

**SOLUCIÓN TECNOLÓGICA PARA CONTROLAR EL FLUJO DE CAJA E
INVENTARIO DE LAS TIENDAS EN BOGOTÁ**

**JOHN JAIRO TAMAYO GUANCHA
NICOLAS MOLINA MONROY**

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA, REDES Y ELECTRÓNICA
PROGRAMA DE TECNOLOGÍA EN INFORMÁTICA
BOGOTÁ D.C.**

**SOLUCIÓN TECNOLÓGICA PARA CONTROLAR EL FLUJO DE CAJA E
INVENTARIO DE LAS TIENDAS EN BOGOTÁ**

Trabajo de grado, para optar al título de Tecnólogo en Informática.

Asesor metodológico: Luis Eduardo Pérez Peregrino

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA, REDES Y ELECTRÓNICA
PROGRAMA DE TECNOLOGÍA EN INFORMÁTICA
BOGOTÁ D.C.**

Nota de aceptación

Firma del presidente del jurado

Firma del jurado No. 1

Firma del jurado No. 2

DEDICATORIA

Dedico este proyecto mi madre la cual ha sido y es un eje principal en mi formación como persona. También a los docentes de la Universidad Minuto de Dios por su formación académica y profesional que me han brindado.

Por último a todos los desarrolladores de software por sus aportes y su noción de compartir libremente sus conocimientos para todo el mundo sin interés monetario alguno por medio de libros, artículos, revistas, tutoriales, guías, vídeos, blogs, foros, chats, etc.

John Jairo Tamayo Guancha

Dedico este proyecto a mis padres por su gran apoyo en mi formación profesional, a los profesores de la Universidad Minuto de Dios por brindarnos su apoyo y conocimientos y a mi novia por su comprensión y apoyo.

También a todos los que hacemos de esta carrera un maravilloso arte y lo hacemos con esfuerzo para solucionar problemas de nuestra sociedad a partir de la tecnología.

Nicolás Molina Monroy

AGRADECIMIENTOS

Agradezco a toda la comunidad de PHP y Laravel Framework por sus aportes al mundo del desarrollo de software, también a las comunidades de desarrollo que se generan por el arte de la programación y creación de tecnologías.

Por último a todos desarrolladores que comparten sus conocimientos por Internet.

John Jairo Tamayo Guancha

Agradezco especialmente a Addy Osmani, desarrollador de Framework de Backbone.js y sus grandes aportes en el mundo del desarrollo de software con JavaScript, y las comunidades de desarrollo en las cuales se comparte el conocimiento sin ánimo de lucro.

Por último a la música (menos el reggaeton) la cual me acompaña de fondo siempre en las horas de desarrollo de software.

Nicolas Molina Monroy

Contenido

RESUMEN	9
INTRODUCCIÓN	11
Título del proyecto	12
Planteamiento del problema	12
Alcance y justificación	13
Objetivos	14
Objetivo general	14
Objetivos específicos.....	14
INGENIERÍA DEL PROYECTO	15
Metodología de desarrollo: Scrum	16
Cronograma	20
Marco tecnológico	21
Consideraciones tecnológicas	30
ANÁLISIS Y DISEÑO	31
Módulo de ventas	31
Módulo de inventario	34
Módulo de compras	35
Módulo de reportes.....	37
Módulo de cuentas por pagar	38
Módulo de cuentas por cobrar	39
Módulo de sincronización	40
Módulo de usuarios	41
Descripción del sistema propuesto	42
Diseño del sistema propuesto.....	44
DESARROLLO	66
GLOSARIO	68
CONCLUSIONES	69
BIBLIOGRAFÍA	70
MANUALES	71

LISTA DE GRÁFICOS

<i>Grafica 1 Ficha sinóptica</i>	19
<i>Grafica 2: Cronograma</i>	20
<i>Grafica 3: Países con mayor adopción en el uso de Smartphone</i>	21
<i>Grafica 4: Estudio de Consumo Digital</i>	22
<i>Grafica 5: PHP</i>	23
<i>Grafica 6: Laravel</i>	23
<i>Grafica 7: MySql</i>	24
<i>Grafica 8: HTML5</i>	24
<i>Grafica 9: JavaScript</i>	25
<i>Grafica 10: Backbone.js</i>	25
<i>Grafica 11: Underscore.js</i>	26
<i>Grafica 12: CSS3</i>	26
<i>Grafica 13: Twitter Bootstrap</i>	27
<i>Grafica 14: jQuery Mobile</i>	27
<i>Grafica 15: Web SQL Database</i>	28
<i>Grafica 16: Estructura de una aplicación realizada con Apache Cordova</i>	28
<i>Grafica 17: Adobe PhoneGap Build</i>	29
<i>Grafica 18: Consideraciones tecnológicas</i>	30

LISTA DE TABLAS

<i>Tabla 1: Integrantes del proyecto</i>	15
------------------------------------------------	----

LISTA DE DIAGRAMAS

<i>Diagrama 1: Hoja de caso de uso (Modulo de Caja)</i>	44
<i>Diagrama 2: Caso de Uso (Modulo de Caja)</i>	45
<i>Diagrama 3: Hoja de Caso de Uso (Modulo de Compras)</i>	46
<i>Diagrama 4: Caso de Uso (Modulo de Compras)</i>	47
<i>Diagrama 5: Hoja de Caso de Uso (Modulo de Cuentas)</i>	48
<i>Diagrama 6: Caso de Uso (Modulo de Cuentas)</i>	49
<i>Diagrama 7: Hoja de Caso de Uso (Modulo de inventario)</i>	50
<i>Diagrama 8: Caso de uso de (Modulo de Inventario)</i>	51
<i>Diagrama 9: Hoja de Caso de Uso (Modulo de Reportes)</i>	52
<i>Diagrama 10: Caso de Uso (Modulo de Reportes)</i>	53
<i>Diagrama 11: Hoja de Caso de Uso (Modulo de Sincronizacion)</i>	54
<i>Diagrama 12: Caso de uso (Modulo de Sincronización)</i>	55
<i>Diagrama 13: Hoja de Caso de Uso (Modulo de Usuarios)</i>	56
<i>Diagrama 14: Caso de Uso (Modulo de Usuarios)</i>	57
<i>Diagrama 15: Hoja de Caso de Uso (Modulo de Ventas)</i>	58
<i>Diagrama 16: Caso de Uso (Modulo de Ventas)</i>	59
<i>Diagrama 17: Diagrama de Clases (Modulo de Caja)</i>	60

<i>Diagrama 18: Diagrama de Clases (Modulo de Compras)</i>	61
<i>Diagrama 19: Diagrama de Clases (Modulo de Inventarió)</i>	62
<i>Diagrama 20: Diagrama de Clases (Modulo de Ventas)</i>	63
<i>Diagrama 21: Diagrama de Componentes</i>	64

RESUMEN

El presente proyecto de grado tiene el propósito de analizar, diseñar y desarrollar una solución tecnológica para facilitar el registro de los ingresos y egresos de una tienda, también facilitar la gestión del inventario de una tienda, brindando información detallada e integral, por consiguiente el tendero puede tomar mejores decisiones para su tienda.

La metodología de desarrollo que se utiliza es SCRUM siendo una metodología ágil, brinda la facilidad de probar, fallar e iterar rápidamente para un desarrollo de mejora continua. También se logra obtener un desarrollo simple, cumpliendo los requerimientos suficientes para un generar un prototipo funcional.

En el análisis del sistema se utiliza el lenguaje unificado de modelado (UML) para diseñar una solución correspondiente a los requerimientos obtenidos, brindando vistas gráficas para interpretar los requerimientos que se desean solventar.

Para el desarrollo del sistema se utilizan tecnologías basadas en Internet (Cloud Computing) y tecnologías móviles brindando escalabilidad y portabilidad a la solución tecnológica planteada.

ABSTRACT

This graduation project aims to analyze, design and develop a technological solution to facilitate the recording of revenues and expenses of a store, also facilitate management of the inventory of a store, providing detailed and comprehensive information, therefore the shopkeeper you can make better decisions for your store.

The development methodology used is an agile methodology SCRUM this methodology provides the facility to try, fail and iterate quickly for development of continuous improvement. It also allow to obtain a simple development, meeting the requirements sufficient for generating a functional prototype.

In system analysis is used Unified Modeling Language (UML) to the design a solution corresponding to the requirements, providing graphical views to interpret the requirements that you want to solve.

To develop the system are used technologies based on Internet (Cloud Computing) and mobile technologies to provide scalability and portability to the proposed technological solution.

INTRODUCCIÓN

Hoy en día uno de los mayores negocios que existen en la ciudad de Bogotá son las tiendas de barrio según el censo de “*establecimientos comerciales de Infocomercio 2011*”, las cuales ofrecen sus servicios y productos a la comunidad. Las tiendas se han mantenido gracias a su amabilidad, atención, economía y facilidad de adquisición de productos para sus clientes.

Sin embargo, actualmente esto ha cambiado, debido a que las grandes cadenas se están expandiendo a las zonas donde las tiendas de barrio han ejercido su labor. Por ejemplo, el director de Éxito Express Nicolás Restrepo dijo lo siguiente: “el Grupo Éxito le apuesta a este tipo de tiendas de conveniencia, pues el formato express responde a necesidades puntuales de los clientes, de encontrar cercanía y agilidad en sus compras en lugares de compra vecinos a su trabajo o residencia”.

Algunas desventajas que tienen actualmente las tiendas de barrio al frente de las grandes cadenas es su mala organización tanto en la contabilidad y en el inventario de sus productos siendo estos últimos el centro de la existencia de este tipo de negocio.

Debido a estos problemas se necesita proponer una solución tecnológica que los solventen para el beneficio de la tienda. Si las tiendas no se organizan o toman mejores decisiones, se verán forzadas a desaparecer.

Por último, se observa el análisis, diseño y desarrollo de la solución tecnológica pertinente, por lo cual se expondrán las tecnologías y herramientas que se utilizan para lograr la solvencia de los problemas anteriormente mencionados, también se observa sus beneficios y aportes para las tiendas de la ciudad de Bogotá.

Título del proyecto

El título del proyecto es “SOLUCIÓN TECNOLÓGICA PARA CONTROLAR EL FLUJO DE CAJA E INVENTARIO DE LAS TIENDAS EN BOGOTÁ”; el nombre está basado en el tipo de solución: (tecnológica), en su principal función: (control del flujo de caja e inventario) y para quién está orientada la solución: (tiendas en la ciudad de Bogotá).

Planteamiento del problema

Los tenderos de la ciudad de Bogotá enfrentan varios problemas diariamente en el control del flujo de caja de sus tiendas; esto perjudica seriamente el crecimiento del negocio y posteriormente el cierre inminente de la tienda.

Uno de los problemas más graves es el pésimo control del flujo de la caja de la tienda; esto conlleva a que el negocio no sea rentable y no aumente su crecimiento. Esto se debe porque los tenderos registran sus ingresos y egresos en formas arcaicas y desorganizadas, tales como el registro manual en una hoja de papel. Algunos llegan a registrar la información en hojas de cálculo de acuerdo a la información de la hoja de papel, este doble proceso se realiza para analizar la información y ayudar a tomar la mejor decisión en beneficio de la tienda.

También se observa la falta de organización en la gestión del inventario de la tienda, esto conlleva al desconocimiento parcial y suposición del número exacto de los productos presentes en la tienda para su correspondiente venta. Algunos productos son más escasos que otros, por lo tanto el tendero desconoce cuáles serían las mejores estrategias y decisiones para la tienda.

Estos problemas reflejan el estado de la tienda y su debilidad para competir con las medianas empresas y grandes cadenas nacionales que actualmente están en la vanguardia tecnológica para solucionar estos problemas y continuar con su crecimiento y rentabilidad del negocio.

Debido a la existencia de estos problemas, se plantea la pregunta: ¿cómo solucionar estos problemas para el beneficio de los tenderos en Bogotá?

Alcance y justificación

El avance de la tecnología ha brindado el fácil acceso y uso para las personas en un mundo interconectado, esto permite desarrollar sistemas tecnológicos tanto en la web y en los dispositivos móviles inteligentes.

Por esta razón se utilizará los beneficios de la tecnología para solucionar los problemas del control de flujo de caja y gestión del inventario de la tienda. Se desarrollará un sistema web para administrar las tiendas y sus respectivos usuarios, también una API REST para la consulta e inserción de la información, por último una aplicación móvil para tabletas la cual el tendero realizará la consulta y registro de la información de su tienda.

Esta solución tecnológica facilitará el registro de los ingresos (ventas) y egresos (compras) permitiendo la sistematización de este proceso para el tendero eliminando el doble registro de la información y evitando la pérdida de datos. También facilitará la gestión del inventario de la tienda, razón por la cual el tendero podrá tener el control de sus productos y como consecuencia podrá tomar mejores decisiones para su negocio.

Objetivos

Se definen los siguientes objetivos para el correcto desarrollo del proyecto:

Objetivo general

Diseñar y desarrollar una solución tecnológica para solventar los problemas de control del flujo de caja, gestión del inventario, desorganización de la información y pérdida de datos de una tienda.

Objetivos específicos

- Gestionar las tiendas y sus respectivos usuarios por medio de un sistema de información utilizando cloud computing.
- Permitir la comunicación de sistemas de información y aplicaciones externas con la base de datos de la tienda por medio de una API REST.
- Gestionar los ingresos que tiene la tienda.
- Gestionar los egresos que tiene la tienda.
- Gestionar el inventario que tiene la tienda.
- Mostrar la información por medio de tablas para brindar un análisis detallado de los datos.

INGENIERÍA DEL PROYECTO

Para un correcto desarrollo de software se debe tener presente los siguientes componentes:

Cuáles son las personas que intervienen y contribuyen en el proyecto (personas), cómo participan en el proyecto (metodología), qué acciones se realizan (requerimientos), cuánto tiempo se dispone (cronograma), cuáles herramientas y tecnologías se necesitan para el desarrollo del proyecto (herramientas y tecnologías).

Integrantes del proyecto

Rol	Responsable	Función
Líder del proyecto	Nicolas Molina	Analizar y organizar las tareas y tiempos del proyecto
Analista de requerimientos	Nicolas Molina	Obtener y analizar los requerimientos del proyecto
Arquitecto de software	John Jairo Tamayo	Diseñar y modelar la arquitectura del proyecto con respecto a los requerimientos
Analista de base de datos	John Jairo Tamayo	Guardar, organizar y asegurar la información del proyecto
Desarrollador back-end	John Jairo Tamayo	Desarrollo en el entorno del servidor
Desarrollador front-end	Nicolas Molina	Desarrollo en el entorno del cliente

Tabla 1: Integrantes del proyecto

Metodología de desarrollo: Scrum

Para el desarrollo de este proyecto, se va a utilizar la Metodología de Desarrollo Ágil SCRUM.

SCRUM es una metodología de desarrollo ágil que nos permite enfocarnos en la generación de valor en el mínimo tiempo. Nos permite tener un control continuo sobre el estado actual de nuestro proyecto.

Una de las características más relevantes es que integra al cliente como parte del grupo de desarrollo en el cual el cliente establece las prioridades y el equipo se auto-organiza para determinar la mejor forma de entregar los requerimientos con más prioridad.

Esta metodología trabaja con sprint lo cual es una etapa en el cual el grupo de desarrollo debe entregar un incremento que está aprobado y terminado para mostrar al cliente. Cada sprint finaliza con componentes entregables que se puede publicar o bien seguir mejorando.

Scrum es una filosofía de trabajo que no sólo implica al desarrollador sino también al cliente dando prioridad a los individuos y a las interacciones sobre los procesos y las tareas, prefiriendo el software funcional sobre la excesiva documentación, promocionando la colaboración con el cliente en lugar de la negociación de contratos y sobre todo teniendo capacidad de respuesta sobre los cambios en lugar de seguir estrictamente una planificación.

Creación de equipos

Scrum considera que un equipo de desarrollo debe funcionar como un equipo deportivo, cada miembro del equipo trabaja de forma independiente, pero hacia el mismo objetivo. Scrum sugiere que un equipo tiene un máximo de 6 a 7 miembros. El facilitador del equipo se llama el "Scrum master". Su trabajo es implementar y gestionar el proceso Scrum en el proyecto. El equipo de Scrum en su conjunto define las prácticas, las reuniones y la terminología de SCRUM, el Scrum Master asegura el cumplimiento de estas "normas". El Scrum master desempeña un papel facilitador y su autoridad es sobre todo indirecta: centra la mayor parte de su tiempo en la gestión de interferencias externas y la solución de los obstáculos externos o "bloqueantes" que no pueden ser resueltos por el equipo Scrum.

