Y que hay "1 o mas" ordenes activas
22     Entonces puede ver la pantalla "principal" de la aplicación
23     Y en el listado de ordenes activas hay "1 o mas" ordenes
24     Cuando elige el pedido con total "0"
25     Entonces debe ver en "Ingredientes" un total de "0"
26     Y un listado con "0" ingredientes
27
28   Escenario: Hay ordenes activas sin ingredientes
29     Dado que el cajero ingresa a la aplicación
30     Y que hay "1 o mas" ordenes activas
31     Entonces puede ver la pantalla "principal" de la aplicación
32     Y en el listado de ordenes activas hay "1 o mas" ordenes
33     Cuando elige el pedido con total "mayor a 0"
34     Entonces debe ver en "Ingredientes" un total de "mayor a 0"
35     Y un listado con "1 o mas" ingredientes
36
37   Escenario: Anular pedido
38     Dado que el cajero ingresa a la aplicación
39     Y que hay "1 o mas" ordenes activas
40     Entonces puede ver la pantalla "principal" de la aplicación
41     Y en el listado de ordenes activas hay "1 o mas" ordenes
42     Cuando elige el pedido con total "1000"
43     Entonces debe ver en "Ingredientes" un total de "1000"
44     Y un listado con "1 o mas" ingredientes
```

45 **Cuando** el cajero presiona el boton de "Anular"
46 **Y** presiona al boton de confirmacion con "Si"
47 **Entonces** en el listado de ordenes activas no puede ver el
 pedido de total "1000"

48
49 **Escenario:** Imprimir pedido
50 **Dado** que el cajero ingresa a la aplicación
51 **Y** que hay "1 o mas" ordenes activas
52 **Entonces** puede ver la pantalla "principal" de la aplicación
53 **Y** en el listado de ordenes activas hay "1 o mas" ordenes
54 **Cuando** elige el pedido con total "1000"
55 **Entonces** debe ver en "Ingredientes" un total de "1000"
56 **Y** un listado con "1 o mas" ingredientes
57 **Cuando** el cajero presiona el boton de "Imprimir"
58 **Y** presiona al boton de confirmacion con "Si"
59 **Entonces** debe ver un mensaje de "imprimiendo"
60 **Y** en el listado de ordenes activas no puede ver el pedido de
 total "1000"

61
62 **Escenario:** Pagar pedido
63 **Dado** que el cajero ingresa a la aplicación
64 **Y** que hay "1 o mas" ordenes activas
65 **Entonces** puede ver la pantalla "principal" de la aplicación
66 **Y** en el listado de ordenes activas hay "1 o mas" ordenes
67 **Cuando** elige el pedido con total "1000"
68 **Entonces** debe ver en "Ingredientes" un total de "1000"
69 **Y** un listado con "1 o mas" ingredientes
70 **Cuando** el cajero presiona el boton de "Pagar"
71 **Y** presiona al boton de confirmación con "Si"
72 **Entonces** debe ver un mensaje de "imprimiendo"
73 **Y** en el listado de ordenes activas no puede ver el pedido de
 total "1000"

Aplicación Caja

10.2.2. Monitor Cocina

1 # language: es
2 Característica: Monitor de cocina
3 Como usuario
4 Quiero poder ingresar al monitor de cocina
5 **y** visualizar el estado de los ingredientes
6
7 **Escenario:** Gráficos en monitor
8 **Dado** que el cocinero hace uso de la aplicación
9 **Entonces** debe poder ver un gráfico del estado actual de "Mesón"
10 **Y** debe poder ver un gráfico del estado actual de "Inventario"
11
12 **Escenario:** Existen ingredientes en mesón

13 Dado que el cocinero hace uso de la aplicación
14 Entonces debe poder ver un gráfico del estado actual de "Mesón"
