



**UNIVERSIDAD MAYOR DE SAN SIMON**  
**FACULTAD DE CIENCIAS Y TECNOLOGÍA**  
**DIRECCIÓN DE POSTGRADO**



## **APLICACIÓN DE SCRUM Y UML PARA EL DESARROLLO DE UN SISTEMA DE VENTAS**

TRABAJO FINAL PRESENTADO PARA OBTENER EL CERTIFICADO DE  
DIPLOMADO EXPERTO EN DESARROLLO DE APLICACIONES  
EMPRESARIALES VERSION III

Autor:  
Saúl Mamani Mamani

Cochabamba – Bolivia

2019

### **Dedicatoria y agradecimientos**

A Dios, a mis padres, a mi novia, y a mis docentes por todo su apoyo

Gracias por compartir su conocimiento

# ÍNDICE GENERAL

Resumen .....	1
1 Introducción .....	2
2 Antecedentes.....	2
2.1 Antecedentes generales .....	2
2.2 Antecedentes específicos .....	3
3 Planteamiento del problema .....	4
4 Metodología de investigación.....	4
4.1 Método bibliográfico.....	4
4.2 Método analítico .....	4
4.3 Método empírico o de campo .....	4
5 Justificación .....	4
6 Aportes .....	5
7 Alcances y limitaciones.....	5
8 Marco teórico.....	6
8.1 Manifiesto ágil.....	6
8.2 Scrum .....	6
8.3 Historias de usuario .....	7
8.4 Lenguaje de Modelado Unificado.....	7
8.5 Product Vision Board .....	8
8.6 Trello .....	8
8.7 Git.....	8
8.8 Heroku.....	9
8.9 Puntos de historia .....	9
9 Marco Práctico.....	10
9.1 Modelo del negocio.....	10
9.1.1 Product Vision Board .....	10
9.2 Roles y stakeholders.....	10
9.3 Historias de usuario .....	10
9.4 Product backlog .....	14
9.5 Estimación del product backlog .....	15
9.6 Definición de hecho .....	16
9.7 Definición de la arquitectura .....	16
9.7.1 Diagrama de componentes.....	16

9.8	Primer sprint .....	17
9.8.1	Sprint planning.....	17
9.8.2	Diagrama de casos de uso .....	17
9.8.3	Diagrama de clases persistentes .....	18
9.8.4	Especificación de casos de uso .....	19
9.8.5	Sprint backlog.....	23
9.8.6	Sprint execution.....	23
9.8.7	Sprint review.....	25
9.8.8	Sprint retrospective.....	27
9.9	Segundo sprint .....	28
9.9.1	Sprint planning.....	28
9.9.2	Diagrama de casos de uso .....	28
9.9.3	Diagrama de clases persistentes .....	29
9.9.4	Especificación de casos de uso .....	29
9.9.5	Sprint backlog.....	32
9.9.6	Sprint execution.....	32
9.9.7	Sprint review.....	34
9.9.8	Sprint retrospective.....	35
9.10	Tercer sprint .....	36
9.10.1	Sprint planning.....	36
9.10.2	Diagrama de casos de uso .....	36
9.10.3	Diagrama de clases persistentes .....	37
9.10.4	Especificación de casos de uso .....	37
9.10.5	Sprint backlog.....	38
9.10.6	Sprint execution.....	39
9.10.7	Sprint review.....	41
9.10.8	Sprint retrospective.....	43
9.11	Diagrama general de casos de uso .....	44
9.12	Diagrama de despliegue.....	45
10	Conclusiones y recomendaciones .....	46
10.1	Conclusiones .....	46
10.2	Recomendaciones .....	46
	Bibliografía.....	47

## ÍNDICE DE TABLAS

Tabla 1: Puntos historia .....	9
Tabla 2: Product Vision Board del proyecto .....	10
Tabla 3: Historia de usuario – Ingresar al sistema como administrador .....	11
Tabla 4: Historia de usuario – Ingresar al sistema como vendedor .....	11
Tabla 5: Historia de usuario – Gestionar usuarios del sistema .....	11
Tabla 6: Historia de usuario – Administrar perfil de usuario .....	11
Tabla 7: Historia de usuario – Cambiar contraseña .....	11
Tabla 8: Historia de usuario – Gestionar productos .....	12
Tabla 9: Historia de usuario – Gestionar clientes .....	12
Tabla 10: Historia de usuario – Listar ventas .....	12
Tabla 11: Historia de usuario – Registrar ventas.....	12
Tabla 12: Historia de usuario – Generar número de ticket .....	13
Tabla 13: Historia de usuario – Generar recibo.....	13
Tabla 14: Historia de usuario – Anular ventas.....	13
Tabla 15: Historia de usuario – Restablecer ventas .....	13
Tabla 16: Historia de usuario – Generar informe económico.....	14
Tabla 17: Historia de usuario – Generar reportes estadísticos.....	14
Tabla 18: Product backlog .....	15
Tabla 19: Product backlog estimado .....	15
Tabla 20: Primer sprint planning .....	17
Tabla 21: Caso de uso iniciar sesión .....	19
Tabla 22: Caso de uso gestionar usuarios .....	20
Tabla 23: Caso de uso gestionar clientes .....	21
Tabla 24: Caso de uso gestionar productos.....	22
Tabla 25: Caso de uso Administrar perfil .....	22
Tabla 26: Caso de uso Cambiar contraseña .....	22
Tabla 27: Primer sprint backlog .....	23
Tabla 28: Segundo sprint planning.....	28
Tabla 29: Caso de uso Listar ventas.....	30
Tabla 30: Caso de uso Registrar ventas .....	30
Tabla 31: Caso de uso Generar número de ticket.....	31
Tabla 32: Caso de uso Generar recibo .....	31
Tabla 33: Segundo sprint backlog.....	32

Tabla 34: Tercer sprint planning .....	36
Tabla 35: Caso de uso Anular ventas .....	37
Tabla 36: Caso de uso Restablecer ventas.....	38
Tabla 37: Caso de uso Generar reportes .....	38
Tabla 38: Tercer sprint backlog.....	38

## ÍNDICE DE ILUSTRACIONES

Ilustración 1: Contextos de cynefin .....	3
Ilustración 2: Scrum framework.....	6
Ilustración 3: Practicas de SCRUM .....	7
Ilustración 4: Modelo de historia de usuario .....	7
Ilustración 5: Diagramas UML.....	8
Ilustración 6: Plantilla Produc Vision Board .....	8
Ilustración 7: Diagrama de componentes .....	16
Ilustración 8: Diagrama de casos de uso primer sprint.....	18
Ilustración 9: Diagrama de clases primer sprint.....	18
Ilustración 10: Tablero Kanban primer sprint.....	23
Ilustración 11: Tablero Kanban primer sprint.....	24
Ilustración 12: Control de versiones del primer sprint.....	24
Ilustración 13: Modelo relacional de la base de datos del primer sprint.....	25
Ilustración 14: Pantalla Login .....	25
Ilustración 15: Pantalla administrar perfil de usuario .....	26
Ilustración 16: Pantalla producto.....	26
Ilustración 17: Seguimiento de historias primer sprint .....	27
Ilustración 18: Burndown chart primer sprint .....	27
Ilustración 19: Diagrama de casos de uso segundo sprint .....	28
Ilustración 20: Diagrama de clases segundo sprint .....	29
Ilustración 21: Tablero Kanban segundo sprint .....	32
Ilustración 22: Tablero Kanban segundo sprint .....	32
Ilustración 23: Control de versiones del segundo sprint .....	33
Ilustración 24: Modelo relacional de la base de datos del segundo sprint .....	33
Ilustración 25: Pantalla registrar ventas .....	34
Ilustración 26: Pantalla generar recibo.....	34
Ilustración 27: Seguimiento de tareas segundo sprint .....	35
Ilustración 28: Burndown chart segundo sprint.....	35
Ilustración 29: Diagrama de casos de uso tercer sprint.....	36
Ilustración 30: Diagrama de clases tercer sprint.....	37
Ilustración 31: Tablero Kanban tercer sprint.....	39
Ilustración 32: Tablero Kanban tercer sprint.....	39
Ilustración 33: Control de versiones del tercer sprint.....	40

Ilustración 34: Modelo relacional de la base de datos del tercer sprint.....	40
Ilustración 35: Pantalla restablecer ventas.....	41
Ilustración 36: Pantalla ventas anuladas.....	41
Ilustración 37: Pantalla reporte de ventas .....	42
Ilustración 38: Pantalla reporte estadístico.....	42
Ilustración 39: Seguimiento de tareas tercer sprint .....	43
Ilustración 40: Burndown chart tercer sprint.....	43
Ilustración 41: Diagrama general de casos de uso.....	44
Ilustración 42: Diagrama de despliegue .....	45

## RESUMEN

La presente monografía muestra cómo una metodología ágil puede trabajar en conjunto con una notación de modelado visual, sin afectar los principios y valores del manifiesto ágil.

Se emplea Scrum como metodología ágil de desarrollo de software; aplicando sus principios, valores y prácticas.

Para modelar de forma visual la funcionalidad, la arquitectura lógica, y la arquitectura física del sistema; se utilizan algunos diagramas del Lenguaje de Modelado Unificado. UML 2.5.

Estos conceptos se ponen en práctica desarrollando un sistema para la venta de comidas y bebidas. Se comprende mejor la visión del proyecto con un product vision board, los requerimientos funcionales son capturados con historias de usuario escritas por el product owner, se organizan y priorizan las historias de usuario en un product backlog, se estiman el esfuerzo y el tiempo de cada historia, y se dividen las historias en tres iteraciones (sprints). El modelado visual de cada iteración se lo realiza con diagramas de casos de uso y diagramas de clases. Además, se realiza el seguimiento de todas las tareas programadas en un tablero Kanban, y se lleva un historial de todo el código fuente con una herramienta de control de versiones. En cada iteración se realiza el sprint review y el sprint retrospectivo.

Para finalizar, se muestra la funcionalidad global del sistema en un diagrama de casos de uso, un diagrama de componentes muestra la arquitectura del sistema y lo componentes necesarios para desarrollarlo, y un diagrama de despliegue muestra la arquitectura física de la red, los servidores y la tecnología necesaria para que el sistema sea puesto en producción.

## 1 INTRODUCCIÓN

La información es considerada el activo más importante en cualquier institución, empresa u organización. La buena o mala administración de este activo es fundamental, principalmente en lo que se refiere a la toma de decisiones, ya que podría determinar el éxito o el fracaso de los proyectos o tareas que realizan la institución. Es por esto que las instituciones dependen cada vez más de la creación, gestión y publicación de sistemas informáticos; capaces de automatizar y administrar su información de manera fiable y eficiente.

En este contexto la ingeniería de software juega un papel muy importante, ya que nos brinda un conjunto de metodologías, métodos, técnicas, y herramientas para desarrollar sistemas software fiables, que cumplan con los requerimientos de los usuarios, y con los estándares de calidad impuestos por la industria del software.

## 2 ANTECEDENTES

### 2.1 Antecedentes generales

El desarrollo de software se puede realizar de muchas maneras. Hay diferentes procesos de desarrollo que los proyectos siguen, desde el modelo de la cascada hasta los procesos basados en objetivos, pasando por los procesos iterativos e incrementales hasta las metodologías ágiles.

Durante mucho tiempo el desarrollo de software siguió un proceso en cascada. En este modelo analizamos los requisitos, diseñamos, desarrollamos, probamos y por último desplegamos el sistema. Una de las principales limitaciones del modelo en cascada es que no permite retroalimentación en cada una de sus etapas, después de algún tiempo cuando terminamos el proyecto y se lo entregamos a los usuarios probablemente el negocio ha experimentado cambios mientras nosotros construimos el sistema, y los usuarios responden "Esto es lo que pedí, pero ya no me sirve".

Por todas las limitaciones del modelo en cascada surge el concepto de desarrollo iterativo e incremental. Un proyecto puede verse como una serie de pequeñas cascadas, realizando entregas incrementales a los usuarios finales, de los cuales recibimos retroalimentación para la siguiente iteración. Siguiendo esta visión aparecen un sin fin de modelos; como la metodología en Espiral, Prototipo, Incremental, el Proceso Unificado de desarrollo (UP o sus variantes RUP, AUP, EUP, etc.), Ingeniería Web basada en UML (UWE), etc. Sin embargo, estos modelos se enfocan en una planificación estricta antes de comenzar el desarrollo del sistema, esto conlleva mucha documentación, burocracia e iteraciones largas; esto es bueno para sistemas cuyos requisitos no cambian mucho, pero es un verdadero problema en sistemas que están en constante cambio.

Por esta razón surgen las metodologías ágiles, como SCRUM, programación extrema (XP), Kanban, etc. cuya visión se centra en el manifiesto ágil<sup>1</sup>. Estas metodologías se enfocan en el trabajo en equipo, se considera al cliente como parte del equipo, y se realizan entregables e iteraciones cortas para que el usuario o el cliente constantemente realice retroalimentación para construir un producto de calidad acorde a sus necesidades.

La comunicación entre los clientes, usuarios y desarrolladores es muy importante para construir el sistema. Los clientes exponen sus necesidades comerciales y los desarrolladores lo plasman en requisitos funcionales y estos requisitos son transformados en código fuente; a este proceso se lo conoce como modelado. El modelado se lo puede realizar de forma

---

<sup>1</sup> El manifiesto ágil es un documento que postula 4 valores y 12 principios para el desarrollo del software con metodologías ágiles

textual, pero se mejora bastante si se utiliza una notación visual como el que nos ofrece el Lenguaje de Modelado Unificado (UML).

## 2.2 Antecedentes específicos

«Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.» Kent Beck.

Scrum es un marco de trabajo ágil que está basado en un pequeño conjunto de valores, principios y prácticas. Scrum ha ganado popularidad en estos últimos años ya que se adapta bien en proyecto cuyos requisitos cambian constantemente, proyectos no predecibles, y proyectos de entorno complejos con gran incertidumbre.