Creación de documentos

Producto o backlog - Actúa como un repositorio de los requisitos del producto. Se suelen contemplar los requisitos de alto nivel.

Incremento - Requerimientos obtenidos del documento de producto identificado y priorizados para la próxima versión.

Sprint - El sprint backlog es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en horas con ninguna tarea de duración superior a 16 horas. Si una tarea es mayor de 16 horas, deberá ser rota en mayor detalle. Las tareas en el sprint backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

La segmentación del proyecto

Todo el proyecto se divide en períodos de tiempo con una duración máxima de 4 semanas. Un período se denomina Sprint y cada equipo recibe una cartera de pedidos a ejecutar en un Sprint determinado.

Reuniones Scrum

Durante el sprint, el equipo lleva a cabo reuniones Scrum diariamente. Las reuniones se realizan en el mismo lugar al mismo tiempo, cada día de trabajo, las reuniones no duran más de 30 minutos.

El scrum master tiene la responsabilidad de pedir a cada miembro del equipo las tres preguntas siguientes:

1. ¿Qué has hecho desde la última reunión de scrum?
2. ¿Hay algo que haya impedido tu trabajo?
3. ¿Qué planeas hacer entre ahora y la próxima reunión scrum?

La conversación se limita a los miembros del equipo de respuesta a las preguntas anteriores. El scrum master es responsable de tomar decisiones de inmediato, si es necesario para eliminar los obstáculos al progreso.

Fases

El proceso de desarrollo Scrum se compone de 5 actividades importantes:

- Planes de lanzamientos.
- Distribución, revisión y ajuste de los estándares de producto
- Sprint
- Revisión del Sprint
- Cierre

Sprint

La fase de Sprint es donde el desarrollo de software se lleva a cabo. Un Sprint consta de las siguientes sub-actividades:

- Elaborar
- Integrar
- Revisar
- Ajustar

Esta fase no tiene una secuencia. A veces un elemento del backlog se tiene que desarrollar, integrar, y revisar cuando otros sólo deben ser revisados o ajustados.

Revisión de Sprint

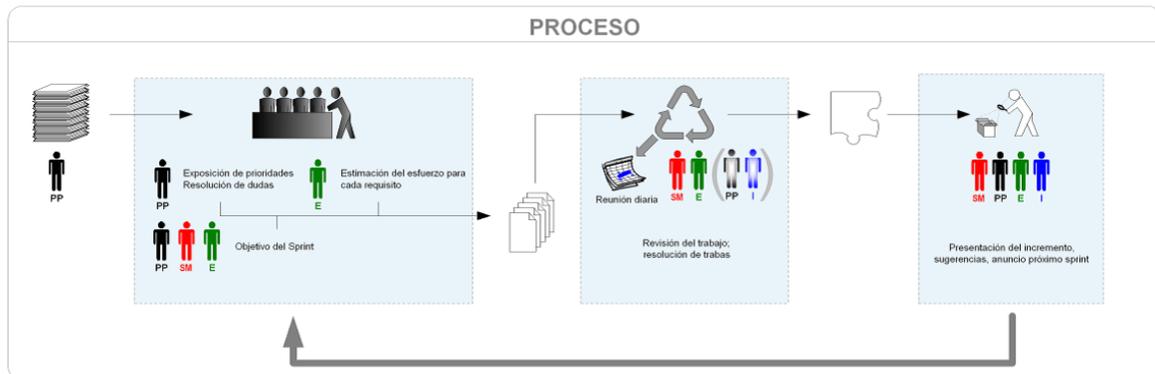
Cada Sprint es seguido por una revisión de Sprint. Durante esta revisión, el software desarrollado en el Sprint anterior se revisa y si es necesario se le añaden nuevos ítems del backlog. El grupo de revisores pueden ser: las partes interesadas del proyecto, gestores, desarrolladores y, en ocasiones los clientes, ventas y marketing. Las actividades, y la revisión de Sprint se repiten hasta que el producto se considera listo para su distribución por los participantes en el proyecto. Luego, el proyecto pasa a la fase de cierre en que el producto se prepara para el lanzamiento y la distribución.

Cierre

En esta fase tienen lugar las actividades de debugging, marketing y promoción. Al acabar esta fase el proyecto quedará cerrado.

SCRUM: FICHA SINÓPTICA

Rev. 0.4



<p style="text-align: center;">ROLES</p> <p>PROPIETARIO DEL PRODUCTO Determina las prioridades. Una sola persona. PP</p> <p>SCRUM MANAGER Gestiona y facilita la ejecución del proceso. SM</p> <p>EQUIPO Construye el producto. E</p> <p>INTERESADOS Asesoran y observan. I</p>	<p style="text-align: center;">COMPONENTES</p> <p>PILA DEL PRODUCTO Relación de requisitos del producto, no es necesario excesivo detalle. Priorizados. Lista en evolución y abierta a todos los roles. El propietario del producto es su responsable y quien decide.</p> <p>PILA DEL SPRINT Requisitos comprometidos por el equipo para el sprint con nivel de detalle suficiente para su ejecución.</p> <p>INCREMENTO Parte del producto desarrollada en un sprint, en condiciones de ser usada (pruebas, codificación limpia y documentada).</p>	<p style="text-align: center;">REUNIONES</p> <p>PLANIFICACIÓN DEL SPRINT 1 jornada de trabajo. El propietario del producto explica las prioridades y dudas del equipo. El equipo estima el esfuerzo de los requisitos prioritarios y se elabora la pila del sprint. El Scrum Manager define en una frase el objetivo del sprint.</p> <p>REUNIÓN DIARIA 15 minutos de duración, dirigida por el Scrum Manager, sólo puede intervenir el equipo: ¿Qué hiciste ayer?, ¿Cuál es el trabajo para hoy?, ¿Qué necesitas?. Se actualiza la pila del sprint.</p> <p>REVISIÓN DEL SPRINT Informativa, aprox. 4 horas, moderada por el Scrum Manager, presentación del incremento, planteamiento de sugerencias y anuncio del próximo sprint.</p>	<p style="text-align: center;">SPRINT</p> <p>Ciclo de desarrollo básico de SCRUM, de duración máxima de 30 días en el que se desarrolla un incremento del producto.</p> <p style="text-align: center;">VALORES</p> <ul style="list-style-type: none"> - Empowerment y compromiso de las personas - Foco en desarrollar lo comprometido - Transparencia y visibilidad del proyecto - Respeto entre las personas - Coraje y responsabilidad
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Proceso ágil de desarrollo iterativo e incremental. Origen: artículo "The New New Product Development Game" (Takeuchi y Nonaka, 1988). Jeff Sutherland fue el primero en implementarlo en para desarrollo de software (1993). Ken Schwaber es su principal difusor.

Juan Palacio

Grafica 1 Ficha sinóptica

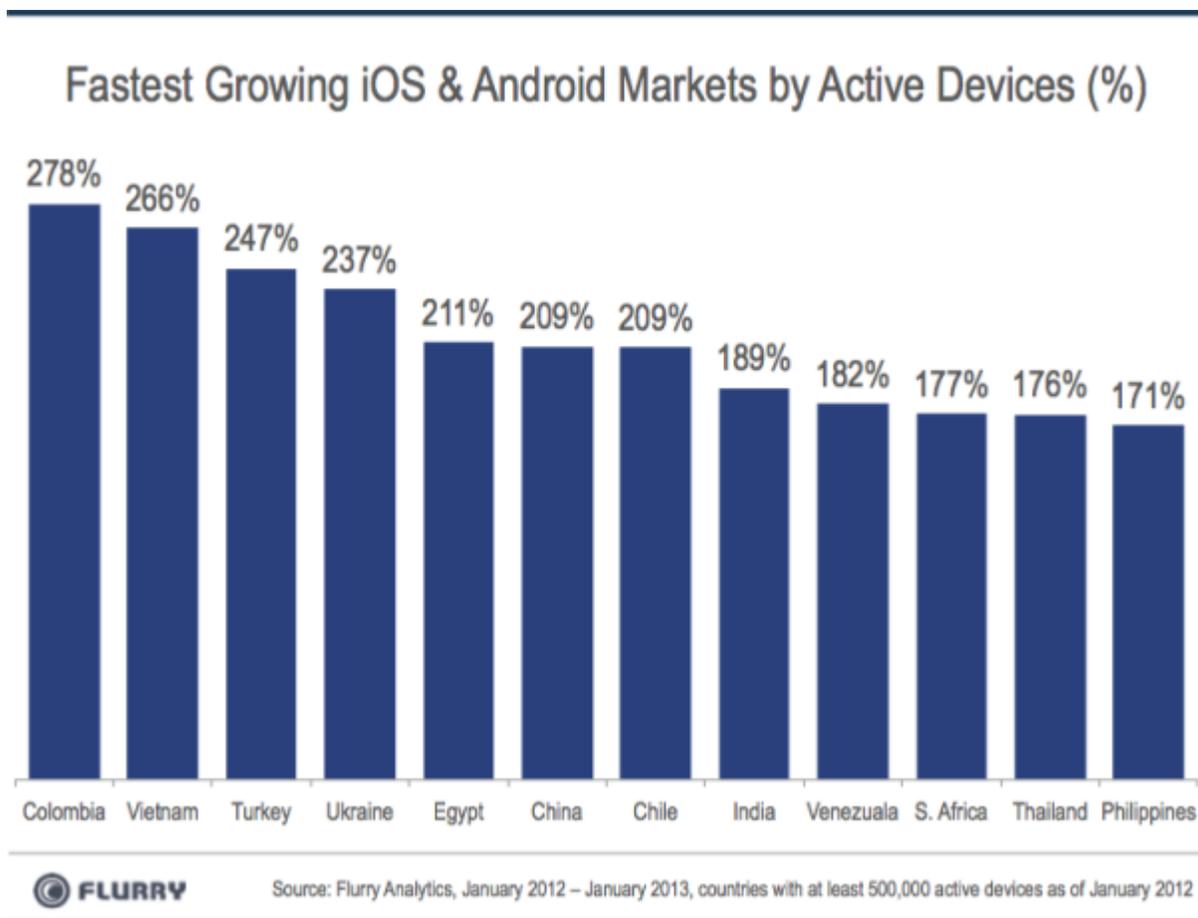
Cronograma

			enero 2015	
Nombre	Fecha de inicio	Fecha de fin	1	2
♀ • Levantamiento de información	20/01/15	20/01/15		
• Requerimientos funcionales	20/01/15	20/01/15		
• Requerimientos de información	20/01/15	20/01/15		
• Requerimientos de restricción	20/01/15	20/01/15		
• Requerimientos no funcionales	20/01/15	20/01/15		
♀ • Análisis y diseño del sistema	20/01/15	20/01/15		
• Objetivos del proyecto	20/01/15	20/01/15		
• Asignación de tareas y tiempos	20/01/15	20/01/15		
• Arquitectura del sistema	20/01/15	20/01/15		
• Arquitectura de base de datos	20/01/15	20/01/15		
• Diseño gráfico del sistema	20/01/15	20/01/15		
♀ • Desarrollo de sistema web	20/01/15	20/01/15		
• Módulo de tiendas	20/01/15	20/01/15		
• Módulo de usuarios	20/01/15	20/01/15		
• Módulo de consulta	20/01/15	20/01/15		
• Desarrollo de API REST	20/01/15	20/01/15		
♀ • Desarrollo de aplicación móvil	20/01/15	20/01/15		
• Módulo de ventas	20/01/15	20/01/15		
• Módulo de caja	20/01/15	20/01/15		
• Módulo de inventario	20/01/15	20/01/15		
• Módulo de compras	20/01/15	20/01/15		
• Módulo de reportes	20/01/15	20/01/15		
• Módulo de cuentas por pagar	20/01/15	20/01/15		
• Módulo de cuentas por cobrar	20/01/15	20/01/15		
• Módulo de sincronización	20/01/15	20/01/15		
• Módulo de usuarios	20/01/15	20/01/15		
• Pruebas	20/01/15	20/01/15		

Grafica 2: Cronograma

Marco tecnológico

La solución planteada utiliza los dispositivos móviles debido a que hoy en día el porcentaje de personas que utilizan los teléfonos inteligentes (Smartphone) está en crecimiento en Colombia; esto lo podemos constatar en el siguiente informe realizado por la Consultora Flurry Mobile con sede en Estados Unidos:



Gráfica 3: Países con mayor adopción en el uso de Smartphone

Como se observa en la *Gráfica 3*, Colombia es el país con mayor crecimiento en el uso de dispositivos móviles inteligentes a nivel mundial. Por consiguiente, se deduce que es muy viable que la solución planteada para este proyecto, esté diseñada y desarrollada para teléfonos inteligentes gracias al uso creciente que tienen los colombianos por esta tecnología; también observamos en el Estudio de Consumo Digital que está creciendo la conexión móvil.



Grafica 4: Estudio de Consumo Digital

Participación del mercado en sistemas operativos móviles en 2013

Los grandes dominadores del mercado actual es iOS y Android con cifras entre el 85% y 90% conjunta, lo cual se registró en el último trimestre del año 2012, actualmente Android es el número uno, por su parte Apple no inventó el Smartphone, ni las tiendas de aplicaciones, ni dispositivos táctiles pero si tuvo la genial idea de poner todas estas características en un Smartphone en el bolsillo.

Android: es un sistema operativo y una plataforma software basada en Linux, desarrollado por Android Inc. para teléfonos móviles, fue comprada por Google en el 2005, pero no fue sino hasta el 2008 que se dio a conocer, debido al proyecto Open Handset Alliance, consorcio formado por 48 empresas de desarrollo de hardware, software y telecomunicaciones que promocionan el software libre siendo Google el que ha contribuido con la mayor parte del código fuente, este sistema es desarrollado para smartphones, tabletas, portátiles, netbooks, Google TV, relojes de pulseras, auriculares y demás dispositivos. Cada una de las versiones de Android recibe el nombre de un postre en inglés. Actualmente, existen más de 400 mil aplicaciones de este sistema operativo.

Herramientas y tecnologías

- **PHP:** Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podrán incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.



Grafica 5: PHP

- **Laravel:** Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.



Grafica 6: Laravel

- **MySQL:** Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation

desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.



Grafica 7: MySQL

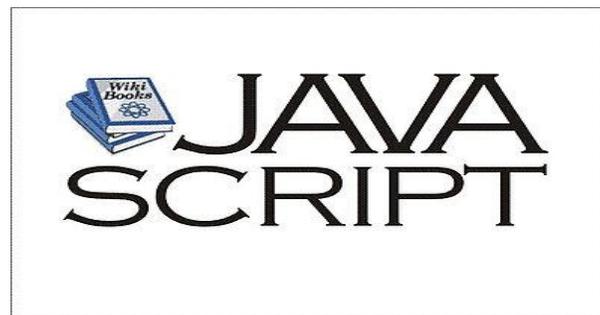
- **HTML5:** (HyperText Markup Language, versión 5) Es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas.



Grafica 8: HTML5

- **JavaScript:** Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador; aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en

documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.



Grafica 9: JavaScript

- **Backbone.js:** Es una herramienta de desarrollo/API para el lenguaje de programación JavaScript con una interfaz RESTful por JSON, basada en el paradigma de diseño de aplicaciones Modelo Vista Controlador. Está diseñada para desarrollar aplicaciones de una única página y para mantener las diferentes partes de las aplicaciones web sincronizadas.



Grafica 10: Backbone.js

- **Underscore.js:** Es una librería que ofrece un gran número de funciones realmente interesantes, en tan solo 4KB. Es muy probable que gran cantidad de las necesidades las resuelva, sin necesidad de perder tiempo desarrollando código que no tienen nada que ver con la lógica de negocio.

UNDERSCORE.JS

Grafica 11: Underscore.js

- **CSS3**: Hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "<style>".



Grafica 12: CSS3

- **Twitter Bootstrap:** Es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.



Grafica 13: Twitter Bootstrap

- **jQuery Mobile:** Es un Framework optimizado para dispositivos táctiles (también conocido como Framework móvil) que está siendo desarrollado actualmente por el equipo de proyectos de jQuery. El desarrollo se centra en la creación de un Framework compatible con la gran variedad de smartphones y tabletas, algo necesario en el creciente, pero heterogéneo mercado de tabletas y smartphones. El Framework de jQuery Mobile es compatible con otros frameworks móviles y plataformas como PhoneGap y Worklight entre otros.