15 Y debe poder ver a lo menos "1" barra de ingredientes en "Mesón"
"
16 Y debe poder ver el porcentaje del ingrediente en "Mesón"
17
18 Escenario: Existen ingredientes en inventario
19 Dado que el cocinero hace uso de la aplicación
20 Entonces debe poder ver un gráfico del estado actual de "
Inventario"
21 Y debe poder ver a lo menos "1" barra de ingredientes en "
Inventario"
22 Y debe poder ver el porcentaje del ingrediente en "Inventario"
23
24 Escenario: Existen ingredientes en meson e inventario
25 Dado que el cocinero hace uso de la aplicación
26 Entonces debe poder ver un gráfico del estado actual de "Mesón"
27 Y debe poder ver un gráfico del estado actual de "Inventario"
28 Y debe poder ver a lo menos "1" barra de ingredientes en "Mesón"
"
29 Y debe poder ver el porcentaje del ingrediente en "Mesón"
30 Y debe poder ver a lo menos "1" barra de ingredientes en "
Inventario"
31 Y debe poder ver el porcentaje del ingrediente en "Inventario"
32
33 Escenario: Existen ingredientes con un estado normal en mesón
34 Dado que el cocinero hace uso de la aplicación
35 Y existe al menos "1" ingrediente con estado normal en "Mesón"
36 Entonces debe poder ver un gráfico del estado actual de "Mesón"
37 Y debe poder ver a lo menos "1" barra de ingredientes en "Mesón"
" de color "Azul"
38 Y debe poder ver el porcentaje del ingrediente en "Mesón" con
un valor "mayor a 30"
39
40 Escenario: Existen ingredientes con un estado medio en mesón
41 Dado que el cocinero hace uso de la aplicación
42 Y existe al menos "1" ingrediente con estado medio en "Mesón"
mayor
43 Entonces debe poder ver un gráfico del estado actual de "Mesón"
44 Y debe poder ver a lo menos "1" barra de ingredientes en "Mesón"
" de color "Naranja"
45 Y debe poder ver el porcentaje del ingrediente en "Mesón" con
un valor "mayor a 20"
46
47 Escenario: Existen ingredientes con un estado critico en mesón
48 Dado que el cocinero hace uso de la aplicación
49 Y existe al menos "1" ingrediente con estado critico en "Mesón"
50 Entonces debe poder ver un gráfico del estado actual de "Mesón"

51 Y debe poder ver a lo menos "1" barra de ingredientes en "Mesón
" de color "Rojo"

52 Y debe poder ver el porcentaje del ingrediente en "Mesón" con
un valor "menor a 20"

53

54 **Escenario:** Existen ingredientes con un estado cero en mesón

55 **Dado** que el cocinero hace uso de la aplicación

56 **Y** existe al menos "1" ingrediente con estado cero en "Mesón"

57 **Entonces** debe poder ver un gráfico del estado actual de "Mesón"

58 **Y** debe poder ver a lo menos "1" barra de ingredientes en "Mesón
" de color "Gris"

59 **Y** debe poder ver el porcentaje del ingrediente en "Mesón" con
un valor "0"

60

61 **Escenario:** Se quita una gran cantidad de un ingrediente en "Mesón
"

62 **Dado** que el cocinero hace uso de la aplicación

63 **Y** existe el ingrediente "Alcachofa" con porcentaje "50" en "Mes
ón"

64 **Entonces** debe poder ver un gráfico del estado actual de "Mesón"

65 **Cuando** se quita una gran cantidad del ingrediente "Alcachofa"
en "Mesón"

66 **Entonces** debe poder ver el ingrediente "Alcachofa" con un
porcentaje "menor a 50"

Aplicación Monitor

10.2.3. Administración

1 # language: es

2 Característica: Sistema de Administracion

3 Como usuario Administrador

4 Quiero poder ingresar al panel de administracion

