

# Aplicación Punto de Venta

## Plan de Verificación y Validación

### V- 1.0

#### Historia de revisiones

Fecha	Versión	Descripción	Autor
4/6/2020	1.0	Estas verificaciones y validaciones fueron realizadas durante el curso del primer sprint de inspección de código fuente. Contiene detalle de las pruebas de funcionalidad, interface de usuario e integridad de datos.	Paulina Gómez, Jesús Estévez, Wilmer Guerrero Nizar Sánchez.

# Contenido

1	<b>Introducción.</b>	3
1.1	Propósito.....	3
1.2	Punto de partida.....	3
1.3	Alcance.....	3
1.4	Estrategia de evolución del plan. ....	4
2	<b>Requerimientos para verificar.</b>	4
3	<b>Estrategia de verificación.</b>	5
3.1	Tipos de pruebas.....	5
3.1.1	Prueba de Interface de Usuario.....	5
3.1.2	Prueba de integridad de datos y base de datos.....	7
3.1.3	Prueba de funcionalidad.....	8
4	<b>Recursos</b>	10
4.1	Roles: .....	10
4.2	Sistema:.....	11
5	<b>Hitos del proyecto de verificación</b>	11
6	<b>Entregables</b>	11
6.1	Modelos de casos de prueba.....	11
6.2	Informes de verificación.....	12
6.3	Evaluación de la verificación.....	12
6.4	Informe final de verificación. ....	13
7	<b>Apéndice.</b>	13
7.1	Niveles de gravedad de error.....	13
7.2	Nivel de aceptación de los elementos verificados.....	14

# 1 Introducción.

## 1.1 Propósito.

El plan de verificación para el proyecto Aplicación Punto de Venta soporta los siguientes objetivos:

- Inspeccionar el código fuente del sistema para detectar malas prácticas que podrían ocasionar un comportamiento no especulado dentro de la lógica de negocios.
- Verificar que la aplicación cumpla con los requisitos de estándares de seguridad.
- Verificar la lógica de negocios, la funcionalidad y seguridad de las operaciones y la integridad de los registros de la base de datos.
- Realizar pruebas que la seguridad y la funcionalidad del sistema.
- Mencionar los documentos corregidos, documentar los errores y detallar las técnicas de corrección.

## 1.2 Punto de partida.

La Aplicación Punto de Venta es un programa que permite la gestión de los productos de cualquier tienda. Esta está desarrollada en el lenguaje Java, y el gestor de base de datos utilizado es PostgreSQL. Consta de un registro de usuarios, y un posterior inicio de sesión para poder acceder al sistema. Esta aplicación contiene un módulo de consultas a los productos, al igual que el módulo de mantenimiento de productos, donde se puede insertar, eliminar y modificar información de la base de datos. Está supuesta a tener un módulo de *Preferencias* para el usuario, una funcionalidad de impresión de reportes, y una sección de *Ayuda* en caso de que el usuario sea inexperto.

## 1.3 Alcance.

Durante la inspección de código, se llevarán a cabo las pruebas de integridad de datos, pruebas de interfaz de usuario, y las pruebas de funcionalidades (que abarca el mantenimiento y consulta de los productos, y la definición de las opciones de preferencias).

Características a verificar:

- Integridad de los datos.
- Seguridad del sistema.
- Performance del sistema.
- Sintaxis e implementación del código.

- Funcionamiento de las vistas.

Restricciones:

- No se pueden añadir nuevos features al sistema.

## **1.4 Estrategia de evolución del plan.**

La responsable del monitoreo del plan de verificación y validación será Paulina Gómez (SCRUM Master). Se realizarán distintas pruebas, divididas en tres sprints de una duración de máximo una semana. Los responsables de realizar las pruebas serán:

El equipo de desarrollo: Nizar Sánchez y Wilmer Guerrero.

El equipo de calidad: Dawin y Jesús Estévez.

