

# Sistema de Información para Empresas Mi Empresa



Jorge Mario Guaqueta Restrepo

Universidad Distrital Francisco José de Caldas  
Ingeniería de Software I  
Facultad de Ingeniería  
Bogotá D.C.  
Junio 10 de 2017

# Resumen

El proyecto consiste en un sistema de información en donde el cliente (**Pega Maz**) puede registrar sus ventas de manera manual, posteriormente al momento de realizar la contabilidad el cliente podrá ingresar a un modulo de visualización de ventas el cual le proporcionará un informe de ventas dentro de un rango de fechas que el usuario seleccione, por ultimo el cliente tendra acceso a la modificación de inventarios en donde podra agregar productos, eliminar productos y agregar cantidades a los productos.

# Tabla de Contenido

<b>Resumen</b>	<b>I</b>
<b>Introducción</b>	<b>1</b>
<b>1. Planeación</b>	<b>2</b>
1.1. Planteamiento del Problema . . . . .	2
1.2. Justificación . . . . .	2
1.3. Objetivos . . . . .	3
1.3.1. Objetivo General . . . . .	3
1.3.2. Objetivos Específicos . . . . .	3
1.4. Alcances . . . . .	3
1.5. Limitaciones . . . . .	3
1.5.1. Marco Metodológico . . . . .	3
1.6. Factibilidad . . . . .	6
1.6.1. Factibilidad Técnica . . . . .	6
1.6.2. Factibilidad Operativa . . . . .	7
1.6.3. Factibilidad Económica . . . . .	7
1.6.4. Factibilidad Legal . . . . .	8
1.7. Cronograma . . . . .	8
<b>2. Análisis</b>	<b>10</b>
2.1. Análisis de Requerimientos . . . . .	10
2.1.1. Requerimientos . . . . .	11
2.2. Definición de Actores . . . . .	11
<b>3. Diseño</b>	<b>15</b>
3.1. Casos de Uso . . . . .	15
3.1.1. Diagramas de Casos de Uso . . . . .	15
3.1.2. Documentacion de casos de uso . . . . .	17
3.2. Diagrama de Clases . . . . .	20
3.3. Diagramas de Secuencia . . . . .	21
3.4. Diagramas de Componentes . . . . .	21
3.5. Diagrama de relaciones . . . . .	21
3.6. Diagrama BPMN . . . . .	21

<b>4. Implementación</b>	<b>27</b>
4.1. Modelo Relacional . . . . .	27
4.1.1. Relaciones del sistema: . . . . .	27
4.2. Diccionario de Datos . . . . .	27
4.2.1. Clase roles de usuario: . . . . .	27
4.2.2. Clase usuario del sistema: . . . . .	28
4.2.3. Clase cliente del sistema: . . . . .	28
4.2.4. Clase impuestos del sistema: . . . . .	28
4.2.5. Clase productos del sistema: . . . . .	29
4.2.6. Clase ordenes del sistema: . . . . .	29
4.2.7. Clase facturas del sistema: . . . . .	29
<b>5. Manual de Instalación</b>	<b>30</b>
5.1. Instalación de python . . . . .	30
5.2. Instalación de pip . . . . .	30
5.3. Instalación de virtualenv . . . . .	30
5.4. Instalación de MySQL server . . . . .	31
5.5. Instalación de Apache . . . . .	31
<b>6. Manual de Usuario</b>	<b>32</b>
6.1. Inicio de sesión . . . . .	32
6.2. Pagina inicial . . . . .	33
6.3. Menú de administración . . . . .	34
6.4. Facturación . . . . .	35
6.5. Inventarios . . . . .	36
6.6. Contabilidad . . . . .	37
<b>Conclusiones</b>	<b>41</b>
<b>Bibliografía</b>	<b>42</b>

# Lista de Figuras

1.1. Ciclo de Scrum . . . . .	6
1.2. Cronograma . . . . .	9
2.1. Diagrama de contexto . . . . .	11
3.1. Diagrama de Casos de Uso de Administrador . . . . .	15
3.2. Diagrama de Casos de Uso de Ingreso . . . . .	16
3.3. Diagrama de Casos de Uso de Inventario . . . . .	16
3.4. Diagrama de Casos de Uso de Facturacion . . . . .	17
3.5. Diagrama de Casos de Uso de Contabilidad . . . . .	17
3.6. Diagrama de Clases . . . . .	21
3.7. Diagrama de secuencia del usuario . . . . .	23
3.8. Diagrama de secuencia del usuario de contabilidad . . . . .	24
3.9. Diagrama de componentes del sistema de información. . . . .	24
3.10. Diagrama de relaciones del sistema de información. . . . .	25
3.11. Diagrama BPMN del proceso de facturación . . . . .	26
6.1. Inicio de sesión de usuario . . . . .	32
6.2. Pagina de inicio . . . . .	33
6.3. Pagina de inicio . . . . .	34
6.4. Facturación . . . . .	35
6.5. Buscador de productos . . . . .	36
6.6. Tabla de orden, con productos agregados . . . . .	37
6.7. Inventarios . . . . .	38
6.8. Contabilidad . . . . .	39
6.9. Facturas encontradas dentro de un rango de fechas . . . . .	40

# Lista de Tablas

1.1. Actividades durante un Sprint de SCRUM . . . . .	4
1.2. Costo de Recursos . . . . .	8
1.3. Costo de Integrantes del Proyecto . . . . .	8
2.1. Tabla de requerimientos funcionales . . . . .	12
2.2. Tabla de requerimientos no funcionales . . . . .	13
2.3. Tabla de definición de los actores . . . . .	14
3.1. Documentación de Caso de Uso . . . . .	17

# Introducción

Durante mucho tiempo las empresas o negocios han llevado su contabilidad de forma manual, así como también sus inventarios y su facturación por lo que se hace necesario implementar sistemas de información en los cuales se pueda realizar una cierta cantidad de actividades que permiten un mejor manejo de los costos, gastos y flujos que se presenten en el negocio.

En cuanto a la necesidad se plantea un sistema de información el cual permitirá facturar los productos que serán vendidos, además de permitir mediante el módulo de inventarios llevar un inventario de los productos que están siendo comercializados y manejar las cantidades de los productos que se encuentren dentro del inventario, y un módulo para obtener las ventas del día, de la semana o del mes, es decir, este módulo permitirá realizar filtros por fechas en donde el usuario ingresa unas fechas y si existen facturas dentro de esta fecha se puede generar el reporte.

# Capítulo 1

## Planeación

### 1.1. Planteamiento del Problema

El problema consiste en que al momento de realizar inventarios se hace necesario llevar a cabo un conteo de las unidades para anotarlas en un cuaderno o manual de inventarios, de igual manera restar las unidades vendidas dependiendo el producto y posteriormente este sera el valor numerico que se encuentra en stock.

Por otra parte se necesitaba hacer diario un registro de ventas y de productos, con la necesidad de que este no se podia acumular por que si se dejaba de hacer por unos dias se convertia en una lista de productos que debian salir del inventario causando errores que pueden llevar a afectar las cantidades en inventario con las unidades en stock.

### 1.2. Justificación

Es valido resolver este problema debido a que con esta solución de software se podra realizar inventario mensual, semanal y diario con una sola tirilla de productos y sus totales vendidos, de igual forma ya no sera necesario la facturación a mano disminuyendo el uso de papel y el porcentaje de error debido a trabajo humano.

Por otra parte se pueden observar con menor dedicación de tiempo parametros como ventas diarias, semanales, mensuales, las cuales se revisan para plantear metas y otros parametros como ventas por usuario las cuales pueden proporcionar informacion de rendimientos.

Por otra parte el modulo de contabilidad proporcionará un reporte conciso el cual sera usado por el usuario de contabilidad o el usuario administrador para revisar las ventas dentro de las fechas establecidas.



## 1.3. Objetivos

### 1.3.1. Objetivo General

Diseñar y desarrollar un prototipo de software basado en django web framework para llevar el manejo de facturación, contabilidad e inventario en un almacén de productos para venta por unidad o al por mayor.

### 1.3.2. Objetivos Específicos

- Desarrollar una sistema de información para negocio.
- Implementar el patron MVC.
- Desarrollar el sistema de informacion mediante web framework django.

## 1.4. Alcances

El proyecto podrá ofrecer un aplicativo editable para diferentes tipos de clientes, es decir, diferentes almacenes. Cada aplicativo tendra la posibilidad de llevar modulos segun el cliente los solicite y ser editado al estilo del almacén.

Asimismo se podran agregar modulos con facilidad para generar nuevas soluciones para requerimientos a petición del cliente.

## 1.5. Limitaciones

El proyecto prestara unicamente soluciones para empresas basadas en productos de venta libre, manejo de facturacion e inventario. Este no podra imprimir facturas, unicamente se realizara mediante correo electronico el envio de facturas.

### 1.5.1. Marco Metodológico

La metodología **SCRUM** es un proceso de desarrollo de software con metodologia agíl y flexible. Su principal objetivo es maximizar la inversión y disminuir el tiempo de desarrollo [1].

Este proceso permite entregar cada fin de la iteración un prototipo funcional y crear expectativas para cada iteración. Esta metodologia permite obtener los siguientes beneficios.

1. **Cumplimento de expectativas:** Esta metodología permite que el cliente genere unas expectativas y el arquitecto de scrum planea cada iteración y genera prioridades.
2. **Flexibilidad a cambios:** Esta metodología presenta una alta capacidad al cambio especialmente en los requerimientos.
3. **Reducción del Time to Market:** Se le puede dar uso a los diferentes prototipos y se puede entregar un prototipo parcial para realizar su evaluación.