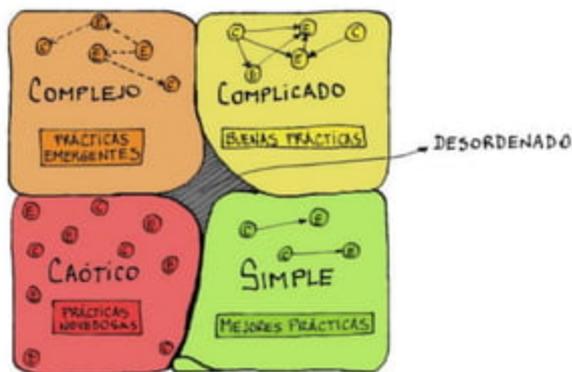


Ilustración 1: Contextos de cynefin  
Fuente: <https://martinalaimo.com/es/blog/cynefin>

El Lenguaje de Modelado Unificado ofrece una notación visual para modelar los artefactos de los sistemas orientados a objetos, se considera un estándar de facto para el análisis y diseño de sistemas. UML 2.5 ofrece un conjunto de diagramas que ayudan a mejorar la comunicación entre todos los involucrados en el desarrollo del proyecto.

Con el fin de aplicar estos conceptos en un ejemplo práctico, esta monografía tiene como objetivo desarrollar un sistema de ventas aplicando la notación de modelado visual UML y la metodología ágil SCRUM.

Para lograr alcanzar este objetivo, se realizarán las siguientes tareas:

- Elaborar un product vision board para entender mejor el sistema que se va a desarrollar y determinar la visión del proyecto.
- Capturar requerimientos funcionales, para determinar el ámbito del sistema a través de historias de usuario escritas por los mismos clientes o por el product owner.
- Planificar las iteraciones necesarias para realizar la entrega de incrementos de calidad que proporcionen valor al cliente.
- Estimar el esfuerzo y el tiempo de las historias de usuarios con el método puntos historia y planning poker.
- Identificar los diagramas UML necesarios para modelar de manera visual los artefactos del sistema.
- Utilizar herramientas de control de versiones, para llevar un control adecuado de los cambios a lo largo del desarrollo del sistema.

- Desarrollar el sistema de ventas empleando las herramientas y la tecnología apropiada.

### 3 PLANTEAMIENTO DEL PROBLEMA

Muchas veces se mal interpreta el manifiesto ágil referente al valor "software que funciona sobre documentación exhaustiva". Esto no significa que se descarte toda documentación, sino, que se logre obtener la documentación adecuada y necesaria que sirva de ayuda tanto para los miembros del equipo scrum como para los stakeholders<sup>2</sup>.

El Lenguaje de Modelado Unificado a priori podría significar mucha burocracia y una carga innecesaria a la documentación del producto cuando se está utilizando una metodología ágil como SCRUM. Sin embargo, esto no es siempre cierto si se lo maneja de manera adecuada modelando solo los diagramas necesarios que ayuden a resolver el problema, ya que ofrece una notación visual que podría ser muy bien aprovechada.

Por este motivo se plantea el siguiente problema:

- ¿Se puede desarrollar un sistema de ventas aplicando UML como notación de modelado y SCRUM como metodología de desarrollo ágil?

### 4 METODOLOGÍA DE INVESTIGACIÓN

Se utilizan los siguientes métodos de investigación para la elaboración del presente trabajo.

#### 4.1 Método bibliográfico

Se realiza la compilación y la lectura de libros relacionados con ingeniería de software, metodologías de desarrollo de software, metodologías ágiles, análisis y diseño de sistemas, programación y base de datos.

#### 4.2 Método analítico

Se revisa y analiza ordenadamente documentos relacionados con el tema de estudio para la realización el presente trabajo.

#### 4.3 Método empírico o de campo

Se utilizan ejemplos de aplicaciones prácticas y experiencias específicas en el desarrollo de sistemas de información.

### 5 JUSTIFICACIÓN

Uno de los valores del manifiesto ágil indica: "Software que funciona sobre documentación exhaustiva". Este manifiesto no indica que se deba prescindir de toda documentación, más bien, se debe elaborar la documentación mínima y necesaria para el desarrollo y el mantenimiento del sistema.

SCRUM basa el desarrollo del proyecto y sus iteraciones en historias de usuario que son narraciones textuales de las características que debe tener el sistema. Las historias de usuario, al ser independientes y pequeñas, a veces se vuelven complejas de comprender cuando se trata de ver el funcionamiento o el objetivo global de la iteración o del sistema; y cómo estas características trabajan entre sí. En este contexto UML, al ser una notación gráfica de modelado, mostraría de manera clara el funcionamiento global del sistema, y ayudaría a comprender mejor los objetivos de la iteración.

---

<sup>2</sup> Stakeholder hace referencia a todas las personas que de alguna forma están involucradas en el desarrollo del sistema, como los clientes, el gerente de la empresa, los contadores, etc. pero que no necesariamente forman parte del equipo Scrum

## 6 APORTES

El empleo de una metodología ágil como SCRUM trabajando en conjunto con una notación de modelado visual como UML, servirán de referencia bibliográfica a los estudiantes o profesionales de las carreras de ingeniería de sistemas o ingeniería informática que cursen el diplomado y quieran elaborar trabajos sobre el módulo de gestión de procesos de desarrollo software.

## 7 ALCANCES Y LIMITACIONES

Para poner en práctica la idea que se quiere mostrar en la presente monografía, se desarrolla un sistema de ventas para restaurants, tiendas o franquicias se dediquen a vender comidas y bebidas.

Se utiliza SCRUM como metodología de desarrollo ágil y UML como notación de modelado visual. Se propone un sistema en plataforma web que se desarrolla bajo el patrón de arquitectura Modelo, Vista y Controlador (MVC), Laravel 5.9 como framework de desarrollo, PHP como lenguaje de programación backend, Bootstrap 4, Java script y VueJS para el desarrollo frontend, y PostgreSQL como gestor de base de datos.

### a) Alcances:

- Administración de usuarios, roles y permisos
- Gestión de productos (comidas y bebidas)
- Gestión de clientes
- Registro de venta de productos
- Anulación y restablecimiento de las ventas
- Generación de un recibo o nota de venta.
- Generación de informes económicos y reportes estadísticos

### b) Limitaciones:

- Para mantener la simplicidad del proyecto, no se implementa un módulo de facturación computarizada.
- La monografía está enfocada en el análisis, diseño y desarrollo del sistema. Por esta razón no se documenta las pruebas realizadas ni la puesta en producción.

## 8 MARCO TEÓRICO

### 8.1 Manifiesto ágil

Como consecuencia de una reunión donde se acuñó el término "Metodologías Ágiles" (febrero 2001), se establecieron los valores de estas metodologías agrupándoles en cuatro postulados; quedando esta agrupación denominada como Manifiesto Ágil. A continuación, se mencionan los cuatro postulados de este manifiesto:

- Valorar más a los individuos y su interacción que a los procesos y las herramientas.
- Valorar más el software que funciona que la documentación exhaustiva.
- Valorar más la colaboración con el cliente que la negociación contractual.
- Valorar la respuesta al cambio que el seguimiento de un plan.

Esto quiere decir que se valora más lo primero, sin dejar de lado lo segundo

### 8.2 Scrum

Scrum es un marco de trabajo para la gestión y desarrollo de productos complejos, en un proceso iterativo e incremental utilizado comúnmente en entornos donde existe gran incertidumbre.

Scrum está basado en un pequeño conjunto de valores, principios, y prácticas; que se basan en los principios de inspección y adaptación, enfocado más en el aspecto humano y el trabajo en equipo donde se involucra totalmente al cliente en todo el proceso.



Ilustración 2: Scrum framework  
Fuente: (Rubin, 2013)

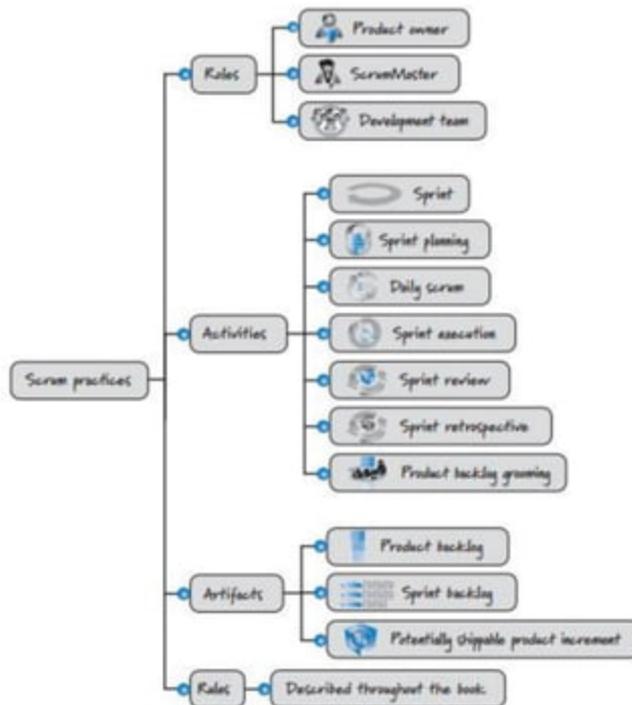


Ilustración 3: Prácticas de SCRUM  
Fuente: (Rubin, 2013)

### 8.3 Historias de usuario

Es una narración que describe una funcionalidad del sistema que tiene valor para un usuario. Las historias de usuario deben ser independientes, negociables, evaluables, estimables, pequeñas, y se las deben probar.

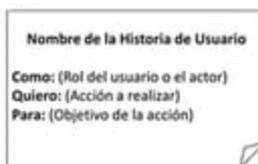


Ilustración 4: Modelo de historia de usuario  
Fuente: Elaboración propia

### 8.4 Lenguaje de Modelado Unificado

UML es un lenguaje de propósito general para visualizar, especificar, construir y documentar los artefactos de un sistema que involucre una gran cantidad de software.

UML es como una caja de herramientas, no es necesario utilizar todos los diagramas, más bien solo los necesarios que te ayudan a resolver el problema.

UML es una notación de modelado para sistemas orientados a objetos, independiente de la metodología o del proceso de desarrollo.

### a) Diagramas UML

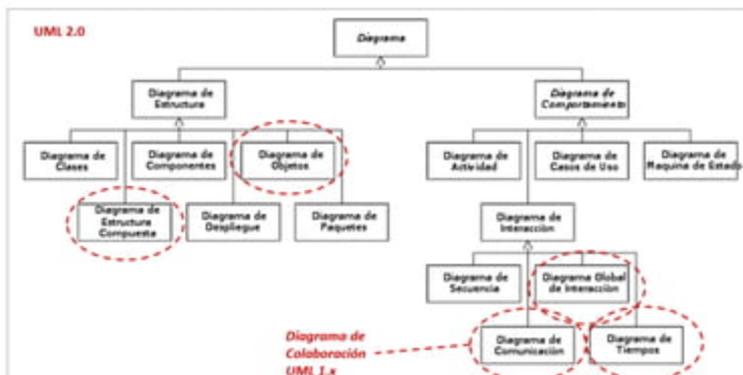


Ilustración 5: Diagramas UML

Fuente: (Rumbaugh, Jacobson, & Booch, El Lenguaje de Modelado Unificado 2.0, 2007)

## 8.5 Product Vision Board

El Product Vision Board (PVB) es una herramienta visual, plasmada en un canvas, creada por Román Pitcher que permite presentar la visión de forma que su entendimiento sea más fluido y además respondiendo preguntas claves que le servirán al equipo para mantener y utilizar la visión.

Codename	Frase representativa		
Grupo de Usuarios	Necesidades	Producto	Valor

Ilustración 6: Plantilla Product Vision Board

Fuente: <https://medium.com/scrumorganico>

## 8.6 Trello

Trello es un software de administración de proyectos con interfaz web y con cliente para iOS y Android para organizar proyectos.

Empleando el sistema Kanban para el registro de actividades con tarjetas virtuales organizar tareas, permite agregar listas, adjuntar archivos, etiquetar eventos, agregar comentarios, y compartir tableros.

## 8.7 Git

Git es un software de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

## 8.8 Heroku

Heroku es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación.

Heroku es una de las primeras plataformas de computación en la nube, que fue desarrollada desde junio de 2007, con el objetivo de soportar solamente el lenguaje de programación Ruby, pero posteriormente se ha extendido el soporte a Java, Node.js, Scala, Python, y PHP. La base del sistema operativo es Debian.

## 8.9 Puntos de historia

El método de los puntos de historia es la forma de realizar estimaciones de esfuerzo y tiempo a las historias de usuario. También nos sirve para medir la velocidad del sprint en la reunión de retrospectiva.

Se escoge una historia de usuario sencilla, una que todo el mundo entiende, para emplearla como referencia. Esa sería la definición de 1 punto de historia en el proyecto. Para estimar las otras historias de usuario, bastará con compararlo con la historia de referencia.

Se emplea la actividad de planning poker para determinar el valor de los puntos y el tiempo de las historias de usuario, según el siguiente cuadro.

SIZE	POINTS	TIME [Days]
XtraSmall (XS)	1	½
Small (S)	2	1
Medium (M)	3	2
Large (L)	5	3
XtraLarge (XL)	8	5

Tabla 1: Puntos historia

## 9 MARCO PRÁCTICO

### 9.1 Modelo del negocio

El sistema de información propuesto según los alcances y limitaciones [Pág. 5], está orientado para cualquier negocio o empresa cuyo rubro principal consiste en la venta de comidas y bebidas; éstas pueden ser restaurant, pensiones, fondas, franquicias de comida rápida, etc.

Como el proyecto no se enfoca en un negocio o empresa en particular, se hace uso del product vision board para comprender mejor la visión del producto que se va a desarrollar.

