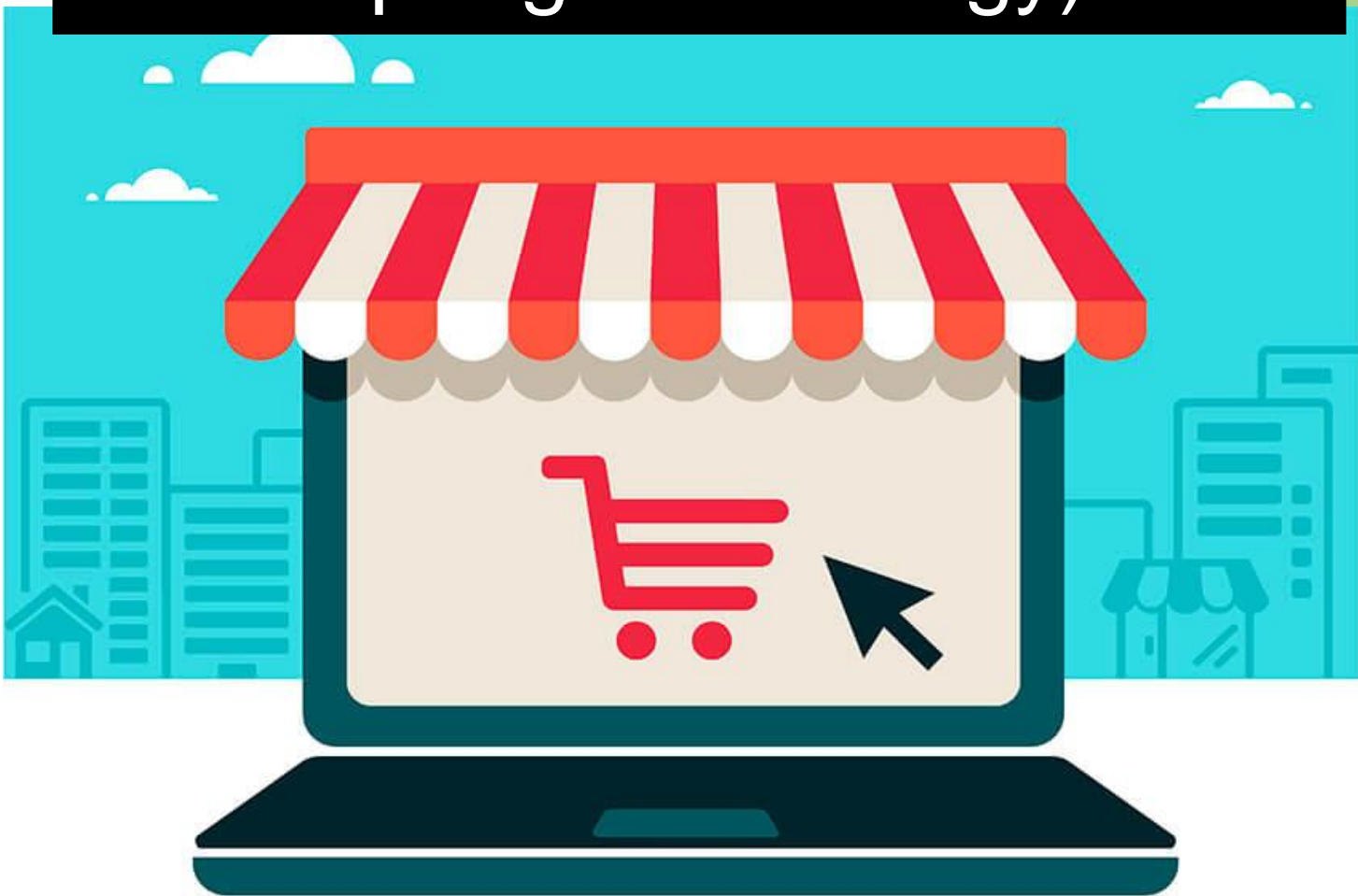


Tienda web Online (by Spring Technology)



Miguel Dominguez Garcia

| | |
|--|-----------|
| CAPÍTULO 1: INTRODUCCIÓN | 3 |
| 1.1. Introducción del Proyecto | 3 |
| 1.2. Propósito | 3 |
| 1.3. Objetivos del Proyecto | 3 |
| 1.4. COSTES del Proyecto | 4 |
| 1.4.1.1. Recurso Informáticos Hardware | 4 |
| 1.4.1.2. Recursos Informáticos Software | 4 |
| 1.4.1.3. Gastos de Personal | 4 |
| 1.4.1.4. Costes de Implantación | 4 |
| CAPÍTULO 2: ANÁLISIS DEL SISTEMA | 5 |
| 2.1. Introducción..... | 5 |
| 2.2. Análisis de Requisitos..... | 5 |
| 2.2.1. Requerimientos Funcionales | 6 |
| 2.2.2. Requerimientos NO Funcionales | 6 |
| 2.3. CASOS de USO: Diagramas y Narrativas de Casos de Uso | 7 |
| 2.3.1. Caso de uso 1 | 9 |
| 2.3.2. Caso de uso 2 | 9 |
| 2.3.3. Caso de uso 3 | 9 |
| 2.3.4. Caso de uso 4 | 10 |
| 2.3.5. Caso de uso 5 | 10 |
| 2.3.6. Caso de uso 6 | 10 |
| CAPÍTULO 3: DISEÑO DEL SISTEMA | 11 |
| 3.1. Introducción..... | 11 |
| 3.2. Diagrama de Clases | 11 |
| 3.3. Diseño de la Base de Datos | 12 |
| 3.4. Diseño Lógico | 12 |

