

# Hermes (Tienda Virtual)

Jesús Andrés Rojas Montenegro, Santiago Díaz González, Cristian David Gómez Muñoz  
No. de Equipo Trabajo: 9

## I. INTRODUCCIÓN

En el presente documento, se realizará la explicación y descripción de una primera versión del proyecto final requerido para la asignatura de estructuras de datos.

Este documento incluirá la descripción del problema a resolver, usuarios del producto, requerimientos funcionales, descripción de la interfaz de usuario, descripción de los entornos de desarrollo y operación y un link al repositorio del proyecto en Github.

## II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

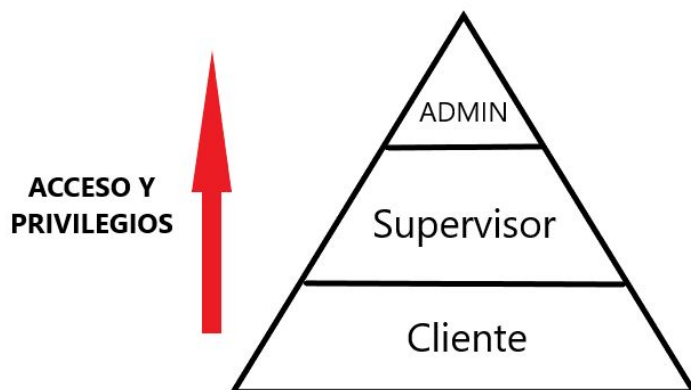
A grande rasgos el objetivo es poder crear una tienda virtual eficiente, accesible e intuitiva a la hora de ser utilizada por los usuarios.

Por lo que, haciendo énfasis en procesos considerados muy importantes en una tienda virtual, como lo son la búsqueda y manejo de productos, el proyecto funcione, en lo posible, ahorrando en memoria utilizada como en tiempo de ejecución; es decir, lograr la mayor eficiencia posible.

## III. USUARIOS DEL PRODUCTO DE SOFTWARE

La caracterización de los roles de usuario se hará por privilegios de acceso al manejo de productos y la respectiva funcionalidad que deba hacer:

- (1) Administrador
- (2) Supervisor de productos
- (3) Supervisor de ventas
- (4) Cliente



## IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

### A. Manejo de productos:

*Descripción:* Los productos están pensados como objetos con varios parámetros básicos, como lo son: ID, Nombre, precio, tags, fotos, reseña del producto, etc. Por lo cual se necesita un **Almacén** de este tipo de objetos donde se puedan eliminar, añadir y editar. Además, que en este Almacén no todo el mundo tenga acceso.

*Acciones iniciadoras, comportamiento esperado y requerimientos funcionales:* Un **Supervisor** está habilitado para manejar el Almacén, es decir, puede crear, añadir, remover y editar productos. El sistema debe rectificar con proveedores en la vida real si los cambios hechos son válidos o no. Por consiguiente, requiere una constante cantidad de actualizaciones y consultas de los datos.

Así, esto requiere que el sistema tenga una sólida comunicación con los datos de la vida real, previniendo errores humanos, y un eficiente manejo de productos.

### B. Uso del carrito de compra:

*Descripción:* Cada **Cliente**, al tener una cuenta, tiene un **Carrito**, el que sirve para apartar los productos que sea posible que compre. Además, este hace que sea fácil la decisión final de qué comprar y qué no comprar.

*Acciones iniciadoras, comportamiento esperado y requerimientos funcionales:* El Cliente necesita entrar al navegador de productos, donde hay únicamente productos válidos, es decir que están revisados por el Supervisor. A lo que, el cliente escoge un producto para añadirlo a su carrito. Va a la página de carrito y escoge si comprar únicamente ese producto, eliminarlo porque ya no lo va a comprar o seguir buscando más productos.

Así, que se requiere una página especial de Carrito, una validación de productos y una muy eficiente consulta y ordenamiento de productos para que el Cliente navegue con facilidad.

### C. Ordenar los productos:

*Descripción:* Teniendo en cuenta los parámetros básicos de un producto, podemos ver que estos pueden ser ordenados con respecto a cualquiera de dichos parámetros. Este orden puede variar en función de la necesidad del cliente o el supervisor de la tienda. Por lo que ambos tiene acceso a ésta función.

*Acciones iniciadoras, comportamiento esperado y requerimientos funcionales:* En el caso de un cliente, puede elegir cómo quiere ver los productos ordenados en su cuenta, para un supervisor, el orden influirá en la página de

inicio de la tienda, afectando así a otros usuarios. En ambos casos, se debe elegir algún parámetro (id, precio, nombre, etc.), luego debe escoger si quiere que se muestran los productos en un orden de mayor a menor o menor a mayor respecto a dicho parámetro. El sistema debe comparar los parámetros de los productos entre sí para poder ordenarlos correctamente. Por lo tanto, se necesita una estructura de datos que permita comparar parámetros de forma eficiente.

#### **D. Búsqueda de producto:**

*Descripción:* Para la búsqueda de un producto, se siguen teniendo en cuenta los parámetros básicos de éste objeto. Un cliente puede buscar un producto para añadirlo a su carrito y comprarlo posteriormente, mientras que un supervisor puede buscarlo para revisar el stock, actualizar algún parámetro o eliminar el producto.