#### 9.1.1 Product Vision Board

<b>ctrlFOOD</b>	Administración y control económico en la venta de alimentos.		
<b>Usuarios</b>	<b>Necesidades</b>	<b>Producto</b>	<b>Valor</b>
<ul style="list-style-type: none"><li>• Restaurants</li><li>• Pensiones</li><li>• Fondas</li><li>• Pollerías</li><li>• Franquicias de comida rápida</li></ul>	<ul style="list-style-type: none"><li>• Controlar las ventas que se realizan</li><li>• Realizar Informes económicos diarios y mensuales</li><li>• Realizar reportes estadísticos para publicidad o descuentos</li></ul>	<ul style="list-style-type: none"><li>• Sistema de información web</li><li>• Administración de usuarios y permisos</li><li>• Gestión de productos</li><li>• Gestión de clientes</li><li>• Administración de las ventas</li><li>• Generación de recibos</li><li>• Generación de informes económicos</li><li>• Generación de reportes estadísticos</li></ul>	<ul style="list-style-type: none"><li>• Control económico fiable</li><li>• Reportes estadísticos confiables</li><li>• Publicidad</li><li>• Descuentos a clientes potenciales</li></ul>

Tabla 2: Product Vision Board del proyecto

### 9.2 Roles y stakeholders

Con fines académicos, los roles de Scrum están representados por una sola persona.

- **Product Owner:** Saul Mamani M.
- **Scrum Master:** Saul Mamani M.
- **Development Team:** Saul Mamani M.
- **Client:** Restaurants, Pensiones, Pollerías, Fondas, Franquicias de comida rápida, etc.

### 9.3 Historias de usuario

Para la determinación de requerimientos y/o características del sistema se recolectan historias de usuario.

<b>HU1: Ingresar al sistema como administrador</b>	
<b>Como</b>	Administrador
<b>Quiero</b>	Ingresar al sistema con una cuenta y un password
<b>Para</b>	Tener el acceso a toda la funcionalidad del sistema, incluyendo la funcionalidad de los vendedores

*Tabla 3: Historia de usuario – Ingresar al sistema como administrador*

<b>HU2: Ingresar al sistema como vendedor</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Ingresar al sistema con una cuenta y un password
<b>Para</b>	Vender productos y generar informes económicos

*Tabla 4: Historia de usuario – Ingresar al sistema como vendedor*

<b>HU3: Gestionar usuarios del sistema</b>	
<b>Como</b>	Administrador
<b>Quiero</b>	Listar, insertar, modificar y eliminar datos de los vendedores
<b>Para</b>	Que utilicen el sistema según su rol seleccionado

*Tabla 5: Historia de usuario – Gestionar usuarios del sistema*

<b>HU4: Administrar perfil de usuario</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Ver y actualizar mis datos personales y mi foto de perfil
<b>Para</b>	Personalizar mi cuenta y cambiar la foto de perfil

*Tabla 6: Historia de usuario – Administrar perfil de usuario*

<b>HU5: Cambiar contraseña</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Cambiar mi contraseña asignada
<b>Para</b>	Mantener seguro mi sesión y mis acciones en el sistema

*Tabla 7: Historia de usuario – Cambiar contraseña*

<b>HU6: Gestionar productos</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Listar, buscar, insertar, modificar y eliminar productos categorizados en comidas y bebidas
<b>Para</b>	Tener un catálogo completo de las comidas y bebidas al momento de registrar las ventas solicitadas

*Tabla 8: Historia de usuario – Gestionar productos*

<b>HU7: Gestionar clientes</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Listar, buscar, insertar, modificar y eliminar datos de los clientes
<b>Para</b>	Registrar los datos del cliente al realizar una venta, llevar un adecuado control de los clientes y generar reportes estadísticos

*Tabla 9: Historia de usuario – Gestionar clientes*

<b>HU8: Listar ventas</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Listar todas las ventas realizadas por rango de fechas, mostrando el total vendido
<b>Para</b>	Llevar un control económico de todo lo que he vendido. Además de, re imprimir los recibos, anular y/o restablecer ventas erradas

*Tabla 10: Historia de usuario – Listar ventas*

<b>HU9: Registrar ventas</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Registrar en el sistema la venta de uno o varios productos del catálogo de comidas o bebidas, y buscar o registrar los datos del cliente
<b>Para</b>	Llevar un control económico adecuado de todas las ventas realizadas

*Tabla 11: Historia de usuario – Registrar ventas*

<b>HU10: Generar número de ticket</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Que el sistema genere un número secuencial de ticket de forma diaria
<b>Para</b>	Llamar al cliente y que recoja su pedido

*Tabla 12: Historia de usuario – Generar número de ticket*

<b>HU11: Generar recibo</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Que el sistema genere un recibo con los datos de la venta realizada, la información del cliente, y un código QR
<b>Para</b>	Tener constancia de que la transacción se ha realizado de manera correcta

*Tabla 13: Historia de usuario – Generar recibo*

<b>HU12: Anular ventas</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Anular las ventas que yo he registrado y que tengan datos erróneos. Las ventas se pueden anular solo el día en el que se realizó la transacción y antes de generar el informe económico
<b>Para</b>	Que no sean tomados en cuenta a la hora de generar el informe económico diario

*Tabla 14: Historia de usuario – Anular ventas*

<b>HU13: Restablecer ventas</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Restablecer una venta que yo he anulado previamente por error. Las ventas se pueden restablecer solo el día en que se realizó la transacción y antes de generar el informe económico diario
<b>Para</b>	Que vuelva a ser tomada en cuenta a la hora de generar el informe económico diario

*Tabla 15: Historia de usuario – Restablecer ventas*

<b>HU14: Generar informe económico</b>	
<b>Como</b>	Vendedor
<b>Quiero</b>	Quiero que el sistema genere informes económicos diarios, mensuales y anuales de todas las ventas realizadas
<b>Para</b>	Llevar un control adecuado de todas las transacciones que se han realizado en un determinado tiempo

Tabla 16: Historia de usuario – Generar informe económico

<b>HU15: Generar reportes estadísticos</b>	
<b>Como</b>	Administrador
<b>Quiero</b>	Que el sistema genere reportes estadísticos económicos y reportes estadísticos de los clientes potenciales
<b>Para</b>	Tener una adecuada toma de decisiones, como lanzar y publicar promociones y descuentos especiales

Tabla 17: Historia de usuario – Generar reportes estadísticos

#### 9.4 Product backlog

La pila del producto pendientes de desarrollar está constituida por las historias de usuario, y ordenamos según prioridad de implementación.

Historia de usuario	Descripción	Prioridad
HU1	Ingresar al Sistema como administrador	1
HU2	Ingresar al sistema como vendedor	1
HU3	Gestionar usuarios del sistema	1
HU4	Administrar perfil de usuario	1
HU5	Cambiar contraseña	1
HU6	Gestionar productos	1
HU7	Gestionar clientes	1
HU8	Listar ventas	2
HU9	Registrar ventas	2
HU10	Generar número de ticket	2
HU11	Generar recibo	2
HU12	Anular ventas	3

<b>HU13</b>	Restablecer ventas	3
<b>HU14</b>	Generar informe económico	3
<b>HU15</b>	Generar reportes estadísticos	3

Tabla 18: Product backlog

### 9.5 Estimación del product backlog

Para estimar el tiempo y el esfuerzo de las historias de usuario se emplea el método de los puntos de historia, y planning poker para asignar valores a cada historia de acuerdo a la Tabla 1.

<b>Sprint</b>	<b>HU</b>	<b>Descripción</b>	<b>Size</b>	<b>Puntos de historia</b>	<b>Tiempo [Días]</b>
<b>Primer Sprint</b>	<b>HU1</b>	Ingresar al Sistema como administrador	XS	1	½
	<b>HU2</b>	Ingresar al sistema como vendedor	XS	1	½
	<b>HU3</b>	Gestionar usuarios del sistema	S	2	1
	<b>HU4</b>	Administrar perfil	XS	1	½
	<b>HU5</b>	Cambiar contraseña	XS	1	½
	<b>HU6</b>	Gestionar productos	S	2	1
	<b>HU7</b>	Gestionar clientes	S	2	1
<b>Segundo Sprint</b>	<b>HU8</b>	Listar ventas	S	2	1
	<b>HU9</b>	Registrar ventas	L	5	3
	<b>HU10</b>	Generar número de ticket	XS	1	½
	<b>HU11</b>	Generar recibo	XS	1	½
<b>Tercer Sprint</b>	<b>HU12</b>	Anular ventas	XS	1	½
	<b>HU13</b>	Restablecer ventas	XS	1	½
	<b>HU14</b>	Generar informe económico	M	3	2
	<b>HU15</b>	Generar reportes estadísticos	M	3	2
<b>Puntos de historia / Tiempo estimado (Time Boxing)</b>				<b>27</b>	<b>15</b>

Tabla 19: Product backlog estimado

De acuerdo al tamaño y a la cantidad de historias de usuario, el tiempo estimado para el desarrollo del sistema es de quince días, las cuales la dividimos en tres iteraciones o sprints de 5 días cada una.

## 9.6 Definición de hecho

Se establece la "definición de hecho" cuando todas las tareas definidas en el sprint backlog de una iteración se hayan completado, se han verificado los criterios de aceptación, y se ha desplegado el product increment en un servidor de pruebas como Heroku.

## 9.7 Definición de la arquitectura

Para modelar y analizar la arquitectura del sistema se utiliza el diagrama de componentes y el diagrama de despliegue del Lenguaje de Modelado Unificado.

### 9.7.1 Diagrama de componentes

El diagrama de componentes muestra la arquitectura lógica del sistema software y los componentes necesarios para construir la aplicación web.

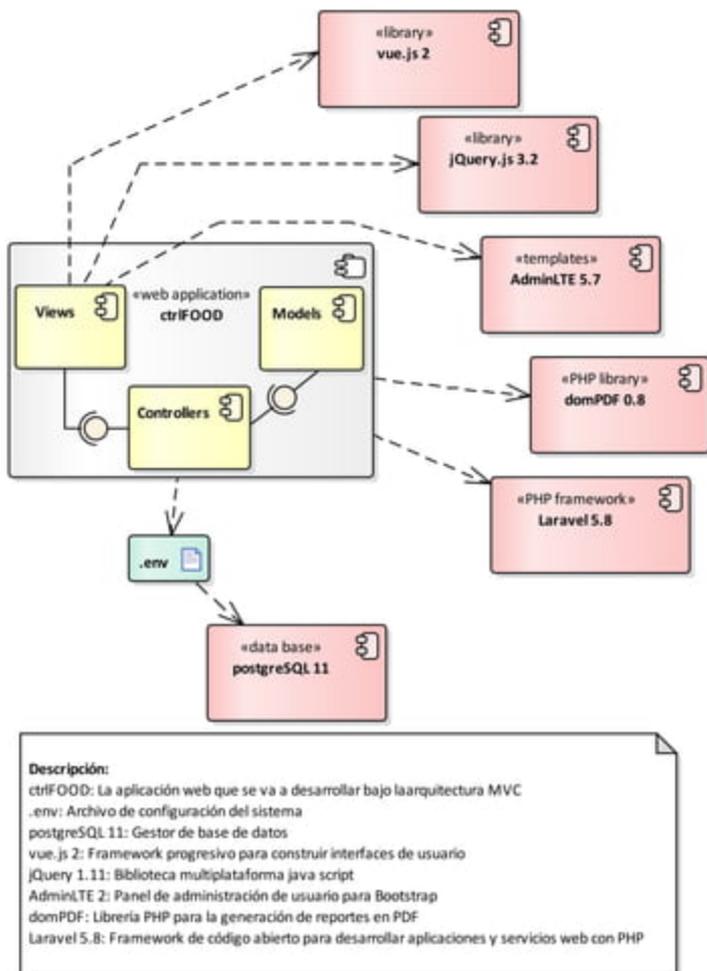


Ilustración 7: Diagrama de componentes  
Fuente: Elaboración propia

## 9.8 Primer sprint

### 9.8.1 Sprint planning

El sprint planning muestra **qué**<sup>3</sup> es lo que se tiene que desarrollar en esta iteración, y los tiempos comprometidos para finalizar cada historia de usuario.

HU	Descripción	Puntos de historia	Tiempo [Días]	Inicio	Finalización
HU1	Ingresar al Sistema como administrador	1	½	02/09/2019	02/09/2019
HU2	Ingresar al sistema como vendedor	1	½	02/09/2019	02/09/2019
HU3	Gestionar usuarios del sistema	2	1	03/09/2019	03/09/2019
HU4	Administrar perfil	1	½	04/09/2019	04/09/2019
HU5	Cambiar contraseña	1	½	04/09/2019	04/09/2019
HU6	Gestionar productos	2	1	05/09/2019	05/09/2019
HU7	Gestionar clientes	2	1	06/09/2019	06/09/2019
<b>Puntos de historia/Tiempo estimado (Time Boxing)</b>		<b>10</b>	<b>5</b>		

Tabla 20: Primer sprint planning

Para completar este sprint se estima un esfuerzo de 10 puntos de historia y tiempo de desarrollo de cinco días.

Para comprender mejor la funcionalidad global del sprint se modela un diagrama de casos de uso, y para modelar el esquema lógico de la base de datos de utiliza un diagrama de clases.

### 9.8.2 Diagrama de casos de uso

El diagrama de casos de uso muestra la funcionalidad global del sprint.

---

<sup>3</sup> La palabra **qué** es lo que se tiene que hacer, hace referencia al **análisis** de sistemas dentro el ciclo de vida un proyecto de desarrollo de software.

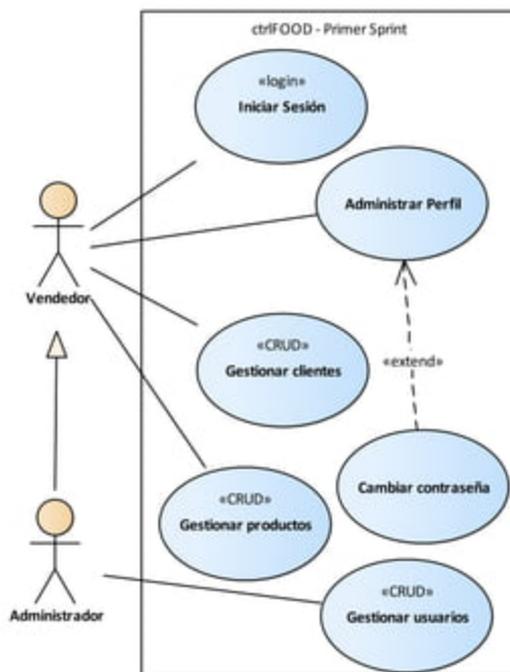


Ilustración 8: Diagrama de casos de uso primer sprint  
Fuente: Elaboración propia

### 9.8.3 Diagrama de clases persistentes

Se utiliza el diagrama de clases para modelar el esquema lógico de la base de datos.