| | | |
|--|---|----|
| 3.5. | Diseño Conceptual | 13 |
| 3.6. | DDL y DML estándar relacional para la BD. | 14 |
| 3.7. | Diseño de la Interfaz | 15 |
| CAPÍTULO 4: IMPLEMENTACIÓN | | 18 |
| 4.1. | Introducción..... | 18 |
| 4.2. | Arquitectura Cliente/Servidor | 18 |
| 4.3. | Lenguajes de Programación | 19 |
| 4.4. | Herramientas de Desarrollo | 19 |
| 4.5. | Codificación..... | 19 |
| CAPÍTULO 5: PRUEBAS DE SOFTWARE..... | | 20 |
| 5.1. | Introducción..... | 20 |
| 5.2. | Técnicas de Pruebas | 20 |
| 5.2.1. | Pruebas de CAJA BLANCA o Enfoque Estructural | 20 |
| 5.2.2. | Pruebas de CAJA NEGRA o Enfoque Funcional..... | 20 |
| CAPÍTULO 6: CONCLUSIONES | | 21 |
| 6.1. | Conclusiones | 21 |
| 6.2. | Propuestas Futuras (Opcional) | 21 |
| CAPÍTULO 7: MANUAL DE INSTALACIÓN | | 22 |
| 7.1. | Creación de un VPS(Virtual Private Server) | 22 |
| 7.2. | Generación del archivo .jar | 23 |
| 7.3. | Configuración en Ubuntu 18.04..... | 25 |
| 7.4. | Instalar la aplicación como un servicio systemd | 25 |
| CAPÍTULO 8: MANUAL DE USUARIO | | 26 |
| CAPÍTULO 9: GLOSARIO DE TÉRMINOS Y ACRÓNIMOS | | 27 |
| CAPÍTULO 10: REFERENCIAS BIBLIOGRÁFICAS (WEBGRAFÍA)..... | | 28 |

Capítulo 1: **INTRODUCCIÓN**

1.1. Introducción del Proyecto

Generación de una tienda web online de bajo coste.

El modelo que se usará es una tienda típica de venta online, que en este caso comercializa hardware y software de terceros.

1.2. Propósito

Una vez desarrollado en su completitud debería ser factible su posible venta como proyecto de despliegue rápido y con un coste reducido debido a las tecnologías implicadas.

1.3. Objetivos del Proyecto

- Realizar un estudio de los requisitos necesarios
- Desarrollo de una aplicación que dé como resultado una aplicación web funcional.
- Desarrollar una interfaz sencilla e intuitiva de cara al usuario.
- Redactar una memoria del proyecto
- Redactar el manual de instalación de la aplicación.
- Redactar el manual de implantación en cloud.
- Redactar el manual de usuario.
- Aprender la utilización de la tecnología Spring basado en Java, para usarlo en generación de páginas web.

1.4. COSTES del Proyecto

El coste y esfuerzo que se requiere para la realización de este proyecto está basado en un trabajo individual de entrenamiento con la herramienta de desarrollo ya que se parte de un desconocimiento previo de la herramienta cuya curva de aprendizaje requiere de un gran esfuerzo de comprensión, pero que a medio y largo plazo ofrece continuidad en su soporte y mejoras.

Los costes relacionados con el hardware y software a utilizar se describen en el siguiente apartado.

1.4.1.1. Recurso Informáticos Hardware

- Equipo portátil para entorno de desarrollo con conexión a Internet.
- Conexión de 100 Mbps simétrico para Crear un entorno virtualizado de desarrollo en local
- Cuenta en el cloud con un saldo mínimo para pruebas, posteriormente cuando esté operativa la página se pasará los cargos de uso.

1.4.1.2. Recursos Informáticos Software

- Generar un VPS o servidor virtualizado en un cloud.
- Crear un entorno virtualizado basado en Ubuntu Linux de desarrollo en local
- Open Source: Java SE 8
- IDE Spring Tool Suite

1.4.1.3. Gastos de Personal

En principio los costes de personal asociado al desarrollo e implantación están orientados 1 o 2 personas puesto que el sistema estará cerrado y desplegado en no más de 1 semana.

1.4.1.4. Costes de Implantación

Estos costes están en relación a un esquema de despliegue rápido y automatizado por lo que los costes de implantación estarán asociados a la negociación de la contratación del servidor virtualizado y una semana laboral de 40 horas para la puesta en marcha por parte de un técnico especializado.

Capítulo 2: **ANÁLISIS DEL SISTEMA**

2.1. Introducción

Se realizará un análisis lo más exhaustivo posible de las necesidades que el web site necesita en el momento actual del desarrollo.

Se tendrá en cuenta la ampliación del producto desarrollado conforme vayamos detectando nuevos requerimientos.

2.2. Análisis de Requisitos

Al finalizar el proyecto debe ser capaz de tener una web online de compras operativa que incluya entre otras funciones:

- Acceso en modo anónimo a la web.
- Acceso en modo identificado.
- Acceso de administrador para gestionar cuentas de usuario.
- Acceso de administrador para gestión de productos y categorías.
- Crear tablas y productos con un almacenamiento permanente.
- Simulación de las ventas de un producto para calcular la puntuación de aceptación de dicho producto.
- Gestión de un carrito de compras con acumulación de productos.
- Gestionar el envío, a través de tercero, del paquete generado.
- Seguimiento del envío.
- Pago mediante TPV virtual, Paypal u otros medios de pago.
- Facturación del pedido realizado
-

2.2.1. Requerimientos Funcionales

Describimos las funciones elementales que utilizara esta versión de la aplicación:

- Acceso sin ninguna restricción a cualquier usuario, en modo consulta de los productos.
- Almacén de productos operativo y funcionando.
- Almacén de categorías operativo y funcionando.
- Algún medio de cálculo de la puntuación del producto sobre las ventas
- CRUD (Creación, Lectura, Modificación y Borrado) en el VPS de los almacenes anteriores.