La Tabla 1.1 presenta las diferentes actividades realizadas con base en las fases de la metodología SCRUM[1].

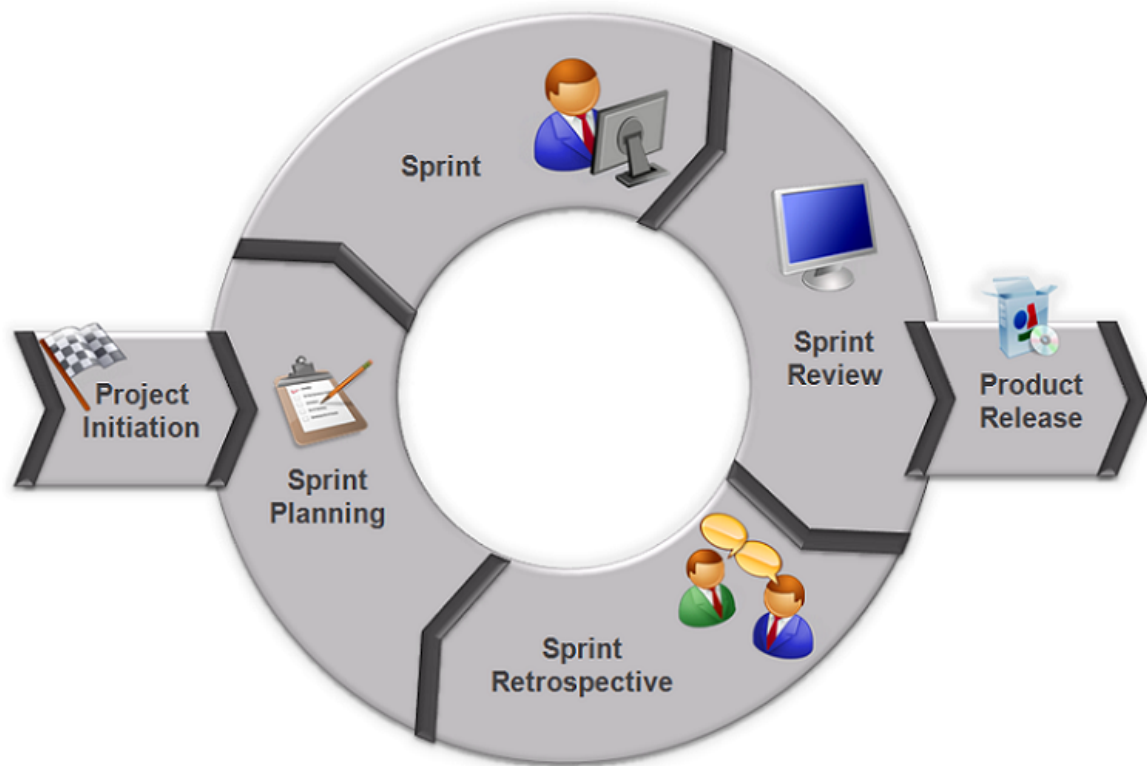
Tabla 1.1: Actividades durante un Sprint de SCRUM

Etapa	Actividad	Descripción
Planeamiento del sprint	Planeación del sprint	Durante la planeación del sprint se llaman a reunión todos los participantes del equipo, durante la consecución de este evento se decide que requerimientos o actividades van a ser asignados a los integrantes. Cada integrante deberá asignar un elemento de tiempo en el cual se va a realizar dicha actividad asignada. Esto permitirá obtener el tiempo en el cual se ejecutará el sprint.
Reunion de Equipo de Scrum	Reuniones diarias	<p>Cada día los elementos del equipo de scrum se reuniran con una duración no maxima a 15 minutos. Estas reuniones se llevaran a cabo en el mismo horario cada día, durante estas reuniones los miembros del equipo deberán responder las siguientes preguntas:</p> <ul style="list-style-type: none"> <li>▪ ¿Qué se hizo ayer?</li> <li>▪ ¿Qué tiene planeado hacer hoy?</li> <li>▪ ¿Qué obstáculos encontró en el camino?</li> </ul> <p>Estas reuniones le permitirán a todos los miembros del equipo realizar una realimentación entre ellos como una manera de apoyo, en caso tal de que alguno tenga inconvenientes, se buscará la manera de resolver el obstáculo; de no ser encontrada una solución se deberá realizar una reunión para buscar solución a este obstáculo.</p>

Refinamiento del Backlog	Revisión del backlog	El director de scrum revisa cada elemento dentro del backlog, esto con el fin de esclarecer dudas que se presenten con respecto a las tareas que tiene el equipo de trabajo. Esto tambien permite recalcular el tiempo y esfuerzo que se va a dedicar para el sprint, y si es necesario ingresar nuevas actividades al backlog.
Revisión del Sprint	Revisión del sprint entre miembros	Los miembros del equipo de desarrollo y los clientes se reúnen para mostrar el prototipo que se obtuvo durante el sprint. Durante esta reunion se realiza una prueba de todos los desarrollos funcionales que se han finalizado. Esta presentacion estará a cargo del Scrum Master y el Product Owner.
Retrospectiva del Sprint	Realimentación del Sprint	<p>Durante esta actividad el Product Owner se reunirá con todo el equipo para dialogar sobre lo que se presente durante el sprint, los puntos principales para esta reunión serán:</p> <ul style="list-style-type: none"> <li>▪ Que cosas se hicieron mal durante el sprint, esto para poder mejorar el proximo</li> <li>▪ Que cosas se hicieron bien para continuar realizando las actividades de esta manera.</li> <li>▪ Que inconvenientes se encontraron y no permitieron avanzar como se planifico.</li> </ul>

Ciclo del scrum, tomado de Manage your complex projects using Agile and Scrum - Scrum Master workshop[2].

Figura 1.1: Ciclo de Scrum



## 1.6. Factibilidad

### 1.6.1. Factibilidad Técnica

Para el desarrollo e implementación del proyecto es necesario contar con los siguientes dispositivos:

- Equipo de computo para desarrollo de aplicaciones.
- Equipo de computo para ejecutar el proyecto.

Así mismo, es necesario contar con los siguientes recursos de software:

- python 3+
- MySQL
- Django
- Navegador web

Por último conexión a internet.

### **1.6.2. Factibilidad Operativa**

Este proyecto será elaborado por un estudiante de Ingeniería de Software I, el cual cuenta con el conocimientos previos en las herramientas python 3, django y MySQL.

### **1.6.3. Factibilidad Económica**

El costo del proyecto se presenta con base en:

1. Costo de los equipos necesarios para el diseño y desarrollo de los prototipos. El costo se desglosa en la Tabla 1.2
2. Costo de mano de obra de los integrantes. El costo se desglosa en la Tabla 1.3.

Tabla 1.2: Costo de Recursos

Recurso	Cantidad	Valor Unitario	Valor total
Equipos de Computo	2	\$1.000.000	\$2.000.000
<b>Total</b>			<b>\$2.000.000</b>

Tabla 1.3: Costo de Integrantes del Proyecto

Integrante	Valor hora	Horas	Valor total
Jorge Mario Guaqueta Restrepo	\$50.000	100	\$5.000.000
<b>Total</b>			<b>\$5.000.000</b>

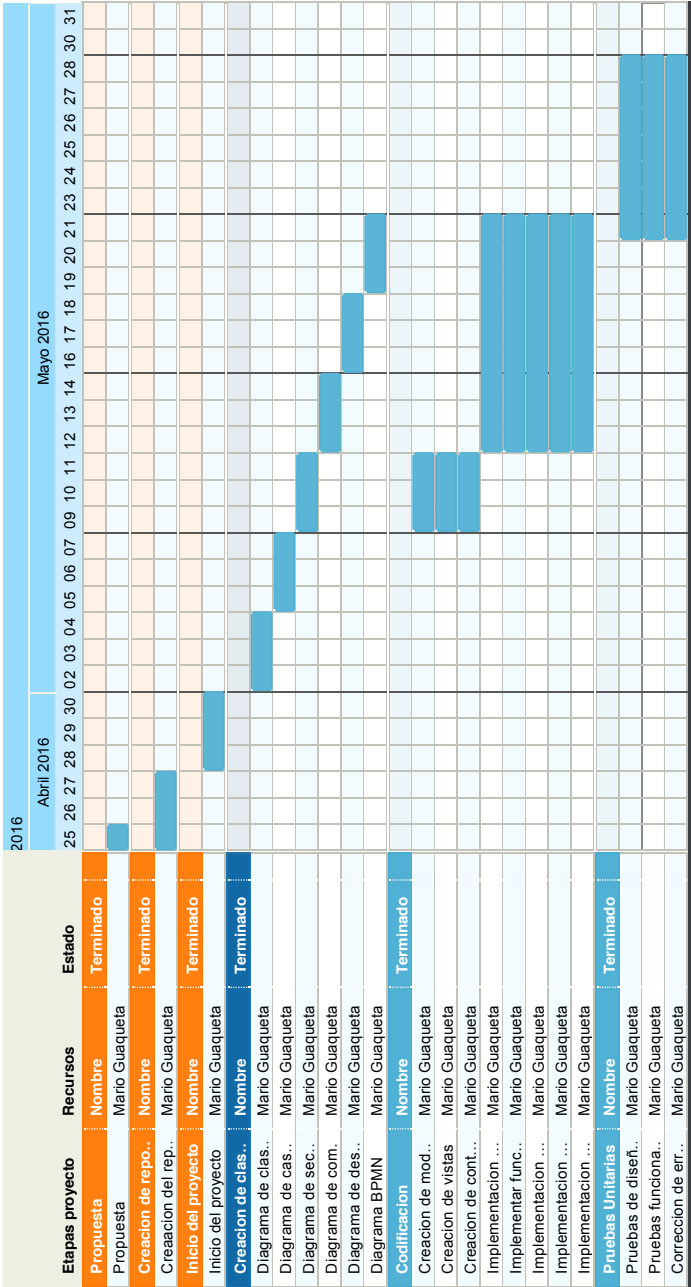
#### 1.6.4. Factibilidad Legal

Se utiliza software libre para el desarrollo del sistema de información y para la base de datos, por lo cual se puede aplicar sin generar costos adicionales.