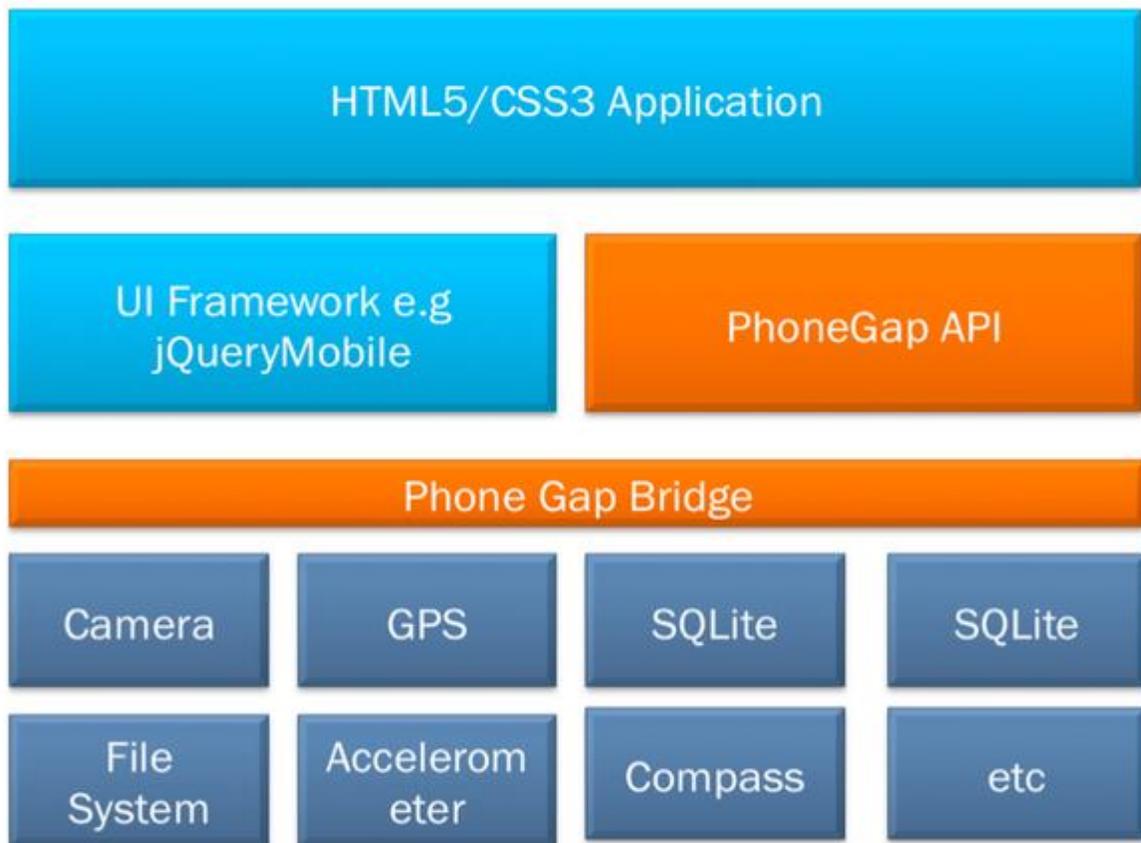
Grafica 14: jQuery Mobile

- **Web SQL Database:** Es una página web de API para almacenar datos en bases de datos que se pueden consultar mediante una variante de SQL. El API es apoyado por Google Chrome, Opera, Safari y el navegador de Android.



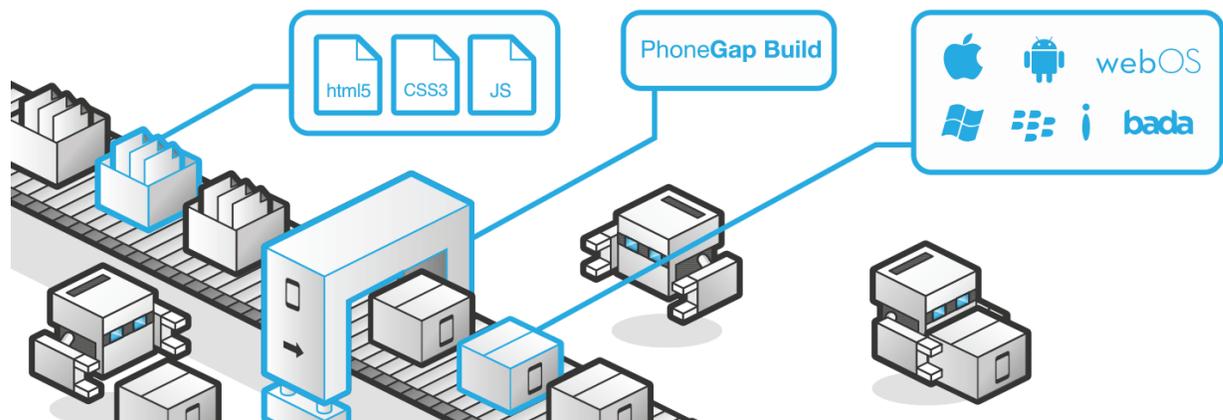
Grafica 15: Web SQL Database

- **Apache Cordova:** Es un conjunto de APIs de dispositivos que permiten a un desarrollador de aplicaciones móviles acceder a las funciones nativas como la cámara o acelerómetro de JavaScript. Combinando con un framework como jQuery Mobile o Dojo Mobile o Sencha Touch, esto permite una aplicación para teléfonos inteligentes desarrollada con sólo HTML, CSS y JavaScript.



Grafica 16: Estructura de una aplicación realizada con Apache Cordova

- **JDK:** Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red. En la unidad de red se pueden tener las herramientas distribuidas en varias computadoras y trabajar como una sola aplicación.
- **Android SDK:** Incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales.
- **Eclipse IDE:** Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.
- **Adobe PhoneGap Build:** Es un servicio que permite crear aplicaciones para dispositivos móviles con tecnologías web estándar (HTML, CSS, JavaScript) y, posteriormente, compilarlas para varias plataformas desde la nube.



Grafica 17: Adobe PhoneGap Build

Consideraciones tecnológicas

La aplicación se realizará con un modelo de prototipo ágil siguiendo la metodología de Scrum, ya que nos permite iterar sobre varias versiones de la aplicación validando directamente con el cliente.

En la *Gráfica 18*, se muestra un comparativo con diferentes criterios relevantes para el desarrollo de la aplicación, comparando el modelo de desarrollo web, nativo e híbrido. La consideración de cada criterio es obtenido gracias a la suma del debate de varios expertos del tema, libros, conferencias, talleres y nuestra propia experiencia como desarrolladores.

	Desarrollo Web	Desarrollo Nativo	Desarrollo Híbrido
Compatibilidad	Compatibilidad con la mayoría de sistemas operativos móviles. Dependiendo del navegador del dispositivo.	100% de compatibilidad con el dispositivo ya que es desarrollado específicamente para el SO.	Compatibilidad parcial con el dispositivo, se hacen ajustes para garantizar el completo funcionamiento.
Usuarios	Se puede llegar a mas número de usuarios por su despliegue a los demás sistemas operativos	Para llegar a mas usuarios de debe realizar un desarrollo para cada SO diferente	Se puede llegar a la mayoría de usuarios de diferentes Sistemas Operativos
Conectividad	Solo trabaja Online	Puede trabajar de forma Online y Offline	Puede trabajar de forma Online y Offline
API	Solo tiene acceso a algunas las funciones nativas que el dispositivo brinde como , cámara, video, microfono etc.	Tiene acceso a todas las funciones nativas que el dispositivo brinde como , cámara, agenda, vibración etc.	Tiene acceso a todas las funciones nativas que el dispositivo brinde como , cámara, agenda, vibración etc.
Proyecto	Un solo proyecto de código que es generado para cualquier sistema operativo móvil.	Varios proyectos de código que es generado para cualquier sistema operativo móvil.	Un solo proyecto de código que es generado para cualquier sistema operativo móvil.
Distribución	No hay tiendas la seguridad dependera del servidor donde este alojada la aplicación	Hay tiendas oficiales y seguras para distribuir y descargar la aplicación	Hay tiendas oficiales y seguras para distribuir y descargar la aplicación
Aplicación	El acceso a la aplicación es desde un navegador web	El acceso a la aplicación es desde las herramientas de interfaz del sistema operativo	El acceso a la aplicación es desde un WebView nativo del sistema operativo
Actualizaciones	Las actualizaciones a la aplicación de hacen una solo vez y les llega a todos los usuarios.	Las actualizaciones a la aplicación de hacen una solo vez y les llega a todos los usuarios, pero dependera del los permisos del usuario y la tienda de distribución	Las actualizaciones a la aplicación de hacen una solo vez y les llega a todos los usuarios, pero dependera del los permisos del usuario y la tienda de distribución
Aprendizaje	La curva de aprendizaje suele ser menor comparada con otros lenguajes	La curva de aprendizaje suele ser mayor comparada con otros lenguajes y además se sumaria el aprendizaje de cada SO	La curva de aprendizaje suele ser menor comparada con otros lenguajes
Mantenimiento	Los costos de mantenimiento son menores debido a que es solo un proyecto el cual hay que mantener	Los costos de mantenimiento son más altos debido a que son varios proyectos los que hay que mantener	Los costos de mantenimiento son menores debido a que es solo un proyecto el cual hay que mantener
Rendimiento	El rendimiento es mucho menor debido a la versión del navegador y conexión a internet	El rendimiento es mas rapido debido a que el sistema operativo esta interpretado lenguaje nativo	El rendimiento es menor ya que depende de la la ejecución del WebView

Gráfica 18: Consideraciones tecnológicas

ANÁLISIS Y DISEÑO

En esta etapa del proyecto se levantan los requerimientos, después se analizan para brindar una solución de acuerdo a la información recogida. Posteriormente, se describe la solución planteada para cumplir con los requerimientos recolectados; en esta descripción se definen diagramas de cómo será la interacción del cliente y el sistema, cuáles con los modelos de información para guardar la información, cuál es la arquitectura que se debe desarrollar, el diseño del sistema y sus componentes.

Definición de requerimientos.

Módulo de ventas

Objetivo del módulo de ventas

Gestionar todas las ventas que se generen desde el negocio y que sean parte de los ingresos a la tienda.

Requerimientos funcionales

1. El sistema debe permitir registrar las ventas de los productos que estén en el inventario de la tienda.
2. El sistema debe permitir agregar uno o más productos al registro de la venta.
3. El sistema debe descontar sistemáticamente del inventario de la tienda la cantidad de cada producto luego de finalizada la venta
4. El sistema debe permitir la búsqueda por nombre y código de barras del producto al momento de agregarlo a la venta.
5. El sistema debe ir actualizando el total de la venta cada vez que haya un cambio en los productos de la venta como eliminar, actualizar o modificar un producto de la venta
6. El sistema debe permitir seleccionar el precio de venta del producto que se va a añadir a la venta.
7. El sistema al finalizar la venta debe mostrar los método de pago.
8. El sistema al finalizar la venta debe permitir el ingreso del pago en efectivo y mostrar el cálculo del cambio.
9. El sistema debe registrar la venta en las deudas del cliente en caso que el método de pago sea "Pendiente de pago".
10. El sistema debe ir mostrando la lista de productos que se van añadiendo a la venta.

11. El sistema al finalizar la venta debe permitir ingresar un valor de descuento a la venta.
12. El sistema debe permitir agregar la información de cliente nuevo sin salirse de la venta actual.

Requerimientos no funcionales

1. El sistema debe permitir al momento de hacer la búsqueda de un producto, auto completar la información relacionada.
2. El sistema debe mostrar cada vez que se agrega un producto una ventana emergente con los campos para agregarlo a la venta.
3. El sistema debe permitir cancelar la venta.

Requerimientos de información

1. El sistema para agregar un producto debe mostrar los campos de: código de barras, nombre del producto, cantidad a solicitar, cantidad disponible y la lista de precios sugeridos.
2. El sistema debe mostrar la lista de productos añadidos a la venta la con la información de: nombre del producto, cantidad solicitada, precio unitario y el precio total.
3. El sistema al finalizar la venta debe mostrar los campos de: Total de venta, campo para registrar el pago en efectivo, campo que muestra el cambio o valor restante, lista de clientes, lista métodos de pago y el campo para el descuento.
4. El sistema debe mostrar al agregar un cliente la siguiente información: Identificación, nombre, apellido, fecha de nacimiento, teléfono, celular, lista de géneros, lista de clasificación del cliente.

Requerimientos de restricción

1. El sistema no debe permitir el ingreso de una cantidad mayor a la que se encuentra en el inventario.
2. El sistema no debe permitir finalizar la venta si el cálculo del cambio es menor a 0.
3. El sistema debe solicitar como campo obligatorio el cliente si se selecciona como método de pago: "Pendiente de Pago"
4. El sistema de no debe permitir ingresar valores menores a 0 en los campos de descuento y el pago en efectivo.
5. El sistema al momento de crear un cliente debe colocar como obligatorios los campos de identificación, nombre y apellido.
- 6.

Módulo de compras

Objetivo del módulo de caja

Debe permitir el cierre de caja de cada sesión para el control del flujo de caja de la tienda.

Requerimientos funcionales

1. El sistema debe recordar al usuario abrir caja con un monto antes de iniciar operaciones en la aplicación.
2. El sistema debe permitir abrir los detalles de la caja luego de validar si esta ya fue abierta.
3. El sistema debe calcular en total de ventas en el día actual.
4. El sistema debe calcular en total de compras en el día actual.
5. El sistema debe calcular la caja esperada en base al total de la compra y al total de la venta.
6. El sistema debe cerrar la sesión de usuario que es autenticado una vez se cierre caja.
7. El sistema debe mostrar un reporte donde agrupe el valor total de ventas por cada método de pago.
8. El sistema debe mostrar un reporte donde agrupe el valor total de compras por cada método de pago.

Requerimientos no funcionales

1. El sistema debe mostrar la alarma para iniciar caja automáticamente, por medio de una ventana emergente.
2. El sistema debe mostrar un aviso que indique que al cerrar caja se cierra la sesión del usuario.
3. El sistema debe mostrar un aviso de confirmación ante de cerrar la caja.

Requerimientos de información

1. El sistema debe mostrar un campo para el inicio de caja.

Requerimientos de restricción

1. El sistema no debe permitir valores menores a 0 en el inicio de caja o en el monto de la caja real.

Módulo de inventario

Objetivo del módulo de inventario

Gestionar todos los productos de la tienda, basados en una base maestra de productos

Requerimientos funcionales

1. El sistema debe mostrar una lista de los productos que se tienen en el inventario.
2. El sistema debe permitir la búsqueda de productos por fabricantes y nombre del producto.
3. El sistema debe crear nuevos productos basados en la información general de una base maestra de productos.
4. El sistema debe gestionar la información de los productos.
5. El sistema debe permitir organizar de menor a mayor por la mínima cantidad en inventario

Requerimientos no funcionales

1. El sistema de debe hacer un carga de máximo ocho productos en la interfaz y a medida que vaya bajando con el scroll ir cargando dinámicamente los ocho productos siguientes.
2. El sistema debe permitir auto completar la información del producto si está en la base maestra de productos.

Requerimientos de información

1. El sistema debe mostrar en la lista de productos la información de: Nombre, Cantidad, Mínima cantidad y Precio Unitario.
2. El sistema al agregar un producto debe mostrar la siguiente información: Código de barras, Nombre, Fabricante, Marca, 5 opciones de precio, la cantidad en el inventario, la cantidad mínima, la cantidad máxima, una lista del estado del producto y un campo para el valor del impuesto.

Requerimientos de restricción

1. El sistema no debe permitir el ingreso de valores menores a 0 en los campos de precio.

2. El sistema no debe permitir la creación de un producto que ya se encuentre registrado en el inventario de la tienda.
3. El sistema no debe permitir la edición de los campos de la base maestra después de haber sido creado el producto.

Módulo de compras

Objetivo del módulo de compras

Gestionar todas las compras que se generen desde el negocio y que sean parte de los egresos de la tienda.

Requerimientos funcionales

1. El sistema debe permitir agregar un producto a la compra y automáticamente debe registrarse en el inventario.
2. El sistema debe permitir el ingreso de tres sugerencias de precio.
3. El sistema debe calcular el primer precio sugerido en base al total de la compra con la cantidad de productos comprados.
4. El sistema debe aumentar en el inventario la cantidad de unidades compradas del producto.
5. El sistema debe permitir agregar un nuevo producto relacionada a la base maestra de productos.
6. El sistema debe autocompletar toda la información relacionada al producto.
7. El sistema debe mostrar la lista de productos añadidos a la compra.
8. El sistema debe actualizar el monto total de la compra cuando se agregue, edite o elimine un producto de la compra actual.
9. El sistema al finalizar la compra debe mostrar una ventana emergente con la información de cierre de la compra.
10. El sistema debe permitir agregar un nuevo proveedor sin necesidad de salirse de la compra actual.
11. El sistema debe registrar como deuda para la tienda si la compra está registrada como: "Pendiente de pago".