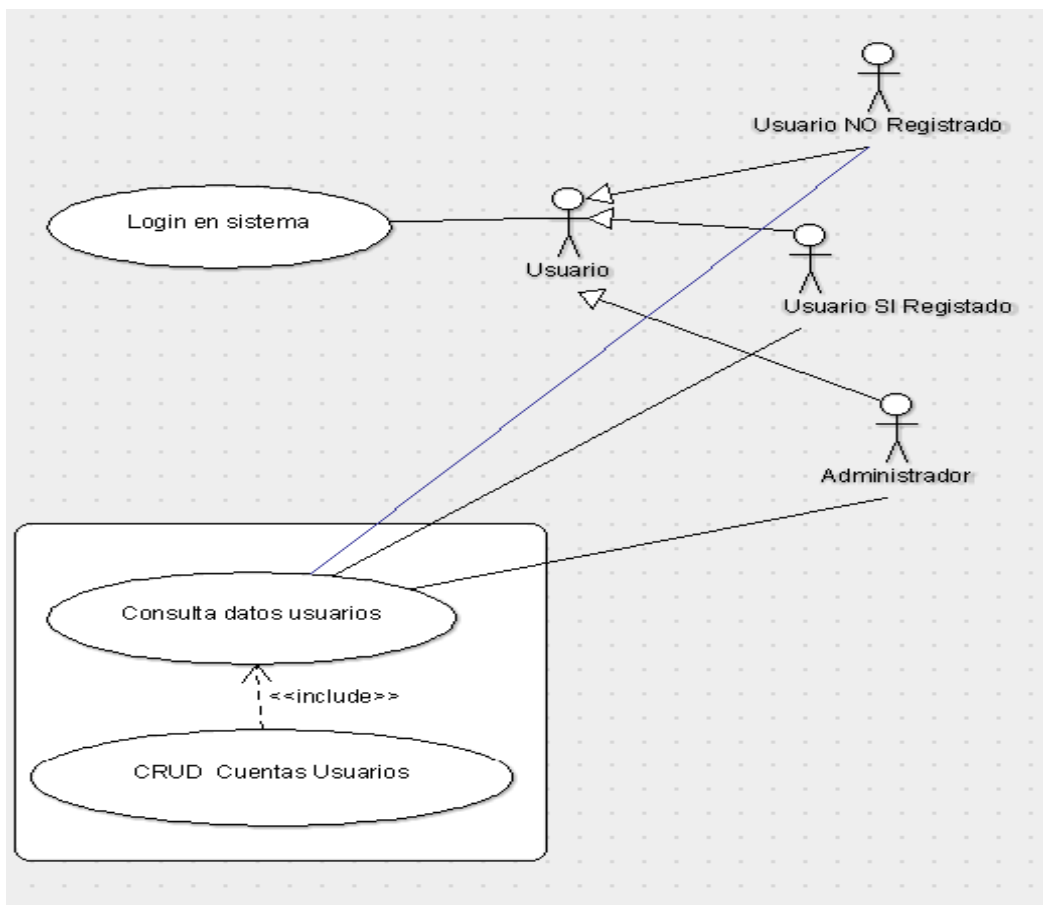
2.2.2. Requerimientos NO Funcionales

Aún no he identificado los requerimientos NO funcionales, en cuanto tenga conciencia de ellos los presentaremos en esta sección.

- A
- B
- C

2.3. CASOS de USO: Diagramas y Narrativas de Casos de Uso

Diagrama de caso de uso Acceso a la Web para Gestión de Usuarios



ERS (Especificación Requisitos del Sistema)

