

Licenciatura en Informática

# INGENIERÍA DE SOFTWARE

Nancy Obed Martínez Miguel

Luis Fernando Vásquez Fabián

Fernando López Salvador

M.T.C.A. ROLANDO PEDRO GABRIEL

# Índice de contenido

0.1	Introducción . . . . .	4
0.2	Próposito del proyecto . . . . .	4
0.3	Viabilidad . . . . .	4
0.4	Factibilidad . . . . .	4
0.5	Ámbito del proyecto . . . . .	5
0.5.1	Validación y Registro de usuario . . . . .	5
0.5.2	Punto de venta . . . . .	5
0.5.3	Ticket . . . . .	5
0.5.4	Agregar registro de los productos . . . . .	5
0.5.5	Modificación del producto . . . . .	6
0.5.6	Eliminar registros . . . . .	6
0.6	Mapa de Navegación . . . . .	7
0.7	Interfaces . . . . .	7
0.8	Requerimientos Funcionales . . . . .	11
0.9	Requerimientos no Funcionales . . . . .	13
0.10	Estándar de Codificación . . . . .	13
0.11	Roles para el Sistema de Ventas . . . . .	20

# Índice de Tablas

1	Requerimientos Funcionales 1 . . . . .	11
2	Requerimientos Funcionales 2 . . . . .	11
3	Requerimientos Funcionales 3 . . . . .	11
4	Requerimientos Funcionales 4 . . . . .	12
5	Requerimientos Funcionales 5 . . . . .	12
6	Requerimientos no Funcionales 1 . . . . .	13
7	Requerimientos no Funcionales 2 . . . . .	13
8	Requerimientos no Funcionales 3 . . . . .	13
9	Requerimientos no Funcionales 3 . . . . .	13
10	Roles del Sistema . . . . .	20

## 0.1 Introducción

En la actualidad las herramientas tecnología se han convertido en uno de los factores más importantes en cuanto a los usos institucionales de las empresas, estas son utilizados diariamente para el manejo de información y su proyección social. Así que dentro del amplio ámbito de la informática se busca que día día se puedan lograr los avances necesarios para poder facilitar la vida de las personas, ya que conforme pasa tiempo surgen las necesidades de cambiar algunas cosas. Por lo que en este proyecto se enfoca en implementar un software de ámbito comercial, este software será diseñado para el control de un sistema de ventas de una tienda de abarrotes , buscando una optimización de los datos que esta maneja.

Por lo que estas aplicaciones mejoran el control administrativo llevando un seguimiento preciso de todas las transacciones que se realizan dentro de la tienda en un tiempo real, para generar reportes detallados de las cuentas que van permitir a los administradores la cantidad correcta de productos, en el momento adecuado lo cual va permitir al negocio o empresa mejorar servicio al cliente reduciendo cualquier anomalía. Los sistemas de ventas tienen la ventaja de ser personalizados para cumplir con las necesidades específicas de un negocio, por ejemplo, en la tienda las organizaciones de venta al menudeo pueden ayudar a localizando precios de venta de los productos y costos actuales de los productos para poder trabajar más rápidamente. En años recientes el mercado ha incrementado y se han ido generando diversos negocios o empresas en todas partes y eso ha llevado a las mismas ofrecer una mejor atención por lo que es muy importante contar con un sistema de ventas rápido y sencillo.

## 0.2 Propósito del proyecto

El sistema de ventas está orientada a la gestión de empresas comerciales o pequeños negocios, es un software practico que va permitir un mejor control del negocio. Se trata de un software para la gestión de clientes y productos. Donde puedes ingresar como administrador y realizar los que se necesite así sea ingresar, eliminar o modificar un producto o de ser un vendedor buscar los productos que el cliente desee y checar precios, cantidades en existencia y realizar su venta. Así mismo generara un informe de la venta y su respectivo ticket. El sistema se hace cargo de registrar las ventas del producto o servicio de la organización, evalúa sus características o necesidades del negocio, llevando un detalle del registro de todas las transacciones.

## 0.3 Viabilidad

Este proyecto es viable debido a que si contamos con las posibilidades para poder trasladarnos a dichos establecimientos al que se realizara el proyecto, todo esto con la finalidad de platicar con el cliente y así nos pueda proporcionar cada una de las necesidades o requerimientos que este quiera que se implemente al software.