Al final de cada sprint, será verificado en equipo que todos los objetivos preestablecidos han sido cumplidos. Estos serán evaluados revisando la lista de objetivos y tomando a consideración la calidad que tuvieron las soluciones. Cualquier cambio o imprevisto será comunicado por medio del grupo, este cambio será evaluado y aprobado por los demás participantes.

## **2 Requerimientos para verificar.**

Será verificado de este proyecto:

- El funcionamiento correcto de registro de usuarios.
- La validación de la información del usuario.
- El funcionamiento del inicio de sesión.
- El funcionamiento de la salida de sesión.
- La calidad de la interfaz de usuario.
- La integridad de los datos.
- La recuperación de errores que tiene el sistema.
- El funcionamiento de las inserciones, modificaciones y eliminaciones a la base de datos.

## 3 Estrategia de verificación.

### 3.1 Tipos de pruebas.

#### 3.1.1 Prueba de Interface de Usuario

##### 3.1.1.1 *Objetivo de la prueba*

Con el fin de comprobar la navegación y distintos elementos relacionados al manejo de CRUD del sistema, se realizaron pruebas de UI siguiendo dos categorías: pruebas manuales y pruebas automatizadas mediante software de terceros. El objetivo de estas pruebas no fue más que el de identificar posibles errores en la interfaz del usuario a tiempo, buscando corregirlos, mejorar la UX y la usabilidad del sistema.

##### 3.1.1.2 *Técnica*

Entre las técnicas aplicadas tanto en los test manuales como automatizados para detectar errores del UI se encuentran: introducir valores que no deberían ser admitidos en ciertos campos, concordancia entre los mensajes de retroalimentación y las funciones realizadas (interactuar con la base de datos o re-factorizar la tabla al filtrar un dato, por ejemplo) o la comparación de ciertos elementos con estándares de UI.

- Análisis de los distintos elementos en el módulo de mantenimiento:

Un error de UI detectado mediante las pruebas manuales en el sistema fue que la vista de las opciones de *Preferencias*, *Ayuda* y *Salir del sistema* estaba muy limitada debido a la mala coordinación entre la paleta de colores de fondo en el UI y dichos hyperlinks.

- HU añadir producto:

Al empezar el sprint este método funcionaba parcialmente bien, pese a que era capaz de añadir los registros tomados por la interfaz en la base de datos, no podía comprobar la validez de los registros que se añadían, tómesese como ejemplo el campo de precio, que admitía sin problemas valores negativos. En este punto del proyecto esta funcionalidad fue optimizada en este aspecto y en el manejo de los hilos de trabajo, ya que el mensaje de confirmación hacía que el hilo central quedase a la espera de la confirmación, esto podría confundir al usuario y hacerle creer que el programa está congelado. Para resolver este problema utilizamos una

instancia de la clase Thread, poniendo así el mensaje de retroalimentación en ese hilo de trabajo.

- HU de modificar producto:

Al igual que la funcionalidad anterior, la misma era capaz de realizar cambios en los registros en la base de datos, esto sin corroborar la veracidad de sus datos, problema que se pudo solventar y que a su vez refleja mensajes de retroalimentación al usuario del mismo modo que se implementó en la funcionalidad anteriormente descrita.

- HU de eliminar producto:

Funciona perfectamente desde el principio, cumple con los objetivos del negocio, como complemento se le añadió un mensaje de retroalimentación utilizando la técnica antes descrita.

- HU de selección de producto:

Funciona perfectamente desde el principio, pudiendo desplegar los datos seleccionados en la caja del CRUD.

- HU de nuevo producto:

Su funcionamiento era muy deplorable, no limpiaba los registros existentes en la caja del CRUD ni mostraba al usuario el código que implementaría este registro nuevo. Pudimos arreglar aquello y en este momento dicha funcionalidad se encuentra en perfectas condiciones.

- HU de búsqueda de archivos:

Anteriormente, al pulsar el botón de buscar el programa era capaz de traer el resultado de la búsqueda, ergo; no era capaz de limpiar la tabla para solo mostrar lo que se pedía, pudimos solventar esto y ahora mismo el usuario ve solamente lo que pide.