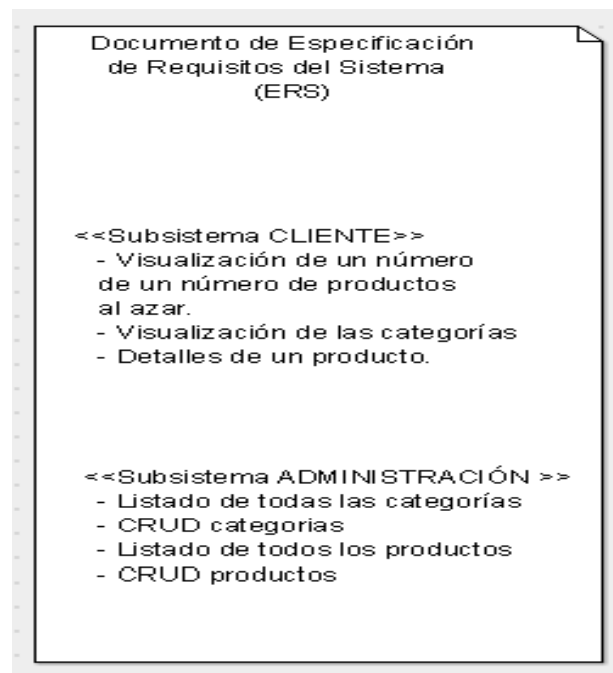
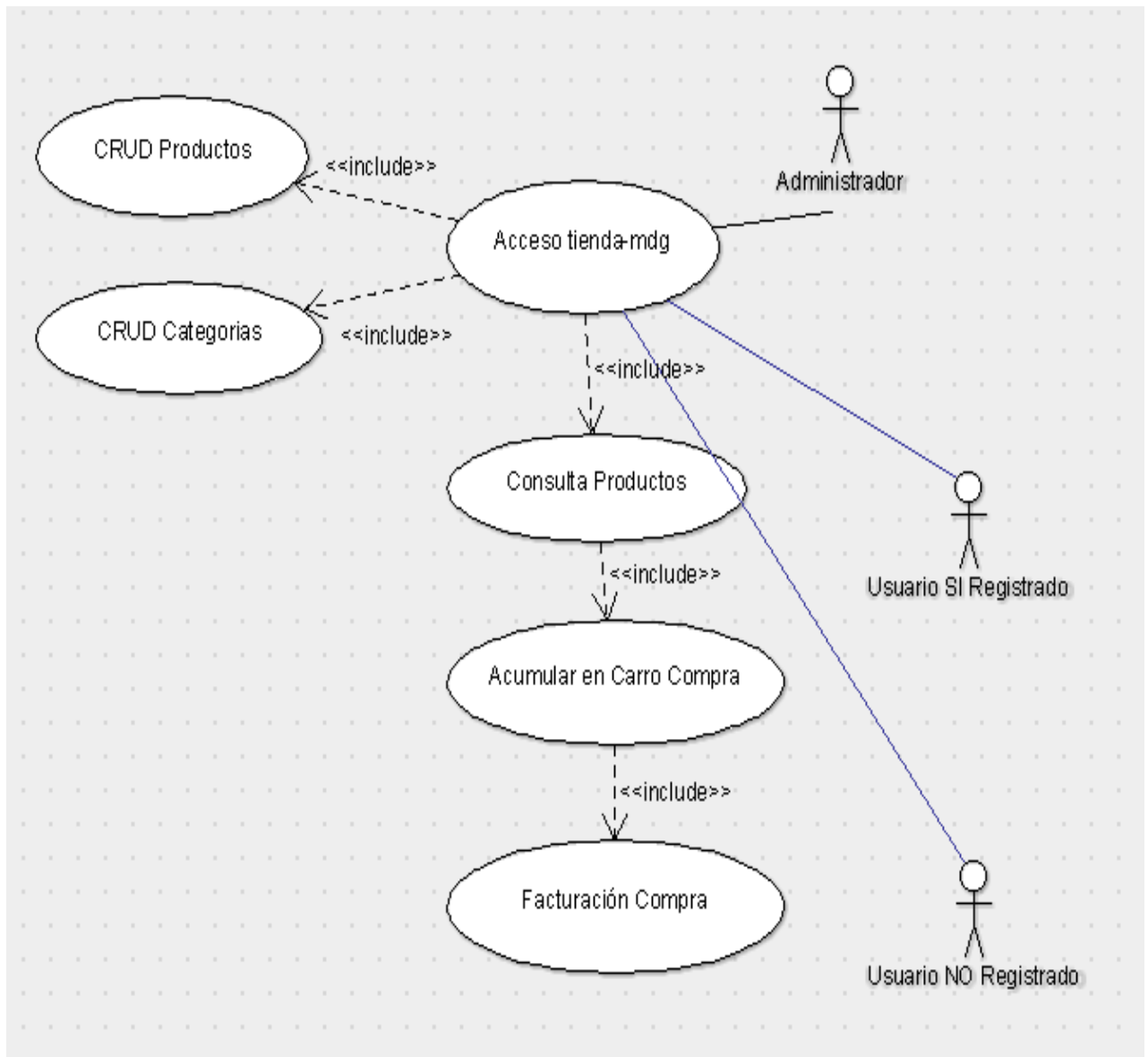


Diagrama de caso de uso Acceso a la Web para Gestión de Productos, Categorías, Compras de productos y facturación.



2.3.1. Caso de uso 1

Nombre del caso de uso: Login en el sistema, entrada a la web en modo anónimo sin identificar.

Actor principal: Usuario NO Registrado

Condiciones de entrada: el usuario no está registrado en el sistema.

Flujo de eventos:

1. A
2. B
3. C

2.3.2. Caso de uso 2

Nombre del caso de uso: Login en la web, entrada a la web con usuario y contraseña.

Actor principal: Usuario SI Registrado

Condiciones de entrada: el usuario si está registrado en el sistema y poseemos un perfil del usuario.

Flujo de eventos:

1. A
2. B
3. C

2.3.3. Caso de uso 3

Nombre del caso de uso: Login en el sistema entrada a la web con usuario y contraseña.

Actor principal: Administrador

Condiciones de entrada: verificar la autenticidad del administrador que accede con permisos de lectura/escritura sobre la base de datos que contiene los usuarios y sus perfiles.

Flujo de eventos:

1. A
2. B
3. C

2.3.4. Caso de uso 4

Nombre del caso de uso: Login en el sistema entrada a la web con usuario y contraseña.

Actor principal: Administrador

Condiciones de entrada: verificación de la autenticidad del administrador

Flujo de eventos:

1. A
2. B
3. C

2.3.5. Caso de uso 5

Nombre del caso de uso: Login en el sistema entrada a la web con usuario y contraseña.

Actor principal: Administrador

Condiciones de entrada: verificación de la autenticidad del administrador

Flujo de eventos:

1. A
2. B
3. C

2.3.6. Caso de uso 6

Nombre del caso de uso: Login en el sistema entrada a la web con usuario y contraseña.

Actor principal: Administrador

Condiciones de entrada: verificación de la autenticidad del administrador

Flujo de eventos:

1. A
2. B
3. C

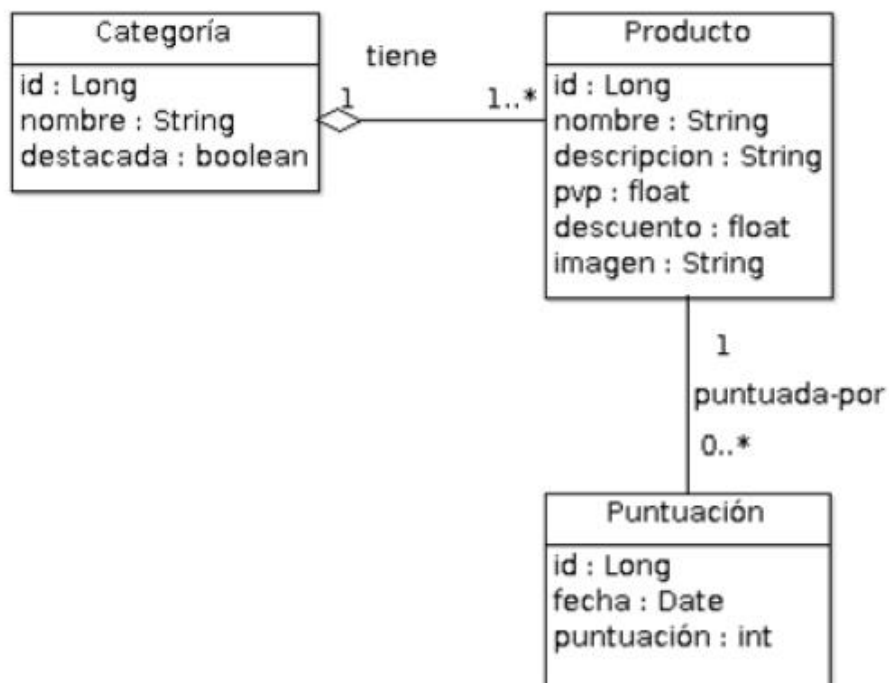
Capítulo 3: DISEÑO DEL SISTEMA

3.1. Introducción

Para llevar el diseño de esta tienda online se ha empezado el ejercicio de creación de categoría y productos con una puntuación del cliente sobre este último en base a su satisfacción en la compra. Como hemos comentado anteriormente la puntuación está simulada.

3.2. Diagrama de Clases

Diseño UML de las clases involucradas en la creación de este proyecto.



3.3. Diseño de la Base de Datos

La estructura de las tablas implicadas en el actual estado del desarrollo es bastante simple ya que sólo incluye 3 elementos: categoría, producto y puntuación.

Las dos primeras son las que permiten la creación del almacén permanente de la tienda.

La tercera es un añadido para puntuar el producto cuando se venda, de momento esto está simulado ya que no se realizan ventas del producto.

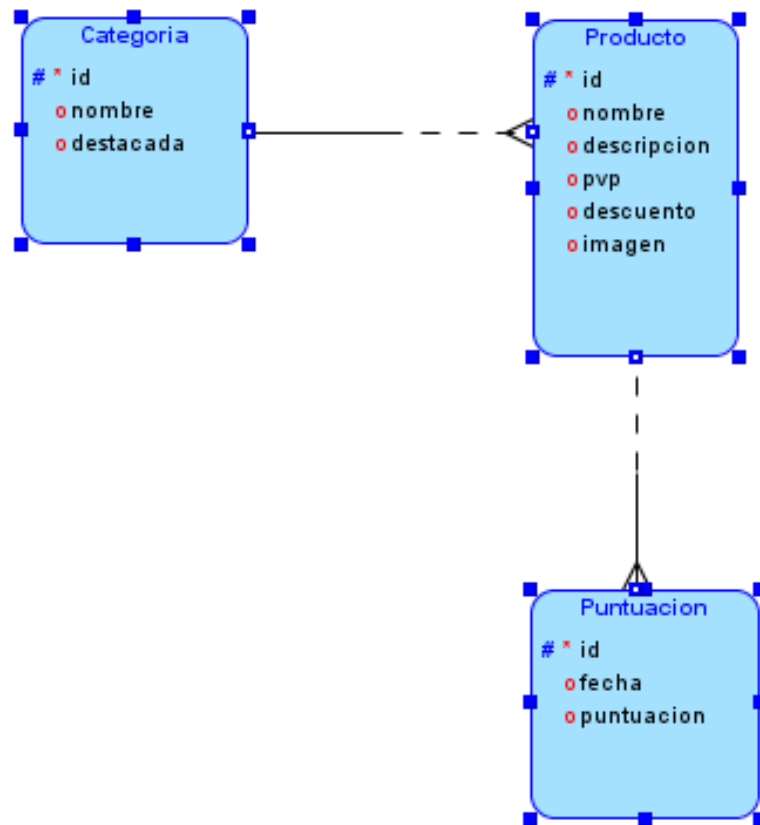
Las relaciones son:

1 producto está en una 1 única categoría

1 producto tiene "0...n" puntuaciones

3.4. Diseño Lógico

Esquema lógico de representación de las tablas implicadas realizado con Datamodeler de Oracle, aunque la representación a través de ORM de Hibernate utiliza el concepto de **entidad** en vez de **tabla** como lo representa el SGBD.



3.5. Diseño Conceptual

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3.6. Diseño Físico

DDL y DML estándar relacional para la base de datos H2 de persistencia embebida en el proyecto con objeto que indicamos anteriormente de estar incluido en un solo paquete compilado.

```
drop table categoria if exists;
drop table producto if exists;
drop table puntuacion if exists;
drop sequence if exists hibernate_sequence;

create sequence hibernate_sequence start with 100 increment by 1;

create table categoria (
    id bigint not null,
    destacada boolean not null,
    imagen varchar(512),
    nombre varchar(512),
    primary key (id)
);

create table producto (
    id bigint not null,
    descripcion clob,
    descuento float not null,
    imagen varchar(512),
    nombre varchar(512),
    pvp float not null,
    categoria_id bigint,
    primary key (id)
);

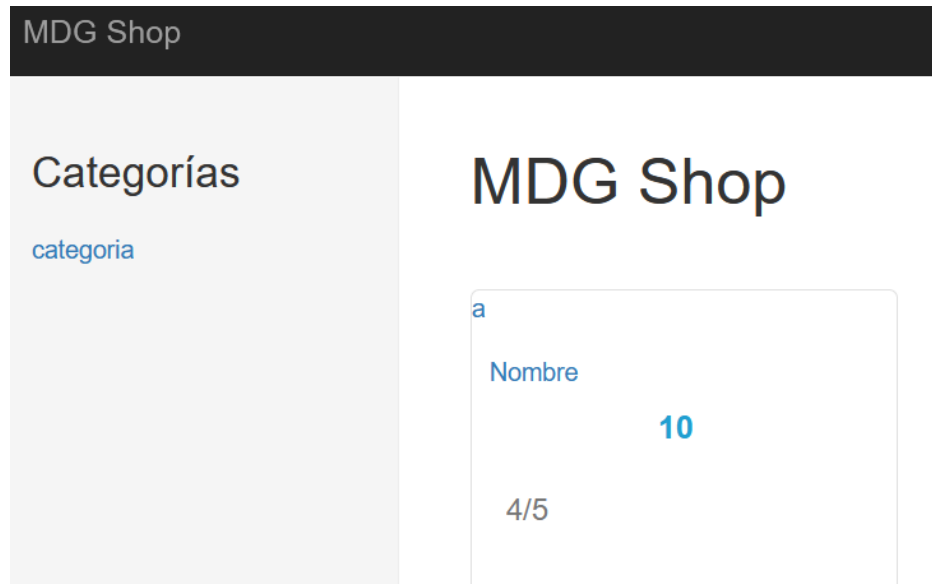
create table puntuacion (
    id bigint not null,
    fecha timestamp,
    puntuacion integer not null,
    producto_id bigint,
    primary key (id)
);
```

```
alter table producto add constraint fk_producto_categoria foreign key (categoria_id) references categoria;
alter table puntuacion add constraint fk_puntuacion_producto foreign key (producto_id) references producto;
```