## 0.4 Factibilidad

El proyecto es factible debido a que en la universidad de la sierra sur contamos con un centro de tecnologías de la informática y comunicación donde contamos con las herramientas necesarias ya que se cuenta con computadoras internet, y la asesoría de los docentes quienes

están altamente calificados para ayudarnos a desarrollar dicho software. A si mímoo también durante nuestra carrera hasta el día de hoy hemos aprendido conocimientos en distintos lenguajes de programación como C, Java, Css, Html y lenguaje ensamblador los cuales podemos aplicar en el proyecto a resolver.

## 0.5 **Ámbito del proyecto**

En la actualidad debido a la falta de conocimiento de muchas personas que administran establecimientos o pequeñas empresas dedicadas a las ventas no tienen en orden la administración de las mismas, como podrían ser las ventas, registros o actualizaciones de productos. Por lo que muchas veces suelen tener pérdidas de ganancias o desperdicio de la misma mercancía.

Pero también se sabe que en un establecimiento deben de quedar de manera ordenada los roles que se desempeñan dentro del lugar, es decir, ver que cosas puede hacer el administrador del negocio y que puede realizar el vendedor. Así que se desea implementar un sistema de control de ventas en una tienda de abarrotes y así dar solución a la problemática ya antes mencionada. Evitando así pérdidas monetarias y descontrol de su inventario, este sistema de venta estará diseñado de manera que el usuario pueda interactuar fácilmente y obtenga buenos resultados al utilizarlo.

Por lo que se describirá la estructura ya pensada de como se va ir desarrollando el sistema, el cual se va a distribuir de la siguiente manera:

### 0.5.1 **Validación y Registro de usuario**

En esta parte se implementará implantará la validación de usuario con la finalidad de mantener seguro los datos del usuario, evitando así que otras personas no autorizadas puedan acceder al sistema haciendo un mal uso del sistema de ventas. También ahora va depender del rol que entren al sistema, así se le irán mostrando sus partes por ejemplo si entra del lado del vendedor:

### 0.5.2 **Punto de venta**

Donde el vendedor va generar una venta es decir el cliente comienza a tomar los productos que serán mostrados desde una tabla y así puede ir viendo si tiene existencia de dicho producto, el precio y entre otras cosas.

### 0.5.3 **Ticket**

En esta parte se piensa que después de que el vendedor genere su venta le muestre como resultado un mensaje donde le detalle el nombre del producto, el precio y de igual manera le muestre la suma total de su compra. En caso contrario y se iniciará sesión como administradores pues lo primero que le mostraría es una ventana con tres botones que le darán acceso a agregar, modificar o eliminar algún producto.

### 0.5.4 **Agregar registro de los productos**

El usuario podrá realizar sus registros diarios de las ventas y adquisición de dispositivos nuevos, además podrá registrar las características de cada producto como nombre, id, cuantos hay existentes, precio entre otras.

### **0.5.5 Modificación del producto**

En cualquier momento el usuario podrá acceder a sus registros para poder hacer las modificaciones necesarias siempre y cuando se requiera. En este caso se le mostrará un pequeño formulario muy similar al de agregar y así podrá modificar lo que necesite y así podrá dar en su botón guardar cambios.

### **0.5.6 Eliminar registros**

El usuario podrá en cualquier momento eliminar los registros pasados que ya no necesite, esto con la finalidad de que no requiera de más espacio de memoria evitando de esta manera la necesidad de hacer más gastos innecesarios. Este apartado estará diseñado de manera simple para que el usuario pueda navegar sin problemas, además de que contará con aviso de alerta cuando el usuario elimine un registro, de esta forma el usuario estará seguro si desea eliminar tal registro, evitando así el riesgo de eliminar registros importantes.

Este sistema tendrá el objetivo de ser una gran herramienta que pueda ayudar al usuario llevar una mejor organización de sus operaciones en su negocio y también está claro que esta también debe cumplir las expectativas de cliente en cuanto al diseño del software ya que el diseño se realizará de forma que pueda ser más agradable y entendible para el usuario, este sistema contará con diferentes herramientas, como el registro de ganancias en donde podrá ver si su negocio está resultando así como también contará con página en donde puedan consultar los modelos productos que ha vendido y así el usuario podrá ver que compras son más recomendables les hará a sus proveedores.

