|  |  |
| --- | --- |
| Logotipo, nombre de la empresa  Descripción generada automáticamente | Programación con Estructuras de Datos.(PED941 G01T) **Ciclo 01 -2022** |

PROYECTO FASE 1

Docente: Carmen Celia Morales Samayoa

Integrantes del equipo:

|  |  |  |
| --- | --- | --- |
| Carné | Nombres |  |
| MM211719 | Molina Daniel Ernesto |  |
| PM211493 | Pérez Molina Edgardo Baltazar |  |
|  |  |  |

Repositorio Daniel: <https://github.com/Practica-ADS/Proyecto-PED-fase-2.git>

Repositorio Edgardo: https://github.com/CanIChangeThisLaterOrNot/PED-Proyecto

## Índice

Introducción……………………………………………………………………………………..3

Marco teorico……………………………………………………………………………………4

Problemática…………………………………………………………………………………….4

Estructuras de datos seleccionadas para resolver la situación problemática y su forma de aplicación……………………………………………………………………………………….5

Diagrama UML…………………………………………………………………………………5

Referencias……… …………………………………………………………………………….6

## Introducción

Cuando nosotros estamos diseñando algún software, lo más seguro es que hemos sentido frustración por un problema lógico, el cual otra persona ya ha pasado por esto mismo, es decir, que otra persona ya haya programado algo similar o igual al problema en que nos encontramos. Mucho antes, en la web no había la cantidad de tutoriales que los hay hoy para un problema lógico de programación, en aquel tiempo, era ser un pionero en programación, encontrar un algoritmo que redujera la cantidad de tiempo que le tomaría a una persona humana hacer la misma acción, por ejemplo, sumar grandes cantidades de números en cuestión de segundos o fracciones del mismo, o efectuar una interfaz en donde se pueda llevar a cabo acciones que pueden afectar de manera automatizada a otras partes del sistema. Situándonos en el presente, cada vez se hace más fácil la programación, debido a los lenguajes de alto nivel que hacen una mímica de lo que una persona quisiera decir para que la maquina lo transcriba a la acción de la palabra que llama a una función dentro del proceso de comunicación entre máquina-usuario. En el presente trabajo aplicando lo aprendido acerca de estos lenguajes buscamos dar solución a una problemática que nos ayudara a poner en práctica lo aprendido y plasmarlo en un solución para un cliente que aunque sea ficticio nos ayudara en el ambiente laboral en el futuro cercano.

## Marco teórico

## Problemática

En una pequeña red de tiendas que busca modernizar su forma de inventariar sus productos, ya que desde que empezaron su negocio han estado registrando todo en libros y a puño y letra, debido a complicaciones que han tenido a la hora de llevar el control de cuantos productos hay disponibles y cuales productos ya se están terminando para ordenar más, ingresar nuevos productos o retirar de los inventarios los productos que ya no se venderán en el supermercado y el cliente expresa su disgusto debido a confusiones que han causado perdidas a la tienda, el cliente busca una forma mas moderna de llevar el inventario de sus productos en venta.

## Estructuras de datos seleccionadas para resolver la situación problemática y su forma de aplicación.

**Base de datos**

Para la base de datos se crearán las siguientes tablas

* Usuarios
* Productos

La base de datos puede estar sujeta a cambios.