Requerimientos no funcionales

1. El sistema al agregar un proveedor debe colocar un formato automático que corresponda a un NIT (Número de identificación tributaria).
2. El sistema al finalizar la compra debe mostrar una ventana emergente con la información para el cierre de la compra.

Requerimientos de información

1. El sistema debe mostrar para crear un producto a la compra la siguiente información: Código de barra, Nombre, Fabricante, Marca, Cantidad, la cantidad disponible en inventario, la cantidad mínima, total de la compra, porcentaje de IVA y tres opciones de precio sugerido.
2. El sistema debe mostrar al finalizar la compra el número de la factura, lista de proveedores, lista de estados de la compra, lista de promociones de la compra y la lista de medios de pago.
3. El sistema debe mostrar al agregar un proveedor la siguiente información: Tipo de documento, NIT (Número de identificación tributaria), nombre, dirección y teléfono.

Requerimientos de restricción

1. El sistema no debe permitir valores menores a 0 en los campos de IVA, precios, cantidad y total de la compra.
2. El sistema al agregar un nuevo proveedor debe colocar como obligatorio el NIT (Número de identificación tributaria) y el nombre.

Módulo de reportes

Objetivo del módulo de reportes

Mostrar al administrador de la tienda la información de estadísticas de compra y venta de productos.

Requerimientos funcionales

1. El sistema debe mostrar un listado de los productos que más se han vendido en la tienda. (Mayor rotación)
2. El sistema debe mostrar un listado de los productos que menos se han vendido en la tienda. (Menor rotación)
3. El sistema debe mostrar un listado de los productos de los cuales genera más ganancia en base al precio de compra y al precio de venta. (Mayor ganancia)
4. El sistema debe mostrar un listado de los productos de los cuales generan menor ganancia en base al precio de compra y al precio de venta de los productos de la tienda. (Menor ganancia)
5. El sistema debe generar un reporte que muestre la suma total en dinero que se tiene en inventario actual.
6. El sistema debe generar un reporte de ventas donde muestre el total de ventas de los tres últimos meses.
7. El sistema debe generar un reporte de compras donde muestre el total de compras de los tres últimos meses.
8. El sistema debe generar un reporte de ganancias donde muestre el total de ganancias de los tres últimos meses.

Requerimientos no funcionales

1. El sistema debe mostrar los reportes de mayor y menor rotación, mayor y menor ganancia en una página.
2. El sistema debe mostrar una consolidada del reporte de tres meses atrás en una página.

Módulo de cuentas por pagar

Objetivo del módulo de cuentas por pagar

Gestiona todas las cuentas por pagar de la tienda.

Requerimientos funcionales

1. El sistema debe mostrar una lista agrupada por proveedor de las cuentas por pagar pendientes del negocio.
2. El sistema debe mostrar la suma total de las cuentas por pagar del negocio.
3. El sistema debe permitir mostrar el detalle de cada deuda a cada proveedor mostrando la lista de productos comprados.
4. El sistema debe permitir abonar un saldo a una cuenta de un proveedor en específico.
5. El sistema debe mostrar el total de la cuenta por pagar de acuerdo a proveedor específico.
6. El sistema debe descontar automáticamente el valor del abono al total de la cuenta por pagar con el proveedor seleccionado.

Requerimientos no funcionales

1. El sistema debe mostrar en una ventana emergente el valor del monto a abonar.

Requerimientos de información

1. El sistema debe mostrar en la lista de cuentas por pagar el nombre del proveedor y el monto total de la deuda.
2. El sistema debe mostrar el detalle de la cuenta por pagar con el proveedor la fecha, el nombre del proveedor y el monto total.

Requerimientos de restricción.

1. El sistema debe sólo tener en cuenta las deudas pendientes y no mostrar las deudas ya canceladas.
2. El sistema no debe permitir ingresar valores menores a 0 en el monto a abonar.

Módulo de cuentas por cobrar

Objetivo del módulo de cuentas por cobrar

Gestiona todas las cuentas por cobrar de la tienda.

Requerimientos Funcionales

1. El sistema debe mostrar una lista agrupada por cliente de las cuentas por cobrar pendientes de la tienda.
2. El sistema debe mostrar la suma total de las cuentas por cobrar de la tienda.
3. El sistema debe permitir mostrar el detalle de cada cliente mostrando la lista de productos vendidos.
4. El sistema debe permitir abonar un saldo a una cuenta de un cliente en específico.
5. El sistema debe mostrar el total de la cuenta por cobrar de acuerdo a cliente específico.
6. El sistema debe descontar automáticamente el valor del abono al total de la cuenta por cobrar del cliente.

Requerimientos no funcionales

1. El sistema debe mostrar en una ventana emergente el valor del monto a abonar.

Requerimientos de información.

1. El sistema de mostrar en la lista de cuentas por cobrar el nombre del cliente y el monto total por cobrar.
2. El sistema debe mostrar en el detalle de la cuenta por cobrar con el cliente mostrando la fecha, el nombre del cliente y el monto total.

Requerimientos de restricción.

1. El sistema debe sólo tener en cuenta las cuentas pendientes y no mostrar las cuentas ya canceladas.
2. El sistema no debe permitir ingresar valores menores a 0 en el monto a abonar.

Módulo de sincronización

Objetivo del Módulo de sincronizar

Gestiona toda la información para la aplicación cuando exista o no conexión a internet.

Requerimientos funcionales

1. El sistema debe permitir la descarga de datos actualizados y nuevos para que se sincronicen con la aplicación móvil.
2. El sistema debe permitir la subida de datos al servidor web de toda la información relacionada con la tienda.
3. El sistema debe permitir trabajar cuando no exista conexión a internet para poder gestionar los datos y luego subirlos al servidor web cuando exista la conexión.
4. El sistema debe mostrar un aviso el cual determina qué tipo de conexión a Internet tiene el dispositivo y si no hay conexión.

Requerimientos no funcionales

1. El sistema debe mostrar en una ventana emergente con un campo para digitar el valor del monto a abonar.
2. El sistema debe mostrar un validador visual cuando ya haya terminado de descargar todos los datos al dispositivo.
3. El sistema debe mostrar un mensaje cuando haya terminado de subir todos los datos al servidor web.

Requerimientos de información

1. El sistema de mostrar una lista con el nombre de la entidad y el número de registros obtenidos.

Requerimientos de restricción

1. El sistema no debe permitir volver a sincronizar datos si ya hay un proceso de sincronización en ejecución.

Módulo de usuarios

Objetivo del módulo de usuarios

Gestiona los usuarios que ingresan a la aplicación

Requerimientos funcionales

1. El sistema debe permitir ingresar al sistema cuando exista conexión a internet así mismo cuando no exista la conexión.
2. El sistema debe sincronizar los usuarios creados desde la aplicación.
3. El sistema debe pedir un cambio de clave si es la primera vez que ingresa al sistema.
4. El sistema debe permitir un método de recuperar la contraseña en caso de pérdida.
5. El sistema debe mostrar una lista de los usuarios que pertenecen al negocio.

Requerimientos de información.

1. El sistema de mostrar una lista con el nombre y el usuario de la aplicación.
2. El sistema debe mostrar la siguiente información del usuario: E-mail, contraseña, identificación, nombre, apellido, fecha de nacimiento, teléfono, celular y lista de géneros.

Requerimientos de restricción.

1. El sistema debe pedir los siguientes campos como obligatorios: E-mail, contraseña, identificación, nombre y apellido.

Descripción del sistema propuesto

La solución tecnológica que se ha propuesto se basa principalmente en una aplicación móvil para tabletas la cual les permite a los tenderos registrar la información de sus ingresos, egresos y consulta de inventario de la tienda.

La aplicación móvil se desarrolla para el sistema operativo Android utilizando las tecnologías de HTML5 como JavaScript, CSS3 y claramente HTML5; también utilizando el proyecto libre de Apache Cordova.

La aplicación móvil se compone de los siguientes módulos:

Módulo de Ventas

Gestiona todas las ventas que se generen desde el negocio y que sean parte de los ingresos a la tienda.

Módulo de Caja

Permite el cierre de caja de cada sesión para el control del flujo de caja de la tienda.

Módulo de Inventario

Gestiona todos los productos de la tienda, basados en una base maestra de productos.

Módulo de Compras

Gestiona todas las compras que se generen desde el negocio y que sean parte de los egresos de la tienda.

Módulo de Reportes

Muestra al tendero estadísticas de compra y venta de productos.

Módulo de Cuentas por Pagar

Gestiona todas las cuentas por pagar del negocio.

Módulo de Cuentas por Cobrar

Gestiona todas las cuentas por cobrar del negocio.

Módulo de Sincronización

Gestiona toda la información para el funcionamiento de la aplicación en modo online y offline.

Módulo de Usuarios

Gestiona los usuarios que ingresan a la aplicación.

Para la persistencia e integridad de la información se ha utilizado la arquitectura API REST, la cual brinda gran escalabilidad y se presta para el desarrollo de múltiples servicios y sistemas de información consumiendo los datos concentrados en la Base de Datos basada en tecnología Cloud Computing.

Para el desarrollo del Sistema de Información y de la API REST se ha utilizado el framework Laravel basado en el lenguaje PHP propicio para el desarrollo web.

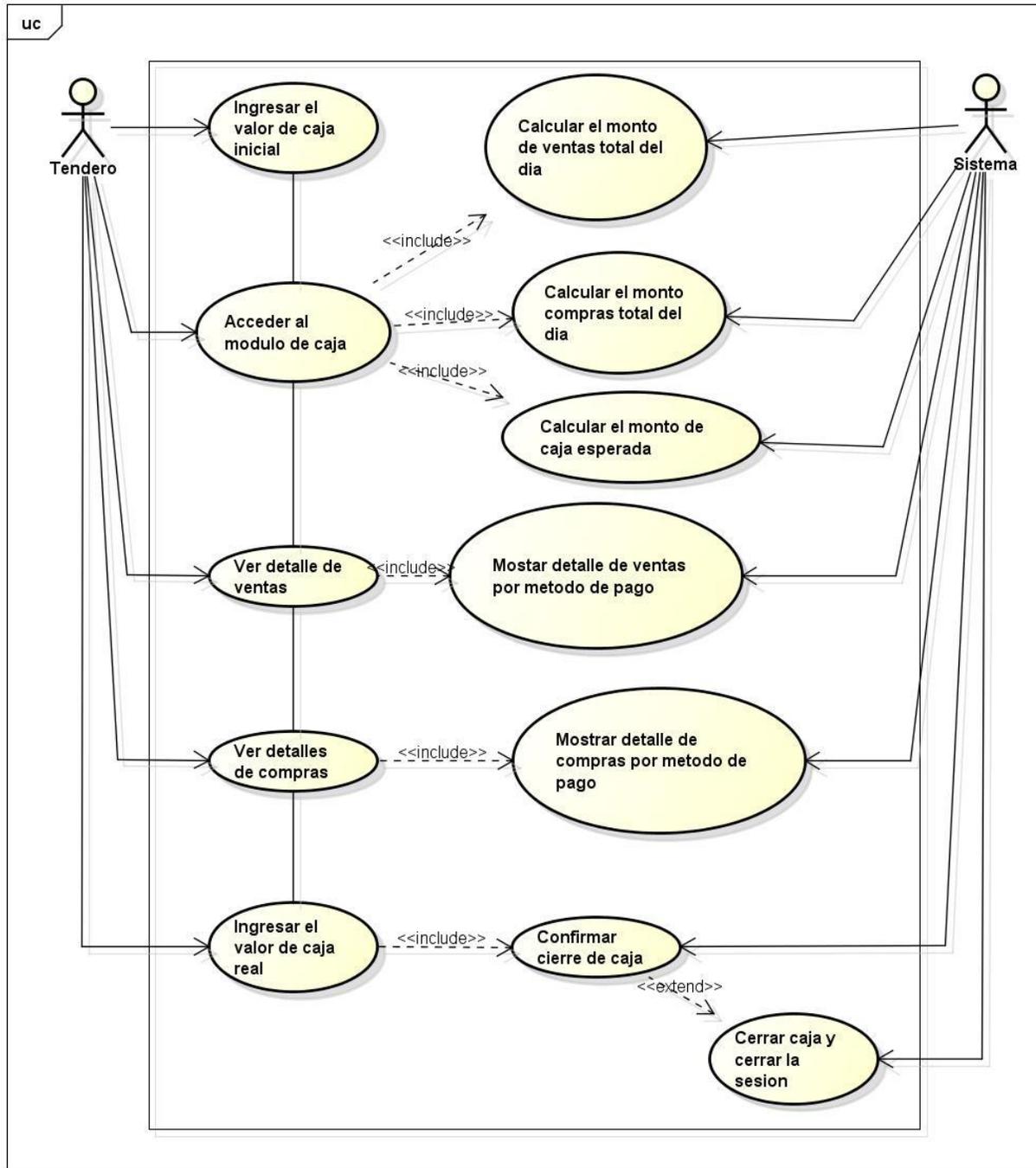
Diseño del sistema propuesto

Diagrama de Casos de Uso

Módulo Caja

Nombre del caso de uso: Registro de Caja	
Autor: John Tamayo y Nicolas Molina	Fecha: 15 de noviembre del 2014
Descripción: Diagrama de caso de uso que muestra el proceso para abrir y cerrar la caja del día	
Autores: Tendero y Sistema	
Precondiciones: Tener sincronizados los datos y estar autenticado en la aplicacion	
Flujo Normal: <ol style="list-style-type: none">1. Ingresar el valor de la caja inicial (Tendero)2. Acceder al modulo de cierre de caja (Tendero)<ol style="list-style-type: none">2.1 Calcular el monto de ventas total del dia (Sistema)2.2 Calcular el monto de compras total del dia (Sistema)2.3 Calcular el monto de caja esperada (Sistema)3. Ver detalle de ventas (Tendero)<ol style="list-style-type: none">3.1 Mostrar detalle de ventas por metodo de pago (Sistema)4. Ver detalle de compras (Tendero)<ol style="list-style-type: none">4.1 Mostrar detalle de compras por metodo de pago (Sistema)5. Ingresar el valor de cierre de caja (Tendero)<ol style="list-style-type: none">5.1 Confirmar el cierre de caja (Sistema)5.2 Cerrar caja y cerrar sesion del usuario (Sistema)	
Flujo Alerno: <ol style="list-style-type: none">2. Ingresar valores menores a cero5. Ingresar valores menores a cero	
Poscondiciones: El usuario queda con la sesion cerrada y el cierre de caja registrado	

Diagrama 1: Hoja de caso de uso (Modulo de Caja)