## 0.6 Mapa de Navegación

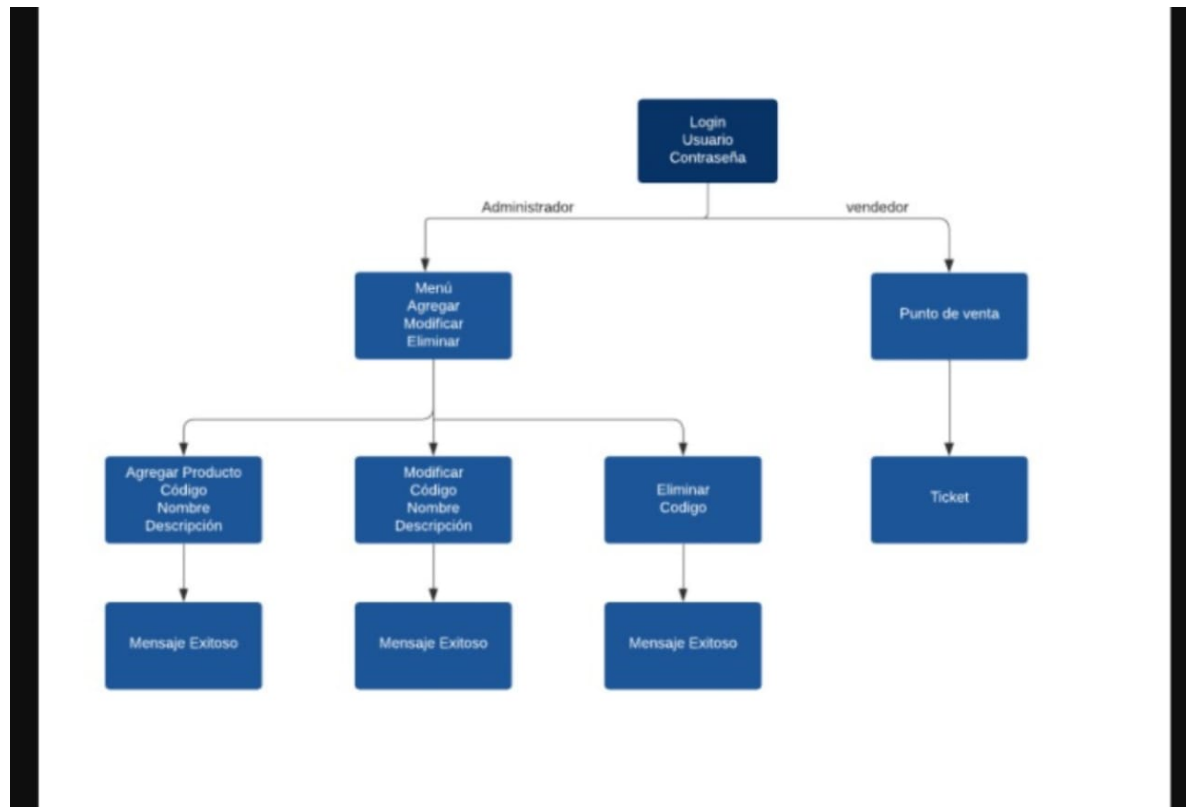


Figure 1: Mapa de Navegación

## 0.7 Interfaces

Una interfaz se utiliza en informática para nombrar a la conexión funcional entre dos sistemas, programas, dispositivos o componentes de cualquier tipo, que proporciona una comunicación de distintos niveles, permitiendo el intercambio de información. Por lo que este proyecto tiene las siguientes interfaces.

## Validación de usuario.

En la siguiente ventana lo que nos muestra es el Login o el inicio de sesión en el cual es necesario ingresar los datos correctos para si poder ingresar al sistema, ya se como Vendedor o como administrador para realizar las acciones necesarias.



