Grocery

Proyecto Computación 1

Iván Uriel Olvera Pérez, José Diego Guerrero, Luis Damián García

20 septiembre de 2021

Índice

Antecedentes históricos

El desarrollo y crecimiento económico en las pequeñas y medianas empresas en México, ha permanecido en constante crecimiento, sin embargo, más de la mitad de las empresas en el sector abarrotero afirman tener problemas de financiamiento que permitan invertir y hacer crecer sus negocios.

Las microempresas son pequeñas organizaciones empresariales, donde los trabajadores y administradores son los mismos dueños.

Planteamiento del problema

La mayoría de las pymes abarroteras tienen problemas estructurales, desfases de información con respecto a los proveedores, falta de administración en el inventario, conteo erróneo de productos, entre otras dificultades.

Por lo cual la implementación de un programa de administración de una tienda con funciones exclusivas para el dueño/administrador les permitirá a microempresas abarroteras facilitarse la administración manteniendo un conteo actual de sus productos.

***Nuestro programa se implementó al 100% en el lenguaje de programación c++**

Justificación

Un aumento en la eficiencia de las empresas abarroteras impactaría directamente en el ámbito local, beneficiando primeramente a los mismos dueños, empleados, proveedores y consumidores.

Debido al alto impacto que tienen las Pymes en México con respecto al PIB y a la generación de empleos, una buena administración permitirá una mejor calidad de vida para los empleados y consumidores.

Objetivos

Objetivo General

- Crear un programa que permita administrar una microempresa abarrotera mediante dos modos: Administrador y Empleado.

Objetivo Específico

- Implementar el uso de clases y objetos en nuestro programa
- Hacer uso de constructores en nuestro código

Hipótesis

La implementación del programa “Grocery” permitirá a micro y pequeñas empresas una administración más ordenada de sus productos y un impacto favorable en sus ganancias.

Trabajos relacionados

Existen múltiples trabajos relacionados encontrados desde la página de GitHub pero la mayor parte de ellos se han centrado en una identificación inteligente de los productos mediante una IA para análisis de imágenes o en aplicaciones con objetivo de crear listas y ordenar las compras de los clientes; la mayoría de ellos han sido creados en Python o en java. Algunos ejemplos de ellos son:

- https://github.com/marcusklasson/GroceryStoreDataset
- https://github.com/tobiagru/ObjectDetectionGroceryProducts
- https://github.com/couchbaselabs/GrocerySync-Android

Metodología

I) Lluvia de ideas

Para tener una idea más clara sobre un buen proyecto comenzamos a pensar en posibles problemáticas actuales

II) Filtración de ideas

Comenzamos a eliminar aquellas ideas que ya hayan sido tema de investigación reciente por alguien más Y aquellas que presenten problemáticas con soluciones computacionales más allá de los temas abordados en este curso para quedarnos con menos de tres posibles proyectos.

III) Definición de la idea

Al tener la idea del proyecto nos enfocamos en bosquejar el código, planteando el uso general de la aplicación, posibles variables que necesitaríamos, clases, objetos y métodos de esas clases.

IV) Investigación

Después de hacer un boceto de nuestra app, comenzamos a buscar recursos bibliográficos e ir realizando a la par, la sección 1 y 2 de este reporte.

V. Implementación del código

- Después de realizar la investigación, plantear la hipótesis, objetivos y demás secciones, comenzamos a realizar el código. Creando un nuevo proyecto en el IDE *“CLion”* con un archivo *main.cpp*.

- Dentro de nuestra función *main();* va el esqueleto del programa donde se hizo uso de los diferentes recursos aprendidos en clase como ciclos do while, for, switch case entre otros.

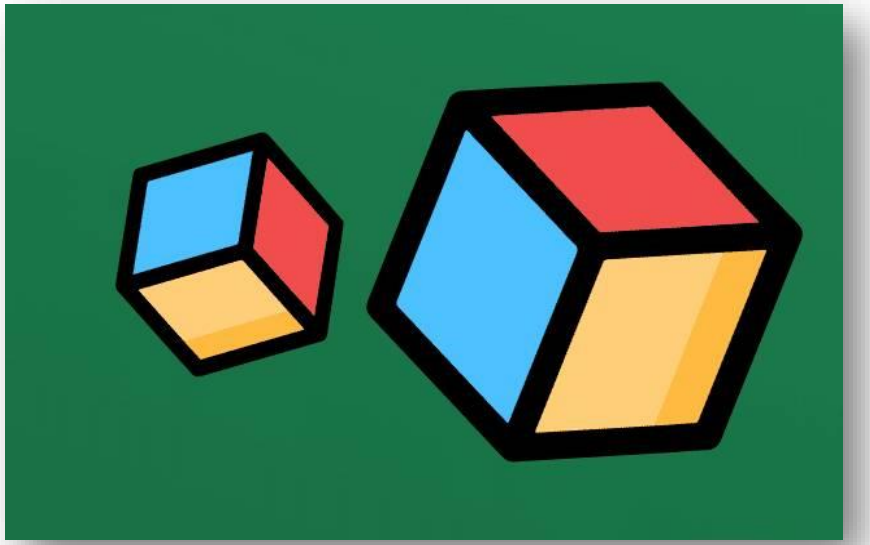
- Los do-while funcionaron para mantener el programa en ejecución. Los switch-case corresponden a las distintas opciones implementadas, desde elección de modo de usuario, hasta las funciones correspondientes a la administración de la tienda.