*Acciones iniciadoras, comportamiento esperado y requerimientos funcionales :* Tanto un administrador como un cliente, ingresarán información en una barra de búsqueda. Ésta información será comparada con los parámetros de los productos para encontrar coincidencias entre éstos y la información ingresada en la barra de búsquedas. Si hay productos con parámetros que coinciden con la información dada, se mostrarán éstos productos en orden de mayor coincidencia a menor coincidencia, si no hay coincidencias, aparecerá un mensaje diciendo “No se ha encontrado el producto”.

Esto significa que se requiere un sistema de comparación eficiente entre los datos introducidos en la barra y los parámetros de los productos.

#### **E. Comprar producto:**

*Descripción:* Establecimiento final y crucial del trato con el cliente para recibir el dinero y en donde se actualizará el Almacén para efectuar la entrega.

*Acciones iniciadoras, comportamiento esperado y requerimientos funcionales :* El cliente va al carrito de compra y selecciona un botón de efectuar compra y si tiene dinero “válido” se contactará con un Supervisor para la generación de una factura, se actualizará el Almacén y se procederá a la entrega del producto.

Si la transacción no es posible se procederá a un Supervisor, pero si el error es del cliente (malos datos, no tiene dinero, no le alcanza, etc.) el sistema le notificará.

Así pues, se requiere acceder a un sistema de dinero externo a la página (que va usar el cliente) y uno interno que controle las cuentas de la tienda. Además de un proceso de generación de facturas que valide los datos (tanto de dinero como de productos).

#### **F. Registro Seguro y fácil manejo de la cuenta para los usuarios:**

*Descripción:* Los usuarios, tanto clientes como supervisores necesitan que su información personal esté segura, además de poder actualizar el estado de su cuenta de manera eficiente. Por lo tanto se requiere un sistema sencillo pero seguro para que puedan manejar dicha información.

*Acciones iniciadoras, comportamiento esperado y requerimientos funcionales :* Para los usuarios administrativos, solo se permitirá crear una cuenta en un

computador con dirección IP en la base de datos de la empresa. Además, deberá haber más personal de la empresa que verifique su identidad. En cuanto al sistema, Primero se deben introducir los datos personales para manejar la cuenta, luego, no sólo se verificará la dirección IP del pc, también enviará un mensaje SMS al celular que el usuario ha puesto en su información personal para verificar su identidad.

En cuanto a los clientes, se les pedirán sus datos personales, y se les enviará un correo de verificación cada vez que alguien intente loguearse a su cuenta. Además, la primera vez que se haga una compra, se hará una verificación con la entidad bancaria correspondiente.

#### **G. Mostrar a los clientes productos seleccionados**

*Descripción:* Los clientes al loguearse, en su página de inicio, se le mostrarán productos recomendados basados en sus compras o en publicidad.

*Acciones iniciadoras, comportamiento esperado y requerimientos funcionales :* El cliente se loguea y ve productos que probablemente le gustarán y si es publicidad estará marcado el producto para dejar claro que es publicidad.

Entonces, el software necesitará un algoritmo de sugerencias basado en los gustos del cliente o en compras anteriores y que pueda ser modificado para poner productos que se quieran (publicidad), el cual debe ser muy eficiente ya que se ejecutará cuando el usuario entre.

#### **H. Los clientes pueden dar a la tienda sugerencias para nuevos productos:**

*Descripción:* La afinidad con los clientes es una prioridad en cada tienda, por lo tanto, los clientes pueden enviar sugerencias a la tienda para que oferten nuevos productos.

*Acciones iniciadoras, comportamiento esperado y requerimientos funcionales :* Desde una cuenta creada en la página de la tienda, los clientes tienen la opción de enviar una sugerencia, la cual es un e-mail que va dirigido al correo electrónico de la tienda, al cual tienen acceso las personas que trabajen en soporte. Esto significa que la tienda requiere de un sistema de correo electrónico, en el que cada cuenta de cliente puede escribir a un único correo que está siendo administrado por el personal.

#### **I. Mostrar datos estratégicos a los dueños de la tienda**

*Descripción:* Los Supervisores y dueños podrán tener consultas para ver estadísticas de los distintos productos.

*Acciones iniciadoras, comportamiento esperado y requerimientos funcionales :* Los Dueños tendrán una página especial donde acceder a las estadísticas de los productos. Por lo que se necesita un algoritmo de Big Data para esto.

## V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Página de inicio:



Página de inicio de sesión:

Al presionar el botón “Log In”, aparecerá el botón “Sign In” y se reducirá el tamaño de la barra de búsqueda, para que aparezcan los campos para nombre de usuario y contraseña.



Página de registro/crear usuario:



Página Carrito:



## VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

El Software será desarrollado en el lenguaje de programación Java. El código será subido a un repositorio en BitBucket, donde será actualizado periódicamente.

Se pretende que el software pueda correr en celulares, tabletas y PC, desde que su sistema operativo sea compatible con Java.

Para la versión final se pretende que el software opere hosteado en la red y una versión de escritorio para los trabajadores de la tienda virtual.

## VII. PROTOTIPO DEL SOFTWARE

Link del repositorio:

<https://Luvakor@bitbucket.org/Luvakor/hermes.git>