Group Name



Correo

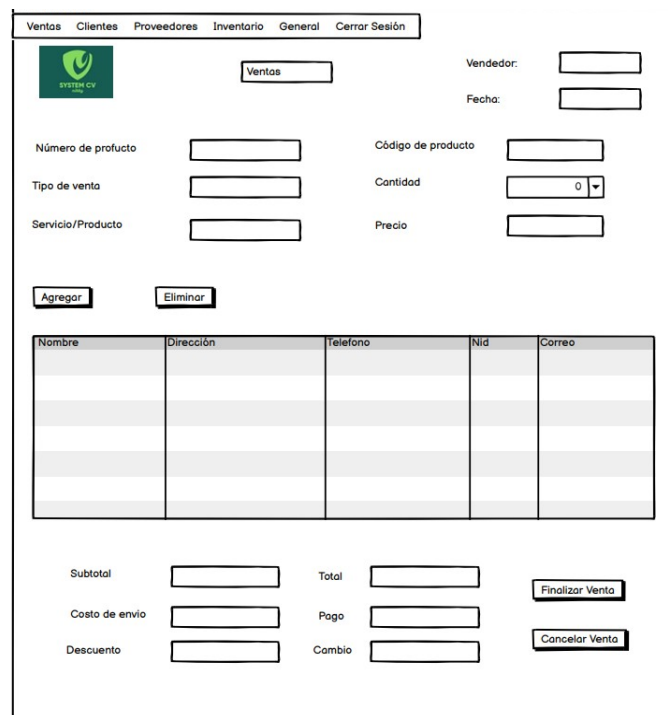
Contraseña

¿Olvidaste tu contraseña?


Figura 1: Login

## Punto de venta.

En caso de ingresar como Vendedor se mostrara la ventana siguiente en la cual podrá realizar las ventas correspondientes al pedido solicitado, en dicha ventana se solicita la siguiente información para poder proceder con la compra: número de producto, código del producto, cantidad, el precio y otros datos ya están dentro de la base de datos por lo cual se agregaran automáticamente. Una vez finalizada la compra en la tabla inferior se mostraran todos los artículos solicitados junto con la cantidad a pagar así como el descuento si es que algún articulo se encuentra en oferta como también el cambio a reponer.



Ventas Clientes Proveedores Inventario General Cerrar Sesión

 Ventas

Vendedor:

Fecha:

Número de producto  Código de producto

Tipo de venta  Cantidad

Servicio/Producto  Precio

Nombre	Dirección	telefono	Nid	Correo

Subtotal  Total

Costo de envío  Pago

Descuento  Cambio

Figura 2: Ventana de Ventas



### Ventana de Clientes y de Inventario.

En caso de que el usuario ingrese como administrador la ventana que se muestra es la de Inventario en la cual el podrá ingresar nuevos artículos o quitar artículos así también tendrá acceso a la ventana de clientes donde podrá modificar la información de los clientes pudiendo modificar los datos de los clientes registrados en la tienda.

Nombre	Dirección	Telefono	Nid	Correo	Contraseña

Figura 4: Ventada de Clientes

Código	Descripción	Precio	Stock

Figura 3: Ventana Inventario

### Ventana de Proveedores.

También contamos con la ventana de Proveedores en la cual almacenamos a todos los proveedores a los cuales compramos mercancía en la cual guardamos el nombre, dirección, teléfono y en la parte inferior contamos con una tabla la cual nos muestra el total de proveedores con los cuales contamos actualmente teniendo así un mejor control de nuestros proveedores.


Nombre	Dirección	Telefono	Nid	Correo

Figura 5: Ventana de Proveedores

**Ventana de Cliente.**

En las opciones de General se muestran tres opciones con las cuales se muestra el total de la información guardada en cada una de las ventanas, la ventana de cliente muestra el total de clientes permitiendo buscar a un cliente en particular sin necesidad de ingresar como administrador solo para poder ver la información si acceso a modificarla.

Group Name



NID Cliente

Fecha


NID	Descripción	Cantidad	Precio Unitario	Precio Venta

Regresar

Figura 6: Ventana Clientes total

**Ventana de Productos.**

En la opción de productos igualmente se muestra el total de los productos que se encuentran en el inventario para facilitar la búsqueda de algún producto en específico.