- HU para impresiones:

El hipervínculo no ejercía ninguna acción. Aún no ha sido implementado.

#### *3.1.1.3 Criterio de aceptación*

Una vez las pruebas manuales y las automatizadas realizadas en los distintos elementos del sistema arrojaron los resultados esperados (corrigiendo los errores descritos en el punto anterior) y tomando en cuenta que cumplieran también los estándares de UX, entonces las pruebas de interfaz de usuario se dieron por completadas.

#### *3.1.1.4 Consideraciones especiales*

No existen consideraciones especiales en este ámbito.

### **3.1.2 Prueba de integridad de datos y base de datos.**

#### *3.1.2.1 Objetivos de la prueba*

Los objetivos principales de esta prueba son detectar los distintos errores que puedan ocurrir en el proceso de mantenimiento de datos, verificar la correcta inserción de los registros, corregir errores de diseño y normalización, y verificar la consistencia y la seguridad que existe dentro del sistema con respecto al acceso a los datos según los roles de usuarios.

#### *3.1.2.2 Técnica*

Se evaluó el módulo de registro con datos válidos y no válidos los cuales los primeros fueron insertados sin problemas justo como debía de ser, en cambio con los datos inválidos no se validaron ciertos casos como la longitud de la contraseña, la inserción de números en campos como nombre, usuario y apellido y el correo electrónico no está correctamente validado ya que permite insertar el mismo correo electrónico varias veces.

Se debe agregar una validación en el módulo de productos en el cual solo permita al usuario en el momento de insertar un producto agregarle una marca de acuerdo a una categoría correspondiente ya seleccionada esto para evitar introducir información inválida, como por ejemplo: se introdujo un producto de categoría golosina de marca que ofrece productos de ferretería, se recomienda hacer un cambio en la base de datos para agregar una relación entre categoría y marca donde una categoría puede tener muchas marcas y una marca pertenece solo a una categoría.

#### *3.1.2.3 Criterio de aceptación*

Los métodos que conllevan el acceso a la base de datos funcionan correctamente, la integridad de los datos se mantiene a través de los diferentes procesos con los datos.

#### *3.1.2.4 Consideraciones especiales*

No existen consideraciones especiales en este ámbito.

### **3.1.3 Prueba de funcionalidad.**

#### *3.1.3.1 Objetivos de la prueba*

Los objetivos principales de esta prueba son mejorar y/o corregir los distintos errores de funcionalidad, resolviendo problemas de sintaxis en el código, y verificando la funcionalidad del módulo de mantenimiento, las consultas a la base de datos, las opciones de preferencias y la salida segura del sistema.

#### *3.1.3.2 Técnica*

Verificación de los archivos marcados con errores de sintaxis por el IDE utilizado (Netbeans).

Se encontró un error de sintaxis en el archivo ControlesBasicos.Java, línea 19, el cual impedía el iniciar el programa, se recomienda cambiar la condición evaluada en "pregunta ==yes" por "pregunta == JOptionPane.OK\_OPTION". Para solucionar el error de sintaxis y así mismo la funcionalidad de salir del sistema.

Se encontró un error al intentar instanciar una clase llamada "PreparedStatement" en el archivo ProductoController.java en la línea 272, esto debido a que no estaba importando el paquete "java.sql.PreparedStatement", se recomienda importar dicho paquete.

Se encontro un error al utilizar una variable no declarada llamada "n" en el archivo ProductoController línea 144, se recomienda declarar la variable y asignarle el resultado de la ejecución de "estado.executeUpdate()" a la misma para ser evaluada.

Se encontró un error ocasionado por una incorrecta validación en RegistroController en las líneas 149-155 en el cual la variable declarada "n" igualada al resultado del método "estado.executeUpdate()" en caso de ser mayor a 0 mostraba un mensaje de error, pero dicho método retornaba siempre valores mayores a 0 en donde 1 hace referencia a modificaciones en alguna fila de la base de datos y 2 que no se modificaba nada. Por lo



cual se recomienda cambiar la validación a "n==1" y el mensaje a mostrar por: "Se ha registrado correctamente".