### 1.7. Cronograma

La Figura 1.2 presenta el cronograma para el desarrollo de la aplicación.

Figura 1.2: Cronograma



## Capítulo 2

# Análisis

### 2.1. Análisis de Requerimientos

Este proyecto permite que la empresa PegaMaz lleve una facturación de forma digital, en donde los usuarios podran almacenar información del cliente, información de la compra y realizar una factura para las ventas.

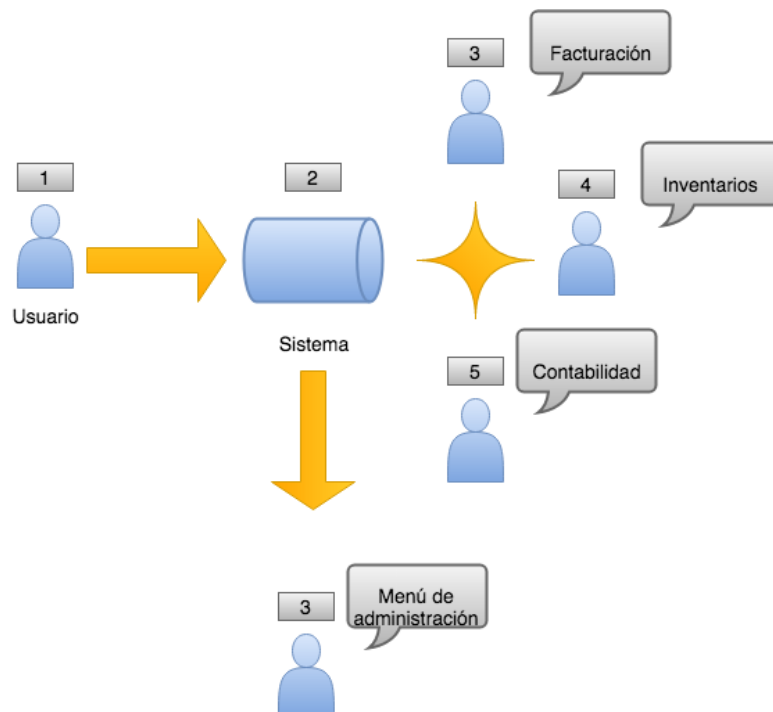
Por otra parte se permitira que los usuarios realicen una busqueda de productos para facturarlos, un modulo de inventario en donde el usuario podra revisar la cantidad de unidades y por último un modulo de ventas en el cual se podra observar las ventas con un filtro de fecha.

La interacción de los usuarios con los modulos contruidos se presenta en la Figura 2.1, la cual describe los siguientes pasos.

- **Usuario:** El usuario es quien realiza los procesos de facturación, ingresa nuevos productos que vienen de los proveedores, al mismo tiempo que visualiza la facturación para informar al cliente y visualización de inventarios para conteos fisicos.
- **Contabilidad:** El usuario de contabilidad podrá visualizar todos lo elementos relacionados con la facturación y las ventas.
- **Cliente:** Recibirá la facturación mediante un correo electrónico.
- **Administrador:** Tendra la facultar de movilizarse por todos los modulos, al igual que podra ingresar productos a la lista de inventario y podra revisar contabilidad.



Figura 2.1: Diagrama de contexto



### 2.1.1. Requerimientos

#### Requerimientos Funcionales

Los requerimientos funcionales describen los diferentes módulos que el proyecto va a contener, La Tabla 2.1 presenta los requerimientos funcionales del sistema.

#### Requerimientos No Funcionales

Los requerimientos no funcionales describen diferentes características adicionales que el proyecto debe satisfacer para poder llevar a cabo los servicios que provee. La tabla 2.2 presenta los requerimientos no funcionales del sistema.

## 2.2. Definición de Actores

Con base en el análisis de requerimientos realizado, se establecen los actores que se presentan en la Tabla 2.3.

Tabla 2.1: Tabla de requerimientos funcionales

Requerimientos funcionales			
Identificador	Nombre	Descripción	Actores
Requerimiento 1	Ingreso	El sistema deberá permitir el ingreso de un usuario mediante su nombre de usuario y contraseña, realizando una validación	<ul style="list-style-type: none"> <li>■ Administrador</li> <li>■ Contador</li> <li>■ Usuario</li> </ul>
Requerimiento 2	Creación de usuario y edición	El sistema deberá permitir, mediante una contraseña de administrador, registrar un usuario el cual podrá tener un nombre de usuario, una contraseña, un email y un rol.	<ul style="list-style-type: none"> <li>■ Administrador</li> </ul>
Requerimiento 3	Creación y edición de productos	El usuario con rol de administrador podrá crear y editar productos.	<ul style="list-style-type: none"> <li>■ Administrador</li> </ul>
Requerimiento 4	Modulo de facturación	<p>Este modulo permitirá que el usuario realice una factura para el cliente con los siguientes aspectos:</p> <ul style="list-style-type: none"> <li>■ Subtotal</li> <li>■ Iva</li> <li>■ Valor final</li> </ul> <p>Por otra parte, permitirá buscar productos en el inventario para ser agregados a la factura, al igual que ingresar un código para agregar el producto.</p>	<ul style="list-style-type: none"> <li>■ Administrador</li> <li>■ Usuario</li> </ul>
Requerimiento 5	Modulo de inventarios	Este modulo permitirá que el usuario realice una revisión de los inventarios con finalidad de solicitud de productos a los proveedores y conteo físico.	<ul style="list-style-type: none"> <li>■ Administrador</li> <li>■ Usuario</li> </ul>
Requerimiento 5	Modulo de contabilidad	Este modulo permitirá que el usuario realice una revisión de ventas por fechas.	<ul style="list-style-type: none"> <li>■ Contador</li> </ul>

Tabla 2.2: Tabla de requerimientos no funcionales

Requerimientos no funcionales		
Identificador	Nombre	Descripción
Requerimiento no funcional 1	Seguridad	El sistema contara con el modulo de autenticación de Django, y contará con el header CSRF.
Requerimiento no funcional 2	Concurrencia	El sistema contara con una concurrencia maxima de 10 personas al mismo tiempo, ya que se utilizara un servidor con poca capacidad y no se implementará encolamiento de tareas.

Tabla 2.3: Tabla de definición de los actores

Actores		
Nombre	Descripción	Atributos
Administrador	Actor encargado de la administración del sistema, el cual tendra la posibilidad de modificar todos campos de los usuarios, gestionará usuarios, eliminacion de productos en inventarios, busqueda de ventas y eliminación de ventas.	<ul style="list-style-type: none"> <li>■ Nombre</li> <li>■ Apellido</li> <li>■ email</li> <li>■ Telefono</li> <li>■ Rol</li> </ul>
Usuario	Actor encargado de la facturación y la revisión de inventarios y actualización de inventarios.	<ul style="list-style-type: none"> <li>■ Nombre</li> <li>■ Apellido</li> <li>■ email</li> <li>■ Telefono</li> <li>■ Rol</li> </ul>
Contador	Actor encargado de la revisión de ventas para el manejo de la contabilidad.	<ul style="list-style-type: none"> <li>■ Nombre</li> <li>■ Apellido</li> <li>■ email</li> <li>■ Telefono</li> <li>■ Rol</li> </ul>
Cliente	Actor encargado de realizar la compra	<ul style="list-style-type: none"> <li>■ Nombre</li> <li>■ Apellido</li> <li>■ email</li> <li>■ Telefono</li> <li>■ Identificación</li> </ul>

## Capítulo 3

# Diseño

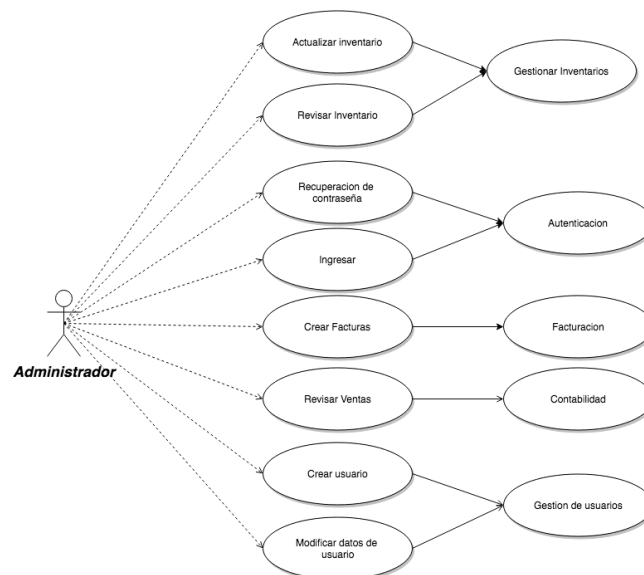
### 3.1. Casos de Uso

#### 3.1.1. Diagramas de Casos de Uso

A través de los diagramas de uso se puede observar los servicios que ofrecerá el sistema.