Buscar Producto:

Regresar

Figura 7: Ventana Productos total

## 0.8 Requerimientos Funcionales

Código de requerimiento	RF01
Nombre	Envío y recepción de archivos
Propósito	Permitir el acceso de información a la base de datos así como la facilidad para enviar información
Descripción	El usuario realiza la encuesta para la información, a continuación acepta los términos para así poder enviar su información a la base de datos
Entrada	Formulario de perfil del cliente
Salida	Mensaje de acción satisfactoria
Prioridad	Alta

Table 1: Requerimientos Funcionales 1

Código de requerimiento	RF02
Nombre	Identificación
Propósito	Para saber si al iniciar sesión es cliente o dueño.
Descripción	Primero da clic en el botón iniciar sesión después podrá poner su contraseña y mirar como administrador o usuario
Entrada	Contraseña
Salida	Regresar a la pantalla principal
Prioridad	Alta

Table 2: Requerimientos Funcionales 2

Código de requerimiento	RF03
Nombre	Operaciones Crud
propósito	Crear, Actualizar, borrar usuarios
descripción	Una vez ubicados en la página inicial del software el usuario debe capturar los datos de cliente
Entrada	Formulario de entrada de llenado de información del cliente
Salida	Mensaje que indique que el registro fue exitoso
Prioridad	Alta

Table 3: Requerimientos Funcionales 3

Código de requerimiento	RF04
Nombre	Facturación de pedidos de venta
propósito	Crear la factura de la venta
descripción	La facturación de pedidos de venta se realizara por medio de una pantalla de pedidos pendientes de facturación.
Entrada	Formulario de entrada de llenado de información de la compra
Salida	Mensaje que indique que el proceso fue exitoso
Prioridad	Media

Table 4: Requerimientos Funcionales 4

Código de requerimiento	RF05
Nombre	Maestro de clientes
propósito	Mantener un mejor control sobre la información de los clientes
descripción	El sistema manejará un registro maestro de clientes. Sólo los usuarios autorizados podrán ingresar nuevos clientes, modificar los datos o eliminarlos.
Entrada	Formulario de entrada de llenado de información del cliente
Salida	Mensaje que indique que el proceso fue exitoso
Prioridad	Alta

Table 5: Requerimientos Funcionales 5

## 0.9 Requerimientos no Funcionales

Código de requerimiento	RNF01
Nombre	Actualización del software
propósito	Realizar mantenimiento y actualizaciones del software
descripción	Ir a la empresa a darle mantenimiento al software y actualizar bibliotecas

Table 6: Requerimientos no Funcionales 1

Código de requerimiento	NRF02
Nombre	Mantenimiento
propósito	Un software con un buen funcionamiento
descripción	Las acciones que se harán para mejorar el software (cambio de contraseña, alguna mejora, limpieza de registro de sistema.

Table 7: Requerimientos no Funcionales 2

Código de requerimiento	NRF03
Nombre	Aplicación para celular
propósito	El software no contara con aplicación para celulares
Prioridad	baja

Table 8: Requerimientos no Funcionales 3

Código de requerimiento	NRF04
Nombre	Ordenes de entrega
propósito	El sistema no contara con opcion para envio a domicilio
Prioridad	baja

Table 9: Requerimientos no Funcionales 3

## 0.10 Estándar de Codificación

# Estándar de codificación Java

## Nomenclatura

El idioma por defecto a la hora de dar sentido funcional al nombre de clases, variables, constantes, etc. será una mezcla entre la nomenclatura tradicional en inglés y la nomenclatura funcional adoptada.

Resumiendo, aquella codificación que por estandarización y/o aceptación se pueda escribir en inglés se mantendrá así por convenio, casos como insert, update, delete, create, retrieve, list, set, get, newInstance, Delegate.

Para la parte funcional se utilizara castellano, por lo tanto la nomenclatura de los métodos será: getListEmpresa en sustitución de getListCompany o insertBanco en lugar de insertarBanco.

## Paquetes

Por defecto todos los paquetes se escribirán en minúsculas y sin utilizar caracteres especiales. El paquete base queda definido como es.gobcantabria, en este paquete no se definirá ninguna clase.

Se tendrá, así mismo, otro nivel extra dentro del paquete definido como el nombre del proyecto o del módulo (Ej. es.gobcantabria.xxxx ).