Se encontró un error de sintaxis en el archivo RegistroController.Java en la línea 43, este ocasionado por colocar el nombre de la variable mal, se recomienda cambiar dicho nombre para adecuarlo a la variable.

Se encontró un error al leer una dependencia, la cual no estaba integrada en el proyecto en el archivo LoginController en la línea 20, con el import "org.apache.commons.codec.digest.DigestUtils;" se recomienda integrar la librería al proyecto, la dependencia al descargar e integrar es "commons-codec-1.14".

Se encontró un fallo al intentar realizar un registro de un nuevo usuario, este error se origina en el archivo RegistroController.java en las líneas 129-132, el error radica en query del insert para los datos, en donde el nombre de la tabla esta incorrecto y el nombre de un campo por igual, estos estaban colocados de la siguiente manera: tabla="usuario" y campo="email" siendo en la base de datos de la siguiente manera: tabla="usuarios" y campo="correo", se recomienda cambiarlos.

Clase LoginController; línea 73, Error de sintaxis: El método perteneciente a la clase DigestUtils presenta un error de sintaxis al tener un 1 intercalado entre las letras "a" y "H" que impiden la correcta llamada del método en cuestión. El mismo fue eliminado para resolver problemas en la compilación del programa.

Con el mismo tenor, la clase RegistroController presenta el mismo error de sintaxis que impide el compilado del programa visto en el comentario anterior, ubicado en la línea 139, al igual que en el caso anterior el 1 intercalado entre el método shaHex fue eliminado.

Clase ProductoController, línea 75: Se ha encontrado un error de sintaxis en la consulta a la base de datos, en dicha consulta hicieron referencia a la tabla categoría como "category", este error impide la carga de datos de dicha tabla, por ende, fue corregido por el nombre correspondiente.

ProductoController, línea 168: Nos encontramos con un error de lógica alojado en el bucle for que itera en esta línea, en este punto el iterador "i" tiene como límite ser menor o igual a rs.getMetaData().getColumnCount()+1, donde rs.getMetaData().getColumnCount() es la cantidad total de columnas traídas por la consulta, al sumarle 1 a este método estamos pidiéndole al iterador que exceda la cantidad de columnas que abarca la consulta y por ende, la consola nos arroja un error de tipo "out of range", para solucionarlo solo bastó con retirar el 1 de la sentencia.

ProductoController, línea 278: Error de sintaxis SQL; La sentencia alojada en el string sql tiene un error al mencionar "INSERT INTO PRODUCT" siendo esta

última palabra la que impedía el funcionamiento de la consulta. Para solucionarlo bastó con escribir correctamente la tabla donde se insertarán los datos: "producto".

ProductoController, error de validación para el input Precio: Este input permitía la inserción de registros negativos y del 0, valores que no deben ser posibles. Además, controlamos una excepción que ocurre al agregar números que excedan el valor máximo de un entero.

### 3.1.3.3 Criterio de aceptación

El programa se ejecuta correctamente sin mostrar errores por pantalla al momento de inicializarlo.

### 3.1.3.4 Consideraciones especiales

No existen consideraciones especiales en este ámbito.

## 4 Recursos

### 4.1 Roles:

En la siguiente tabla se muestra la composición de personal para el proyecto Aplicación Punto de Venta en el área de Verificación del Software.

Rol	Responsabilidades
<b>Responsable de verificación</b> Paulina Gómez	<ol style="list-style-type: none"><li>1. Crear el plan de verificación.</li><li>2. Generar el modelo de prueba.</li><li>3. Evaluar las actividades de cada sprint.</li><li>4. Recibir los informes de verificación.</li><li>5. Evaluar la verificación final del sistema.</li></ol>
<b>Asistente de verificación</b> Wilmer Guerrero	<ol style="list-style-type: none"><li>1. Ejecuta las pruebas.</li><li>2. Registra los resultados.</li><li>3. Documenta las necesidades de cambio.</li></ol>
<b>Administrador de base de datos</b> Dawin Pérez	<ol style="list-style-type: none"><li>1. Gestiona y mantiene la base de datos.</li></ol>