3.7. Diseño de la Interfaz

Imágenes del aspecto visual que tendrá para el usuario las diferentes vistas:

Vista Principal con una plantilla básica de Bootstrap => index.html




Vista detalle de cada producto => detail.html



Formulario de categorías => form-categoria.html

MDG Shop



categoría

Nombre

URL de la imagen

☐ **Es destacada**

Enviar

Formulario de productos => form-producto.html

MDG Shop

Opciones ▾

producto

Nombre

Errores

URL de la imagen

Errores

Descripción

Descripción

PVP

Errores

Descuento

Categoría

Errores

Enviar

Vista del CRUD para la gestión de categorías => list-categoria.html

MDG Shop

No se puede borrar una categoría que tiene asociados productos

Nueva categoría

| Imagen | Nombre | Operaciones |
|--------|--------|-------------------------------------|
| imagen | | <div>Editar</div> <div>Borrar</div> |

Vista del CRUD para la gestión de productos => list-producto.html

MDG Shop

Nuevo producto

| Imagen | Nombre | Pvp | Operaciones |
|--------|--------|-----|-------------------------------------|
| imagen | | | <div>Editar</div> <div>Borrar</div> |

Capítulo 4: **IMPLEMENTACIÓN**

4.1. Introducción

Se ha creado este proyecto mediante la utilización de un entorno de desarrollo específico para el framework Spring dominado STS (Spring Tool Suite), que permite la integración de diferentes módulos ampliables.

4.2. Arquitectura Cliente/Servidor



Desde el punto de vista del *cliente* se utilizan los navegadores web más conocidos tanto en sistemas Windows, Linux y Apple, ya que se trata de un aplicación que no depende del sistema operativo sino del navegador instalado en el equipo del cliente que a su vez soporta de forma nativa las peticiones HTTP/HTTPS para la comunicación con el servidor.



Desde el punto de vista del **servidor**, para la publicación de la web se ha utilizado un proveedor de nombres de bajo coste que nos ha permitido generar el nombre miguel-dominguez-garcia.shop



También desde el punto de vista del **servidor**, para este proyecto se ha usado un modelo de **arquitectura clásico** con el objeto de que la publicación del sitio web completo en el VPS esté simplificado al máximo, ya que la base de datos está embebida dentro del archivo .jar así como el código de ejecución de control y visualización de datos.

Esto permite que instalando una instancia del archivo .jar en el servidor este opere funcionalmente si ningún otro servicio.

<https://www.digitalocean.com/>

4.3. Lenguajes de Programación



Java SE 1.8 JDK como lenguaje de base para la generación del código que está embebido dentro de la herramienta de desarrollo a la hora de generar los beans (servicios, repositorios, que controlan toda

4.4. Herramientas de Desarrollo

STS (Spring Tool Suite) framework que incluye Spring, MVC, etc.



como IDE específico de este para Spring Boot, Spring

Bootstrap las vistas de



como plantilla simple responsive para las tablas

4.5. Codificación

Todo el código desarrollado se encuentra disponible en el pendrive que se adjunta con esta memoria.

Capítulo 5: **PRUEBAS DE SOFTWARE**

5.1. Introducción

En el estado actual del desarrollo de la aplicación la pruebas a efectuar son de correcta ejecución de los CRUDs de las tablas del sistema y su correcta visualización a través de los formularios desarrollados.

5.2. Técnicas de Pruebas

5.2.1. Pruebas de CAJA BLANCA o Enfoque Estructural

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

5.2.2. Pruebas de CAJA NEGRA o Enfoque Funcional

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Capítulo 6: **CONCLUSIONES**

6.1. Conclusiones

Este proyecto sirve como entrenamiento y aprendizaje de una tecnología que tiene perspectivas de futuro y admite la ampliación continuada del producto.

La simpleza de esta presentación de proyecto está basada en la obtención de conocimiento para seguir ampliándolo.

6.2. Propuestas Futuras (Opcional)

- Seguridad de acceso a la web, mediante identificación de usuario y contraseña
- Seguridad de acceso con certificado digital
- Seguridad de acceso con DNI electrónico
- Acumulación de productos en un carrito de compra
- Cobro mediante TPV electrónico.
- Cobro mediante PayPal
- Publicidad de sponsor
- Etc.


Capítulo 7: **MANUAL DE INSTALACIÓN**

7.1. Creación de un VPS(Virtual Private Server)

Creamos una cuenta en <https://cloud.digitalocean.com/> con un saldo pequeño, en mi caso 5\$ para la creación del VPS que en este entorno de denomina *droplet*.

Droplets



Name

 **tienda-mdg-2021-12-16-s-1vcpu-1gb-fra1-01**
1 GB / 25 GB Disk / FRA1 - Ubuntu tienda-mdg-2021-12-16

Lo guardamos como *snapshot* para no consumir recursos si no se está usando, por lo que procedemos a la destrucción del VPS si no se está usando.