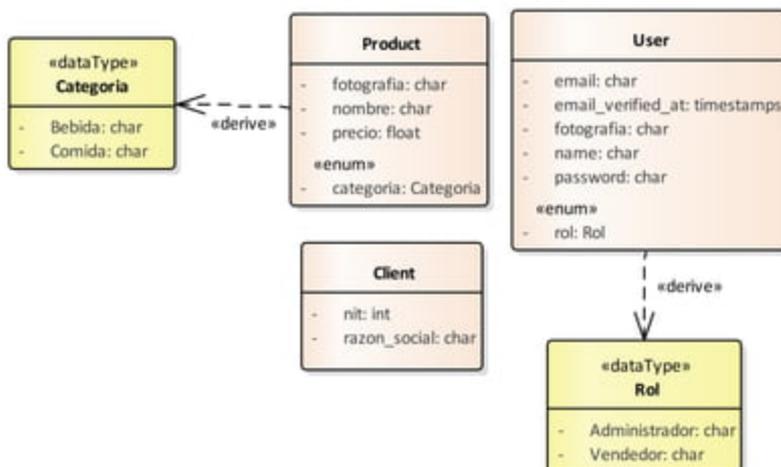


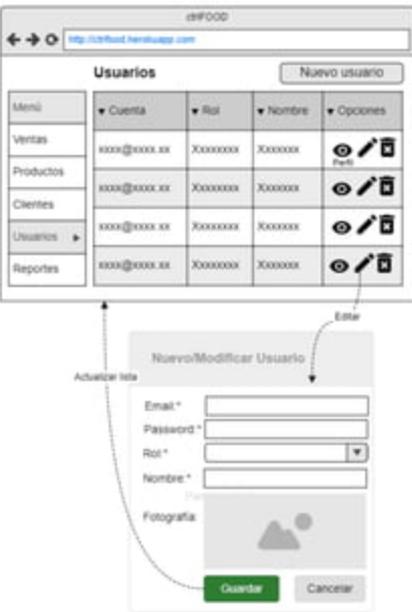
Ilustración 9: Diagrama de clases primer sprint  
Fuente: Elaboración propia

### 9.8.4 Especificación de casos de uso

Para la especificación de los casos de uso utilizamos historias de usuario detallados.

<p><b>Caso de uso:</b> Iniciar Sesión</p> 	<p><b>Historia de usuario:</b> HU1, HU2</p>	
	<p><b>Actor:</b> Vendedor</p>	<p><b>Depende de:</b> HU3</p>
	<p><b>Descripción:</b> Ingresar al sistema con una cuenta y un password</p>	
	<p><b>Criterio de Aceptación:</b></p> <ul style="list-style-type: none"> <li>- El actor ingresa su email y una contraseña y presiona el botón Aceptar.</li> <li>- El sistema busca las credenciales en la base de datos.</li> <li>- Una vez verificada las credenciales, el sistema abre la pantalla principal de acuerdo al rol del actor. Si es vendedor tiene acceso solo a la funcionalidad mostrada en diagrama de casos de uso, y si es administrador tiene acceso a toda la funcionalidad del sistema.</li> </ul>	
<p><b>Elaborado por:</b> Saul Mamani</p>	<p><b>Responsable:</b> Saul Mamani</p>	<p><b>Versión:</b> 1.0</p>

Tabla 21: Caso de uso iniciar sesión

<p><b>Caso de uso:</b> Gestionar usuarios</p> 	<p><b>Historia de usuario:</b> HU3</p>	
	<p><b>Actor:</b> Administrador</p>	<p><b>Depende de:</b></p>
	<p><b>Descripción:</b> Crear, modificar, eliminar y listar usuarios del sistema</p>	
	<p><b>Criterio de Aceptación:</b></p> <ul style="list-style-type: none"> <li>- El sistema muestra la lista de todos los usuarios registrados y un menú con las siguientes opciones:</li> </ul> <p><b>Nuevo</b></p> <ul style="list-style-type: none"> <li>- El actor presiona el botón Nuevo</li> <li>- El sistema abre un formulario con los datos del usuario en base a la clase "User" del diagrama de clases</li> <li>- El actor llena los datos y presiona el botón Guardar</li> <li>- El sistema inserta el registro en la base de datos y lanza un mensaje</li> </ul> <p><b>Modificar</b></p> <ul style="list-style-type: none"> <li>- El actor selecciona un registro presiona el botón Modificar</li> <li>- El sistema abre un formulario con los datos del usuario seleccionado</li> </ul>	

	<ul style="list-style-type: none"> <li>- El actor modifica uno o más campos y presiona el botón Guardar</li> <li>- El sistema modifica los datos y lanza un mensaje</li> </ul> <p><b>Eliminar</b></p> <ul style="list-style-type: none"> <li>- El actor selecciona un registro presiona el botón Eliminar</li> <li>- El sistema muestra un mensaje de confirmación</li> <li>- El actor confirma el mensaje</li> <li>- El sistema elimina el registro de la base de datos</li> </ul>	
<b>Elaborado por:</b> Saul Mamani	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0

Tabla 22: Caso de uso gestionar usuarios

<b>Caso de uso:</b> Gestionar clientes	<b>Historia de usuario:</b> HU7
	<b>Actor:</b> Vendedor <b>Depende de:</b>
	<b>Descripción:</b> Crear, modificar, eliminar y listar clientes
	<b>Criterio de Aceptación:</b>
	<ul style="list-style-type: none"> <li>- El sistema muestra la lista de todos los clientes registrados y un menú con las siguientes opciones:</li> </ul> <p><b>Nuevo</b></p> <ul style="list-style-type: none"> <li>- El actor presiona el botón Nuevo</li> <li>- El sistema abre un formulario con los datos del cliente en base a la clase "Client" del diagrama de clases</li> <li>- El actor llena los datos y presion el botón Guardar</li> <li>- El sistema inserta el registro en la base de datos y lanza un mensaje</li> </ul> <p><b>Modificar</b></p> <ul style="list-style-type: none"> <li>- El actor selecciona un registro presiona el botón Modificar</li> <li>- El sistema abre un formulario con los datos del cliente seleccionado</li> <li>- El actor modifica uno o más campos y presiona el botón Guardar</li> <li>- El sistema modifica los datos y lanza un mensaje</li> </ul>

	<p><b>Eliminar</b></p> <ul style="list-style-type: none"> <li>– El actor selecciona un registro presiona el botón Eliminar</li> <li>– El sistema muestra un mensaje de confirmación</li> <li>– El actor confirma el mensaje</li> <li>– El sistema elimina el registro de la base de datos</li> </ul>	
<p><b>Elaborado por:</b> Saul Mamani</p>	<p><b>Responsable:</b> Saul Mamani</p>	<p><b>Versión:</b> 1.0</p>

Tabla 23: Caso de uso gestionar clientes

<p><b>Caso de uso:</b> Gestionar productos</p>	<p><b>Historia de usuario:</b> HU6</p>
	<p><b>Actor:</b> Vendedor   <b>Depende de:</b></p>
	<p><b>Descripción:</b> Crear, modificar, eliminar y listar producto</p>
	<p><b>Criterio de Aceptación:</b></p> <ul style="list-style-type: none"> <li>– El sistema muestra la lista de todos los productos registrados y un menú con las siguientes opciones:</li> </ul>
	<p><b>Nuevo</b></p> <ul style="list-style-type: none"> <li>– El actor presiona el botón Nuevo</li> <li>– El sistema abre un formulario con los datos del producto en base a la clase "Product" del diagrama de clases</li> <li>– El actor llena los datos, carga una imagen no más de 2MB de tamaño y presión el botón Guardar</li> <li>– El sistema inserta el registro en la base de datos y lanza un mensaje</li> </ul>
	<p><b>Modificar</b></p> <ul style="list-style-type: none"> <li>– El actor selecciona un registro presiona el botón Modificar</li> <li>– El sistema abre un formulario con los datos del producto seleccionado</li> <li>– El actor modifica uno o más campos y presiona el botón Guardar</li> <li>– El sistema modifica los datos y lanza un mensaje</li> </ul>
	<p><b>Eliminar</b></p> <ul style="list-style-type: none"> <li>– El actor selecciona un registro presiona el botón Eliminar</li> <li>– El sistema muestra un mensaje de confirmación</li> </ul>



	<ul style="list-style-type: none"> <li>- El actor confirma el mensaje</li> <li>- El sistema elimina el registro de la base de datos</li> </ul>	
<b>Elaborado por:</b> Saul Mamani	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0

Tabla 24: Caso de uso gestionar productos

<b>Caso de uso:</b> Administrar perfil	<b>Historia de usuario:</b> HU4	
	<b>Actor:</b> Vendedor	<b>Depende de:</b> HU3
	<b>Descripción:</b> Administrar datos del usuario logeado	
	<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- Al actor presiona el botón Perfil de usuario de la pantalla principal</li> <li>- El sistema llena los datos del usuario logeado</li> <li>- El actor modifica uno o más datos, carga una foto de no más de 2MB, y presiona Aceptar</li> <li>- El sistema actualiza la información del usuario en la base de datos, y refresca el sistema.</li> </ul>	
<b>Elaborado por:</b> Saul Mamani	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0

Tabla 25: Caso de uso Administrar perfil

<b>Caso de uso:</b> Cambiar contraseña	<b>Historia de usuario:</b> HU5	
	<b>Actor:</b> Vendedor	<b>Depende de:</b> HU3, HU4
	<b>Descripción:</b> Cambiar contraseña de un usuario logado	
	<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- El sistema muestra la pantalla para cambiar contraseña</li> <li>- El actor llena la contraseña actual, la nueva contraseña y la verificación de la contraseña</li> <li>- El sistema verifica las credenciales en la base de datos, actualiza la contraseña, y lanza un mensaje de éxito. <ul style="list-style-type: none"> <li>o Las contraseñas deben estar protegidas por un algoritmo de cifrado.</li> </ul> </li> </ul>	
<b>Elaborado por:</b> Saul Mamani	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0

Tabla 26: Caso de uso Cambiar contraseña

### 9.8.5 Sprint backlog

Para determinar **cómo**<sup>4</sup> se va a desarrollar el sprint, se identifican una o varias tareas por historia de usuario, las cuales son agrupadas en un sprint backlog y asignadas a los miembros del equipo de desarrollo.

HU	Descripción	Tareas
HU1	Ingresar al Sistema como administrador	T1. Diseñar las migraciones en Laravel T2. Desarrollar el Login.
HU2	Ingresar al sistema como vendedor	T3. Diseñar el IU del sistema T4. Controlar los permisos en el menú
HU3	Gestionar usuarios del sistema	T5. Desarrollar el CRUD de usuarios
HU4	Administrar perfil	T6. Diseñar la pantalla de perfil de usuario
HU5	Cambiar contraseña	T7. Desarrollar el cambio de contraseña
HU6	Gestionar productos	T8. Desarrollar el CRUD de productos
HU7	Gestionar clientes	T9. Desarrollar el CRUD de clientes

Tabla 27: Primer sprint backlog

### 9.8.6 Sprint execution

Para controlar la ejecución del sprint y el avance de las tareas, empleamos un tablero Kanban con la herramienta Trello.

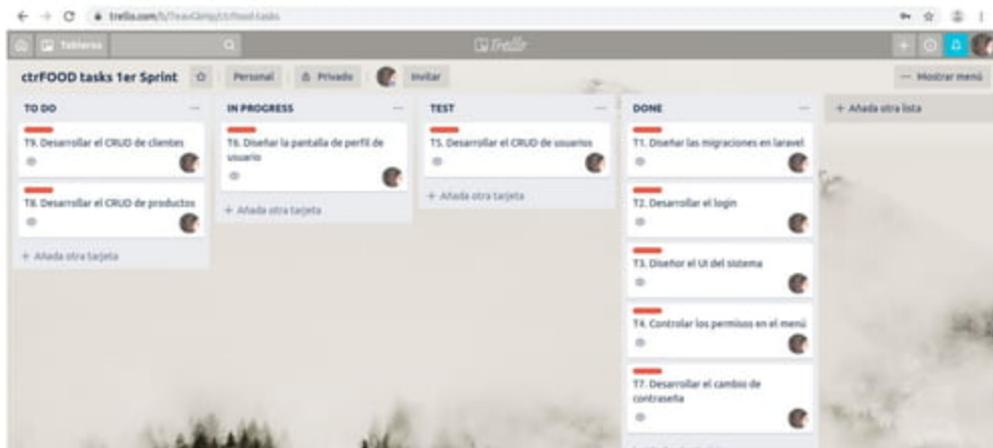


Ilustración 10: Tablero Kanban primer sprint  
Fuente: Elaboración propia

<sup>4</sup> La palabra **cómo** se va a desarrollar, hace referencia al **diseño** de sistemas dentro el ciclo de vida un proyecto de desarrollo de software.