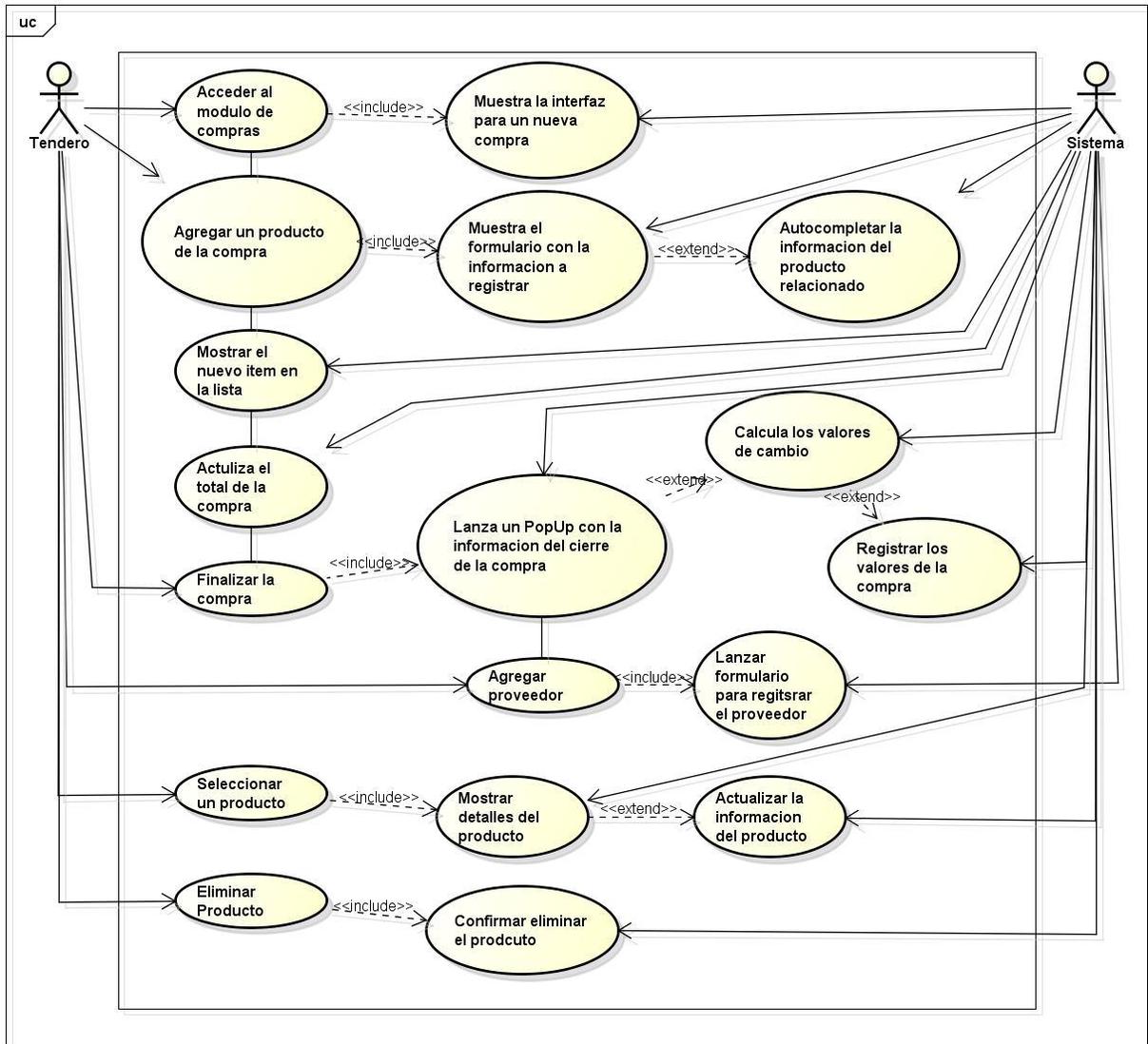
powered by Astah

Diagrama 2: Caso de Uso (Modulo de Caja)

Módulo de Compras

Nombre del caso de uso: Registro de Compras	
Autor: John Tamayo y Nicolas Molina	Fecha: 15 de noviembre del 2014
Descripción: Diagrama de caso de uso que muestra el proceso de registro de productos a una compra nueva	
Autores: Tendero y Sistema	
Precondiciones: Tener sincronizados los datos y estar autenticado en la aplicacion	
Flujo Normal: <ol style="list-style-type: none">1. Acceder al modulo de compras(Tendero)<ol style="list-style-type: none">1.1 Muestra la interfaz para una nueva compra(Sistema)2. Agregar un producto a la compra (Tendero)<ol style="list-style-type: none">2.1 Muestra el formulario con la informacion a registrar(Sistema)2.2 Autocompleta la informacion del producto relacionado(Sistema)3. Mostrar los cambios en la lista de productos (Sistema)4. Actuliza el total de la compra (Sistema)5. Finalizar Venta (Tendero)<ol style="list-style-type: none">5.1 Lanza un PopUp con la informacion del cierre de la compra(Sistema)5.2 Calcular los valores de cambio (Sistema)5.3 Registrar los valores de compra (Sistema)6. Agregar proveedor (Tendero)<ol style="list-style-type: none">6.1 Lanzar formulario para registrar proveedor (Sistema)7. Seleccionar un producto (Tendero)<ol style="list-style-type: none">7.1 Mostrar detalles del prodcuto(Sistema)7.2 Actualizar la informacion del prodcuto8. Elimnar Producto(Tendero)	
Flujo Alerno: <ol style="list-style-type: none">2. No se puede agregar un producto por su disponibilidad en el inventario5. No se llena la informacion requerida para finalizar la compra	
Poscondiciones: La compra queda registrada en la base de sistema con toda su informacion y lo deja la interfaz lista para seguir con otra compra nueva	

Diagrama 3: Hoja de Caso de Uso (Modulo de Compras)



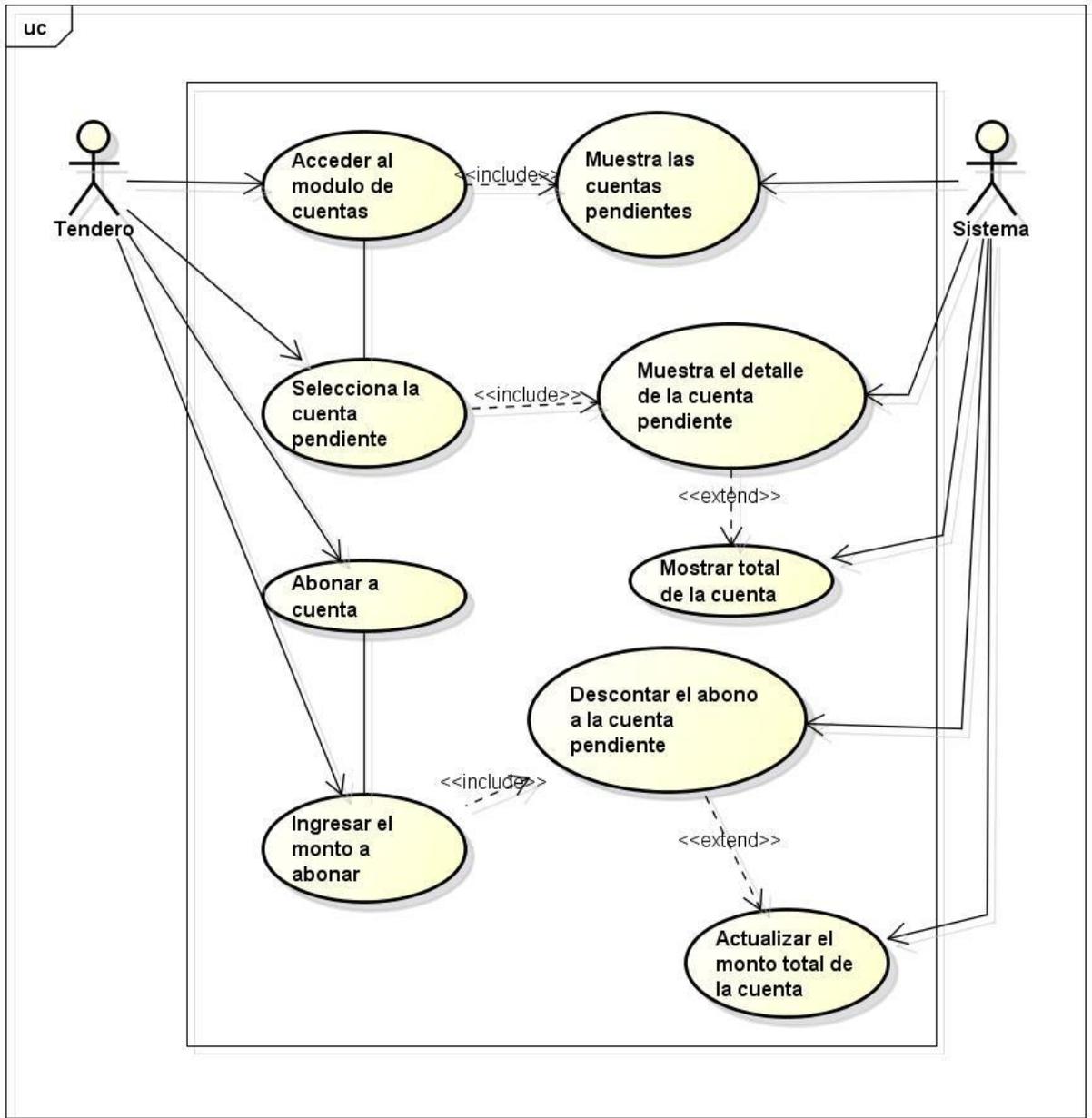
powered by Astah

Diagrama 4: Caso de Uso (Modulo de Compras)

Módulo de Cuentas

Nombre del caso de uso: Gestionar Cuentas pendientes	
Autor: John Tamayo y Nicolas Molina	Fecha: 15 de noviembre del 2014
Descripción: Diagrama de caso de uso que muestra los procesos para gestionar las cuentas pendientes	
Autores: Tendero y Sistema	
Precondiciones: Tener sincronizados los datos y estar autenticado en la aplicacion	
Flujo Normal: <ol style="list-style-type: none">1. Acceder al modulo de cuentas(Tendero)<ol style="list-style-type: none">1.1 Muestra las cuentas pendientes2. Seleccionar la cuenta pendiente(Tendero)<ol style="list-style-type: none">2.1 Muestra el detalle de la cuentas pendiente (Sistema)2.2 Muestra el total de la cuenta(Sistema)3. Abonar a cuenta (Tendero)4. Ingresar el monto a abonar (Tendero)<ol style="list-style-type: none">4.1 Descontar el abono a la cuenta pendiente(Sistema)4.2 Actualizar el monto total de la cuenta (Sistema)	
Flujo Alternativo: <ol style="list-style-type: none">4. Ingresar valores menores a 0	

Diagrama 5: Hoja de Caso de Uso (Modulo de Cuentas)



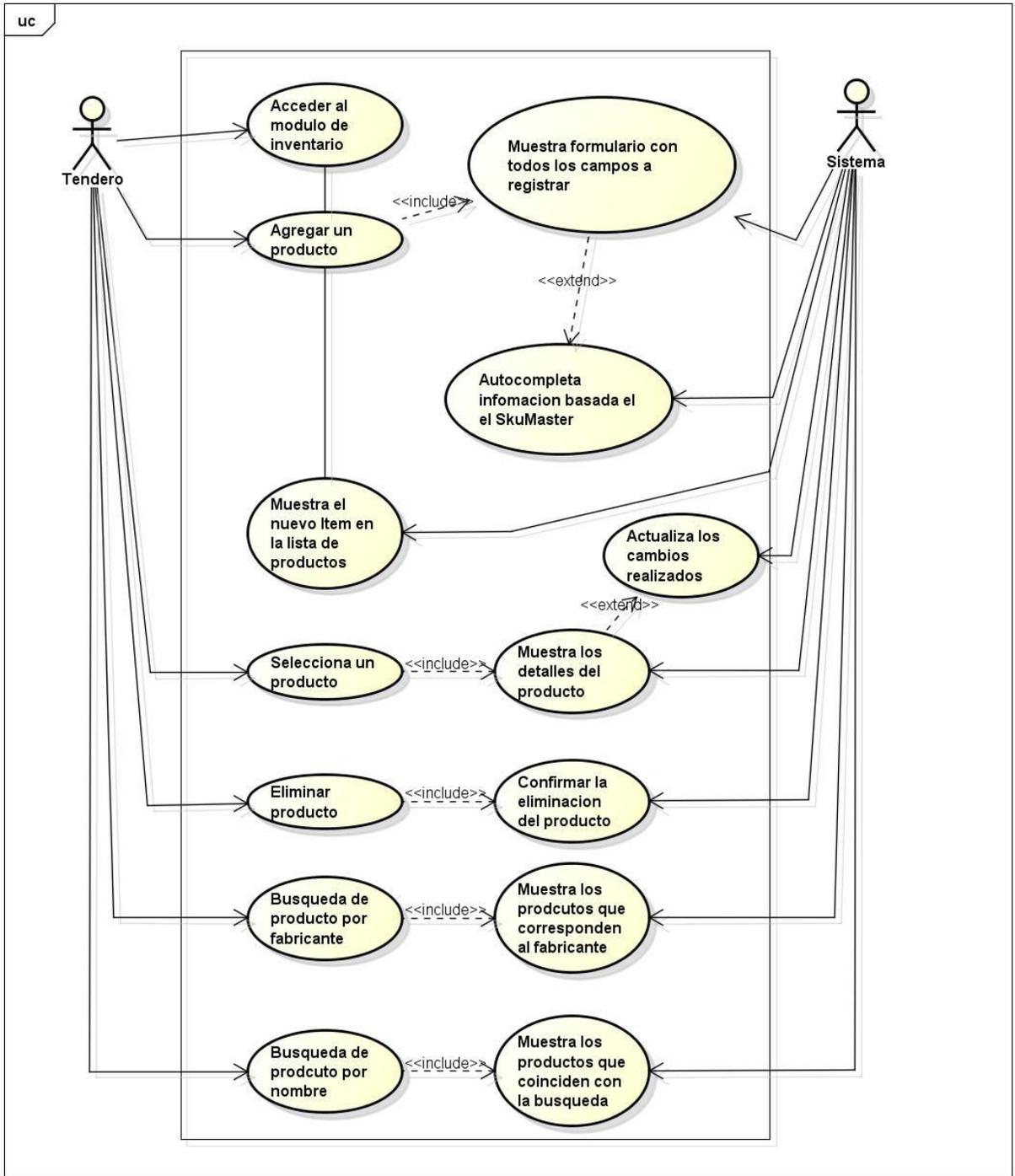
powered by Astah

Diagrama 6: Caso de Uso (Modulo de Cuentas)

Módulo de inventario

Nombre del caso de uso: Modificar el inventario	
Autor: John Tamayo y Nicolas Molina	Fecha: 15 de noviembre del 2014
Descripción: Diagrama de caso de uso que muestra los procesos para agregar, editar, buscar y eliminar un producto del inventario	
Autores: Tendero y Sistema	
Precondiciones: Tener sincronizados los datos y estar autenticado en la aplicacion	
Flujo Normal: <ol style="list-style-type: none">1. Acceder al modulo del inventario(Tendero)2. Agregar un producto (Tendero)<ol style="list-style-type: none">2.1 Muestra formulario con todos los campos a registrar (Sistema)2.2 Autocompleta la informacion basada en sku master (Sistema)3. Muestra el nuevo item en la lista de productos (Sistema)4. Selecciona un producto(Tendero)<ol style="list-style-type: none">4.1 Muestra los detalles del producto (Sistema)4.2 Actuliza los cambios realizados (Sistema)5.Eliminar un producto (Tendero)<ol style="list-style-type: none">5.1 Confirmar la eliminacion del producto (Sistema)6. Busqueda de producto por fabricante(Tendero)<ol style="list-style-type: none">6.1 Muestra los productos que corresponden al fabricante (Sistema)7. Busqueda del producto por nombre (Tendero)<ol style="list-style-type: none">7.1 Muestra los productos que coinciden con la busqueda (Sistema)	
Flujo Alerno: <ol style="list-style-type: none">2. Ingresar valores que no corresponden a los requisitos de restricciones4.2. Ingresar valores que no corresponden a los requisitos de restricciones7.1 Busqueda de un prodcuto que no existe	

Diagrama 7: Hoja de Caso de Uso (Modulo de inventario)