Programación Orientada a Objetos: ¿Por qué?

¿Cómo se puede programar una tienda?

El proyecto cuenta con clases separadas, cada una con su interface en headers y las declaraciones de sus métodos en archivos .cpp separados del main.

- Versatilidad
- Módulos
- Trabajo cooperativo
- Seguridad
- Capacidad de programar lo que sea



Programación Orientada a Objetos: Clases

Tienda

El objeto tienda tiene un nombre y un invetario. Implementa los métodos get() y set() necesarios para crear una tienda.

Producto

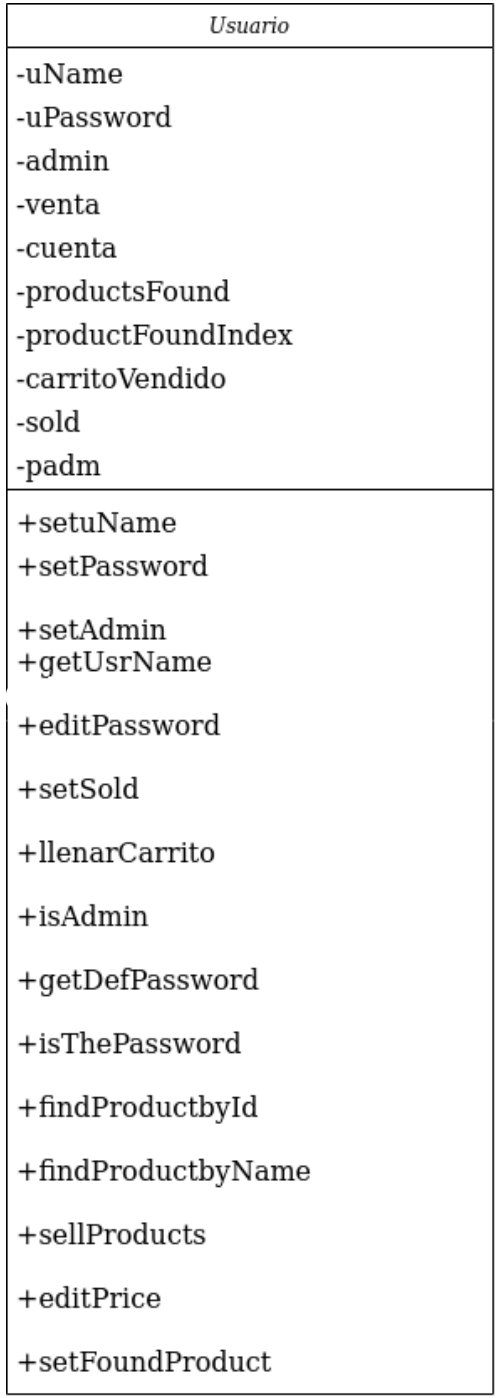
Cuenta con un nombre, id, fecha de caducidad. Implementa métodos get() y set() para crear un Producto.

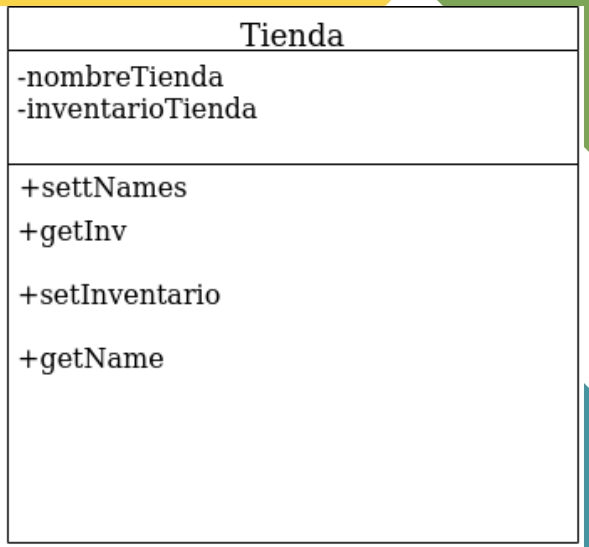
Usuario

El más complejo, se relaciona con y modifica los tres objetos. Además de implementar los get() y set(), cuenta con métodos para editar precios, remover del inventario, buscar productos lo cual logra por medio de reutilización de los métodos get() y set() propios y de los otros objetos.