Snapshots

Droplets Volumes

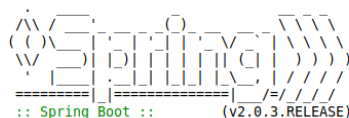
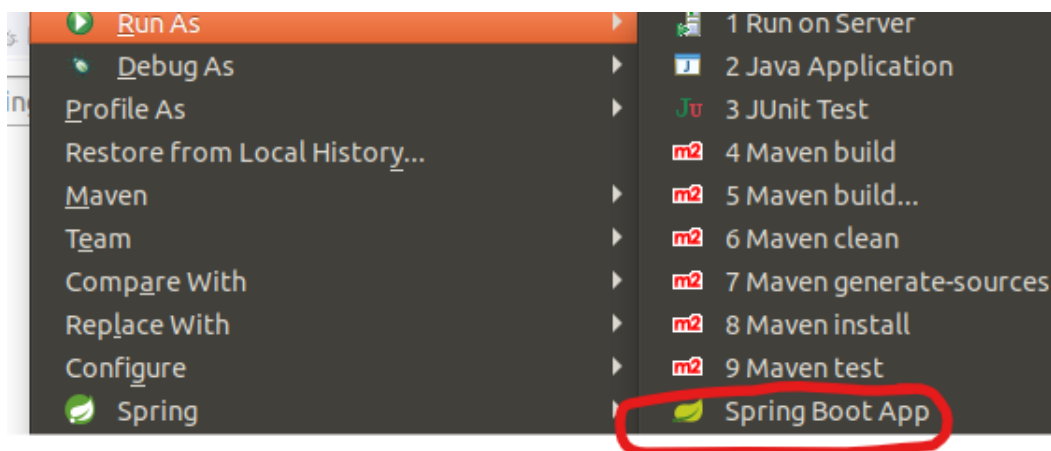
| Name | Size |
|---|---------|
|  tienda-mdg-2021-12-16 Created from tienda-mdg | 3.74 GB |
|  tienda-mdg-2021-12-10 Created from tienda-mdg | 3.63 GB |

7.2. Generación del archivo .jar

El entorno de desarrollo STS está instalado en un equipo virtualizado con VMware y es donde se ha realizado toda la generación del código que se ha subido a github:

<https://github.com/miguel-dominguez-garcia/tienda-mdg.git>

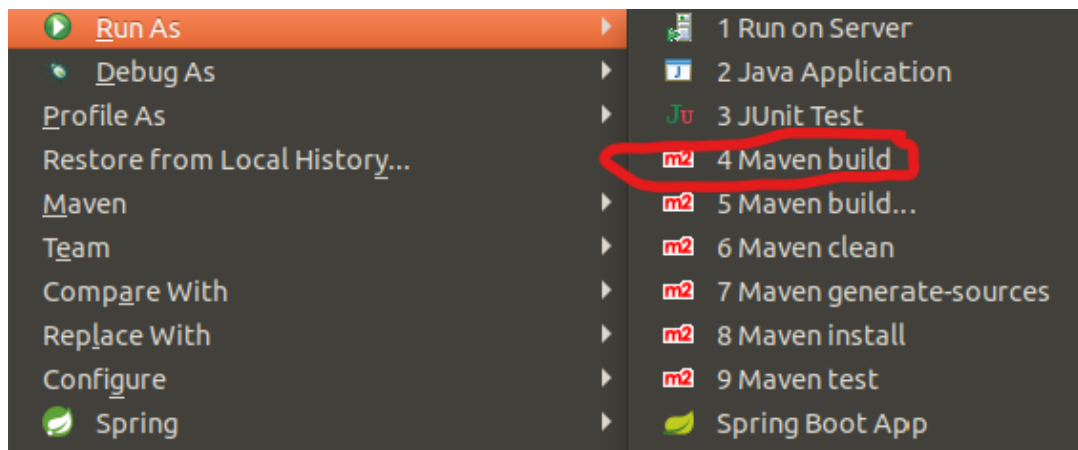
Durante el desarrollo se ha utilizado la opción de ejecución como “Spring Boot App”, con objeto de ir realizando pruebas.



```

2022-02-21 18:11:13.905 INFO 10099 --- [main] com.mdominguez.proyecto.Application : Starting Application on ubuntu with PID 10099 (/home/usuario/git
2022-02-21 18:11:13.907 INFO 10099 --- [main] com.mdominguez.proyecto.Application : No active profile set, falling back to default profiles: default
2022-02-21 18:11:13.929 INFO 10099 --- [main] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext
2022-02-21 18:11:14.484 INFO 10099 --- [main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTransaction
2022-02-21 18:11:14.641 INFO 10099 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 9000 (http)
2022-02-21 18:11:14.651 INFO 10099 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-02-21 18:11:14.651 INFO 10099 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.31
2022-02-21 18:11:14.654 INFO 10099 --- [ost-startStop-1] o.a.c.c.c.[Tomcat].[/] : The APR based Apache Tomcat Native library which allows optimal
2022-02-21 18:11:14.697 INFO 10099 --- [ost-startStop-1] o.s.web.context.ContextLoader : Initializing Spring embedded WebApplicationContext
2022-02-21 18:11:14.764 INFO 10099 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Root WebApplicationContext: initialization completed in 769 ms
2022-02-21 18:11:14.764 INFO 10099 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/]
2022-02-21 18:11:14.765 INFO 10099 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Servlet webServlet mapped to [/h2-console/*]
2022-02-21 18:11:14.767 INFO 10099 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to: [/]
2022-02-21 18:11:14.767 INFO 10099 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to: [/]
2022-02-21 18:11:14.767 INFO 10099 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter' to: [/]
2022-02-21 18:11:14.767 INFO 10099 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to: [/]
2022-02-21 18:11:14.767 INFO 10099 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'resourceUrlEncodingFilter' to: [/]
2022-02-21 18:11:14.846 INFO 10099 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-02-21 18:11:14.914 INFO 10099 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-02-21 18:11:14.935 INFO 10099 --- [main] j.LocalContainerEntityManagerFactoryBean : Building JPA container EntityManagerFactory for persistence unit
2022-02-21 18:11:14.941 INFO 10099 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [
    name: default
    ...]
2022-02-21 18:11:15.025 INFO 10099 --- [main] org.hibernate.Version : HHH000412: Hibernate Core {5.2.17.Final}
2022-02-21 18:11:15.025 INFO 10099 --- [main] org.hibernate.cfg.Environment : HHH000206: hibernate.properties not found
2022-02-21 18:11:15.040 INFO 10099 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
  