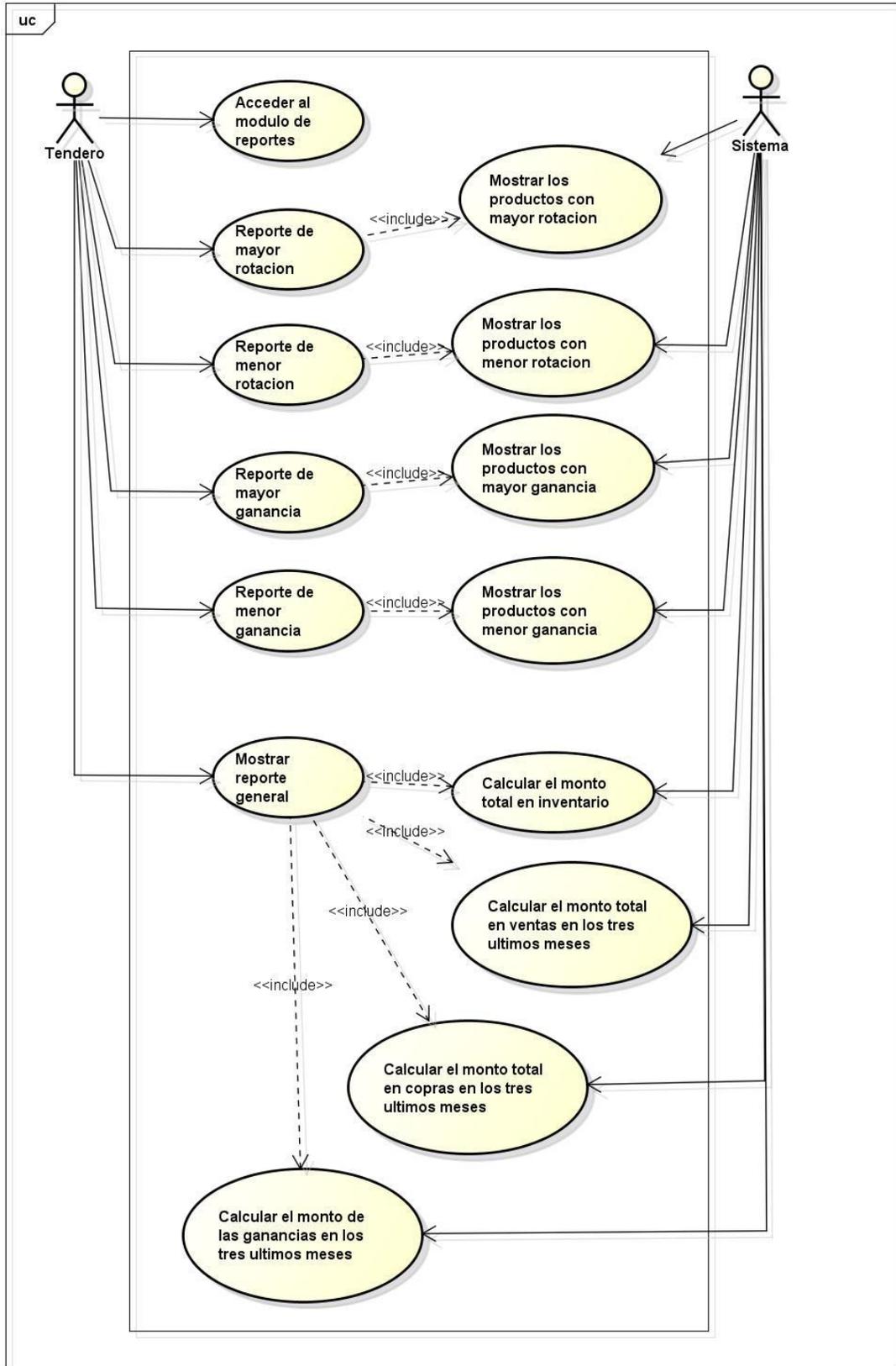
powered by Astah

Diagrama 8: Caso de uso de (Modulo de Inventario)

Módulo de Reportes

Nombre del caso de uso: Visualizar reportes	
Autor: John Tamayo y Nicolas Molina	Fecha: 15 de noviembre del 2014
Descripción: Diagrama de caso de uso que muestra los procesos para ver 4 tipos de reportes	
Autores: Tendero y Sistema	
Precondiciones: Tener sincronizados los datos y estar autenticado en la aplicacion	
Flujo Normal: <ol style="list-style-type: none">1. Acceder al modulo de reportes(Tendero)2. Reporte de mayor rotacion(Tendero)<ol style="list-style-type: none">2.1 Muestra los productos con mayor rotacion (Sistema)3. Reporte de menor rotacion (Sistema)<ol style="list-style-type: none">3.1 Muestra los productos con menor rotacion (Sistema)4. Reporte de mayor ganancia (Tendero)<ol style="list-style-type: none">4.1 Muestra los productos con mayor ganancia(Sistema)5. Reporte de menor ganancia (Tendero)<ol style="list-style-type: none">5.1 Muestra los productos con menor ganancia(Sistema)6. Mostrar reporte general(Tendero)<ol style="list-style-type: none">6.1 Calcular el monto total del inventario (Sistema)6.2 Calcular el monto total en ventas en los ultimos tres meses (Sistema)6.3 Calcular el monto total en compras en los ultimos tres meses(Sistema)6.4 Calcular el monto total en ganancias en los ultimos tres meses (Sistema)	

Diagrama 9: Hoja de Caso de Uso (Modulo de Reportes)



powered by Astah

Diagrama 10: Caso de Uso (Modulo de Reportes)

Módulo de Sincronización

Nombre del caso de uso: Sincronizar datos	
Autor: John Tamayo y Nicolas Molina	Fecha: 15 de noviembre del 2014
Descripción: Diagrama de caso de uso que muestra los procesos para sincronizar los datos	
Autores: Tendero y Sistema	
Precondiciones: Conexion a internet y estar autenticado en la aplicacion	
Flujo Normal: <ol style="list-style-type: none">1. Acceder al modulo de sincronizacion(Tendero)<ol style="list-style-type: none">1.1 Mostrar lista de entidades a sincronizar(Sistema)2. Subir Datos(Tendero)<ol style="list-style-type: none">2.1 Actualizar en la nube los cambios hechos en modo offline(Sistema)2.2 Mostrar mensaje de finalizacion(Sistema)3. Descargar Datos (Tendero)<ol style="list-style-type: none">3.1 Descargar los datos correspondientes al usuario (Sistema)3.2 Mostrar el progreso de la descarga de datos (Sistema)	
Flujo Alternativo: <ol style="list-style-type: none">1. No tiene acceso a internet3. Ya tiene datos descargados	
Poscondiciones: El usuario queda con todos los datos descargados para trabajar online y sube los datos para ser alojados y respaldados en la nube.	

Diagrama 11: Hoja de Caso de Uso (Modulo de Sincronización)

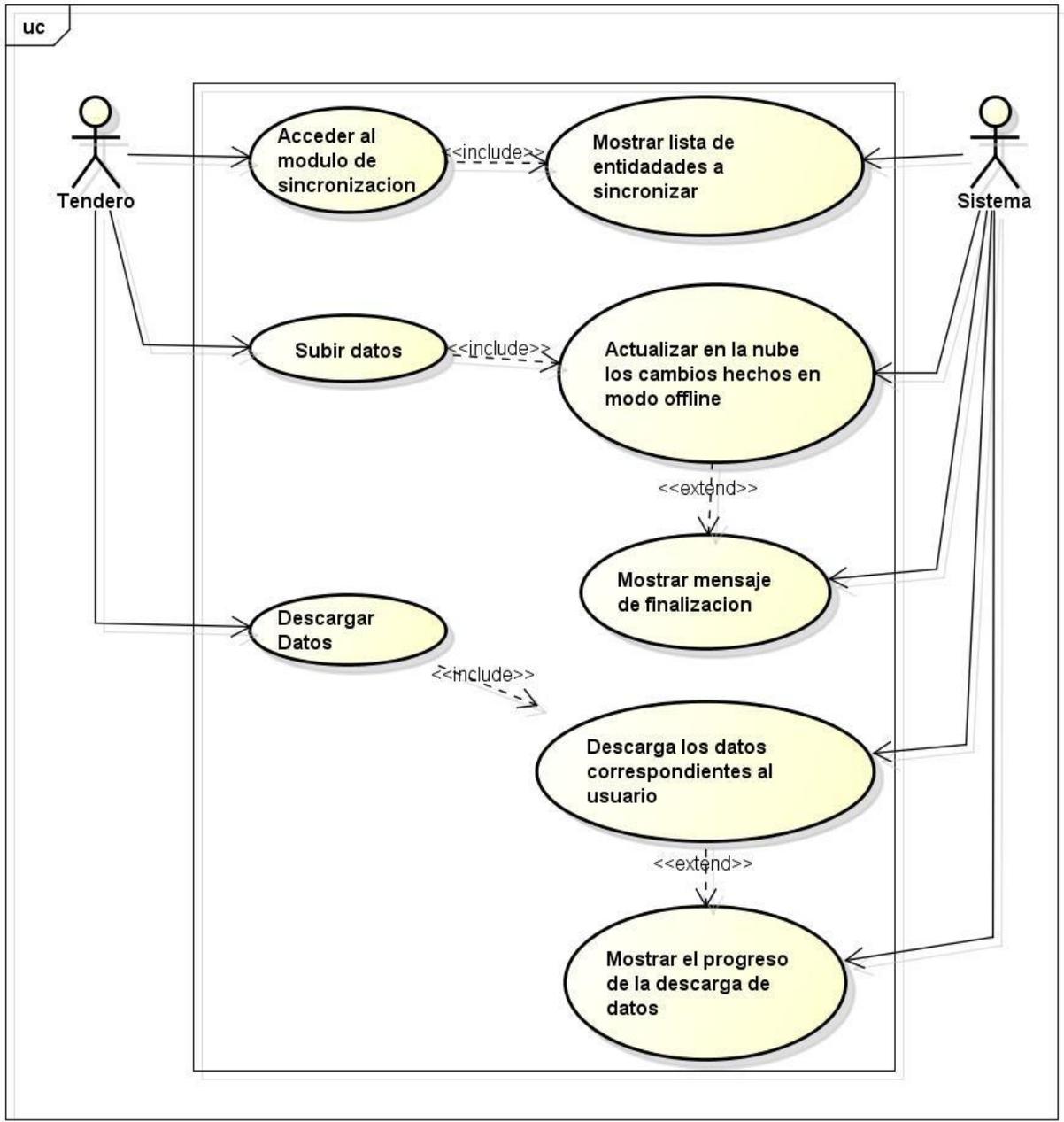


Diagrama 12: Caso de uso (Modulo de Sincronización)

Módulo de Usuarios

Nombre del caso de uso: Gestionar Usuarios	
Autor: John Tamayo y Nicolas Molina	Fecha: 15 de noviembre del 2014
Descripción: Diagrama de caso de uso que muestra los procesos para gestionar los usuarios	
Autores: Tendero y Sistema	
Precondiciones: Estar autenticado en la aplicacion	
Flujo Normal: <ol style="list-style-type: none">1. Acceder al modulo de usuarios (Tendero)<ol style="list-style-type: none">1.1 Mostrar lista de usuarios(Sistema)2. Agregar usuarios(Tendero)<ol style="list-style-type: none">2.1 Mostrar formulario con la informacion a registrar(Sistema)3. Mostrar nuevo usuario a la lista de usuarios(Sistema)4. Seleccionar usuario(Tendero)<ol style="list-style-type: none">4.1 Mostrar detales del usuario(Sistema)4.2 Actualizar los cambios realizados (Sistema)5. Eliminar usuario (Tendero)<ol style="list-style-type: none">5.1 Confirmar eliminacion (Sistema)5.2 Eliminar usuario del sistema (Sistema)6. Ingreso (Tendero)<ol style="list-style-type: none">6.1 Ingresar usuario y clave(Tendero)6.2 Validar acceso a la aplicacion (Sistema)	
Flujo Alterno: <ol style="list-style-type: none">6.2 No tiene permisos para acceder a la aplicacion	
Poscondiciones: Queda autenticado en la aplicacion	

Diagrama 13: Hoja de Caso de Uso (Modulo de Usuarios)

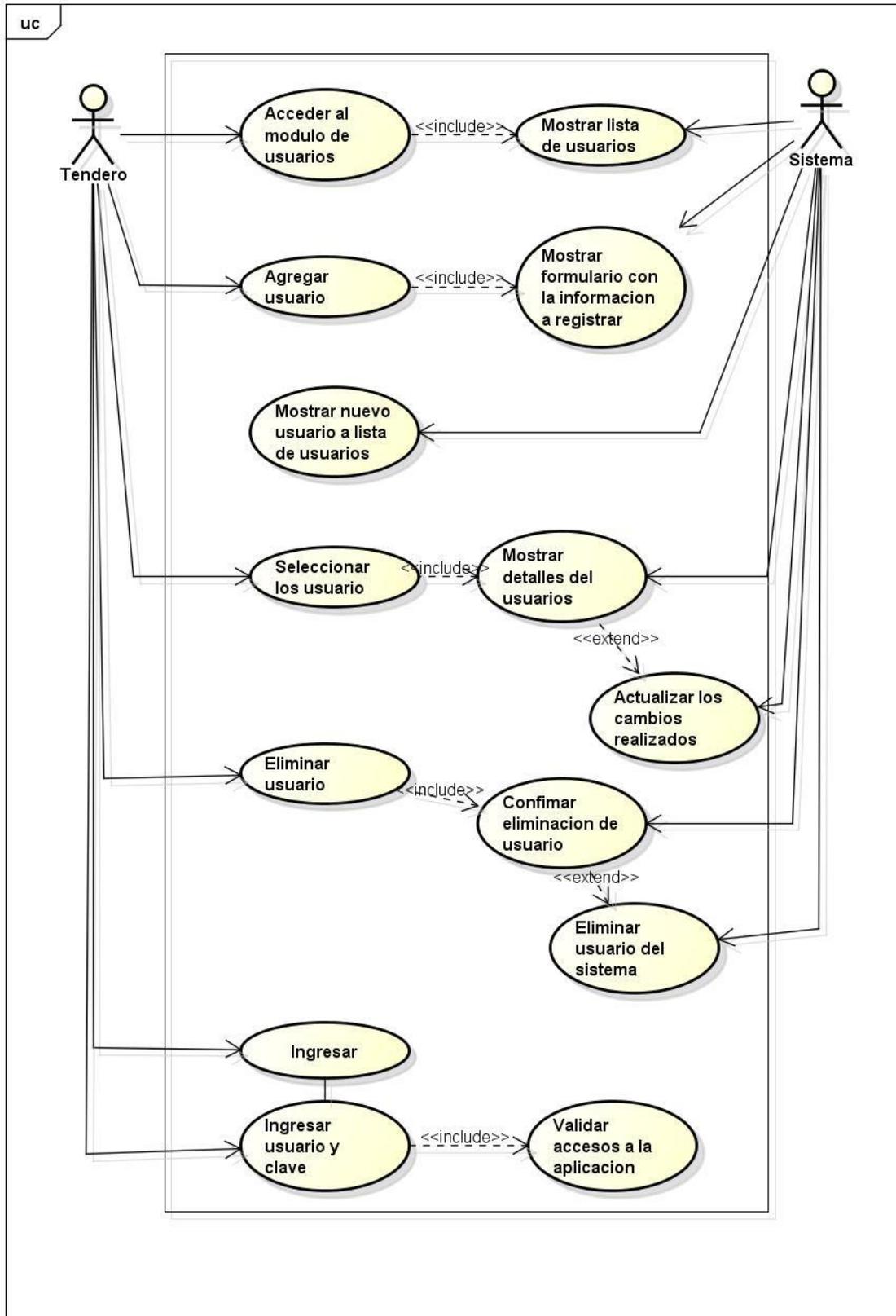
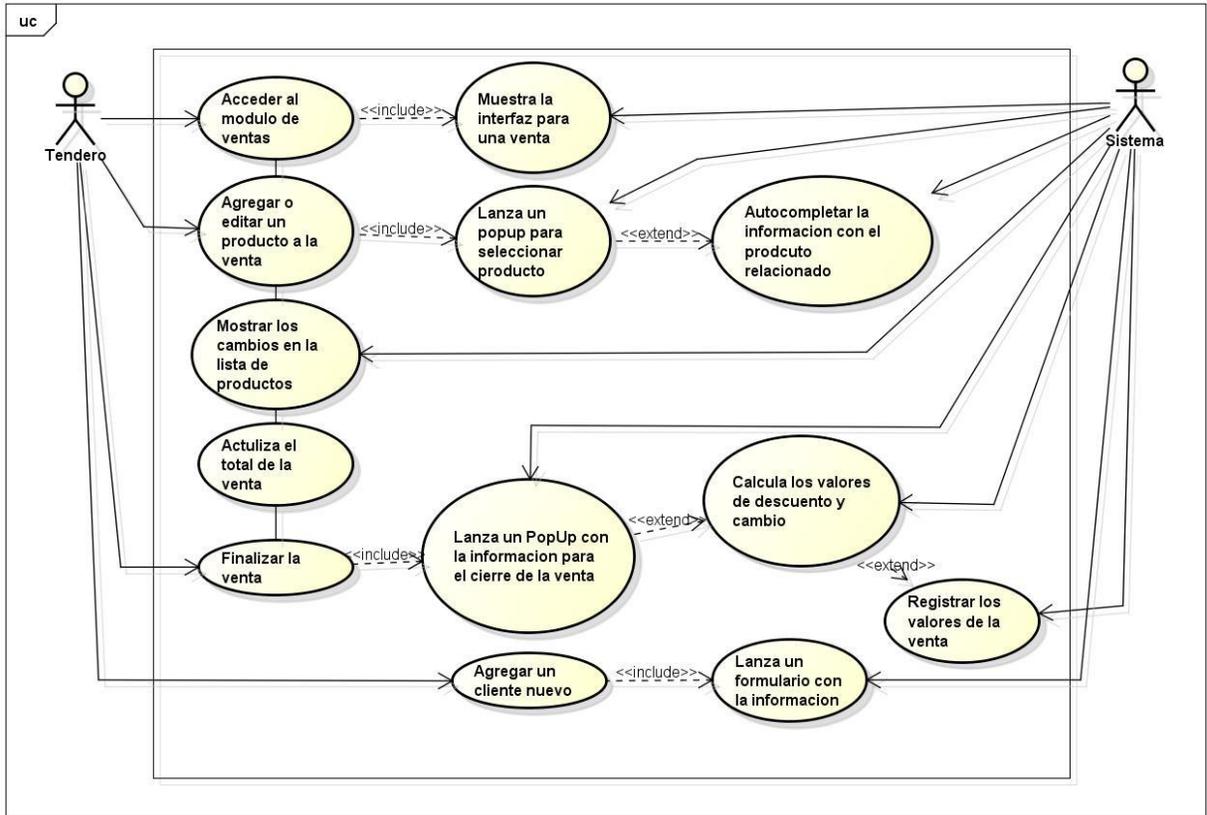