**Programa de inventario en C#**

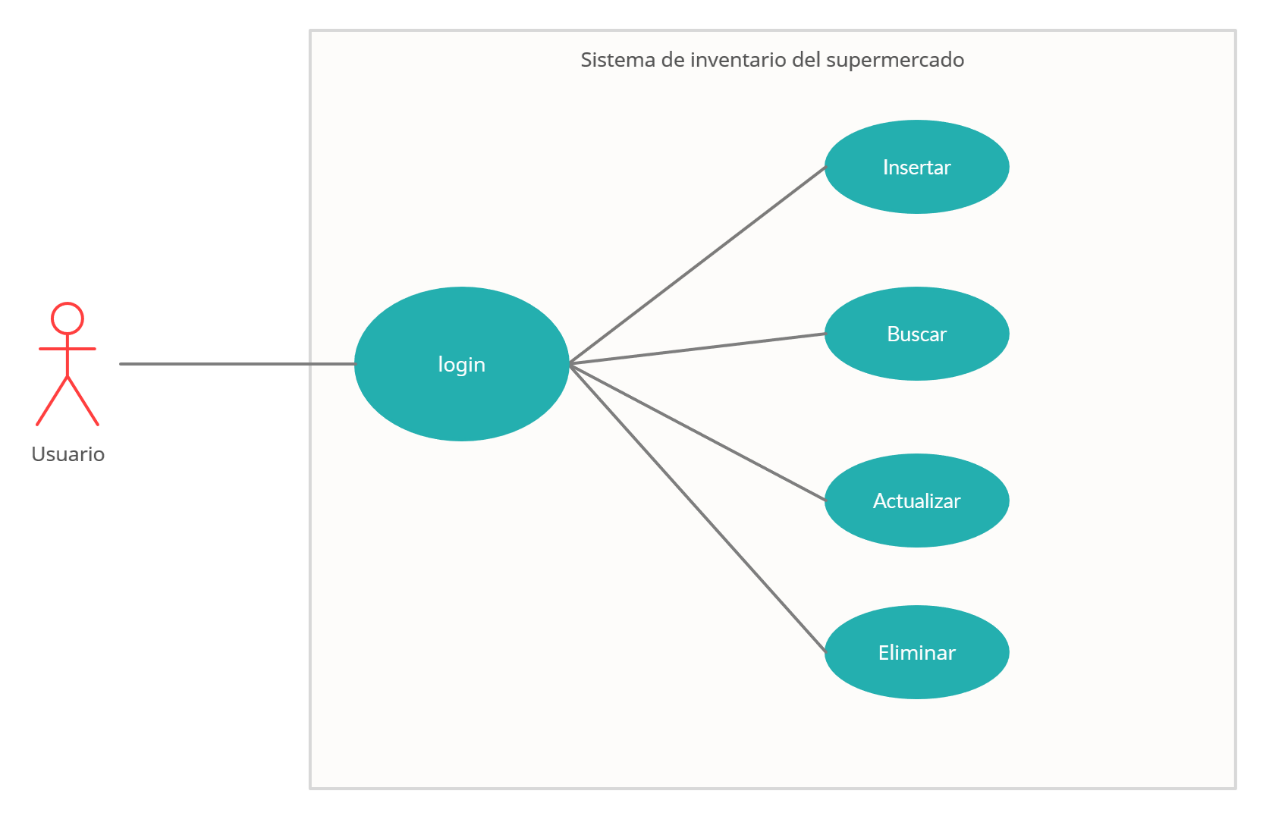
* El programa contendrá una ventana de Login donde se pedirán las credenciales del empleado para verificar su identidad.
* El programa en C# contendrá 4 funciones que se utilizara para llevar el inventario del supermercado y están detalladas a continuación:

1. Se necesitará registrar los siguientes datos del producto en la base de datos:

* Nombre del producto
* Cantidad
* Tipo
* Marca
* Fecha de ingreso
* Fecha de caducidad
* Precio
* Enlatado (Si/No)
* Código de barras (Opcional).

1. Se necesita la función para buscar un producto único mediante su código caso contrario buscar por medio de un árbol binario de datos, procurando que sus resultados se aproximen conforme vaya descartando datos por medio de la recursividad lo cual seria la parte principal en la que se basarían actualizar y borrar.
2. Se necesitará la función Actualizar, se buscará un producto único como en el caso de la función de búsqueda los cuales tendrán características similares en la parte de verificar el producto en el inventario luego de encontrar el resultado se modificará la información del producto encontrado.
3. Se necesitará de la función borrar que buscara un producto especifico como en el caso de actualizar o buscar y tendrán características similares, luego de encontrar el resultado de la búsqueda se procederá a mostrar una advertencia al momento de dar click en la opción de eliminar si está seguro de proceder con la acción ye que una vez eliminado no se puede recuperar.