5 y modificar, agregar o eliminar características

6

7 **Escenario:** Ingreso a aplicacion

8 **Dado** que el administrador ingresa a la aplicacion

9 **Entonces** debe poder ver una pantalla de identificación

10 **Y** debe poder ver un campo de "Usuario"

11 **Y** debe poder ver un campo de "Contraseña"

12

13 **Escenario:** Ingreso a aplicacion con credenciales incorrectas

14 **Dado** que el administrador ingresa a la aplicacion

15 **Entonces** debe poder ver una pantalla de identificación

16 **Y** en el campo de "Usuario" ingresa "cualquierUsuario"

17 **Y** en el campo de "Contraseña" ingresa "cualquierContraseña"

18 **Y** presiona el boton de "Ingresar"

19 **Entonces** debe ver un mensaje de error con "Credenciales invá
lidas"

20

21 **Escenario:** Ingreso a aplicacion con credenciales correctas

22 **Dado** que el administrador ingresa a la aplicacion

23 **Entonces** debe poder ver una pantalla de identificación

24 **Y** en el campo de "Usuario" ingresa "Admin"

25 **Y** en el campo de "Contraseña" ingresa "PasswordAdmin"

26 **Y** presiona el boton de "Ingresar"

27 **Entonces** debe ver el elemento "Admin"

28 **Y** el boton "Exit"

29

30 **Escenario:** Editar Precio de Ingrediente

31 **Dado** que el administrador ingresa a la aplicacion

32 **Y** que se encuentra autenticado con credenciales "Admin" y "
PasswordAdmin"

33 **Entonces** debe poder ver la pantalla de inicio

34 **Cuando** va a la sección de "Ingredientes"

35 **Y** escoge la opción "Editar Precios"

36 **Entonces** puede ver la sección de "Agregar Ingrediente"

37 **Cuando** en el campo "Lechuga" completa con "4000"

38 **Y** presiona el boton "Actualizar"

39 **Entonces** puede ver un mensaje de "Sus ingredientes han sido
actualizados"

40 **Cuando** va a la sección de "Ingredientes"

41 **Y** escoge la opción "Editar Precios"

42 **Entonces** puede ver en el campo "Lechuga" el valor "4000"

43

44 **Escenario:** Añadir Ingrediente

45 **Dado** que el administrador ingresa a la aplicacion

46 **Y** que se encuentra autenticado con credenciales "Admin" y "
PasswordAdmin"

47 **Entonces** debe poder ver la pantalla de inicio

48 **Cuando** va a la sección de "Ingredientes"

49 **Y** escoge la opción "Agregar Ingrediente"

50 **Entonces** puede ver la sección de "Agregar Ingrediente"

51 **Cuando** en el campo "Nombre" completa con "Queso Mantecoso"

52 **Y** en el campo "Precio Unitario" completa con "3500"

53 **Y** de el campo "Tipo" selecciona la opción "Quesos"

54 **Y** presiona el boton "Añadir"

55 **Entonces** puede ver un mensaje de "Ha Añadido un nuevo
Ingrediente"

56 **Cuando** va a la sección de "Ingredientes"

57 **Y** escoge la opción "Editar Precios"

58 **Entonces** puede ver el ingrediente "Queso Mantecoso"

59

60 **Escenario:** Sección Administrador

61 **Dado** que el administrador ingresa a la aplicacion

62 **Y** que se encuentra autenticado con credenciales "Admin" y "
PasswordAdmin"

63 **Entonces** debe poder ver la pantalla de inicio
64 **Cuando** va a la sección de "Administrador"
65 **Entonces** puede ver la lista de usuarios "Administrador"
66 **Y** puede ver un boton de agregar usuarios
67 **Y** puede ver un boton de eliminar usuarios
68
69 **Escenario:** Sección Administrador
70 **Dado** que el administrador ingresa a la aplicacion