Si existiera una parte común a varios de estos módulos, el nombre de los paquetes comenzarán por: es.gobcantabria.common

La estructura en árbol de los paquetes siguientes a partir de este último se define como sigue: (presuponiendo que se trata de una aplicación web multicapa)

- bussines
  - dao
  - domain
  - service
  - helper
  - exception
- util
- web
  - controller
  - model
  - view

## Nombres de Interfaces

Los nombres de interfaces utilizarán el sufijo **Interface** y estarán compuestos por palabras con la primera letra en mayúscula (CamelCase). Se debe evitar el uso de abreviaciones que dificulten la comprensión del código.

Ejemplo: ConexionInterface, ComponenteTablaInterface

## Nombres de clases

Los nombres de clases deben ser mezclas de mayúsculas y minúsculas, con la primera letra de cada palabra interna en mayúsculas (CamelCase).

Debemos intentar mantener los nombres de clases simples y descriptivos.

Debemos usar palabras completas y evitar acrónimos y abreviaturas (se permiten DAO, DTO, URL, HTML, etc.).

Si la clase cumpliera algún patrón determinado o tuviera una funcionalidad específica es recomendable definirlo en el nombre.

Paquete	Funcionalidad	Nombre
bussines.dao	Data Access Object (Interface)	<nombre> DAO
bussines.dao.impl	Data Access Object (Implementation)	<nombre> DAOImpl
bussines.exception	Excepciones	<nombre> Exception
bussines.service	Service	<nombre> Service
bussines.helper	Helper	<nombre> Helper
bussines.dto	Data Transfer Objects	<nombre> DTO
util	Clases de constantes.	<scope> Keys <nombre> Keys
web.controller	Controller	<nombre> Controller
web.filter	Filter	<nombre> Filter
web.model	Model	<nombre> Model
web.listener	Listener	<nombre> Listener

## Nombres específicos de gestiones

Cuando se trata de gestionar una entidad determinada (Ej. Usuario) se definen los nombres de clases, demás ficheros implicados con la siguientes reglas:

Clase: <<FuncionalidadGenerica>><<Entidad>><<Especificación de Clase>>

Ejemplo: Usuario

UsuarioAction, FindUsuarioAction

## Métodos

Los métodos deberán ser verbos (en infinitivo), en mayúsculas y minúsculas con la primera letra del nombre en minúsculas, y con la primera letra de cada palabra interna en mayúsculas (lowerCamelCase).

No se permiten caracteres especiales.

El nombre ha de ser lo suficientemente descriptivo, no importando a priori la longitud del mismo.

## Variables

Los nombres de las variables tanto de instancia como estáticas reciben el mismo tratamiento que para los métodos, con la salvedad de que aquí sí importa más la relación entre la regla mnemónica y la longitud del nombre.

Ejemplo:

- Correctos: diaCalculo, fechaIncorporacion
- Incorrectos: dC, Dcal, fl, Fl¿

Se evitará en la medida de lo posible la utilización de caracteres especiales, así como nombre sin ningún tipo de significado funcional.

Las excepciones son las variables utilizadas en bucles for, para esos casos se permite utilizar i, j, k, l y siempre en ese orden de anidamiento.

El primer bucle siempre será el que tenga la variable i como iterador. (Esta variable se definirá para el bucle en cuestión).

## Constantes

Los nombres de constantes de clases deberían escribirse todo en mayúsculas con las palabras separadas por subrayados ("\_"). Todas serán declaradas como public static final

```
public static final String PROPERTY_URL_SERVICIO = "urlServicio";
```

## Estilo de codificación

### Comentarios

Los comentarios serán utilizados para dar información adicional al desarrollador sobre la implementación del diseño de la clase. Se tiene, por tanto, que evitar referencias al diseño funcional de la misma.

El uso abusivo de los comentarios es desaconsejable, principalmente por el trabajo extra necesario para su correcto mantenimiento. Es preferible rediseñar el código para una mejor compresión del mismo.

Estándar de Codificación JEE 13ESTÁNDAR DE CODIFICACIÓN JEE



Se tienen que evitar el uso de caracteres especiales dentro de los comentarios así como el uso de cajas u otro tipo de gráfico creado mediante códigos ASCII.

La estructura de los diferentes tipos de comentarios y su uso general se presenta en la clase base de codificación.

## Comentarios de Documentación (JavaDoc)

Como norma es obligatorio proporcionar un comentario de documentación por cada clase / interface, método, propiedad o constante creado.

