[[1]](#footnote-1)

**SISTEMA DE GESTION DE PRODUCTOS (noviembre de 2024)**

Autores: Torres Pachón Juan Esteban, González Gonzáles Dinalut y Ruiz Rincon Jessica Jineth estudiantes de la fundación universitaria de San gil Unisangil (SEDE-CHIQUINQUIRA)

***Resumen*—En el presente documento hemos diseñado un algoritmo en Python, usando la programación modular para crear un sistema de gestión de productos establecidos en un inventario. Por lo que aplicamos una metodología en la que establecimos los requisitos y limitaciones del algoritmo. Al ejecutar el programa logramos identificar que el algoritmo calcula de forma precisa el área de las figuras geométricas.**

**Palabras clave-programación modular, sistema de gestión, inventario, clases, atributos, métodos, Python.**

**INTRODUCCIÓN**

 En este documento de muestra el proceso para el diseño de un algoritmo para la gestión de productos de un inventario por lo que es necesario aplicar una metodología la cual esta desarrollada a lo largo del documento con una descripción muy detallada con la finalidad de hacer uso y crear un algoritmo dentro del lenguaje de programación Python.

1. ***ANÁLISIS***

***Contexto***

Desarrollar un algoritmo Sistema de gestión de productos: En un sistema de gestión de inventarios, la información sobre productos debe ser organizada y manipulada con eficiencia

***B. Población***

Especialista de inventario.

***C. Limitaciones y alcance***

Las limitaciones y alcances que se pudieron identificar son las siguientes:

**Limitaciones**

* No genera factura
* No se pueden actualizar o almacenar nuevos productos.
* Cualquier persona que quiera era usar el código puede acceder a él.

**Alcances**

* Se puede gestionar la información requerida.
* El usuario puede interactuar con la consola.
* La interfaz es simple y fácil de usar

**III. OBJETIVOS**

***Objetivo general***

Diseñar un algoritmo donde se pueda hacer gestión de control dentro de un inventario donde se simulan compras y ventas con el fin de modificar el stock.

***Objetivos específicos***

* Gestionar productos del inventario.
* Gestionar la compra de productos (Aumentar stock).
* Gestionar la venta de productos (Disminuir stock).
* Facilitar la interacción del usuario.
* Presentar al usuario un programa que pueda manejar de forma comprensible y simple.

**IV. ESPECIFICACIÓN DE REQUERIMIENTOS.**

***CON RESPECTO AL DISPOSITIVO***

* Mínimo 4 GB de RAM.
* Sistema operativo: Windows 8 en adelante.
* CPU: Intel Core i3.
* Disco: HDD o SSD.

1. ***CON RESPECTO AL USUARIO.***

*Requisitos mínimos*

* *El usuario debe tener conocimientos previos acerca del funcionamiento del programa al momento de usarlo.*
* *El usuario debe saber para qué sirve el código*

**V. DOCUMENTACIÓN.**

1. ***Interna.***

*Toda la información relacionada con el desarrollo del código se encuentra almacenada en un repositorio en GitHub.*

*Enlace:*

1. ***Técnica.***

***Diccionario de datos:***

* ***Producto*** *que al relacionarlo directamente con la programación seria la clase.*
  + ***Nombre, Precio y Cantidad*** *seria n los atributos (cualidades del producto)*
* ***Inventario*** *contiene todos los productos disponibles.*
* ***Datos ingresados por el usuario el*** *usuario interactúa con la consola ingresado el nombre del producto al pie de la letra y la cantidad.*
* ***Tabla representación*** *gráfica del inventario*
* ***Mensajes*** *Alertar del que el usuario está ejecutando mal una función.*

***Descripción del problema:***

*Se identifico la necesidad de crear un sistema de gestión de inventarios, la información sobre productos debe ser organizada y manipulada con eficiencia*

**VI. DISEÑO DEL ALGORITMO**

Utilizamos la programación modular para hacer un análisis el cual utilizamos como guía, para luego adaptar en el lenguaje de programación Python, siendo este el estructurador del software

requerido.