Diagrama 14: Caso de Uso (Modulo de Usuarios)

Módulo de Ventas

Nombre del caso de uso: Registro de Ventas	
Autor: John Tamayo y Nicolas Molina	Fecha: 15 de noviembre del 2014
Descripción: Diagrama de caso de uso que muestra el proceso de registro de productos a una venta nueva	
Autores: Tendero y Sistema	
Precondiciones: Tener sincronizados los datos y estar autenticado en la aplicación	
Flujo Normal: <ol style="list-style-type: none">1. Acceder al modulo de ventas (Tendero)<ol style="list-style-type: none">1.1 Muestra la interfaz para una venta (Sistema)2. Agregar, editar o eliminar un producto a la venta (Tendero)<ol style="list-style-type: none">2.1 Lanza un Poup para seleccionar el producto(Sistema)2.2 Autocompletar la informacion con el producto relacionado(Sistema)3. Mostrar los cambios en la lista de productos (Sistema)4. Actuliza el total de la venta (Sistema)5. Finalizar Venta (Tendero)<ol style="list-style-type: none">5.1 Lanza un Poup con la informacion del cierre de la venta (Sistema)5.2 Calcula los valores de descuento y cambio (Sistema)5.3 Registrar los valores de la venta (Sistema)6. Agregar un cliente nuevo(Tendero)<ol style="list-style-type: none">6.1 Lanzar un popup con la informacion (Sistema)	
Flujo Alterno: <ol style="list-style-type: none">2. No se puede agregar un producto por su disponibilidad en el inventario5. No se llena la informacion requerida para finalizar la venta	
Poscondiciones: La venta queda registrada en la base de sistema con toda su informacion y lo deja la interfaz lista para seguir con otra venta nueva	

Diagrama 15: Hoja de Caso de Uso (Modulo de Ventas)

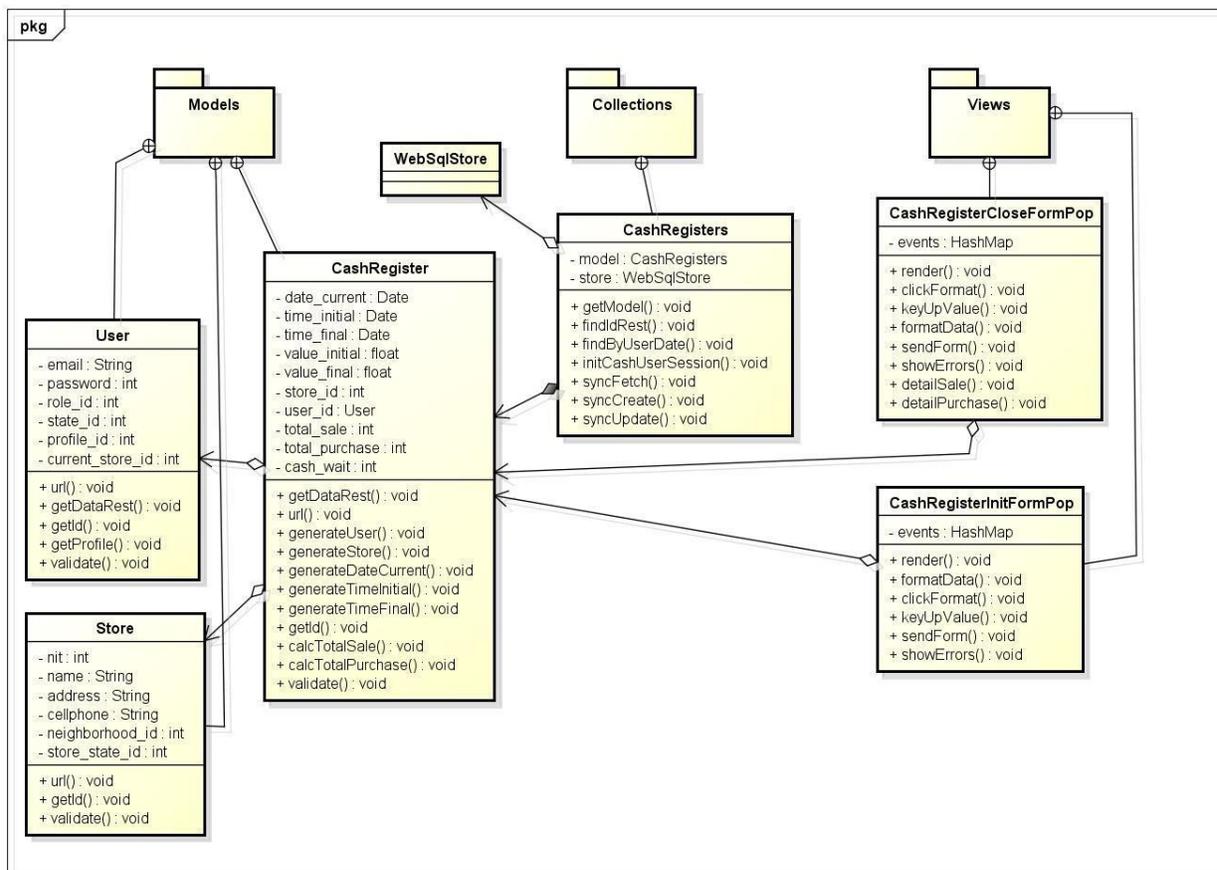


powered by Astah

Diagrama 16: Caso de Uso (Modulo de Ventas)

Diagrama de Clases

Módulo de Caja



powered by Astah

Diagrama 17: Diagrama de Clases (Modulo de Caja)

Módulo de Compras

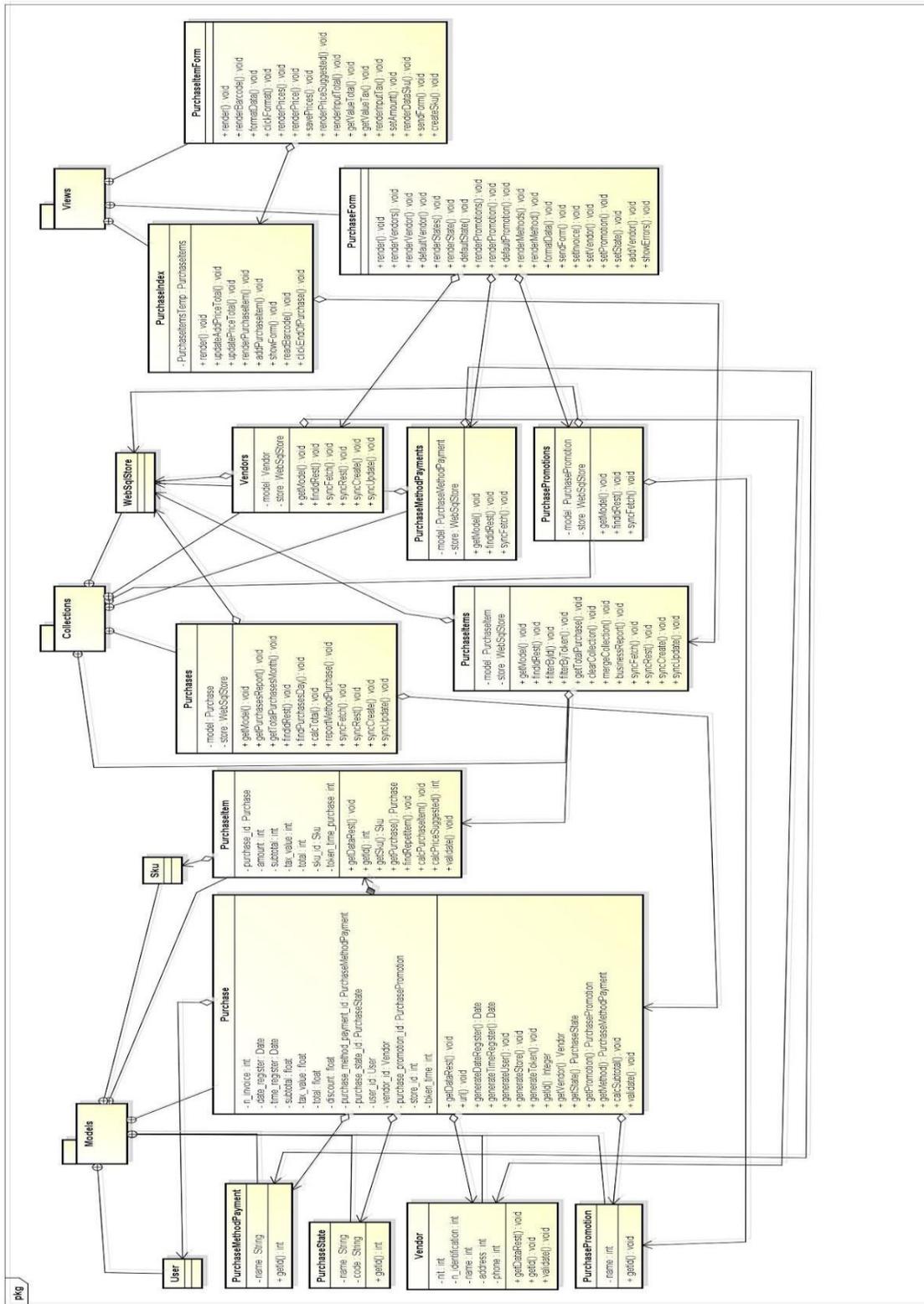


Diagrama 18: Diagrama de Clases (Modulo de Compras)

Módulo de Inventario

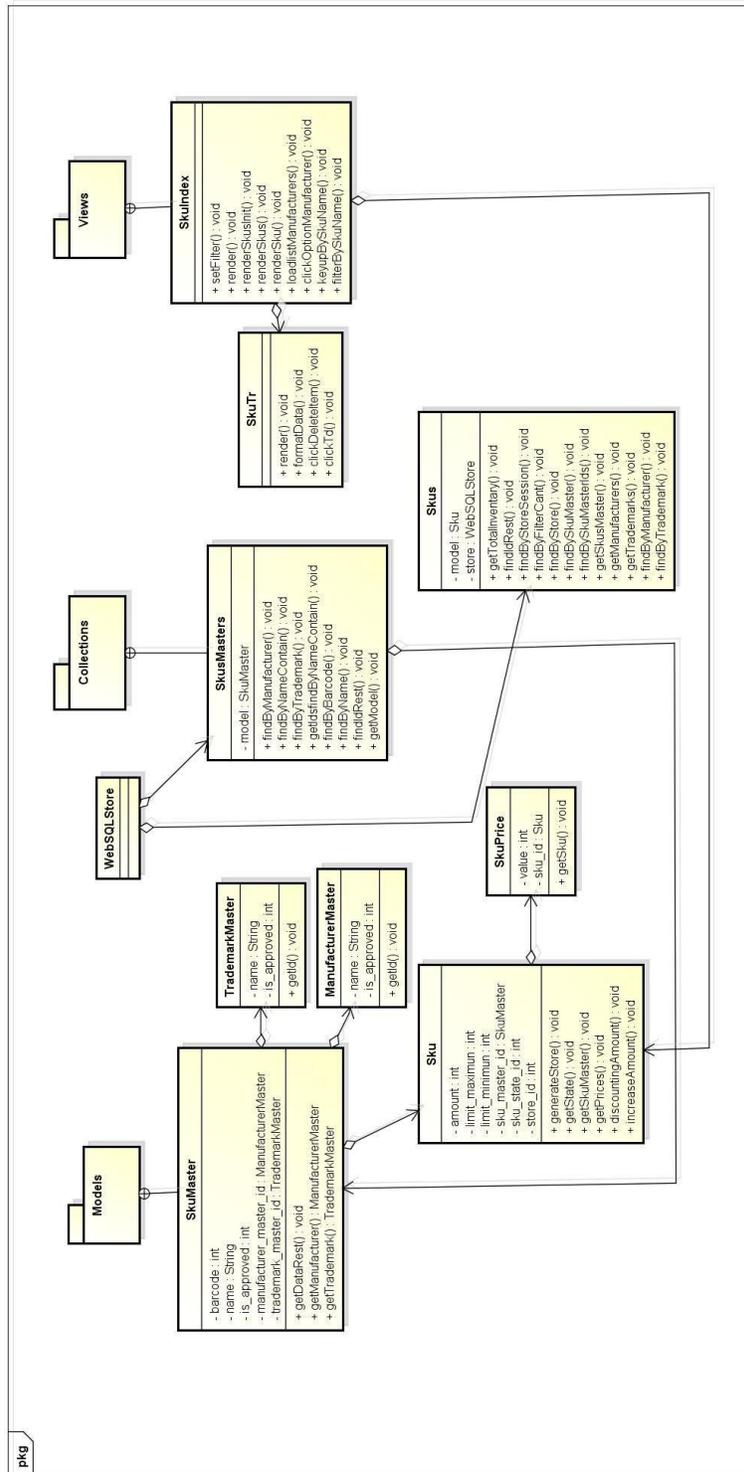


Diagrama 19: Diagrama de Clases (Modulo de Inventario)

Módulo de Ventas

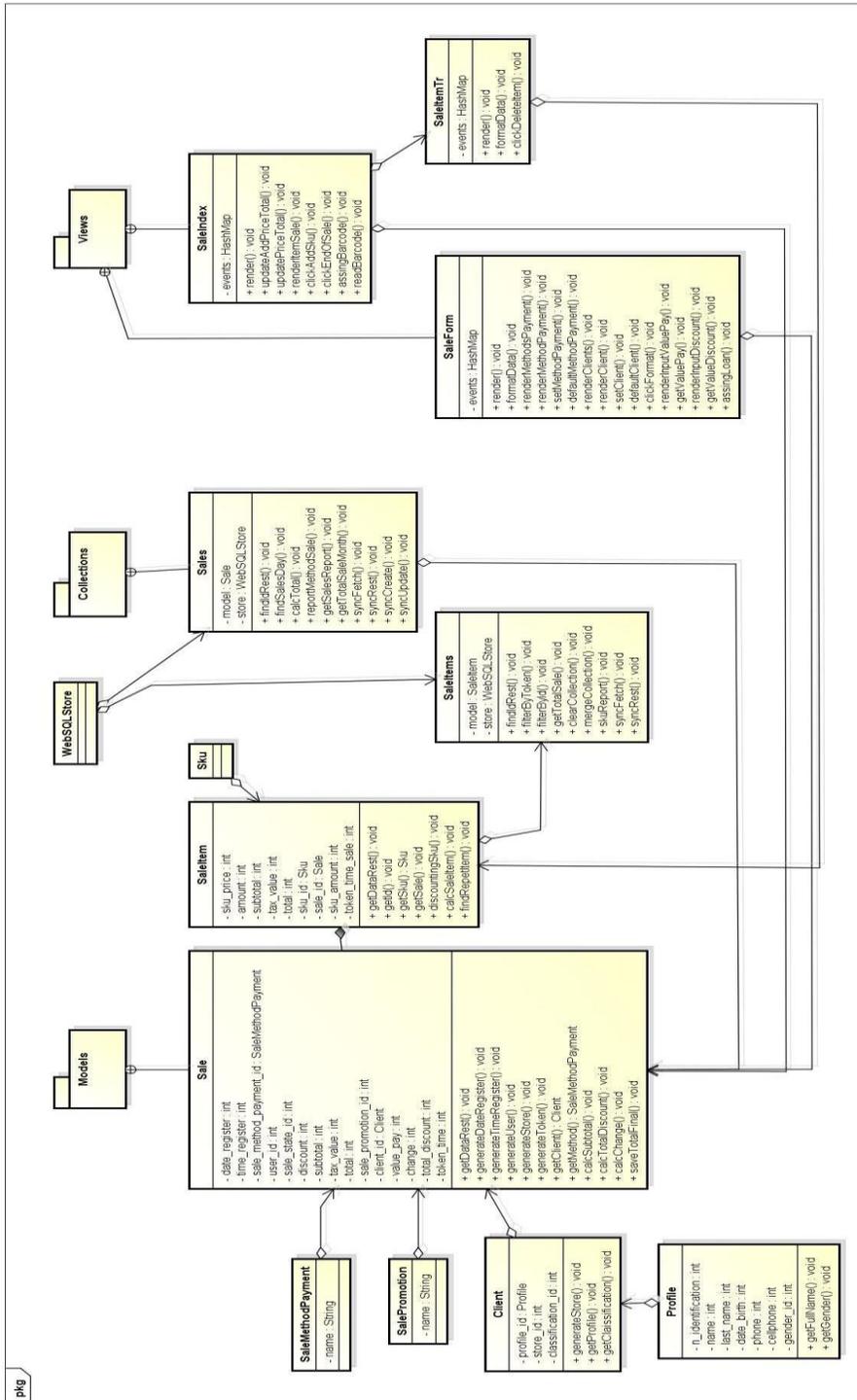


Diagrama 20: Diagrama de Clases (Modulo de Ventas)