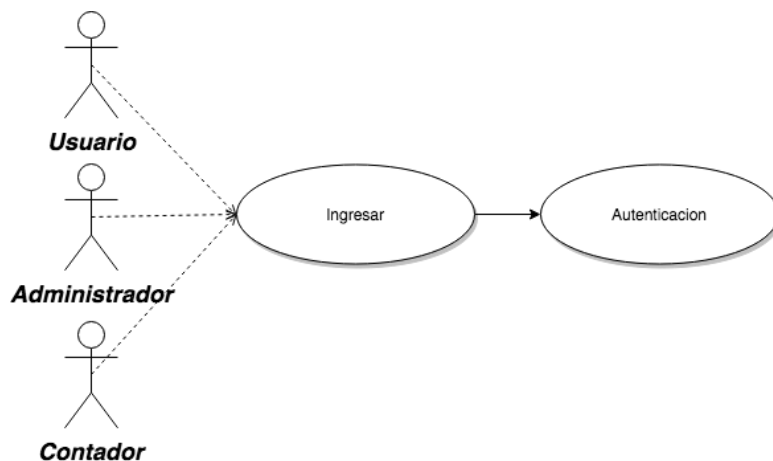
La Figura 3.1 presenta el diagrama de casos de uso para la gestión del administrador.

Figura 3.1: Diagrama de Casos de Uso de Administrador



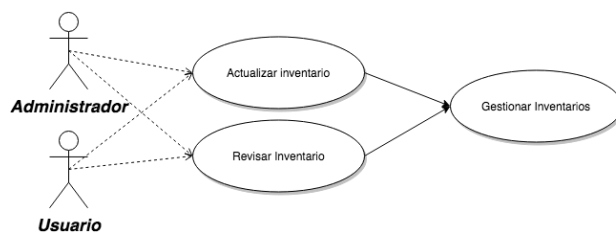
La Figura 3.2 presenta el diagrama de casos de uso para la gestión de las sesiones.

Figura 3.2: Diagrama de Casos de Uso de Ingreso



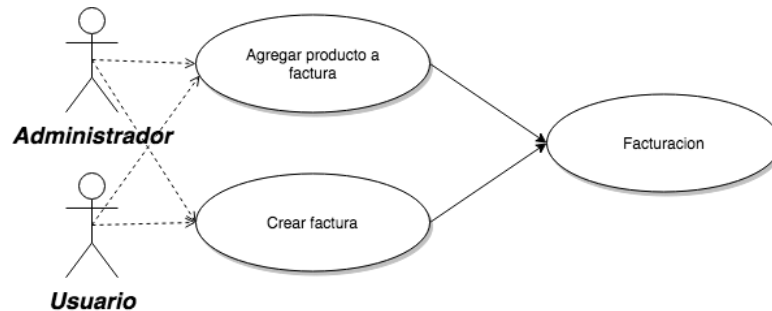
La Figura 3.3 presenta el diagrama de casos de uso para la gestión de inventarios.

Figura 3.3: Diagrama de Casos de Uso de Inventario



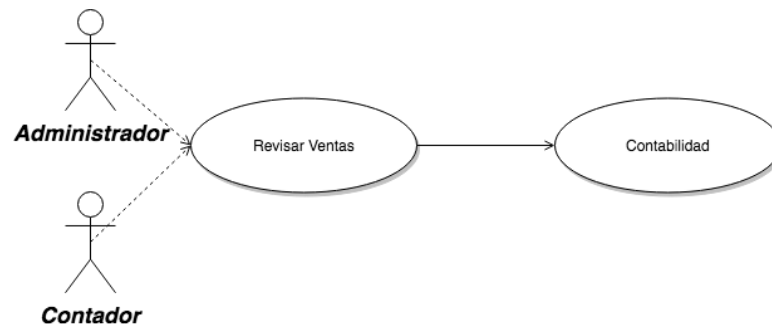
La Figura 3.4 presenta el diagrama de casos de uso para la gestión de facturación.

Figura 3.4: Diagrama de Casos de Uso de Facturación



La Figura 3.5 presenta el diagrama de casos de uso para la gestión de la contabilidad.

Figura 3.5: Diagrama de Casos de Uso de Contabilidad



### 3.1.2. Documentación de casos de uso

En la siguiente descripción se encontrarán las características propias de cada caso, al igual que el flujo de interacción de cada uno de ellos, de igual forma el flujo de eventos entre el actor y el sistema. La tabla 3.1 describe los casos de uso del sistema.

Tabla 3.1: Documentación de Caso de Uso

ID	Caso de uso 1
Nombre	<b>Administrador</b>
Descripción	El usuario de administración podrá interactuar con todos los módulos del sistema
Actor	Administrador
Precondición	El rol del usuario debe ser administrador
Poscondición	El usuario puede interactuar con todos los módulos del sistema

Eventos	Actor	Sistema
	1. Ingresa los datos de usuario	
		2. Valida la información
		3. Ingresa la información
		4. Muestra todos los modulos del sistema
Excepción	Ninguno	
Autor	Mario Guaqueta	
Fecha	Abril de 2017	
ID	Caso de uso 2	
Nombre	Ingreso	
Descripción	El usuario ingresa los datos para iniciar la sesión en el sistema.	
Actor	Administrador, Usuario, Contabilidad	
Precondicion	El usuario ingresará los datos para inciar la sesión.	
Poscondicion	El usuario podra ver los modulos dependiendo el rol.	
Eventos	Actor	Sistema
	1. El usuario ingresa los datos	
		2. Valida información del usuario
		3. Inicia sesión en caso exitoso
		4. Muestra los modulos dependiendo el rol
Excepción	Si los datos no son correctos no inicia sesión	
Autor	Mario Guaqueta	
Fecha	Abril de 2017	
ID	Caso de uso 3	
Nombre	Manejo de inventarios	
Descripción	El Administrador agrega o elimina productos al ivnentario, el usuario agregar productos a el inventario.	
Actor	Administrador, Usuario	
Precondicion	El administrador y el usuario deben estar autenticados.	
Poscondicion	El inventario es modificado	
Eventos	Actor	Sistema
	1. Agrega o elimina producto al inventario	
		2. Verifica que no exista o que exista dependiendo el caso
	3. Acepta agregar el producto e ingresa las cantidades	
		4. Almacena el producto en el inventario



<b>Excepción</b>	Si el producto no existe no puede ser eliminado, si el producto ya existe se suman las cantidades a las existentes.
<b>Autor</b>	Mario Guaqueta
<b>Fecha</b>	Abril de 2017

<b>ID</b>	<b>Caso de uso 4</b>
<b>Nombre</b>	<b>Facturación</b>
<b>Descripción</b>	El Administrador o el usuario agrega productos a la factura y procede a generar factura.
<b>Actor</b>	Administrador, Usuario
<b>Precondicion</b>	El administrador y el usuario deben estar autenticados, los productos deben tener existencias en el inventario.
<b>Poscondicion</b>	Se agregan productos a la factura

<b>Eventos</b>	<b>Actor</b>	<b>Sistema</b>
	1. Agrega o elimina producto a la factura	
		2. Verifica que si exista el producto, verifica que las cantidades sean mayores a las solicitadas.
		3. Agrega las cantidades a la factura
	4. Debe generar factura	
		5. Debe eliminar las cantidades de la base de datos

<b>Excepción</b>	Si no hay cantidades del producto este no sera agregado a la factura, o si las cantidades solicitadas son mayores tampoco sera agregado al sistema.
<b>Autor</b>	Mario Guaqueta
<b>Fecha</b>	Abril de 2017

<b>ID</b>	<b>Caso de uso 5</b>
<b>Nombre</b>	<b>Contabilidad</b>
<b>Descripción</b>	El Administrador o el usuario de contabilidad podran generar un reporte de ventas para su posterior revisión.
<b>Actor</b>	Administrador, Usuario de contabilidad
<b>Precondicion</b>	El administrador y el usuario deben estar autenticados.
<b>Poscondicion</b>	El usuario debe agregar las fechas en las cuales desea ver el reporte.