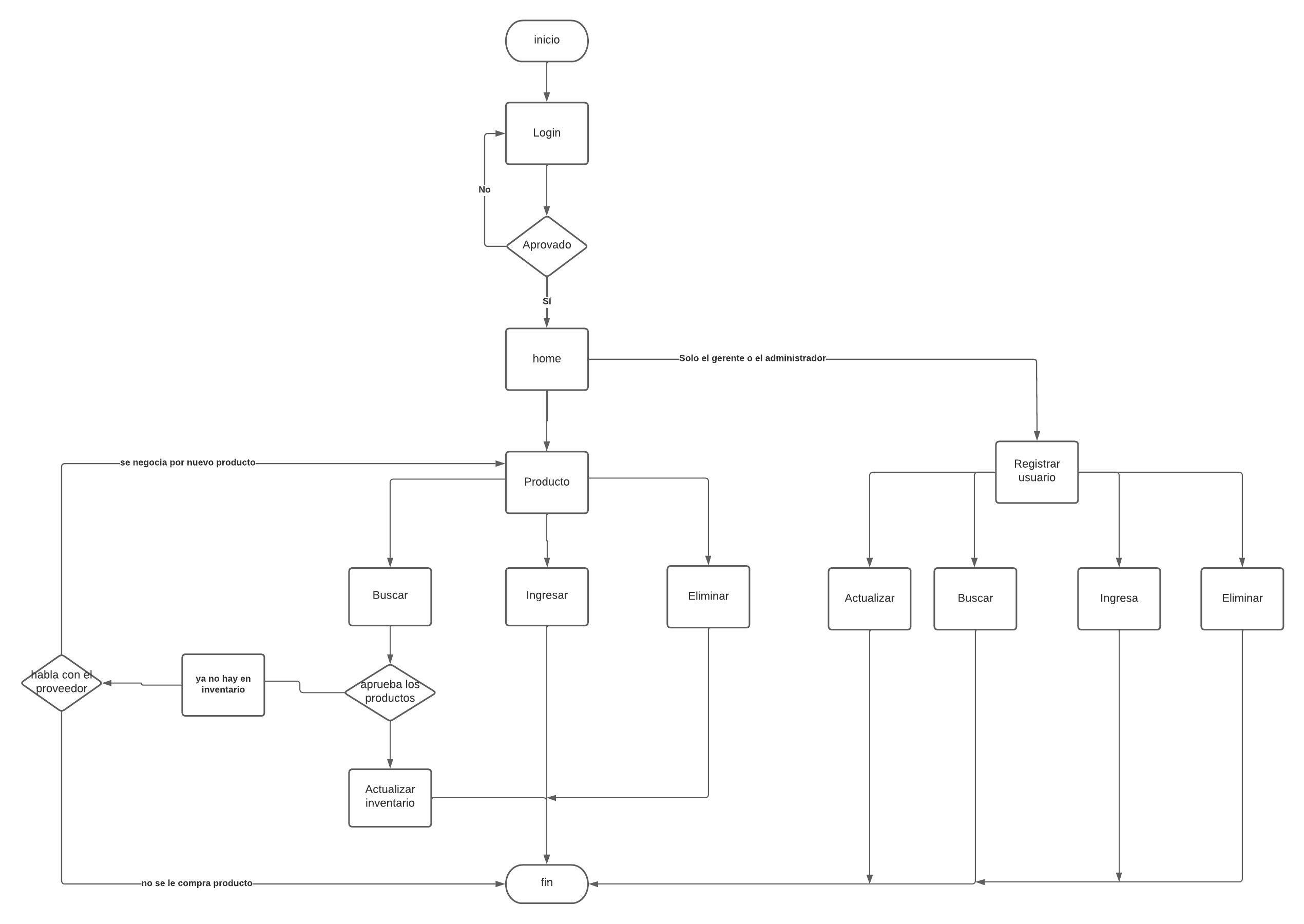
## Diagrama UML



## Diseño del sistema

## Diseño del proyecto

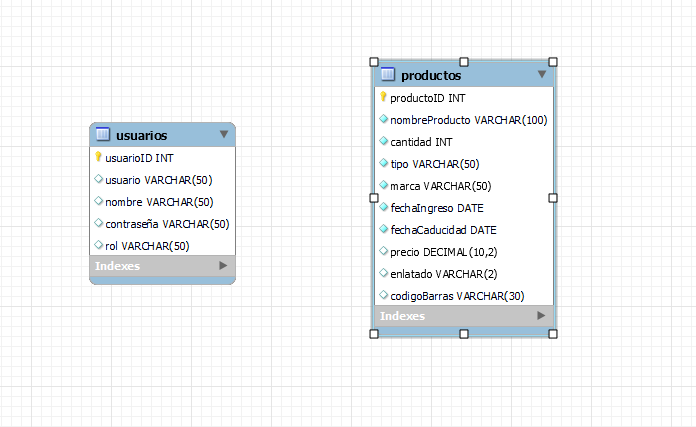
El proyecto en cuestión debía de ser una interfaz amigable para un usuario inexperto en el uso de las computadoras para que sea fácil de usar, su finalidad es ayudar al usuario para llevar un control de su inventario de modo que evite tener perdidas de producto que le hagan perder dinero a nuestro cliente, después de varias reuniones los diseñadores decidieron dividir sus esfuerzos y mientras uno de ellos se encarga de la seguridad del programa el otro trabajaría en la visualización y manipulación de los productos para después unir la interfaz en una sola depurando fallas y entregando a tiempo el producto final.



* **Login**: es la primera pantalla que ve el usuario y donde mediante sus credenciales puede ingresar al sistema si son validadas como correctas.
* **Home:** es la pantalla principal del programa donde se pueden acceder a múltiples opciones del sistema
* **Producto**: Es la pantalla mediante la cual el usuario o el gerente pueden trabajar en el inventario de la tienda.
* **Registrar usuario**: Es la pantalla en la cual el gerente puede ingresar nuevos empleados buscar o modificar los datos de sus empleados y eliminarlos cuando dejen su puesto.

**Funciones**

* **Insertar:** Función mediante la cual se agrega un nuevo producto al inventario o el gerente puede unir un nuevo empleado al staff.
* **Actualizar:** Función mediante la cual el usuario puede actualizar la información de un producto y el gerente puede actualizar la información de un empleado.
* **Borrar:** Función mediante la que el usuario puede eliminar un producto del inventario y el gerente puede eliminar del sistema a un empleado que ya no deba tener acceso al sistema