Diagrama de Componentes

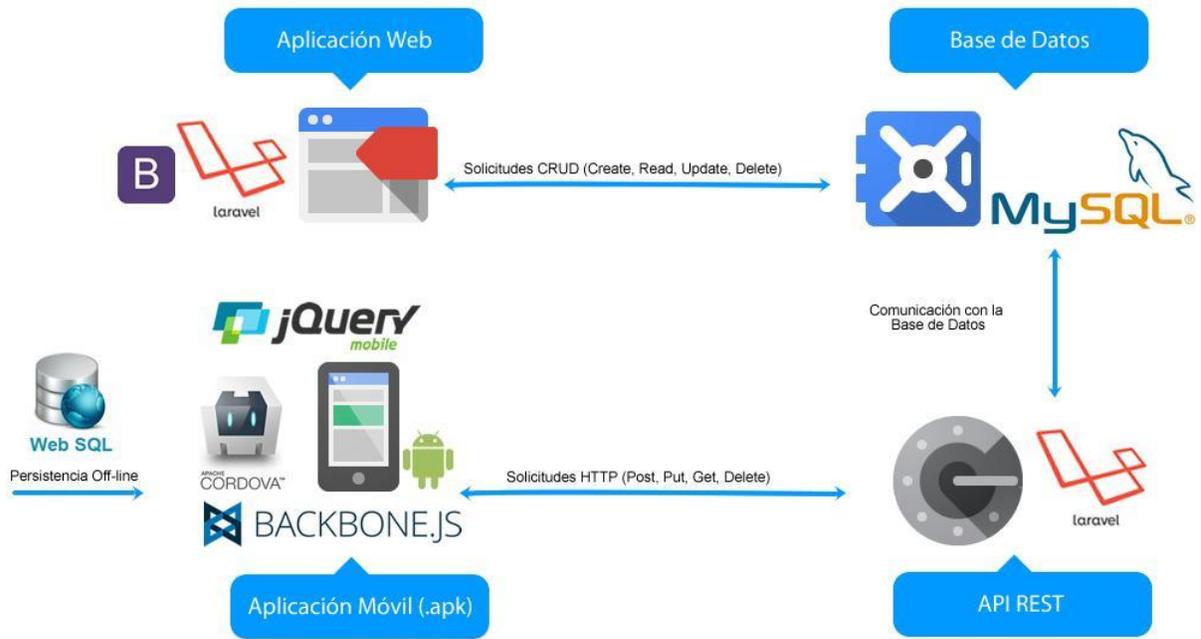
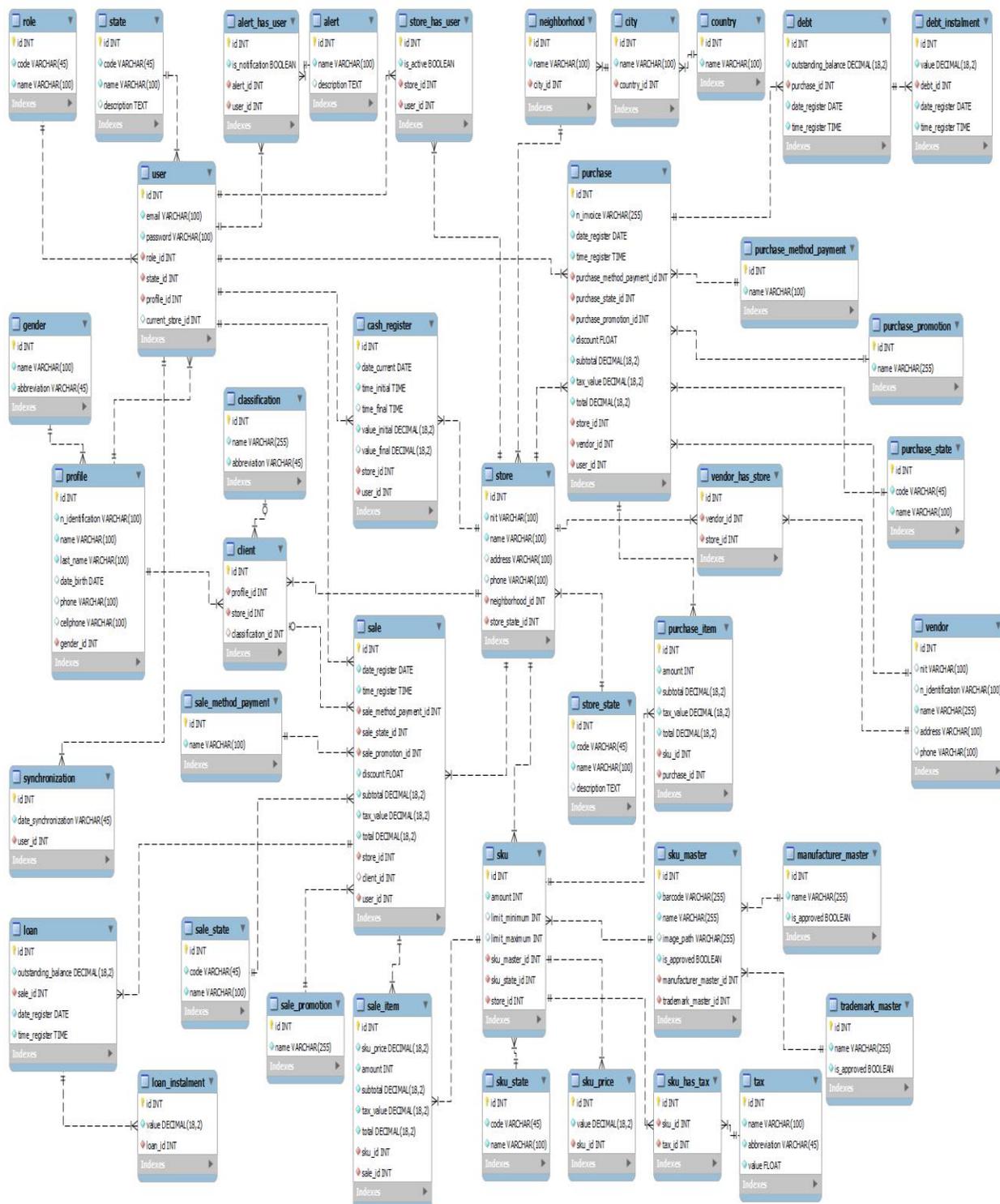


Diagrama 21: Diagrama de Componentes

Diagrama Entidad - Relación



DESARROLLO

En la etapa de desarrollo del proyecto, se plantean dos aspectos fundamentales para llevar a cabo su correcta gestión: especificaciones técnicas de Hardware y de Software, debido a que permite medir la capacidad de funcionalidad, desarrollo y procesamiento que genera su implementación.

Especificaciones técnicas

El desarrollo del proyecto necesita tener en cuenta aspectos técnicos, investigativos y de desarrollo.

Software

Para el desarrollo de la aplicación móvil se ha utilizado las siguientes herramientas, y tecnologías:

- Sublime Text
- Android SDK
- Apache Cordova
- HTML5
- CSS3
- JavaScript
- jQuery Mobile
- Backbone.js
- Underscore.js

Para el desarrollo del sistema de información, API REST y Base de Datos se utilizaron las siguientes herramientas y tecnologías:

- Servidor web XAMPP
- MySQL Workbench
- Gestor de Base de Datos MySQL
- FileZilla
- Putty
- Google Chrome
- PHP
- Composer
- Framework Laravel
- Twitter Bootstrap

- jQuery
- REST Console

Hardware

Para el sistema de información se debe contar con un espacio de hosting en un VPS o Hosting dedicado.

También se puede utilizar en un equipo local:

- Procesador Intel Core i3
- Memoria RAM: 4 Gb
- Disco Duro: 500 Gb
- Teclado
- Mouse

GLOSARIO

Ingeniería del Software: Es una disciplina de ingeniería que comprende todos los aspectos de producción del software.

Requerimientos: Especifican que es lo que el sistema debe hacer (sus funciones) y propiedades esenciales deseables.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

Servidor: Es una aplicación informática que provee servicios a otras aplicaciones denominadas Clientes.

XAMPP: Es un servidor libre independiente, con el cual podemos instalar apache, aparte que contiene MySQL, e interpreta el lenguaje de programación PHP.

Hosting: Es un servicio que provee a los usuarios, almacenar información, para que pueda acceder a ella desde cualquier lugar.

Base de Datos: Es un conjunto de datos pertenecientes a un mismo contexto y almacenados para un posterior uso.

MySQL: Es un motor de bases de datos libre, el cual puede ser utilizado por cualquier persona. Maneja varias herramientas para el manejo de datos.

PHP: Es un lenguaje de programación interpretado para crear páginas web dinámicas, su uso no es local, por lo que se requiere un servidor para su funcionamiento.

CONCLUSIONES

Se desarrolló el Sistema de Información basado en Cloud Computing, también se implementó la arquitectura API REST sin ninguna dificultad aportando beneficios el desarrollo de servicios o sistemas de información que requieran conectarse con la Base de Datos del presente proyecto.

También se desarrolló la aplicación móvil para tabletas, brindando comodidad, portabilidad y simpleza a solución propuesta.

Se realizó las pruebas pertinentes en ambiente de producción mostrando los siguientes resultados:

- El tendero gestionó los ingresos de su tienda de manera adecuada.
- Gestionó los egresos que tiene la tienda observando en tiempo real sus ganancias.
- Gestionó el inventario que tiene la tienda y logró saber qué tenía y qué le hacía falta a su tienda.
- Observó la información por medio de tablas y logró tomar mejores decisiones en beneficio de su tienda.

BIBLIOGRAFÍA

Deemer, P. Benefield, G. Larman, C. Vodde, B. (2009) Información Básica de SCRUM (The SCRUM Primer). Versión 1.1
San Francisco, California, USA.

Schmuller, J. (2000) Aprendiendo UML en 24 horas
Pearson Educación.

Arango, T. (2013, Enero 12). Tiendas de barrio siguen teniendo mayor presencia.
Recuperado el 30 de Noviembre de 2014 de:
http://www.larepublica.co/consumo/tiendas-de-barrio-siguen-teniendo-mayor-presencia_28962
Periódico La República.

Colprensa (2014, Septiembre 19). Grandes cadenas apuestan por las tiendas de barrio.
Recuperado el 30 de Noviembre de 2014 de:
<http://www.elpais.com.co/elpais/economia/noticias/grandes-cadenas-apuestan-por-tiendas-barrio>
Diario del País.

Fernández, A. (2011, Marzo 28). Aplicaciones para el desarrollo.
Recuperado el 30 de Noviembre de 2014 de:
<http://www.emprendedores.es/gestion/apps/aplicaciones-para-el-desarrollo>
Website emprendedores.es

iniciativamexico.org. Frogtek: Ayudando a las Tienditas Mexicanas.
Recuperado el 30 de Noviembre de 2014 de:
http://www.iniciativamexico.org/perfiles/frogtek_ayudando_a_las_tienditas_mexicanas/
Website iniciativamexico.org

MANUALES



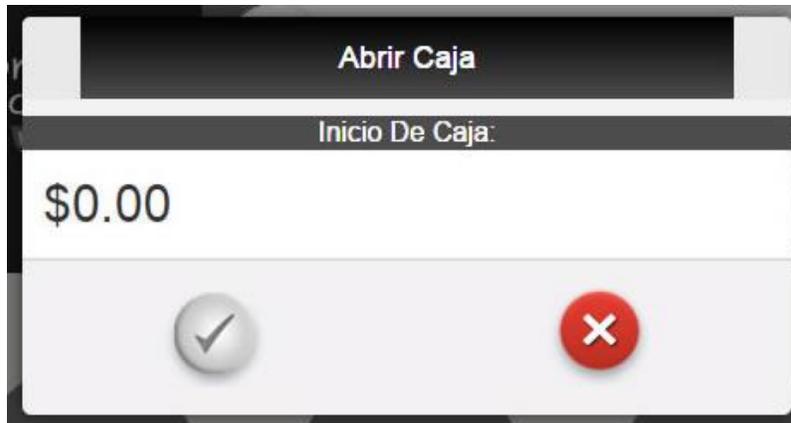
Ingreso

El usuario debe ingresar su usuario y clave para poder autenticarse en la aplicación, lo primera vez que ingrese debe tener acceso a internet.



Pantalla de inicio

Al autenticarse exitosamente se mostrará la pantalla de inicio con los cinco módulos para la gestión del negocio.



Inicio de Caja

Al ingresar la aplicación automáticamente le lanzara un aviso para saber con cuanto inicia la caja en el día actual.



Módulo de ventas

En el módulo de ventas se mostrará toda la interfaz para hacer una venta, al inicio aparece vacía pero a medida que se vayan agregando productos a la venta este los va agregando a la lista de productos de la compra, desde ahí podrá eliminar o editar un ítem de la venta.

Agregar Un Producto

Codigo De Barras: 7702090034035

Nombre De Sku: Bebida Energizante Peak 2

Cantidad: 1 En Inventario: 5

Precio: \$1,800.00

✓ ✗

Agregar un ítem a la venta

Cuando se agrega un nuevo producto a la venta se lanza una ventana donde puede buscar y escoger el producto así como la cantidad y el precio a vender.

Finalizar Venta

Total: \$12,800.00

Paga Con: \$20,000.00

Cambio: \$7,200.00

Cliente: Cliente Uno + Agregar

Método De Pago: Tarjeta Crédito

Descuento: 0%

Cerrar Venta Imprimir Cancelar

Finalizar la venta

Cuando finaliza la venta, debe llenar la información requerida para ser registrada y descontada del inventario.



Módulo de compra

En el módulo de compras se registran todos los gastos a proveedores que tenga la tienda, a medida que se van agregando se puede ir modificando o eliminar un producto.

Agregar Sku
Código De Barra:

7702090032505

Nombre:

Gaseosa Postobón La 10 de 500MI

<small>Fabricante:</small>	<small>Marca:</small>
Postobón S.A.	Postobón

<small>Cantidad:</small>	<small>En Inventario:</small>	<small>Cantidad Mínima:</small>
1	0	0

<small>Total Compra:</small>	<small>Iva:</small>
\$0.00	16%

Precio De Venta 1:

\$0.00

Precio De Venta 2:

\$0.00

Precio De Venta 3:

\$0.00

✓
✗

Agregar un producto a la compra

Al agregar un producto a la tienda aparece toda la información para registrarlo y que éste se aumente en el inventario de la tienda.



Finalizar la compra

Al finalizar una venta se deberá registrar la información para ser registrada en las cuentas del negocio.



Módulo de inventario

En este módulo se puede gestionar toda la información de los productos de la tienda así como sus precios de venta o la cantidad mínima en la tienda.

Crear Sku	
Codigo De Barra:	
Nombre:	
Fabricante:	Marca:
Precio De Venta 1:	
\$0.00	
Precio De Venta 2:	
\$0.00	
Precio De Venta 3:	
\$0.00	
Precio De Venta 4:	
\$0.00	
Precio De Venta 5:	
\$0.00	
Cantidad:	
1	

Crear Sku

Para agregar un nuevo producto solo se debe colocar la información del código de barras y este la auto completará la información si está en la base de datos de skus maestra.

Cerrar Caja	
Inicio De Caja:	
\$5,000.00	
Total De Ventas:	
\$0.00	Detalles
Total De Compra:	
\$0.00	Detalles
Caja Esperada:	
\$0.00	
Caja Real:	
\$0.00	
Recuerde que si cierra caja se cerrará su sesión	
	

Cierre de caja

En este módulo se puede el reporte final del negocio antes de cerrar caja, debe ingresar el valor de caja real que al final quedó en la tienda.