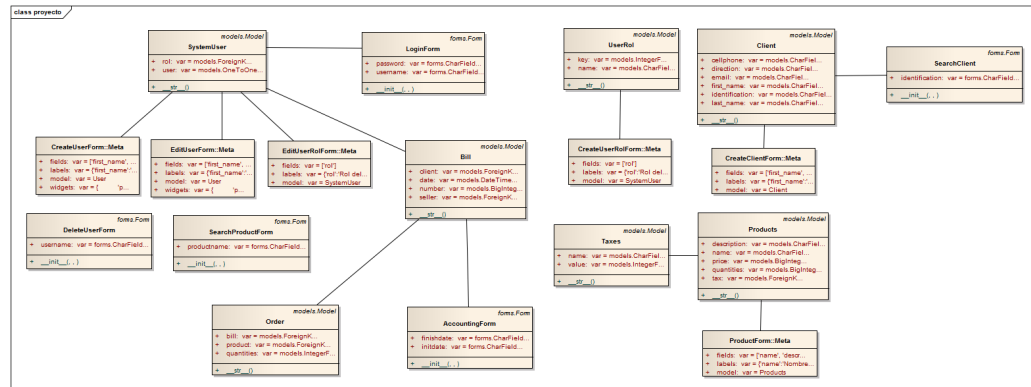
<b>Eventos</b>	<b>Actor</b>	<b>Sistema</b>
	1. Agrega las fechas de inicio y fin	
		2. Busca las ventas entre los rangos.
		3. Genera el reporte
	4. Decide que hacer con el reporte	
<b>Excepción</b>	Si no hay ventas en las fechas de filtro no generará nada.	
<b>Autor</b>	Mario Guaqueta	
<b>Fecha</b>	Abril de 2017	

### 3.2. Diagrama de Clases

Con base en los casos de uso y su documentación se crea el diagrama de clases que se presenta en la Figura 3.6. El diagrama de clases cuenta con las siguientes clases:

- **Usuario.** Esta clase representa los atributos del usuario.
- **Rol.** Esta clase representa los atributos del rol de cada usuario.
- **Facturas.** Esta clase representa los atributos de las facturas.
- **Productos.** Esta clase representa los atributos de los productos.
- **Djangoauth.** Esta clase representa la clase authentication de Django, la cual es una contribucion con el paquete.
- **Facturación.** Esta clase contiene los metodos y atributos que son utilizados al momento de realizar una factura.
- **Inventario.** Esta clase representa los atributos que son necesarios al momento de verificar los inventarios.
- **Editar usuario.** Esta clase representa los atributos necesarios y los servicios al momento de que un administrador va a editar un usuario.
- **Ingreso.** Esta clase representa los atributos necesarios para que un usuario pueda acceder a utilizar el sistema de información.

Figura 3.6: Diagrama de Clases



### 3.3. Diagramas de Secuencia

En las figuras ??, 3.7, 3.8 se observa la secuencia que sigue el proceso al momento de que el usuario, dependiendo de su rol, realiza una acción relacionada a su rol. Por otra parte se observan las secuencias que seguidamente el sistema al momento de realizar cada acción.

En el sistema se encuentran tres tipos de secuencia para cada tipo de usuario dependiendo su rol.

### 3.4. Diagramas de Componentes

Para el diagrama de componentes se visualizan los componentes que se encuentran en el sistema, los cuales son modificados por cada usuario y lo que puede realizar el usuario en el componente. De igual forma se realiza una relación de como funciona el servicio.

En la figura 3.9 se aprecia el diagrama de componentes del cual esta hecho el sistema de información, incluyendo su capa de persistencia.

### 3.5. Diagrama de relaciones

Para el diagrama de relaciones, que se observa en la figura 3.10 , se encuentran las relaciones que se van a dar en la base de datos del sistema.

### 3.6. Diagrama BPMN

Para los diagramas de BPMN se observan cuatro diagramas, los cuales muestran el flujo de los procesos de negocio, para cada actividad realizada y el usuario encargado.

En la figura 3.11 se observa el diagrama BPMN para la actividad de facturación en la cual el usuario puede ser el usuario normal o el administrador, involucrando otro actor denominado cliente, el cual da inicio a la actividad. Al momento de agregar el producto a la factura se valida si hay existencias de este producto y las existencias que se tengan sean mayores o iguales a las necesarias por el cliente.

Figura 3.7: Diagrama de secuencia del usuario

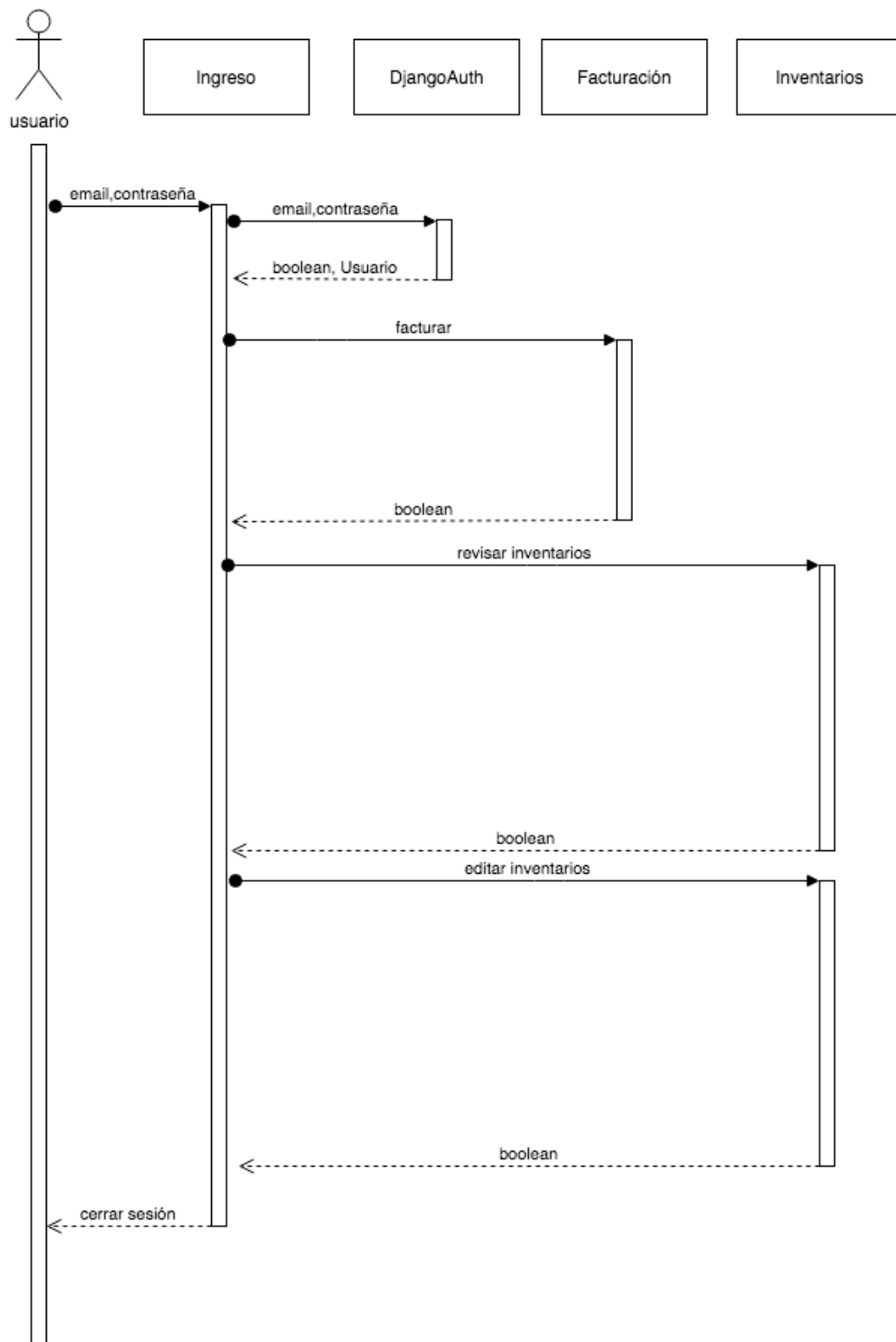


Figura 3.8: Diagrama de secuencia del usuario de contabilidad

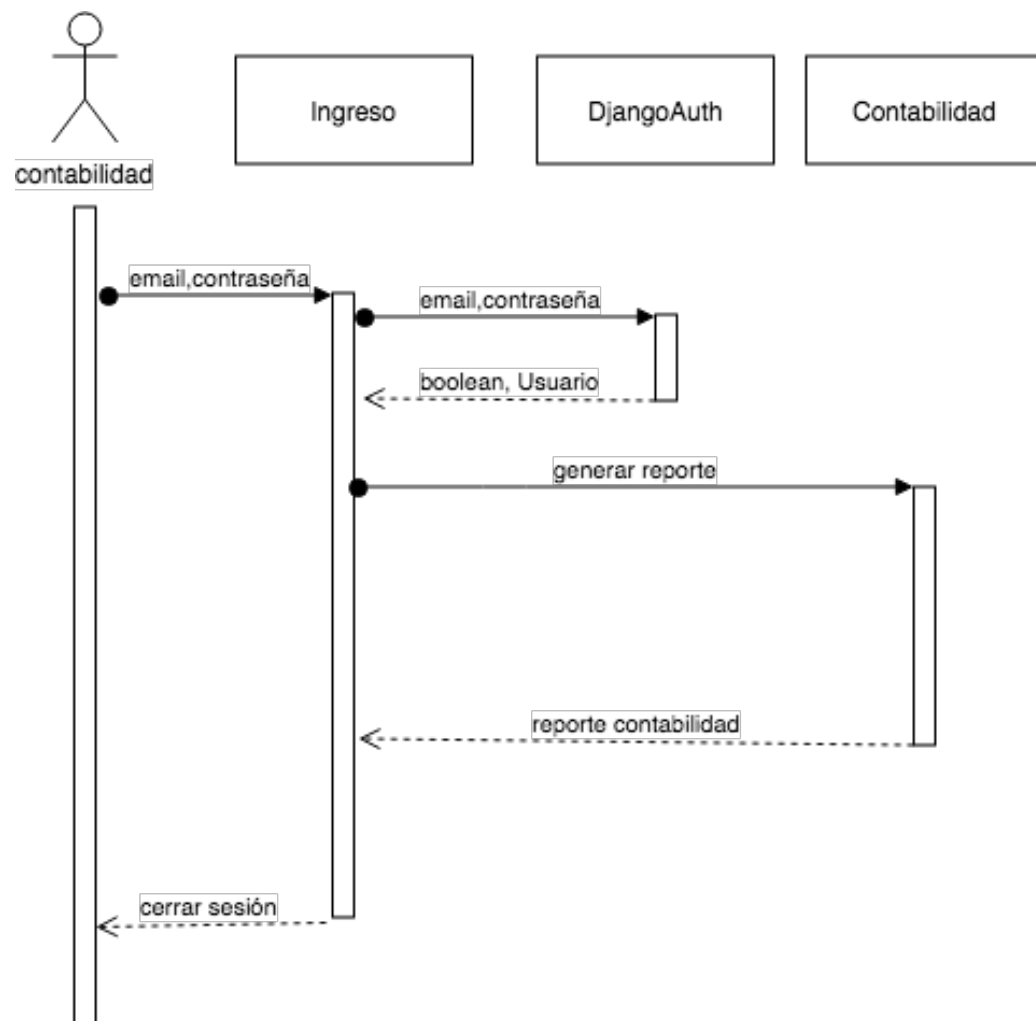


Figura 3.9: Diagrama de componentes del sistema de información.