Son obligatorios los siguientes comentarios de documentación:

Comentario de la clase / interface:

- Prescripción genérica de la clase y su responsabilidad.
- Autor.

Todas las variables tipo `private` o `protected` han de ser obligatoriamente comentadas.

Reglas generales a la hora de escribir comentarios de documentación.

- Siempre se escribe en tercera persona.
- Los caracteres especiales tales como tildes y eñes se han de codificar con su código HTML correspondiente.
- Las descripciones siempre deberían empezar por un verbo.

En cuanto a los tag's utilizados el orden de los mismos es el siguiente:

- **@param** seguido de el nombre del parámetro e indentada la descripción del mismo. Usualmente esta descripción será una frase corta que comienza definiendo el tipo del parámetro.

@param alturaCaja      Entero que define la altura de la caja en píxeles

@param longitudCaja    Flotante que define la longitud de la caja en píxeles

- **@return** este tag no aparece para aquellos métodos que retornan void. Por lo demás se comporta como el tag anterior
- **@throws** Descripción breve de la posible causa de la excepción.
- **@see** Su uso queda restringido en cuanto a cantidad de los mismos como con el atributo **@link**.

Los demás tags permitidos (**@since**, **@serial**, etc..) el uso es menos común y por lo tanto no se define una manera de utilizarlos.

Usar el atributo `<code>` para las palabras reservadas de java, nombres de clases, métodos, interfaces, propiedades, argumentos y ejemplos de código.

El uso del atributo **@link** tendrá que ser mínimo, para evitar llenar el documento de enlaces. Como norma general solo se incluirán links cuando la referencia sea necesaria y sólo en la primera aparición de la misma.

Para distinguir entre dos métodos con el mismo nombre pero diferentes parámetros de entrada se utilizarán el nombre del método seguido por los argumentos del mismo entre paréntesis.

## Declaraciones

Para la declaración de las variables se utiliza una declaración de cada vez y no se permiten dejar variables locales sin inicializar salvo en el caso de que sean propiedades de un objeto bean.

La codificación correcta sería:

```
public static Integer entero = new Integer(0);
```

La declaración de las variables locales a una clase, método o bloque de código se realizan al principio del mismo y no justo antes de necesitarse la utilización de la variable.

La única excepción a esta regla son las variables que gestionan los bucles for.

Las variables de avance de bucles for no podrán ser modificadas de ninguna manera fuera de la propia sentencia del bucle.

La duplicidad de los nombres de variables en diferentes niveles dentro de la misma clase se tiene que evitar.

## Sentencias

Normas básicas son :

- Una sentencia por línea de código.
- Todo bloque de sentencias entre llaves, aunque sea una sola sentencia después de un if.

La declaración de los bucles for serán usualmente de la forma :

```
for (int i = 0; i < condicion ; i++)
```

Son obligatorias las tres condiciones del bucle for: inicialización, condición de finalización y actualización del valor de la variable de avance.

La variable de avance del bucle nunca podrá ser modificada dentro del propio bucle.

## Buenas prácticas

### Constantes

Como norma general todas las constantes numéricas no deberían codificarse directamente, salvo la excepción de -1, 0 y 1.

### Propiedades

El acceso/modificación de las propiedades de una clase (no constantes) siempre mediante métodos de acceso get/set.

La asignación de variables / propiedades no podrá ser consecutiva.

Variable1 = variable2 = "hola mundo"

No válido

No utilizar el operador asignación en sitios donde se pueda confundir con el operador igualdad. Ni dentro de expresiones complejas.

## Métodos

Como norma general no se debe acceder a un método estático desde una instancia de una clase, debemos utilizar la clase en sí misma.

## 0.11 Roles para el Sistema de Ventas

Nombre		Roles
Nancy Obed Martínez Miguel		Master scrum
Luis Fernando		Desarrollador
Fernando López Salvador		Auxiliar
Entrada		Formulario de perfil del cliente
M.T.C.A.Rolando	Pedro	Usuario Final
Grabriel		
M.T.E.Everardo	de	Jesús
Pacheco Antonio		Programador
M.C.CLirio Ruiz Guerra		Administrador de la Base de Datos

Table 10: Roles del Sistema