* **Buscar:** Función del sistema mediante la cual el usuario puede buscar un producto del inventario para ver si existe en el inventario o si tienen unidades de dicho producto y en el apartado de registrar usuario puede ser usado para encontrar la información de un empleado.



**Diccionario de datos.**

* **usuarioID:** Código único del empleado y es autogenerado por la base de datos.
* **Usuario:** Nombre único que se ingresa junto con la contraseña del empleado para validar su identidad.
* **nombre:** registro del nombre del empleado a quien se le asigna el código del empleado.
* **contraseña:** palabra o dígitos secretos que le permiten al empleado validar su identidad como parte del staff
* **rol:** Que tipo de usuario es.
* **productoID:** Codigo único delproducto.
* **nombreProducto:** Nombre del producto en el inventario.
* **cantidad:** Cantidaes de los productos
* **tipo:** De que clase de producto es.
* **marca:** la imagen representativa de la empresa que lo fabrico.
* **fechaingreso:** Fecha que llego a la tienda.
* **fechacaducidad:** Hasta que fecha aproximadamente se puede consumir el producto.
* **precio:** Cuanto cuenta el producto.
* **enlatado:** Verifica si el producto es enlatado para decidir el método de almacenaje.
* **código de barras:** Combinación de letras o de números que identifican un producto.

**Seguridad empleada en el sistema.**

* **SHA512:** es un conjunto de funciones hash criptográficas, diseñadas por la [Agencia de Seguridad Nacional](https://es.wikipedia.org/wiki/National_Security_Agency), en ese sistema se utilizo para cifrar la contraseña de los usuarios para mas seguridad.

## Fuentes de consulta

* Herramienta de Diagrama UML

<https://creately.com/es/lp/herramienta-de-diagrama-uml/>

* Normas Apa 2016 – Edición 6

<https://normasapa.net/normas-apa-2016/>

* **Setup Git and GitHub**

[**https://cfss.uchicago.edu/setup/github/**](https://cfss.uchicago.edu/setup/github/)