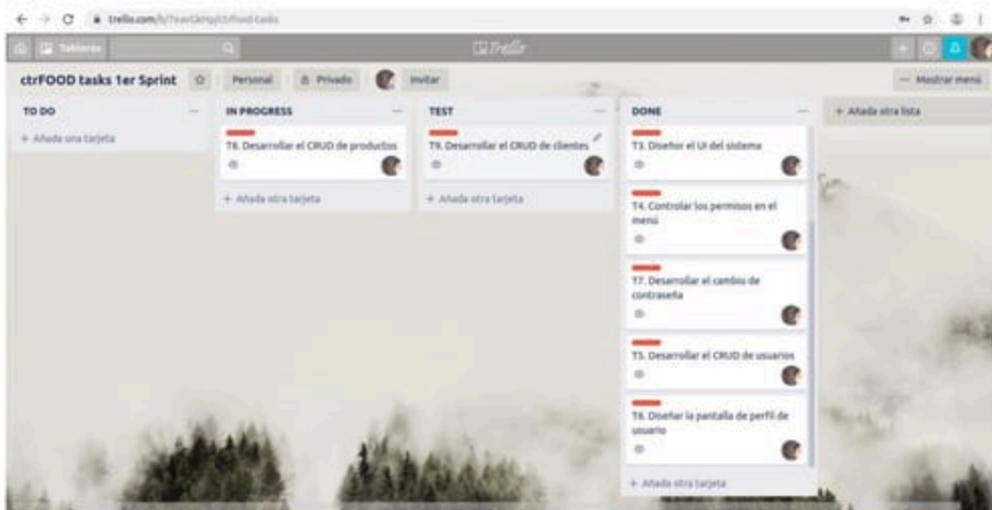


Ilustración 11: Tablero Kanban primer sprint  
Fuente: Elaboración propia

Para controlar el avance del código fuente se utiliza el sistema de control de versiones GIT, el cual se encuentra en GitHub bajo en el siguiente repositorio: <http://github.com/saulmamani/ctrlFood>

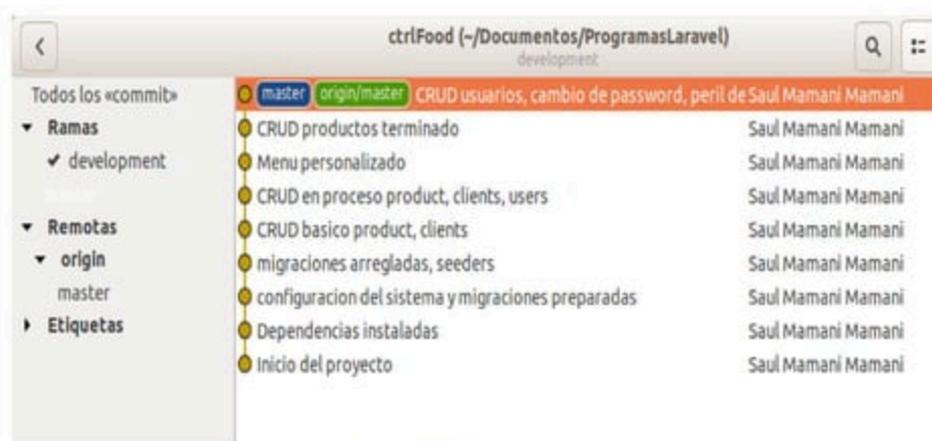


Ilustración 12: Control de versiones del primer sprint  
Fuente: Elaboración propia

El modelo relacional de la base de datos que se ha implementado en base al diagrama de clases persistentes.



Ilustración 13: Modelo relacional de la base de datos del primer sprint  
Fuente: Elaboración propia

### 9.8.7 Sprint review

El **product increment** representa los entregables realizados al product owner y al cliente, estos entregables reflejan el cumplimiento del primer sprint.

Se ha desplegado el sistema en la siguiente dirección URL, a modo de prueba para su revisión: <https://ctrlfood.herokuapp.com>



Ilustración 14: Pantalla Login  
Fuente: Elaboración propia

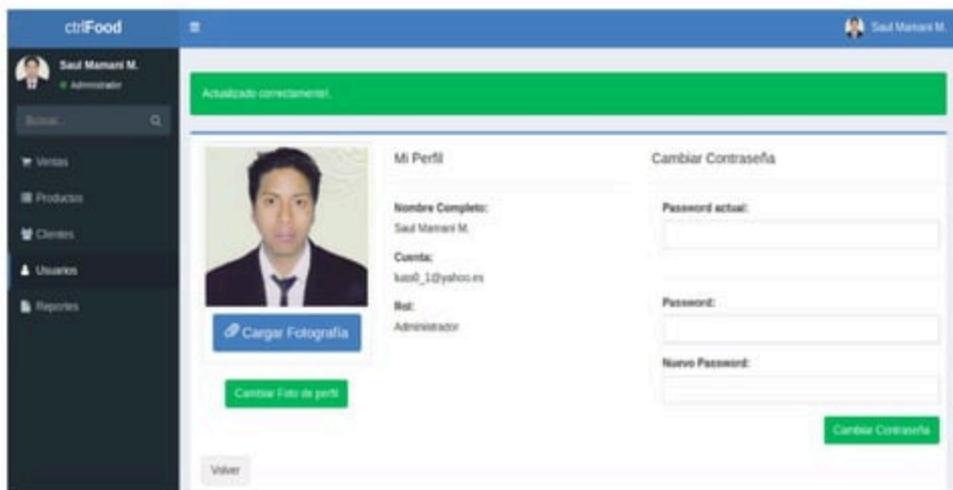


Ilustración 15: Pantalla administrar perfil de usuario  
Fuente: Elaboración propia

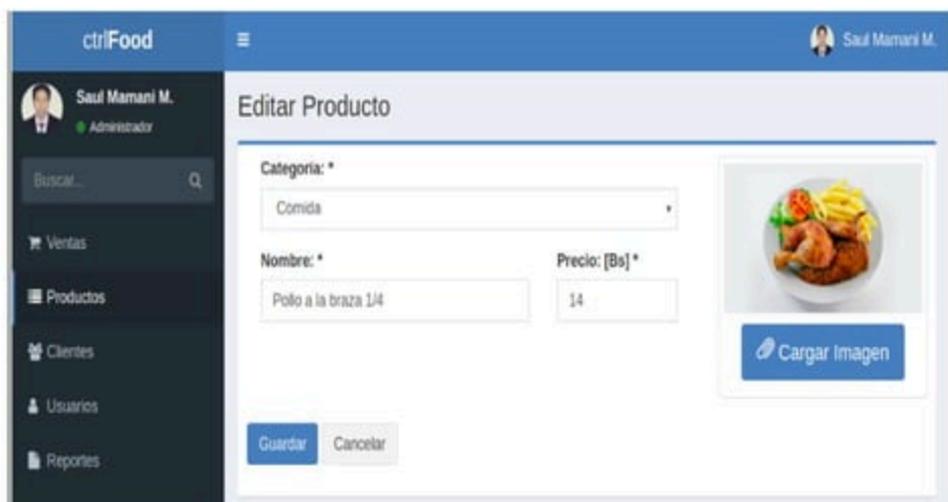


Ilustración 16: Pantalla producto  
Fuente: Elaboración propia

### 9.8.8 Sprint retrospective

El desarrollo del primer sprint se concluyó de manera adecuada cumpliendo las tareas, los tiempos establecidos, y la definición de hecho establecido al inicio del sprint. Esta evolución se puede apreciar en el gráfico burndown.



Ilustración 17: Seguimiento de historias primer sprint  
 Fuente: Elaboración propia

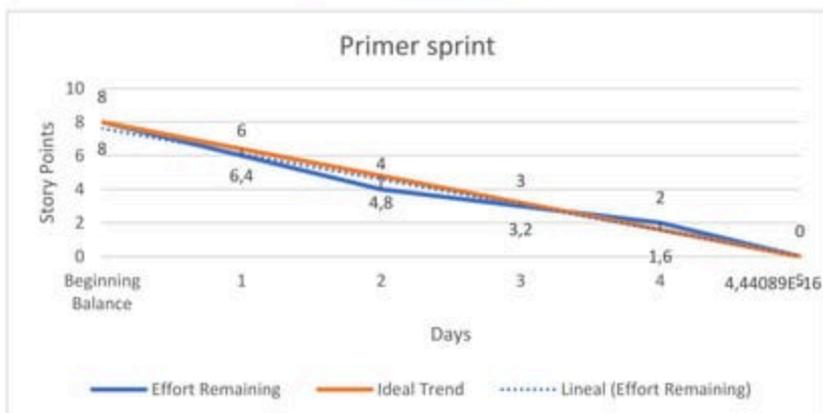


Ilustración 18: Burndown chart primer sprint  
 Fuente: Elaboración propia

## 9.9 Segundo sprint

### 9.9.1 Sprint planning

El sprint planning muestra **qué** es lo que se tiene que desarrollar en esta iteración, y los tiempos comprometidos para finalizar cada historia de usuario.

HU	Descripción	Puntos de historia	Tiempo [Días]	Inicio	Finalización
HU8	Listar ventas	2	1	08/09/2019	09/09/2019
HU9	Registrar ventas	5	3	10/09/2019	12/09/2019
HU10	Generar nro. de ticket	1	½	13/09/2019	13/09/2019
HU11	Generar recibo	1	½	13/09/2019	13/09/2019
<b>Puntos de historia / Tiempo estimado (Time Boxing)</b>		<b>9</b>	<b>5</b>		

Tabla 28: Segundo sprint planning

Para completar este sprint se estima un esfuerzo de 9 puntos de historia y tiempo de desarrollo de cinco días.

### 9.9.2 Diagrama de casos de uso

El diagrama de casos de uso muestra la funcionalidad global del sprint.

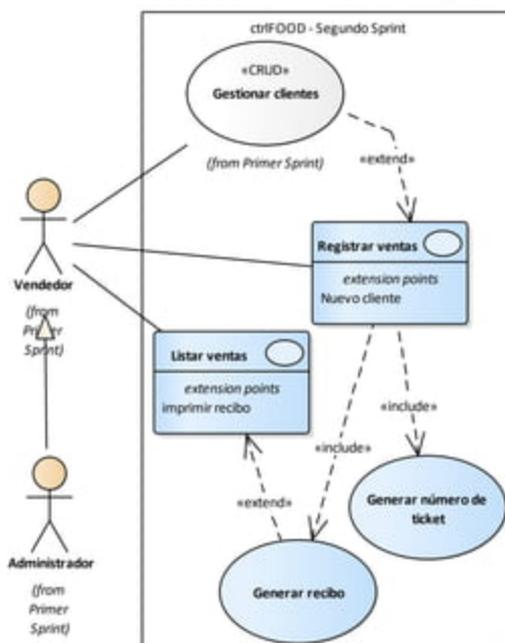


Ilustración 19: Diagrama de casos de uso segundo sprint  
Fuente: Elaboración propia

### 9.9.3 Diagrama de clases persistentes

Se emplea el diagrama de clases para modelar el esquema lógico de la base de datos.

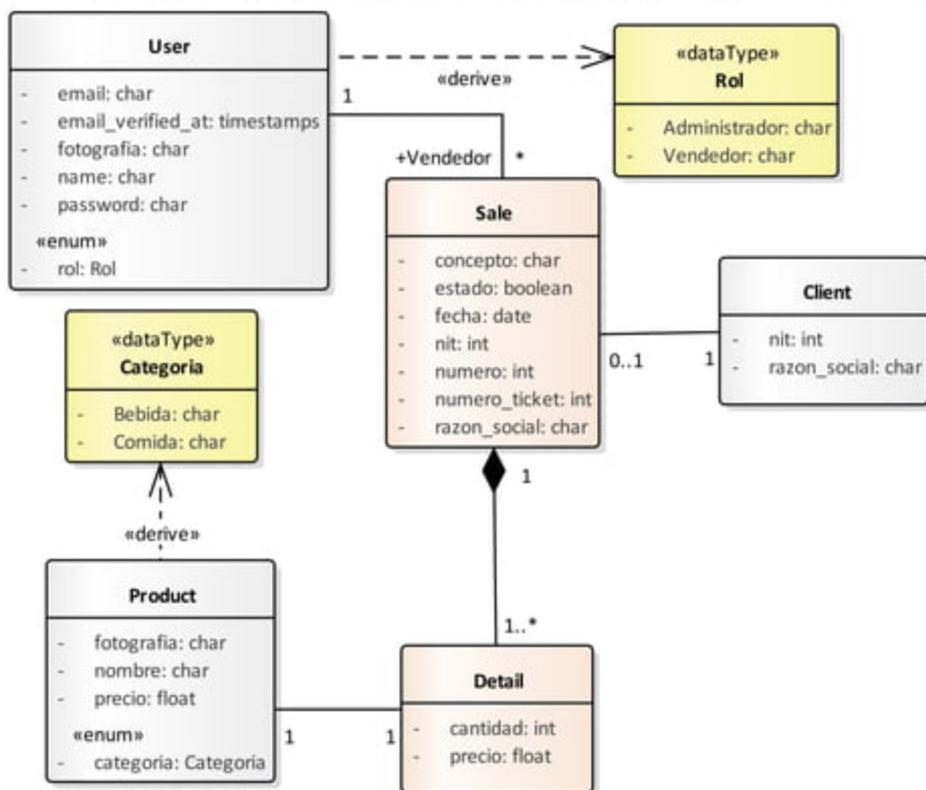


Ilustración 20: Diagrama de clases segundo sprint  
Fuente: Elaboración propia

### 9.9.4 Especificación de casos de uso

Para la especificación de los casos de uso utilizamos historias de usuario detallados.