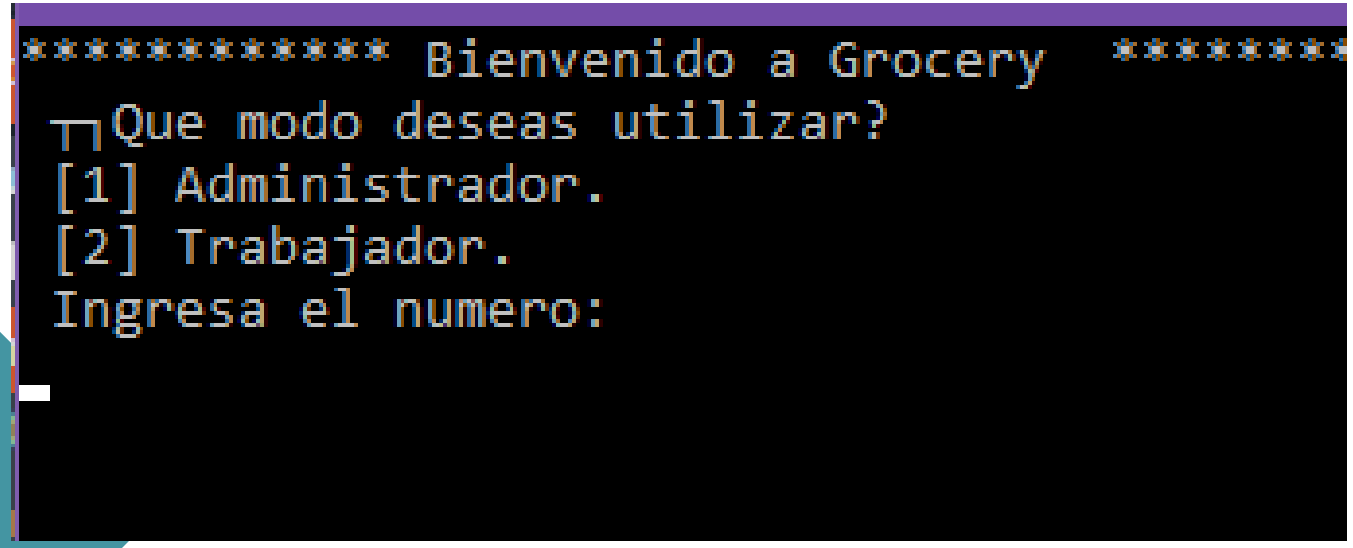
UML de las clases

En un diagrama UML se representan gráficamente las clases de un programa OOP. El nuestro consta de tres clases en las cuales indicamos su métodos y sus variables, indicando si son públicas, privadas o protegidas, con los símbolos +, - y # respectivamente.





Resultados



Se logró crear un programa funcional cumpliéndose el objetivo de generar una herramienta en la que puedan interactuar tanto un administrador como un usuario para acceder a disntintas funciones en búsqueda de lograr un mayor control y una mejor organización en la microempresa

Segunda Entrega

Bugs arreglados

1. Se arregló bucle infinito en varias opciones.
2. Ya no se interrumpe la ejecución en la opción de vender productos.

Bugs nuevos

1. Sale un error al ejecutar la opción de vender productos o añadir productos.

```
double free or corruption (out)

Process finished with exit code 134 (interrupted by signal 6: SIGABRT)
```

Mejoras

1. Las funciones ahora trabajan pasando objetos por referencia. Gracias a esto se pudieron implementar funciones para modificar usuarios y productos, además de añadir nuevos.

2. Se usan apuntadores para poder modificar y usar los objetos como producto y usuario sin extraerlos de sus respectivos vectores.

3. En conjunto con los puntos 2 y 1 se añadieron destructores para la opción vender productos. Se destruye desde un vector de apuntadores (carrito vendido).

Bibliografía

- de León Vázquez, I., Cruz Lugo, A., Laffit Anaya, A., Sosa Serrano, N. and Vega Hernández, M., n.d. Microempresas del sector abarrotero. *Xikua Universidad Autónoma del estado de Hidalgo*, 3.
- Conde, C. (2003). Fuentes de financiamiento para la microempresa en México. Portes, 68-69
- Romero Sánchez, B., Vargas Matamoros, K., González Hidalgo, K., Castañeda Gutiérrez, J. and Rodríguez Lozada, M., 2019. SITUACIÓN ECONÓMICA DE LAS MIPYMES DE ABARROTES EN XALOZTOC, TLAXCALA MÉXICO Y SU CAPITALIZACIÓN. *Revista de investigación interdisciplinaria en métodos experimentale*, 8(1).