Figura 3.10: Diagrama de relaciones del sistema de información.

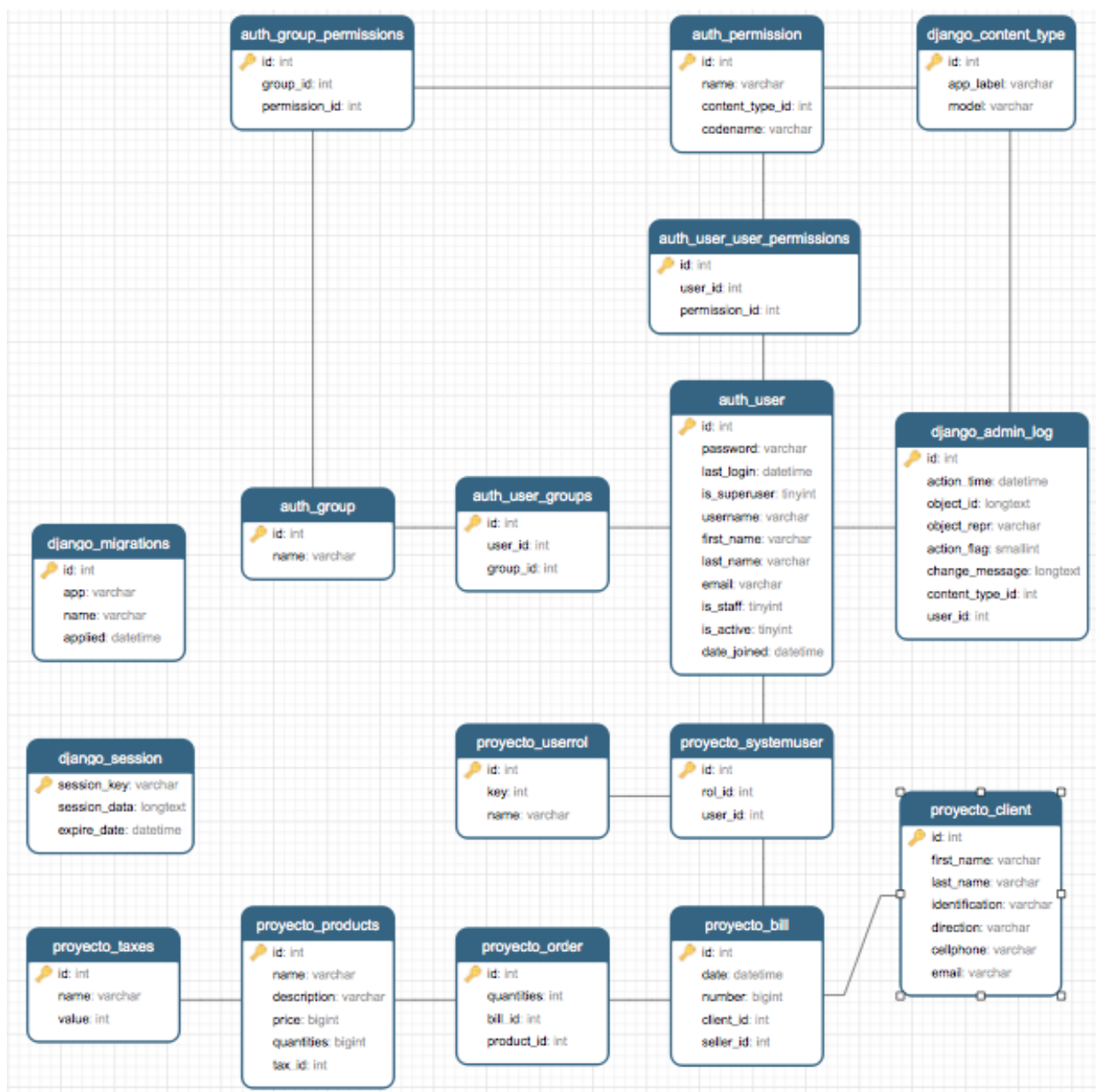
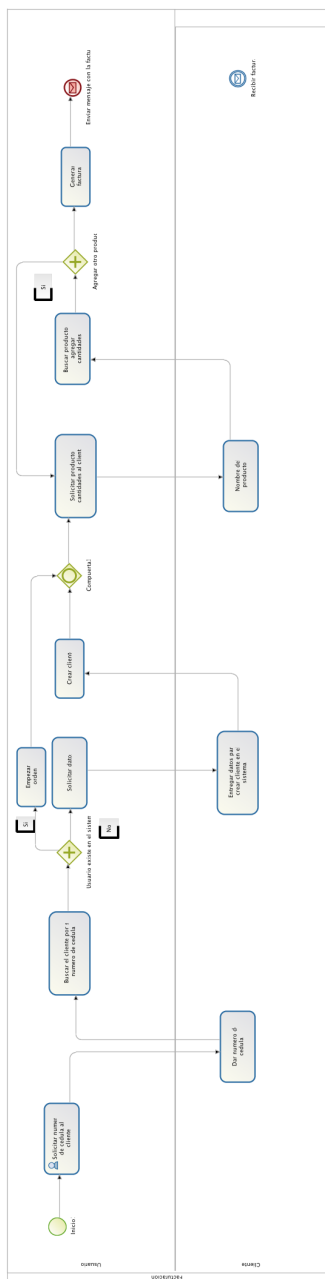


Figura 3.11: Diagrama BPMN del proceso de facturación





## Capítulo 4

# Implementación

### 4.1. Modelo Relacional

#### 4.1.1. Relaciones del sistema:

Para el desarrollo del sistema se hacen necesarias las siguientes relaciones en base a el modelo de negocio:

- **Relación usuario con roles:** Se relaciona el usuario que trae por defecto el framework con unos roles del sistema los cuales proporcionan la funcionalidad que va a tener el sistema dependiendo el rol.
- **Relación usuario con factura:** Se relaciona el usuario con la factura ya que se hace necesario mantener la información de quien vendio que producto.
- **Relación factura con producto:** La relación de factura con producto se hace necesario implementar la clase orden la cual va a tener esta relación.
- **Relación producto con impuesto:** La relación de producto con impuesto se hace necesaria ya que se van a tener diferentes impuestos para cada producto.
- **Relación cliente con factura:** Cada factura debe tener obligatoriamente un cliente, ya que este va a ser entregado al cliente.

### 4.2. Diccionario de Datos

#### 4.2.1. Clase roles de usuario:

En la clase roles de usuario se encuentran los siguientes datos:

- **Key:** Es la llave dependiendo el rol, este dato es de tipo entero, para este sistema se definio 1 para administrador, 2 para vendedor y 3 para contabilidad.
- **name:** Es el nombre del rol.

#### 4.2.2. Clase usuario del sistema:

En la clase usuario del sistema se encuentran los siguientes datos:

- **user:** Es la relación uno a uno con el modelo de autenticación de Django, que facilita la autenticación y el modelo.
- **rol:** Es la relación foranea con la clase roles de usuario, en donde cada usuario tendra un rol.

#### 4.2.3. Clase cliente del sistema:

En la clase cliente del sistema se encuentran los siguientes datos:

- **first\_name:** Es el campo de texto de los nombres del cliente.
- **last\_name:** Es el campo de texto de los segundos nombres del cliente.
- **identification:** Es el campo de texto de la identificación del cliente.
- **direction:** Es el campo de texto de la dirección del cliente.
- **cellphone:** Es el campo de texto del celular del cliente.
- **email:** Es el campo de texto del email del cliente.

#### 4.2.4. Clase impuestos del sistema:

En la clase impuestos del sistema se encuentran los siguientes datos:

- **name:** Es el campo de texto del nombre del impuesto.
- **value:** Es el campo de tipo entero del valor del impuesto.

#### 4.2.5. Clase productos del sistema:

En la clase productos del sistema se encuentran los siguientes datos:

- **name:** Es el campo de texto del nombre del producto.
- **description:** Es el campo de texto de la descripción del producto
- **price:** Es el campo de tipo entero de gran tamaño del precio del producto
- **quantities:** Es el campo de tipo entero de las cantidades del producto
- **tax:** Es la relación foranea con la clase impuestos, ya que cada producto debe llevar un impuesto.