<p><b>Caso de uso: Listar ventas</b></p>	<p><b>Historia de usuario: HU8</b></p> <p><b>Actor:</b> Vendedor</p> <p><b>Depende de:</b> HU9</p> <p><b>Descripción:</b> Muestra la lista de las ventas realizadas</p> <p><b>Criterio de Aceptación:</b></p> <ul style="list-style-type: none"> <li>El sistema muestra la lista de todas las ventas realizadas y un formulario de búsquedas por rango de fechas, estado, y clientes.</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>- El actor llena los datos del formulario y pulsa el botón Buscar</li> <li>- El sistema actualiza el listado según los parámetros de búsqueda</li> </ul>		
<b>Elaborado por:</b> Saul Mamani	<table border="1"> <tr> <td><b>Responsable:</b> Saul Mamani</td> <td><b>Versión:</b> 1.0</td> </tr> </table>	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0
<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0		

Tabla 29: Caso de uso Listar ventas

<b>Caso de uso:</b> Registrar ventas	<b>Historia de usuario:</b> HU9	
	<b>Actor:</b> Vendedor	<b>Depende de:</b> HU6
	<b>Descripción:</b> Registro de ventas de uno más productos	
	<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- El sistema muestra la pantalla de ventas con el catálogo de los productos habilitados para vender.</li> <li>- El actor selecciona uno o más productos y los añade al carrito de ventas, además puede modificar las cantidades en el carrito.</li> <li>- El sistema calcula el subtotal y el total del monto a cobrar en bolivianos</li> <li>- El actor llena los datos de la venta, en base a las clases "Sale" y "Detail" del diagrama de clases.</li> <li>- Al llenar el campo Nit el sistema busca la razón social asociada a ese Nit, si lo encuentra lo llena en el campo correspondiente; y si no lo encuentra deja en campo vacío.</li> <li>- El actor presiona el botón Finalizar venta.</li> <li>- El sistema <b>genera un número de ticket</b>, inserta la nueva venta en la base de datos y prepara la impresión del recibo.</li> </ul>	
<b>Elaborado por:</b> Saul Mamani	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0

Tabla 30: Caso de uso Registrar ventas

<b>Caso de uso:</b> Generar número de ticket	<b>Historia de usuario:</b> HU10	
<b>Actor:</b> Vendedor	<b>Depende de:</b> HU9	
<b>Descripción:</b> Genera el número de ticket de la transacción		
<b>Criterio de Aceptación:</b>		
<ul style="list-style-type: none"> <li>Al registrar una nueva venta, el sistema genera un número de ticket que se va incrementando de uno en uno, y empieza en 1 al inicio de cada jornada.</li> </ul> <p><i>* esta historia de usuario consiste en un algoritmo y no tiene una evidente interfaz de usuario</i></p>		
<b>Elaborado por:</b> Saul Mamani	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0

Tabla 31: Caso de uso Generar número de ticket

<b>Caso de uso:</b> Generar recibo	<b>Historia de usuario:</b> HU11																					
<div style="border: 1px solid black; padding: 10px; margin: 10px;"> <p style="text-align: center;">SALI Restaurant Plan 500 Av. Heroes del Chaco #232 NIT: 3095304010 TEL/CEL: 76137269</p> <p style="text-align: center;"><b>R E C I B O</b></p> <p>Nro. Recibo: 00      Fecha: 00/00/0000 NIT / CI: 000000 Razón Social: Xxxxxx xxxxxxx xxxxx</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>▼ Cantidad</th> <th>▼ Descripción</th> <th>▼ Precio</th> <th>▼ Importe</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00</td> <td style="text-align: center;">Xxxxxxxx</td> <td style="text-align: center;">00</td> <td style="text-align: center;">00</td> </tr> <tr> <td style="text-align: center;">00</td> <td style="text-align: center;">Xxxxxxxx</td> <td style="text-align: center;">00</td> <td style="text-align: center;">00</td> </tr> <tr> <td style="text-align: center;">00</td> <td style="text-align: center;">Xxxxxxxx</td> <td style="text-align: center;">00</td> <td style="text-align: center;">00</td> </tr> <tr> <td style="text-align: center;">00</td> <td style="text-align: center;">Xxxxxxxx</td> <td style="text-align: center;">00</td> <td style="text-align: center;">00</td> </tr> </tbody> </table> <p>son: Xxxxxxx xxxxxxxxx x x x xxxxxx</p> <div style="text-align: right; margin-top: 10px;">  </div> </div>	▼ Cantidad	▼ Descripción	▼ Precio	▼ Importe	00	Xxxxxxxx	00	00	<b>Actor:</b> Vendedor	<b>Depende de:</b> HU9, HU10												
	▼ Cantidad	▼ Descripción	▼ Precio	▼ Importe																		
	00	Xxxxxxxx	00	00																		
	00	Xxxxxxxx	00	00																		
00	Xxxxxxxx	00	00																			
00	Xxxxxxxx	00	00																			
<b>Descripción:</b> Generación o impresión de la nota de venta o recibo																						
<b>Criterio de Aceptación:</b>																						
<ul style="list-style-type: none"> <li>Después de realizar una venta, el sistema genera y muestra en una pantalla el recibo con todos los datos de la venta correspondiente, y crea un código QR con los datos del cliente, el número y el total vendido.</li> <li>El actor presiona el botón imprimir</li> <li>El sistema imprimir el recibo en una impresora pre instalada.</li> </ul>																						
<b>Elaborado por:</b> Saul Mamani	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0																				

Tabla 32: Caso de uso Generar recibo

### 9.9.5 Sprint backlog

Para determinar **cómo** se va a desarrollar el sprint, se identifican una o varias tareas por historia de usuario, las cuales son agrupadas en un sprint backlog y asignadas a los miembros del equipo de desarrollo.

HU	Descripción	Tareas
HU8	Listar ventas	T1. Diseñar el Formulario de búsquedas T2. Desarrollar los listados
HU9	Registrar ventas	T3. Listar los productos a ser vendidos T4. Registrar en el carrito de ventas T5. Calcular los totales T6. Registrar la venta
HU10	Generar nro. de ticket	T7. Generar el número de ticket
HU11	Generar recibo	T8. Diseñar el recibo integrando un código QR

Tabla 33: Segundo sprint backlog

### 9.9.6 Sprint execution

Para controlar la ejecución del sprint y el avance de las tareas, empleamos un tablero Kanban con la herramienta Trello.

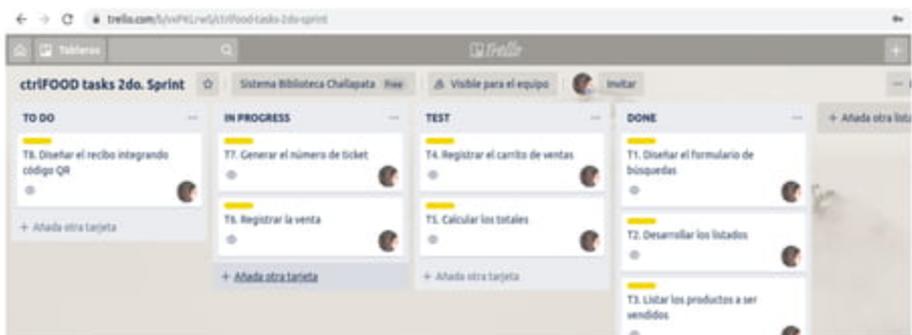


Ilustración 21: Tablero Kanban segundo sprint  
Fuente: Elaboración propia

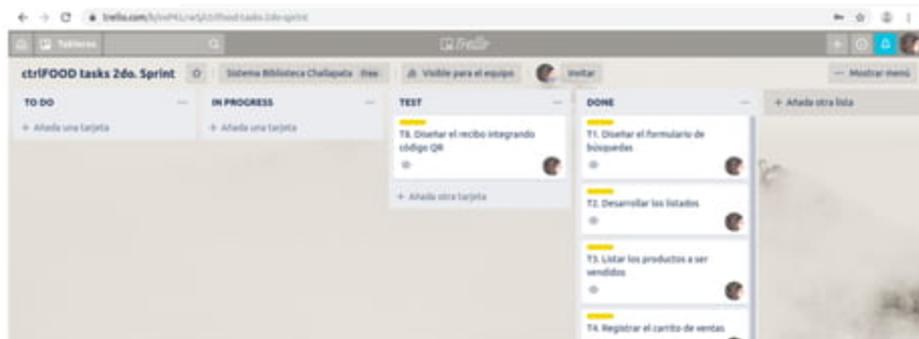


Ilustración 22: Tablero Kanban segundo sprint  
Fuente: Elaboración propia

Para controlar el avance del código fuente se utiliza el sistema de control de versiones GIT, el cual se encuentra en GitHub bajo en el siguiente repositorio: <http://github.com/saulmamani/ctrlFood>

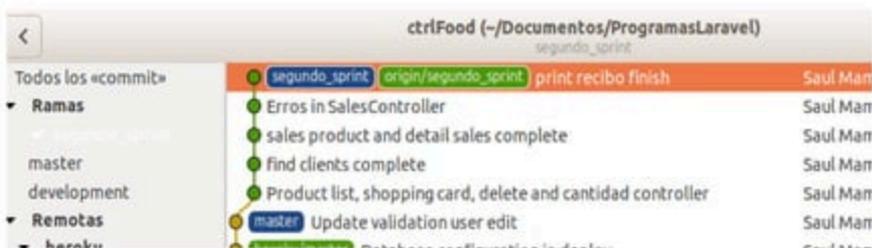


Ilustración 23: Control de versiones del segundo sprint  
Fuente: Elaboración propia

El modelo relacional de la base de datos se ha implementado en base al diagrama de clases persistentes.



Ilustración 24: Modelo relacional de la base de datos del segundo sprint  
Fuente: Elaboración propia

### 9.9.7 Sprint review

El **product increment** representa los entregables realizados al product owner y al cliente, estos entregables reflejan el cumplimiento del segundo sprint.

Se ha desplegado el sistema en la siguiente dirección URL, a modo de prueba para su revisión: <https://ctrfood.herokuapp.com>

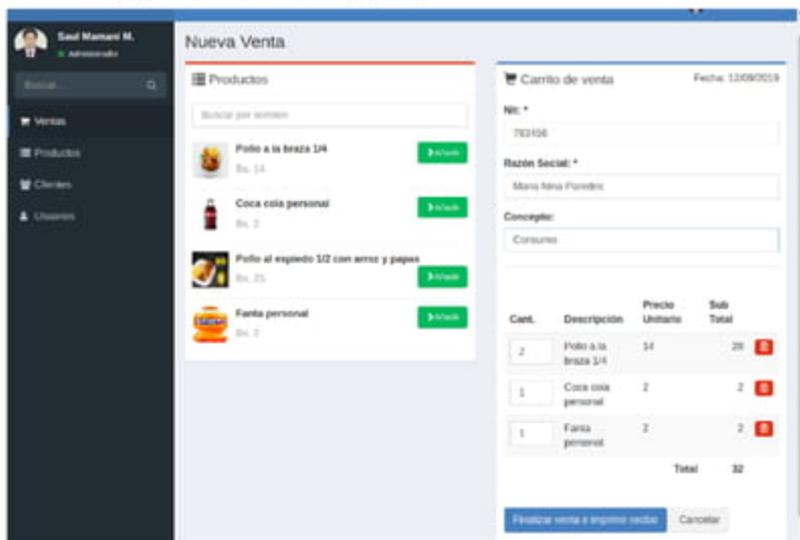


Ilustración 25: Pantalla registrar ventas  
Fuente: Elaboración propia



Ilustración 26: Pantalla generar recibo  
Fuente: Elaboración propia

### 9.9.8 Sprint retrospective

El desarrollo del segundo sprint se concluyó de manera adecuada cumpliendo las tareas, los tiempos establecidos, y la definición de hecho establecido al inicio del sprint. Esta evolución se puede apreciar en el gráfico burndown.

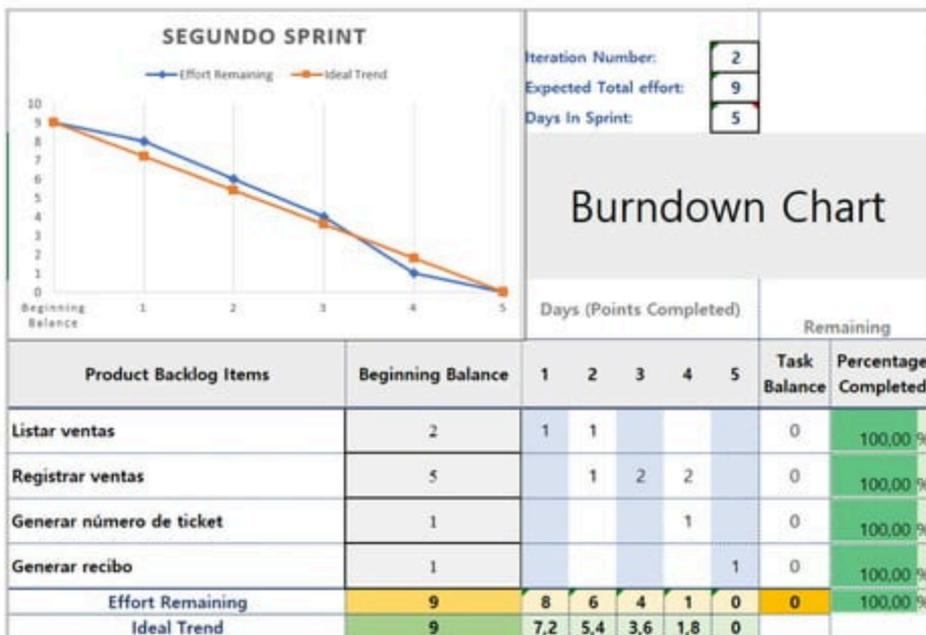


Ilustración 27: Seguimiento de tareas segundo sprint  
 Fuente: Elaboración propia



Ilustración 28: Burndown chart segundo sprint  
 Fuente: Elaboración propia

## 9.10 Tercer sprint

### 9.10.1 Sprint planning

El sprint planning muestra **qué** es lo que se tiene que desarrollar en esta iteración, y los tiempos comprometidos para finalizar cada historia de usuario.

HU	Descripción	Puntos de historia	Tiempo [Días]	Inicio	Finalización
HU12	Anular ventas	1	½	16/09/2019	16/09/2019
HU13	Restablecer ventas	1	½	16/09/2019	16/09/2019
HU14	Generar informe económico	3	2	17/09/2019	18/09/2019
HU15	Generar reportes estadísticos	3	2	19/09/2019	20/09/2019
<b>Puntos historia/Tiempo estimado (Time Boxing)</b>		<b>8</b>	<b>5</b>		

Tabla 34: Tercer sprint planning

Para completar este sprint se estima un esfuerzo de 8 puntos de historia y tiempo de desarrollo de cinco días.

### 9.10.2 Diagrama de casos de uso

El diagrama de casos de uso muestra la funcionalidad global del sprint.