```

Una vez finalizado el proceso se ha generado un proyecto Maven que genera un .jar que es el que enviamos al VPS.



El objetivo es crear un archivo para programar el levantamiento automático si intervención de operadores humanos.

```
-----< com.mdominguez:sbtienda-mdg-2022-02-09 >-----
Building sbtienda-mdg 0.0.1-SNAPSHOT
-----[ jar ]-----

--- maven-resources-plugin:3.0.1:resources (default-resources) @ sbtienda-mdg-2022-02-09 ---
Using 'UTF-8' encoding to copy filtered resources.
Copying 1 resource
Copying 10 resources

--- maven-compiler-plugin:3.7.0:compile (default-compile) @ sbtienda-mdg-2022-02-09 ---
Nothing to compile - all classes are up to date

--- maven-resources-plugin:3.0.1:testResources (default-testResources) @ sbtienda-mdg-2022-02-09 ---
Using 'UTF-8' encoding to copy filtered resources.
Copying 0 resource

--- maven-compiler-plugin:3.7.0:testCompile (default-testCompile) @ sbtienda-mdg-2022-02-09 ---
Nothing to compile - all classes are up to date

--- maven-surefire-plugin:2.21.0:test (default-test) @ sbtienda-mdg-2022-02-09 ---

--- maven-jar-plugin:3.0.2:jar (default-jar) @ sbtienda-mdg-2022-02-09 ---
Building jar: /home/usuario/git/repository/sbtienda-mdg-2022-02-09/target/sbtienda-mdg-2022-02-09.jar

--- spring-boot-maven-plugin:2.0.3.RELEASE:repackage (default) @ sbtienda-mdg-2022-02-09 ---
BUILD SUCCESS
-----
Total time: 1.530 s
Finished at: 2022-02-21T18:08:51+01:00
-----
```

7.3. Configuración en Ubuntu 18.04

Copiamos el archivo compilado .jar desde el equipo en se encuentra el IDE Spring, que como anteriormente hemos indicado, habíamos compilado como proyecto Maven, en el directorio [/var/www/sboot](#).

Cambiamos el nombre de la aplicación a de [sbtienda-mdg-YYYY-mm-dd.jar](#) a [sbtienda-mdg.jar](#), con objeto de simplificar su uso en el entorno de Linux y a la hora de crear scripts.

7.4. Instalar la aplicación como un servicio systemd

En el directorio [/etc/systemd/system](#) se crea un fichero de configuración: [sbtienda-mdg.service](#) con la siguiente descripción

1. [Unit]
2. Description=sbtienda-mdg
3. After=syslog.target
- 4.
5. [Service]
6. User=bootapp
7. ExecStart = /usr/lib/jvm/jdk1.8.0_202/bin/java -jar
/var/www/sboot/sbtienda-mdg.jar
8. SucessExitStatus
- 9.
10. [Install]
11. WantedBy=multi-user.target

Capítulo 8: **MANUAL DE USUARIO**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Capítulo 9: **GLOSARIO DE TÉRMINOS Y ACRÓNIMOS**

- Cloud → conocida también como servicios en la nube, informática en la nube, nube de cómputo o simplemente «la nube». Es el uso de una red de servidores remotos conectados a internet para almacenar, administrar y procesar datos, servidores, bases de datos, redes y software.
- STS → Spring Tool Suite
- VPS → Virtual Private Server
- IDE → Integrated Development Environment
- SGBD → Sistema de Gestión de Base de Datos, conjunto de sistemas que permite la gestión de una base de datos relacional.
- RDBMS → Relational Database Management System, es el nombre en inglés del SGBD.
- SQL → Structured Query Language, lenguaje de consulta en SGBDs.
- ORM → Object Relational Mapping, sistema que permite mapear un objeto como una tabla en SQL.
- CRUD → Create, Read, Update and Delete, hace referencia a todos los procesos de gestión de una entidad/tabla para su correcto mantenimiento.
- Droplet → en lenguaje cloud de la empresa Digital Ocean, hace referencia a un VPS operativo y funcionando que genera facturación mensual por costes generados.
- Snapshot → En este proyecto es la imagen de un VPS inactivo y hace de copia de seguridad en el cloud. Se suele tener desactivado el VPS para evitar el consumo económico y se activa cuando se necesita tener operativo y atendiendo a usuarios.
- Ubuntu Linux → Sistema operativo basado en Debian, cuya versión 18.04 stable es usada en este proyecto.

...

Capítulo 10: **REFERENCIAS BIBLIOGRÁFICAS (Webgrafía)**

<https://openwebinars.net/academia/aprende/spring-boot/>

<https://openwebinars.net/academia/aprende/spring-framework/>

<https://openwebinars.net/academia/aprende/introduccion-thymeleaf/>

<https://openwebinars.net/academia/aprende/crud-java-ee/>

<https://openwebinars.net/academia/aprende/servidores-web/>

<https://blogging-techies.com/es-posible-obtener-alojamiento-de-namecheap-gratis/>

<https://www.digitalocean.com/>