#### 4.2.6. Clase ordenes del sistema:

En la clase ordenes del sistema se encuentran los siguientes datos:

- **bill:** Es la relación foranea entre una orden y una factura, ya que la factura tiene varias ordenes.
- **product:** Es la relación foranea ya que la orden debe tener un producto.
- **quantities:** Es el campo de tipo entero de las cantidades de la orden.

#### 4.2.7. Clase facturas del sistema:

En la clase facturas del sistema se encuentran los siguientes datos:

- **client:** Es la relación foranea entre una factura y un cliente, ya que la factura tiene un cliente que esta solicitando la factura.
- **seller:** Es la relación foranea de usuario, ya que la factura debe tener un usuario quien vendio el producto o los productos.
- **number:** Es el campo de tipo entero que llevara el numero de facturas que se tienen para establecer un orden y unas referencias que pueden ser utilizadas

## Capítulo 5

# Manual de Instalación

### 5.1. Instalación de python

Se debe instalar python en el sistema operativo, ya que es el lenguaje en el cual se encuentra escrita la aplicación. Para la instalación del mismo se debe acudir a la siguiente dirección de enlace:

**Python**

### 5.2. Instalación de pip

Para la instalación de pip se hace necesario tener python, lo cual seria el segundo paso a seguir, este se puede instalar, una vez instalado python, con la siguiente linea de código desde el simbolo del sistema en windows o el terminal en mac o linux.

**python pip-get.py**

Mas referencias en: **Documentación de PIP**

### 5.3. Instalación de virtualenv

Para la instalación se hace necesario haber ejecutado el anterior comando e instalado **PIP** ya que mediante esta herramienta se puede instalar virtualenv mediante el siguiente comando en el simbolo del sistema o el terminal en mac o linux:

**pip install virtualenv -p python3**

Lo ultimo se hace necesario ya que en la aplicación web se utiliza python 3.

## 5.4. Instalación de MySQL server

Para la instalación de MySQL server se hace necesario utilizar la herramienta **PIP** o **BREW** dependiendo el sistema operativo, esta herramienta se hace necesaria para el manejo de bases de datos. Una vez instalado el motor de bases de datos se hace necesario crear una base de datos con el nombre de **ingsoft** en localhost por el puerto 3306 y un usuario con los siguientes datos:

- **Nombre de usuario:** ingsoft
- **Contraseña:** abc123456

Por otra parte en nuestro virtualenv se hace necesario instalar MySQL cliente para poder dar uso a esa base de datos.

## 5.5. Instalación de Apache

Por ultimo se debe instalar el servidor http apache para poder tener el sistema conectado con apache y no estar monitoreandolo cada momento.

Para esto se hace necesario ver el siguiente **tutorial**

En este tutorial se explica como se configuran los puertos, como se configuran los archivos estaticos y los permisos para que apache pueda acceder a la aplicación.

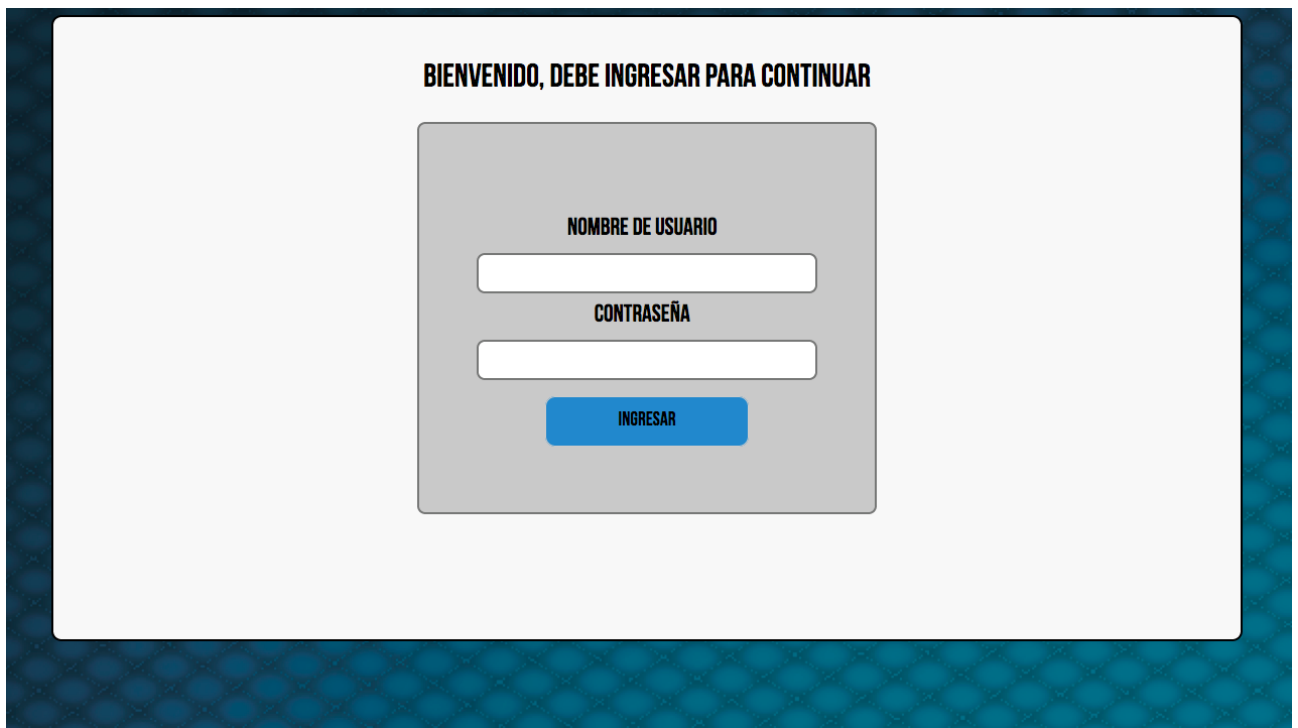
## Capítulo 6

# Manual de Usuario

### 6.1. Inicio de sesión

En la figura 6.1 se observa la pagina principal, en donde el usuario podra dar inicio a su sesión por medio de un nombre de usuario asignado y una contraseña.

Figura 6.1: Inicio de sesión de usuario

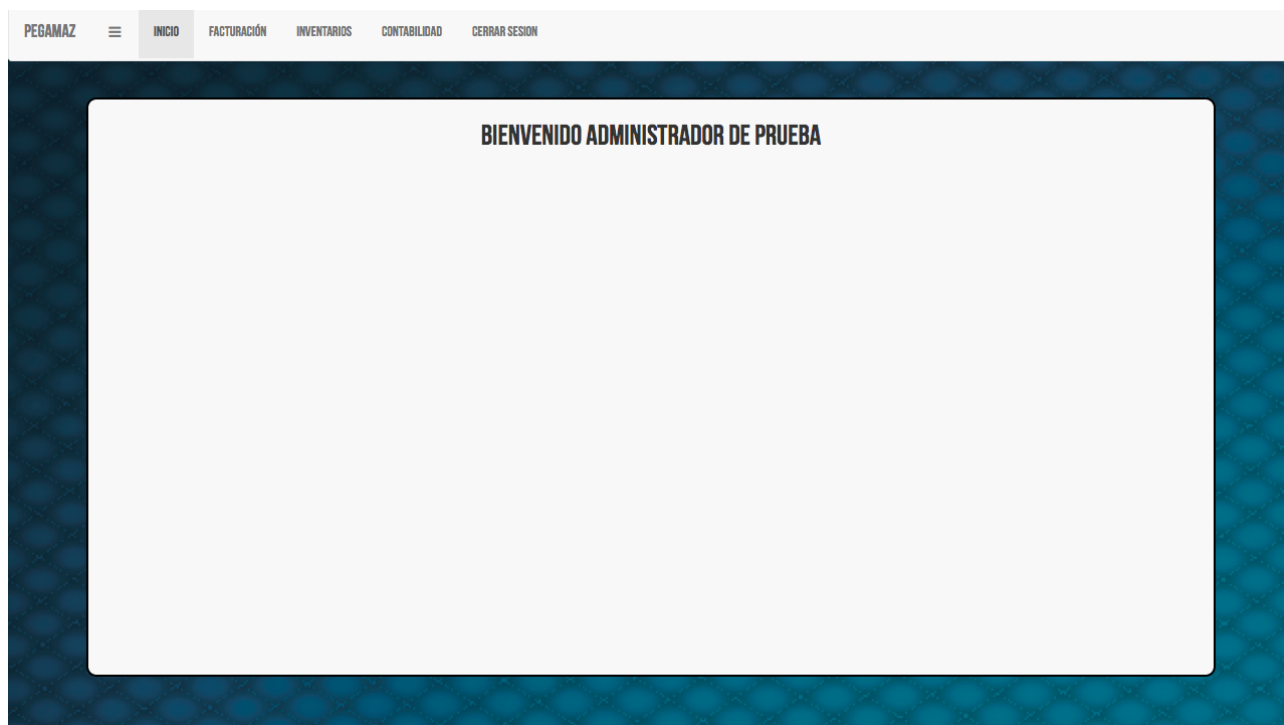


The image shows a login form interface. At the top, the text "BIENVENIDO, DEBE INGRESAR PARA CONTINUAR" is displayed in bold. Below this, there is a gray rectangular box containing the login fields. Inside the box, the label "NOMBRE DE USUARIO" is above a white input field. Below that, the label "CONTRASEÑA" is above another white input field. At the bottom of the gray box is a blue button with the text "INGRESAR" in white. The entire form is set against a light gray background, which is itself within a larger frame with a dark blue, patterned border.