71 **Y** que se encuentra autenticado con credenciales "Admin" y "
 PasswordAdmin"
72 **Entonces** debe poder ver la pantalla de inicio
73 **Cuando** va a la sección de "Administrador"
74 **Entonces** puede ver la lista de usuarios "Administrador"
75 **Cuando** presiona el boton "agregar usuarios"
76 **Entonces** puede ver la sección de "Agregar Usuarios"
77 **Cuando** en el campo "email" rellena con "emailprueba@gmail.com"
78 **Y** en el campo "usuario" rellena con "SerAdmin"
79 **Y** en el campo "contraseña" rellena con "SerAdminPassword"
80 **Y** presiona el boton "Añadir"
81 **Entonces** puede ver un mensaje de "exito" indicando "Ha añadido
 un nuevo Administrador"
82
83 **Escenario:** Revisión de Dashboard
84 **Dado** que el administrador ingresa a la aplicacion
85 **Y** que se encuentra autenticado con credenciales "Admin" y "
 PasswordAdmin"
86 **Entonces** debe poder ver la pantalla de inicio
87 **Cuando** va a la sección de "Dashboard"
88 **Entonces** puede ver a lo menos "1" gráfico con información histó
 rica
89
90 **Escenario:** Revisión de Ventas
91 **Dado** que el administrador ingresa a la aplicacion
92 **Y** que se encuentra autenticado con credenciales "Admin" y "
 PasswordAdmin"
93 **Entonces** debe poder ver la pantalla de inicio
94 **Cuando** va a la sección de "Ventas"
95 **Entonces** puede ver la lista con el detalle de las ventas histó
 ricas

Aplicación Administración

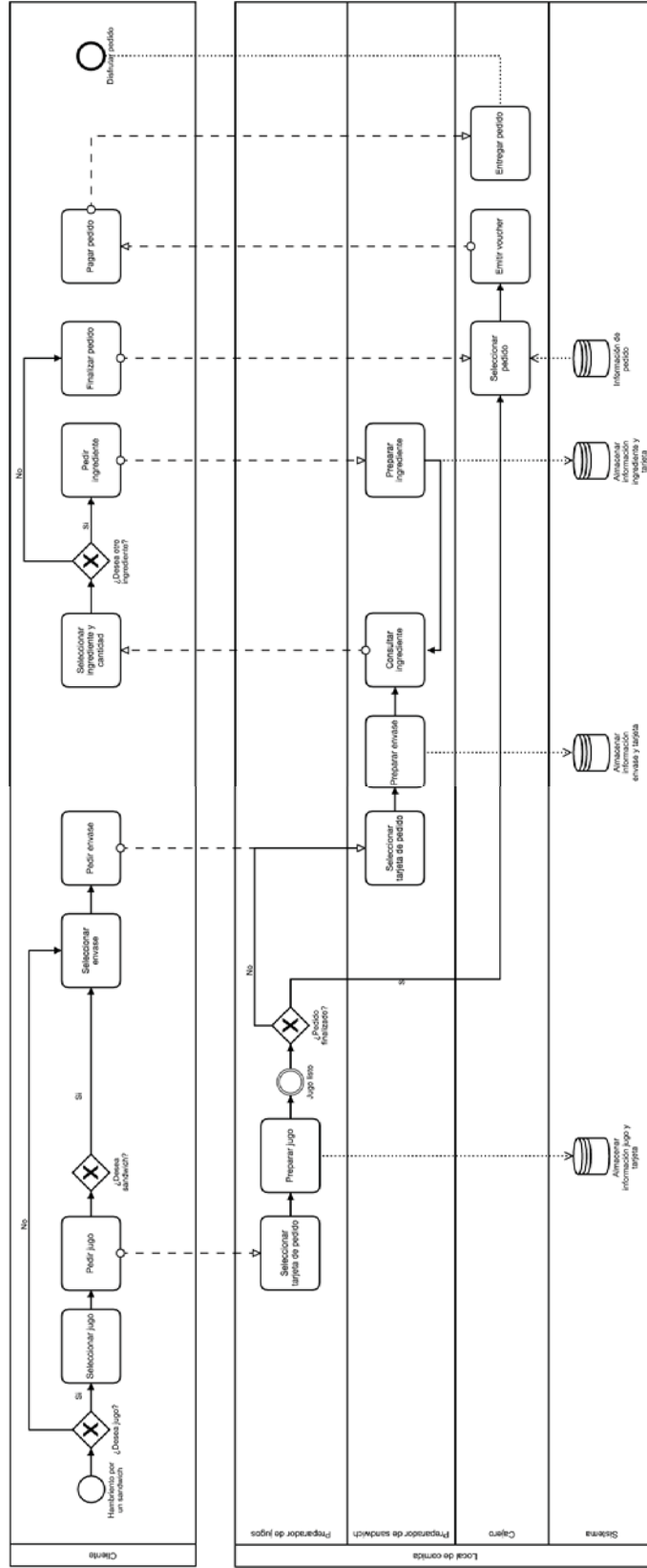


Figura 11: Modelo de proceso de negocio

Referencias

- [1] Architecture of a Database System. <http://www.agilemodeling.com/essays/agileArchitecture.html>, 2007. [Online; accedido 04-October-2016].
- [2] Agile Architecture: Strategies for Scaling Agile Development. <http://db.cs.berkeley.edu/papers/fntdb07-architecture.pdf>, 2008. [Online; accedido 04-October-2016].
- [3] Desarrollo de un prototipo para el seguimiento y cuidado de jardines urbanos basado en la recopilación de datos a través de la plataforma de arduino. <http://repositorio.puce.edu.ec/bitstream/handle/22000/11267/Tesis.pdf?sequence=1>, 2009. [Online; accedido 04-October-2016].
- [4] Kanban vs Scrum How to make the most of both . <https://www.crisp.se/file-uploads/Kanban-vs-Scrum.pdf>, 2009. [Online; accedido 04-October-2016].
- [5] The cucumber book . <https://wiki.kuali.org/download/attachments/329384409/The%20Cucumber%20Book.pdf?version=1&modificationDate=1407504510768&api=v2>, 2009. [Online; accedido 04-October-2016].
- [6] 7 Key Principles of Lean Software Development. <http://www.allaboutagile.com/7-key-principles-of-lean-software-development-2/>, 2010. [Online; accedido 04-October-2016].
- [7] Lean Software Development. <https://www.computer.org/csdl/mags/so/2012/05/mso2012050022.pdf>, 2012. [Online; accedido 18-October-2016].
- [8] MQTT Tutorial for Raspberry Pi, Arduino, and ESP8266. <https://www.baldengineer.com/mqtt-tutorial.html>, 2014. [Online; accedido 28-October-2016].
- [9] Distribuidor de Arduino y accesorios. <http://www.amgkits.com/>, 2015. [Online; accedido 04-October-2016].
- [10] Docker - Build, Ship, and Run Any App, Anywhere. <https://www.docker.com/>, 2015. [Online; accedido 04-October-2016].
- [11] Introducing BDD. <https://dannorth.net/introducing-bdd/>, 2015. [Online; accedido 10-October-2016].