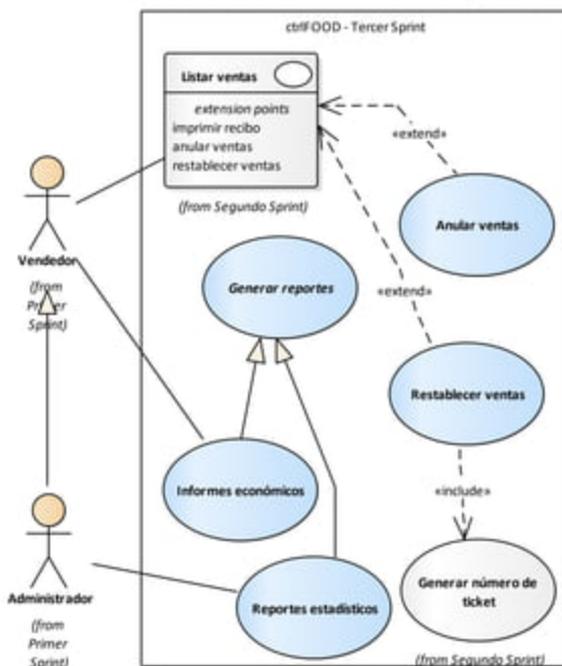


Ilustración 29: Diagrama de casos de uso tercer sprint  
Fuente: Elaboración propia

### 9.10.3 Diagrama de clases persistentes

Se emplea el diagrama de clases para modelar el esquema lógico de la base de datos.

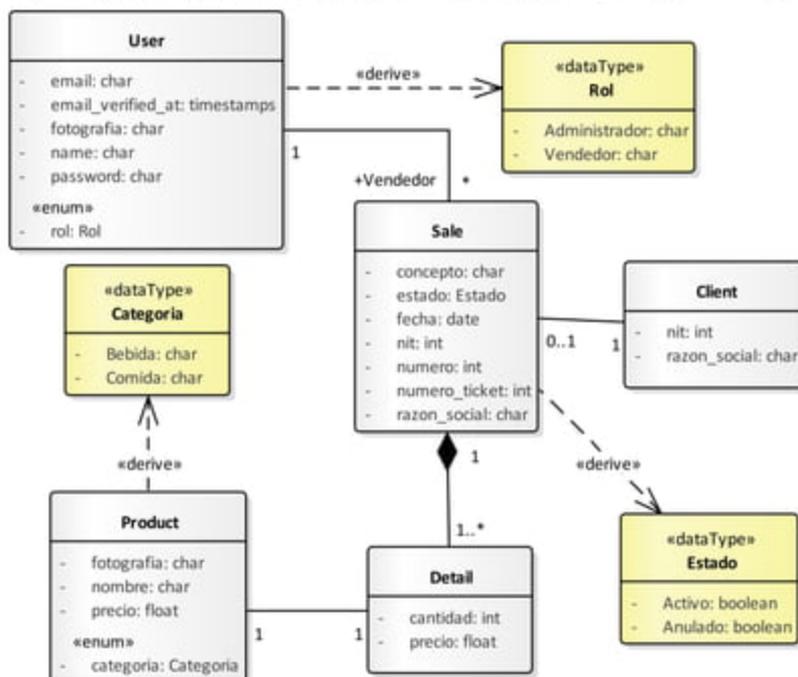


Ilustración 30: Diagrama de clases tercer sprint  
Fuente: Elaboración propia

### 9.10.4 Especificación de casos de uso

Para la especificación de los casos de uso utilizamos historias de usuario detallados.

<b>Caso de uso: Anular ventas</b>  	<b>Historia de usuario: HU12</b>	
	<b>Prioridad:</b> Baja	<b>Depende de:</b> HU8
	<b>Descripción:</b> Anulación de ventas erróneas	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>El sistema muestra la lista de ventas realizadas de la fecha actual, con el botón para Anular</li> <li>El actor pulsa el botón Anular</li> <li>El sistema cambia el estado de la venta en la base de datos y recarga la lista.             <ul style="list-style-type: none"> <li>Las ventas anuladas no son contabilizadas tomadas en cuenta para el reporte económico.</li> </ul> </li> </ul>		
<b>Elaborado por:</b> Saul Mamani	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0

Tabla 35: Caso de uso Anular ventas

<b>Caso de uso:</b> Restablecer ventas	<b>Historia de usuario:</b> HU13	
	<b>Prioridad:</b> Baja	<b>Depende de:</b> HU12
	<b>Descripción:</b> Restablece ventas anuladas por error	
	<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>– El sistema muestra la lista de ventas anuladas de la fecha actual.</li> <li>– El actor pulsa el botón Restablecer</li> <li>– El sistema cambia el estado de la venta en la base de datos y recarga la lista.</li> </ul>	
<b>Elaborado por:</b> Saul Mamani	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0

Tabla 36: Caso de uso Restablecer ventas

<b>Caso de uso:</b> Generar reportes	<b>Historia de usuario:</b> HU14, HU15	
	<b>Prioridad:</b> Baja	<b>Depende de:</b>
	<b>Descripción:</b> Generar informes y reportes estadísticos	
	<b>Criterio de Aceptación:</b> <p><b>Reporte económico</b></p> <ul style="list-style-type: none"> <li>– En la lista de ventas (HU8) el sistema muestra un botón para imprimir el reporte, y el actor pulsa el botón.</li> <li>– El sistema genera el reporte económico en PDF.</li> </ul> <p><b>Reporte estadístico</b></p> <ul style="list-style-type: none"> <li>– El actor pulsa el botón Reportes del menú principal</li> <li>– El sistema genera dos reportes estadísticos: <ul style="list-style-type: none"> <li>o Gráfica del total de las ventas realizadas agrupadas por mes</li> <li>o Lista de clientes potenciales (clientes con más compras realizadas)</li> </ul> </li> </ul>	
<b>Elaborado por:</b> Saul Mamani	<b>Responsable:</b> Saul Mamani	<b>Versión:</b> 1.0

Tabla 37: Caso de uso Generar reportes

### 9.10.5 Sprint backlog

Para determinar cómo se va a desarrollar el sprint, se identifican una o varias tareas por historia de usuario, las cuales son agrupadas en un sprint backlog y asignadas a los miembros del equipo de desarrollo.

HU	Descripción	Tareas
HU12	Anular ventas	T1. Desarrollar anular ventas
HU13	Restablecer ventas	T2. Desarrollar restablecer ventas
HU14	Generar informe económico	T3. Diseñar el reporte económico en PDF
HU15	Generar reportes estadísticos	T4. Diseñar el reporte de los clientes potenciales T5. Diseñar el reporte de ingresos económicos por año

Tabla 38: Tercer sprint backlog

### 9.10.6 Sprint execution

Para controlar la ejecución del sprint y el avance de las tareas, empleamos un tablero Kanban con la herramienta Trello.

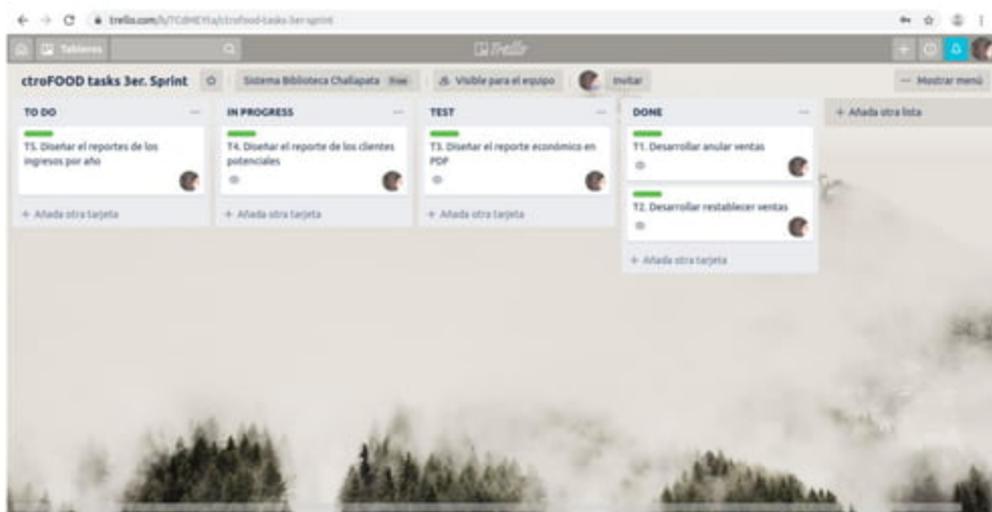


Ilustración 31: Tablero Kanban tercer sprint  
Fuente: Elaboración propia

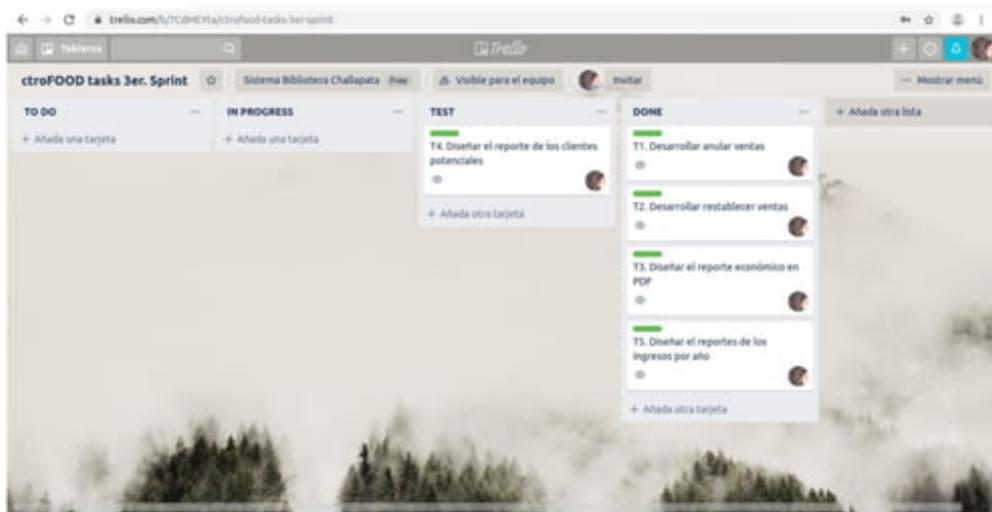


Ilustración 32: Tablero Kanban tercer sprint  
Fuente: Elaboración propia

Para controlar el avance del código fuente se utiliza el sistema de control de versiones GIT, el cual se encuentra en GitHub bajo en el siguiente repositorio: <http://github.com/saulmamani/ctrlFood>

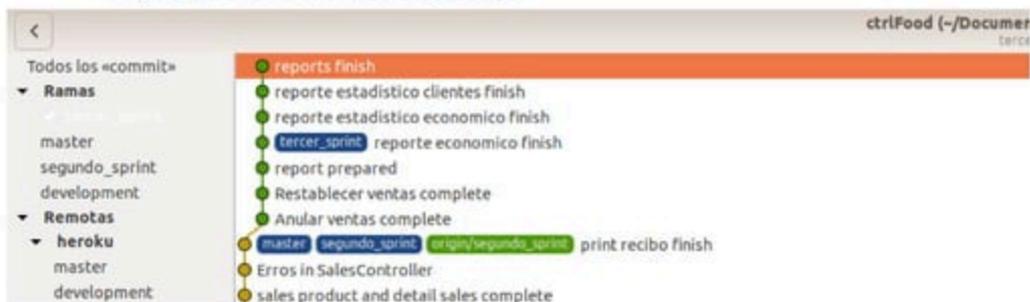


Ilustración 33: Control de versiones del tercer sprint  
Fuente: Elaboración propia

El modelo relacional de la base de datos que se ha implementado en base al diagrama de clases persistentes.

Se puede observar que no ha sufrido cambios desde el sprint anterior, por lo tanto, se concluye que el base de datos fue concluido en el segundo sprint.



Ilustración 34: Modelo relacional de la base de datos del tercer sprint  
Fuente: Elaboración propia

### 9.10.7 Sprint review

El **product increment** representa los entregables realizados al product owner y al cliente, estos entregables reflejan el cumplimiento del tercer sprint.

Se ha desplegado el sistema en la siguiente dirección URL, a modo de prueba para su revisión: <https://ctrlfood.herokuapp.com>

Número de Ticket	Fecha	Cliente	Total [Bs]	Registrado por	Estado
6	11/09/2019 21:26:36	Pedro Lamas No. 1	25.00	luis0_1@yahoo.es	Activo
8	11/09/2019 21:54:23	Joselina Berrios No. 1	2.00	luis0_1@yahoo.es	Activo
3	11/09/2019 20:59:36	Lake Macia No. 3000000000	25.00	luis0_1@yahoo.es	Activo
2	11/09/2019 20:48:29	Saul Mamani Mamani No. 3000000000	39.00	luis0_1@yahoo.es	Activo
1	11/09/2019 20:44:29	Saul Mamani Mamani No. 3000000000	39.00	luis0_1@yahoo.es	Activo

Ilustración 35: Pantalla restablecer ventas  
Fuente: Elaboración propia

Número de Ticket	Fecha	Cliente	Total [Bs]	Registrado por	Estado
4	11/09/2019 21:08:22	Jobikts No.	2.00	luis0_1@yahoo.es	Anulado
2	11/09/2019 20:48:29	Saul Mamani Mamani No. 3000000000	39.00	luis0_1@yahoo.es	Anulado

Ilustración 36: Pantalla ventas anuladas  
Fuente: Elaboración propia

localhost:8000/reporte\_economico?txtBuscar=&txtEstado=1&txtDesde=08%2F09%2F2019&txtHasta=15%2F09%2F2019

**SALI Restaurant**  
 Plan 500 Av. Héroes del Chaco #232  
 NIT: 3095304010  
 TEL/CEL: 76137269

**REPORTE ECONÓMICO**

Desde: 08/09/2019 - Hasta: 15/09/2019  
 Estado: Activo - Filtrado por:

Número de Ticket	Fecha	Cliente	Total [Bs]	Registrado por
1	15/09/2019 09:08:38	Saul Morales Miranda Nc: 3095304	32.00	luarfi_1@yahoo.es
1	12/09/2019 11:43:07	Maria Nina Paredes Nc: 763456	32.00	luarfi_1@yahoo.es
6	11/09/2019 21:26:35	Pedro Lamas Nc: 0	25.00	luarfi_1@yahoo.es
5	11/09/2019 21:16:21	Josefina Berrios Nc: 0	2.00	luarfi_1@yahoo.es
3	11/09/2019 20:59:36	Lidia Marce Nc: 3767932012	25.00	luarfi_1@yahoo.es
2	11/09/2019 20:48:29	Saul Mamani Mamani Nc: 3095304010	39.00	luarfi_1@yahoo.es
1	11/09/2019 20:44:29	Saul Mamani Mamani Nc: 3095304010	30.00	luarfi_1@yahoo.es

**Total vendido [Bs]: 185**

Ilustración 37: Pantalla reporte de ventas  
 Fuente: Elaboración propia



Ilustración 38: Pantalla reporte estadístico  
 Fuente: Elaboración propia

### 9.10.8 Sprint retrospective

El desarrollo del tercer sprint se concluyó de manera adecuada cumpliendo las tareas, los tiempos establecidos, y la definición de hecho establecido al inicio del sprint. Esta evolución se puede apreciar en el gráfico burndown.