## 6.2. Pagina inicial

En la figura 6.2 se observa la pagina principal del sistema, en donde se muestran todas las funcionalidades dependiendo el usuario, para el administrador quien tiene el control total de la aplicación se muestra el menú de administración ??, por otra parte si los usuarios son de tipo vendedor o contabilidad veran sus modulos.

Figura 6.2: Pagina de inicio

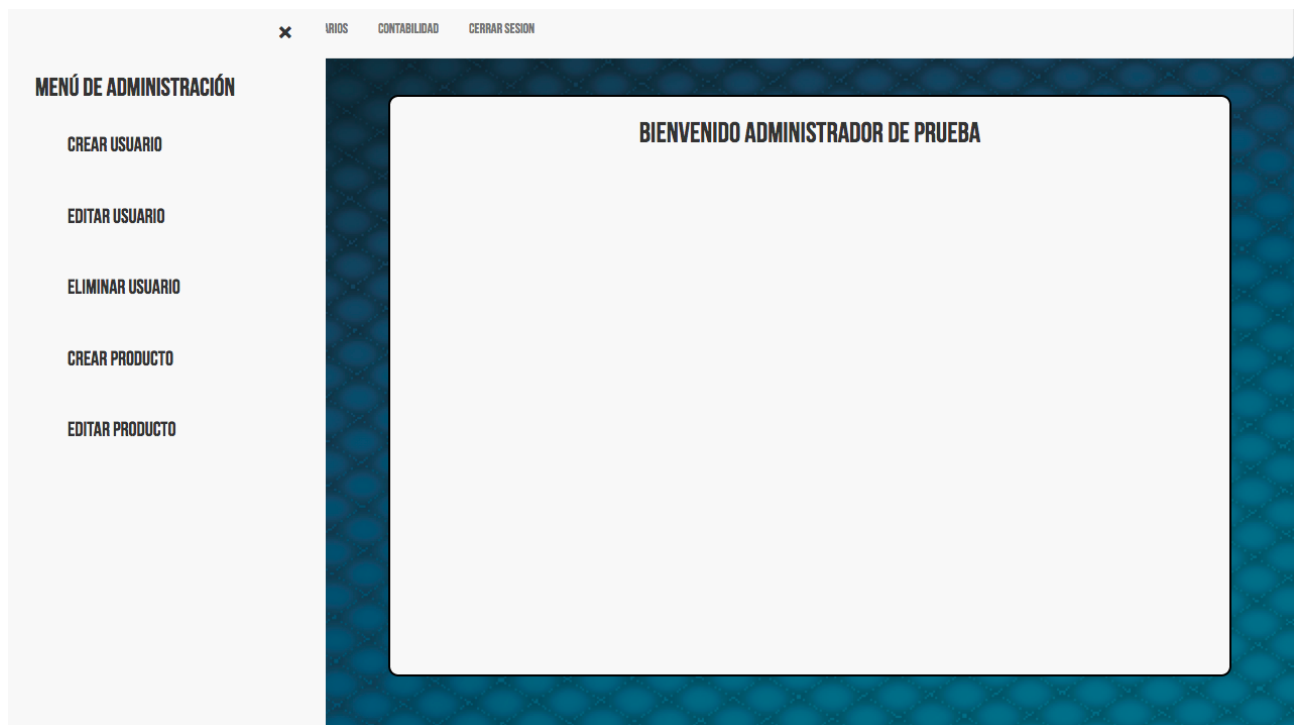


### 6.3. Menú de administración

En la figura 6.3 se observa la pagina principal del sistema, en este caso en especifico se observa la pagina inicial del administrador, el cual tiene todas las funcionalidades, incluida el menu de administración en donde se implementan las funcionalidades de:

- Crear Usuario
- Editar Usuario
- Eliminar Usuario
- Crear Producto
- Editar Producto

Figura 6.3: Pagina de inicio





## 6.4. Facturación

En la figura 6.4 se observa la pagina de facturación en donde se debe primero solicitar al cliente el número de cedula para dar inicio a la facturación, posteriormente si el usuario existe se genera la factura, pero si no existe se debe crear el usuario, posteriormente iniciar la busqueda de productos en el formulario, estos productos se buscan por su nombre 6.5 , posteriormente si el sistema arroja productos por ese nombre buscado se deben agregar las cantidades y luego agregar a la factura, se observara abajo la tabla de la orden que se tiene hasta el momento como en la figura 6.6, por ultimo una vez se culmine la factura se presiona el boton de generar factura y se enviara un mensaje al correo del cliente con el resumen de la compra.

Figura 6.4: Facturación

PEGAMAZ

INICIO FACTURACIÓN INVENTARIOS CONTABILIDAD CERRAR SESION

Número de identificación del cliente\*

Buscar

Figura 6.5: Buscador de productos

PEGAMAZ

INICIO FACTURACIÓN INVENTARIOS CONTABILIDAD CERRAR SESIÓN

Busca el producto por su nombre

Nombre del producto\*

Buscar

NOMBRE	DESCRIPCIÓN	PRECIO SIN IVA	CANTIDAD	AGREGAR
Pegante 1	Pegante para suela	15000	<input type="text"/>	AGREGAR A PEDIDO
Pegante 2	Pegante para tela	10000	<input type="text"/>	AGREGAR A PEDIDO
Pegante 3	Pegante para cuero	12000	<input type="text"/>	AGREGAR A PEDIDO

Tu compra

PRODUCTO	DESCRIPCIÓN	PRECIO SIN IVA	CANTIDADES
----------	-------------	----------------	------------

## 6.5. Inventarios

En la figura 6.7 se observa la pagina de inventarios en donde se observaran todos los productos en el sistema, su descripción, su precion y sus cantidades para que los usuarios lleven el inventario.

Figura 6.6: Tabla de orden, con productos agregados

PEGAMAZ

INICIO

FACTURACIÓN

INVENTARIOS

CONTABILIDAD

CERRAR SESIÓN

Busca el producto por su nombre

Nombre del producto\*

Buscar

Tu compra

PRODUCTO	DESCRIPCIÓN	PRECIO SIN IVA	CANTIDADES
Pegante 1	Pegante para suela	15000	100

TOTAL

1,502,400.0

GENERAR FACTURA

## 6.6. Contabilidad

En la figura 6.8 se observa la pagina de contabilidad en la cual el usuario podra observar las facturas dentro de unas fechas establecidas mediante el buscador6.9.

Figura 6.7: Inventarios

**Inventario Existente**

PRODUCTO	DESCRIPCIÓN	PRECIO SIN IVA	CANTIDADES EXISTENTES
Pegante 1	Pegante para suela	15000	19
Pegante 2	Pegante para tela	10000	8
Pegante 3	Pegante para cuero	12000	200

127.0.0.1:8000/inventories

Figura 6.8: Contabilidad

PEGAMAZ    INICIO    FACTURACIÓN    INVENTARIOS    CONTABILIDAD    CERRAR SESIÓN

## Contabilidad

Fecha Inicial\*

yyyy-mm-dd

Fecha Final\*

yyyy-mm-dd

Buscar

127.0.0.1:8000/accounting

Figura 6.9: Facturas encontradas dentro de un rango de fechas

PEGAMAZ				
<div> <div></div> <div>INICIO</div> <div>FACTURACIÓN</div> <div>INVENTARIOS</div> <div>CONTABILIDAD</div> <div>CERRAR SESIÓN</div> </div>				
Contabilidad				
CLIENTE		FECHA	VENDEDOR	NÚMERO DE FACTURA
NOMBRE	IDENTIFICACIÓN	June 7, 2017, 8:35 p.m.	Administrador De Prueba	10
Jorge Mario	1032459576			
NOMBRE	IDENTIFICACIÓN	June 7, 2017, 10:24 p.m.	Jorge Mario Guaqueta Restrepo	11
Jorge Mario	1032459576			
NOMBRE	IDENTIFICACIÓN	June 9, 2017, 10:35 p.m.	Administrador De Prueba	12
Jorge Mario	1032459576			

# Conclusiones

- Utilizar Django Web Framework como herramienta para el desarrollo del sistema de información, ya que esta herramienta permite realizar modificaciones a los modelos de manera rapida, permite utilizar plantillas para ser adaptadas a los proyectos y a su necesidad.
- Utilizar MySQL como administrador de bases de datos, ya que es un servicio implementable en Django y facilmente utilizable, con bajos costos de utilización.
- Django web framework permite implementar una misma plantilla para toda la pagina web y cambiar su contenido y funcionalidad mediante la etiqueta **extends**, de igual forma, este permite incluir plantillas que se hagan necesarias dependiendo la pagina, mediante la etiqueta **include**
- Mediante Django se disminuye el riesgo de SQLInjection el cual es un riesgo de seguridad, ya que Django tiene la herramienta QuerySet la cual permite hacer las solicitudes a la capa de persistencia sin necesidad de utilizar consultas creadas por el usuario.
- Mediante Django se disminuye el riesgo en autorización y autenticación ya que este exhibe el modulo de autenticación en el cual es usuario se autentica sin necesidad de realizar consultas a la capa de persistencia directamente desde la vista. Por otra parte Django exhibe el servicio de manejo de sesiones para que el usuario valide si esta o no autenticado.

# Bibliografía

- [1] W. Lara, “Scrum: Cómo funciona la metodología de trabajo scrum.”
- [2] SOFTTENG, “Metodología scrum para desarrollo de software aplicaciones complejas.”