## 4.2 Sistema:

Recurso	Nombre/Tipo
Servidor de base de datos	PostgreSQL
Nombre del servidor	InspeccionPostgresql
Nombre de la base de datos	Inspeccion
Máquina virtual #1	Req. Medios
Máquina virtual #2	Req. Mínimos
Máquina virtual #3	Recomendados

## 5 Hitos del proyecto de verificación

Primer Sprint			
Actividad	Esfuerzo	Fecha de inicio	Fecha de fin
Planificar la verificación	3	25/05/2020	25/05/2020
Elaborar casos de prueba	2	26/05/2020	26/05/2020
Ejecutar la verificación	5	03/06/2020	03/06/2020
Evaluar la verificación	3	04/06/2020	04/06/2020

Segundo Sprint			
Actividad	Esfuerzo	Fecha de inicio	Fecha de fin
Elaborar casos de prueba	2	05/06/2020	05/06/2020
Ejecutar la verificación	5	11/06/2020	11/06/2020
Evaluar la verificación	3	12/06/2020	12/06/2020

## 6 Entregables

### 6.1 Modelos de casos de prueba

Documento	<b>Modelo de Casos de Prueba</b>
Creado por	Paulina Gómez
Para quien	Equipo de desarrollo y calidad.

Fecha de liberación	26/05/2020
---------------------	------------

## 6.2 Informes de verificación

Documento	<b>Informe de verificación de prueba unitaria - Prueba de Funcionalidad</b>
Creado por	Jesús Estévez y Wilmer Guerrero
Para quien	Paulina Gómez
Fecha de liberación	04/06/2020

Documento	<b>Informe de verificación del sistema - Prueba de Funcionalidad</b>
Creado por	Jesús Estévez y Wilmer Guerrero
Para quien	Paulina Gómez
Fecha de liberación	04/06/2020

Documento	<b>Informe de verificación del sistema - Prueba de Integridad de datos</b>
Creado por	Nizar Sánchez y Dawin Pérez
Para quien	Paulina Gómez
Fecha de liberación	04/06/2020

Documento	<b>Informe de verificación de prueba unitaria - Prueba de Integridad de datos.</b>
Creado por	Nizar Sánchez y Dawin Pérez.
Para quien	Paulina Gómez
Fecha de liberación	04/06/2020

## 6.3 Evaluación de la verificación

Documento	<b>Evaluación de la verificación - Prueba de interfaz de usuario.</b>
-----------	---

Creado por	Wilmer Guerrero.
Para quien	Equipos de desarrollo y calidad.
Fecha de liberación	04/6/2020

Documento	<b>Evaluación de la verificación – Prueba de integridad de datos.</b>
Creado por	Dawin Pérez.
Para quien	Equipos de desarrollo y calidad.
Fecha de liberación	04/6/2020

Documento	<b>Evaluación de la verificación – Prueba de Funcionalidad.</b>
Creado por	Jesús Estévez.
Para quien	Equipos de desarrollo y calidad.
Fecha de liberación	04/6/2020

## 6.4 Informe final de verificación.

Documento	<b>Informe final de verificación.</b>
Creado por	Paulina Gómez
Fecha de liberación	14/06/2020

## 7 Apéndice.

### 7.1 Niveles de gravedad de error

Error	Nivel
Errores de sintaxis.	Catastrófico.
Errores de validaciones.	Marginal.
Errores de hilos de trabajo.	Marginal.
Errores de integridad de los datos.	Crítico.

## 7.2 Nivel de aceptación de los elementos verificados

Elemento	Nivel
Registro de usuarios.	Aprobado.
Inicio de sesión.	Aprobado.
Salida de sesión.	Aprobado.
Inserción de productos.	Aprobado.
Modificación de productos.	Aprobado.
Actualización de datos.	Aprobado.
Eliminación de productos.	Aprobado.
Sintaxis del código.	Aprobado con observaciones.
Prácticas de desarrollo.	Aprobado con observaciones.