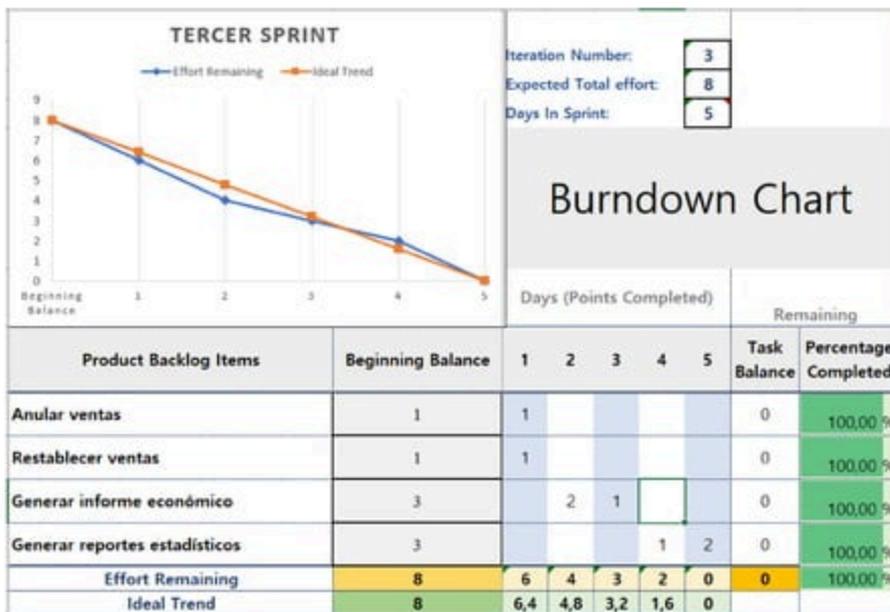


Ilustración 39: Seguimiento de tareas tercer sprint  
 Fuente: Elaboración propia

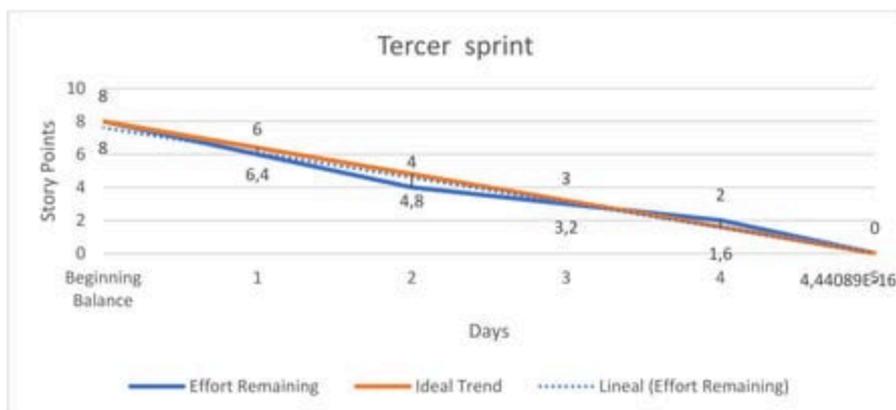


Ilustración 40: Burndown chart tercer sprint  
 Fuente: Elaboración propia

## 9.11 Diagrama general de casos de uso

Se puede ver la funcionalidad global del sistema, y su interacción con los actores, en el siguiente diagrama de casos de uso.

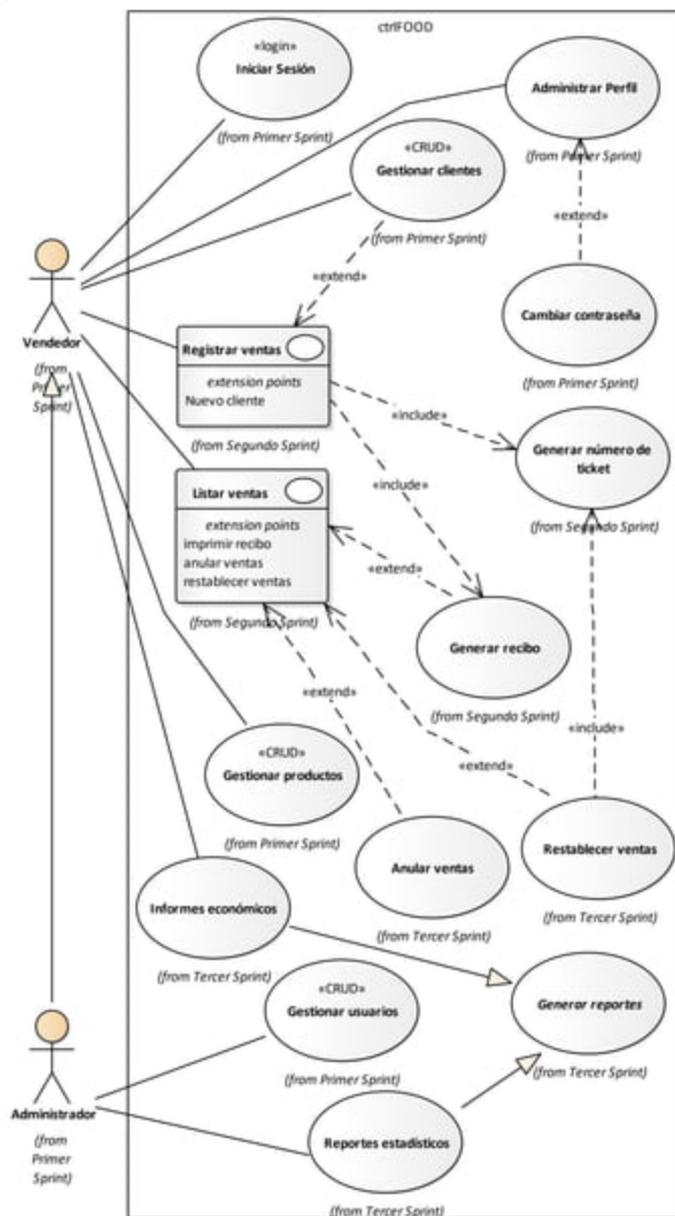


Ilustración 41: Diagrama general de casos de uso  
Fuente: Elaboración propia

## 9.12 Diagrama de despliegue

El diagrama de despliegue muestra la arquitectura física de red y los servidores donde el software será puesto en producción, además de, las terminales de los clientes que van a utilizar el sistema.

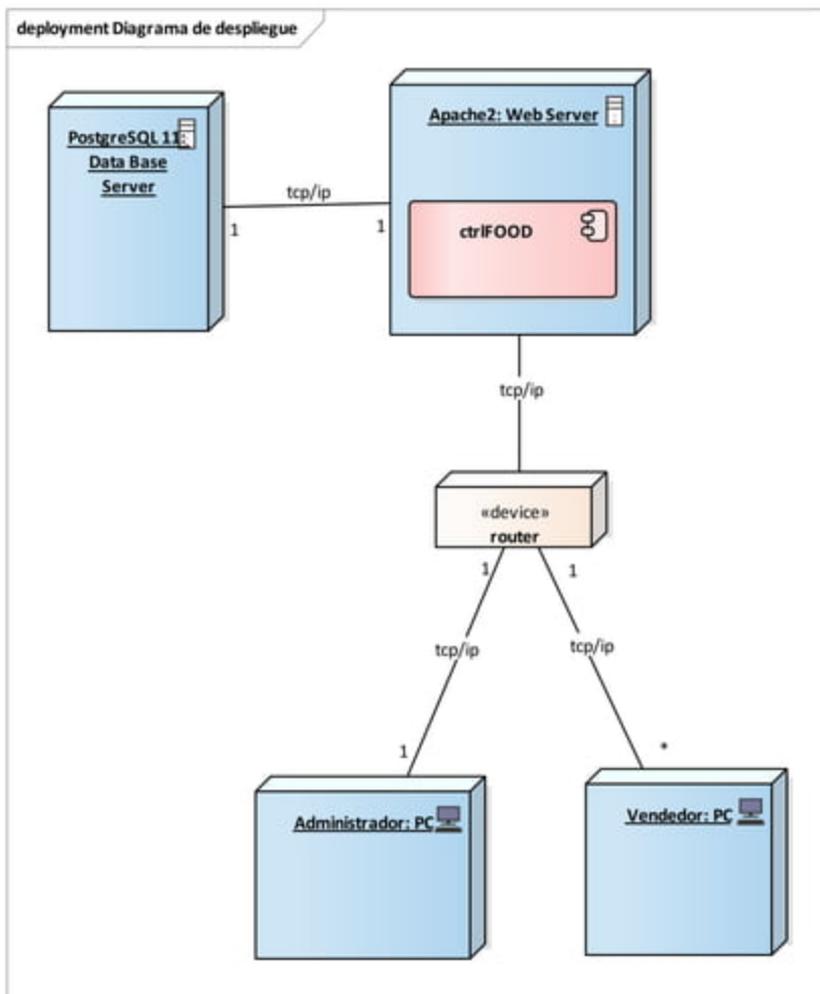


Ilustración 42: Diagrama de despliegue  
Fuente: Elaboración propia

## 10 CONCLUSIONES Y RECOMENDACIONES

Después de haber culminado las diferentes etapas del desarrollo de la monografía, se puede llegar a las siguientes conclusiones y realizar las algunas recomendaciones:

### 10.1 Conclusiones

Es posible trabajar con una metodología ágil como scrum en combinación con una notación de modelado visual como UML, sin dañar el valor del manifiesto ágil que dicta: "Software funcionando sobre documentación exhaustiva", utilizando solamente los diagramas necesarios que ayuden a comprender mejor la funcionalidad, las interacciones, y la lógica del sistema.

UML, al ser una notación de modelado visual, mejora la comprensión del problema y el objetivo de las iteraciones, además, mejora la comunicación entre los diferentes miembros del equipo scrum y los stakeholders, puesto que con una notación visual se puede observar mejor los distintos niveles de abstracción de un sistema.

Se han cumplido también las tareas planteadas al principio del proyecto. [Pág. 3]

- Con la ayuda del product vision board se logró comprender la visión del producto que fue desarrollado como ejemplo para la monografía.
- El ámbito del sistema fue determinado con las historias de usuario escritas por el product owner.
- Se ha cumplido con la planificación descrita en el product backlog, siguiendo los plazos estimados dentro las iteraciones y las entregas establecidas en el despliegue.
- Se han modelado diagramas de casos de uso, para tener una visión global de la funcionalidad dentro de cada iteración o de cada sprint, y diagramas de clases para representar la lógica de la base de datos. Además, se ha modelado el diagrama de componentes y el diagrama de despliegue para mostrar la arquitectura del sistema, la arquitectura física, los servidores, y la tecnología que se necesita para la puesta en producción del sistema.
- Para controlar el avance de las tareas identificadas en el sprint backlog de cada iteración, se ha utilizado un tablero virtual Kanban, apoyándonos en la herramienta online Trello.
- Se tiene un historial completo de todos los cambios realizados al código del sistema con la herramienta de control de versiones Git.
- El sistema fue desarrollado de acuerdo a lo planificado y se encuentra desplegado Heroku como un servidor de pruebas. <https://ctrifood.herokuapp.com>

### 10.2 Recomendaciones

Si se va utilizar una notación de modelado visual como UML con una metodología de desarrollo ágil, se recomienda aplicar solo los diagramas que sean estrictamente necesarios para resolver y comprender mejor el problema; de esta forma no dañará los valores y los principios del manifiesto ágil.

## BIBLIOGRAFÍA

- Conallen, J. (2011). *Building Web Applications with UML*. Boston: Addison-Wesley.
- Group, O. M. (13 de 09 de 2019). *omg*. Obtenido de omg: <https://www.omg.org/spec/UML/About-UML/>
- Hiromoto, H. (31 de 12 de 2017). *scrumorganico*. Obtenido de scrumorganico: <https://medium.com/scrumorganico>
- Manifiesto, A. (13 de 09 de 2019). *agilemanifiesto*. Obtenido de agilemanifiesto: <https://agilemanifiesto.org/iso/es/manifiesto.html>
- Mercurial, A. (5 de 9 de 2019). *Agile Mercurial*. Obtenido de Agile Mercurial: <https://agile-mercurial.com/excel-templates/burndown-chart/>
- Moreno, G. R. (2004). *UML con Rational Rose*. Lima: MEGABYTE SAC.
- Rubin, K. S. (2013). *Essential Scrum*. Boston: Addison Wesley.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2007). *El Lenguaje de Modelado Unificado 2.0*. Madrid: Perarson Addison Wesley.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2007). *El lenguaje de Modelado Unificado Manual de Referencia*. Madrid: PEARSON Addison Wesley.
- Saiz, J. (2018 de 09 de 13). *js*. Obtenido de js: <http://jorgesaiz.com>
- Sommerville, I. (2005). *Ingenieria de Software*. Madrid: PEARSON Addison Wesley.