***Modelos de análisis:***

Como modelo de análisis podemos identificar que se utilizó la programación como se muestra en la ilistacion1

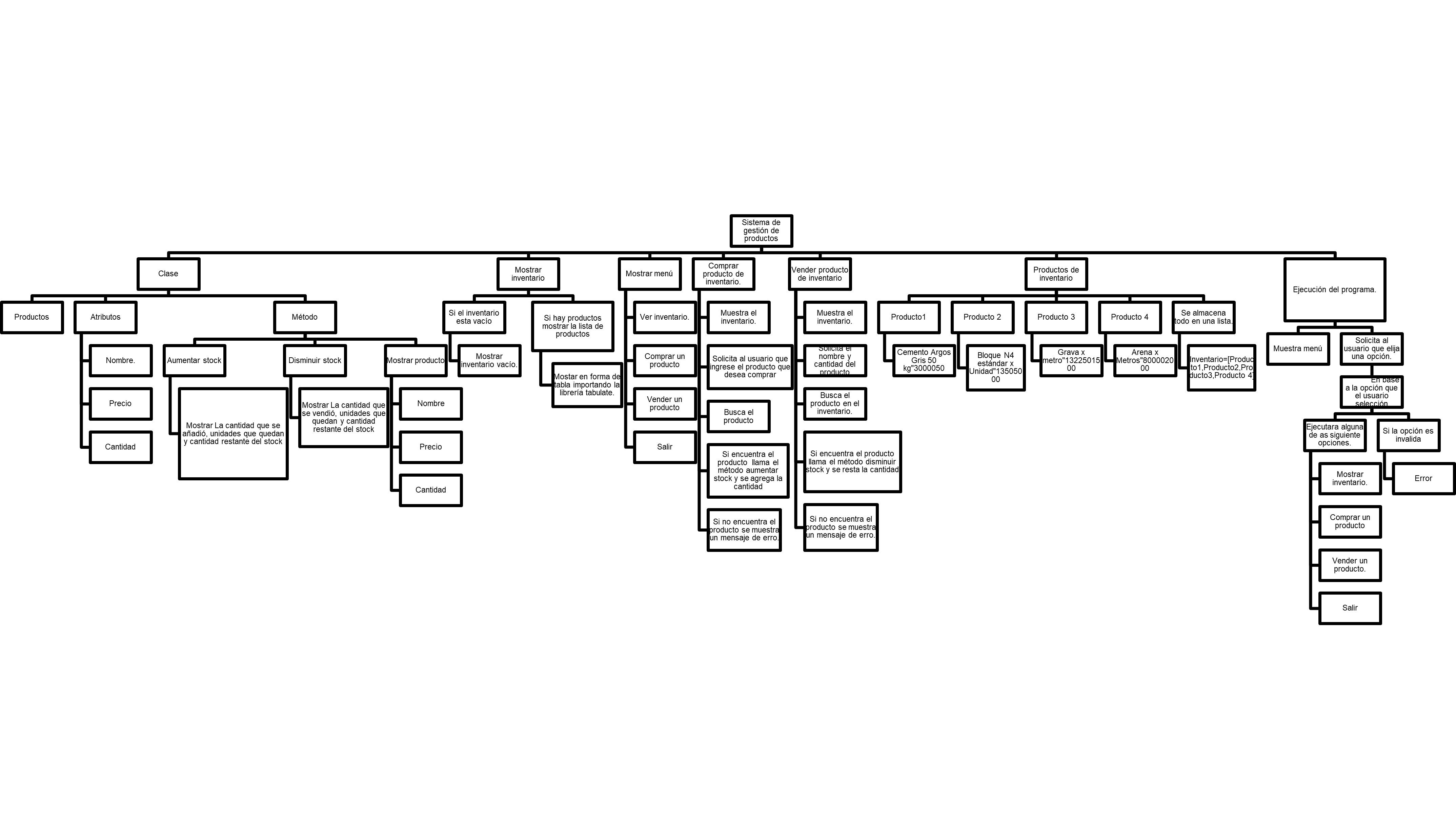


Ilustración 1.Esquema en el cual se almacena el análisis del problema y su solución.

Tabla

Descripción generada automáticamente

Ilustración 2.Diagrama de programación orientada a objetos.

**VIICODIFICACIÓN.**

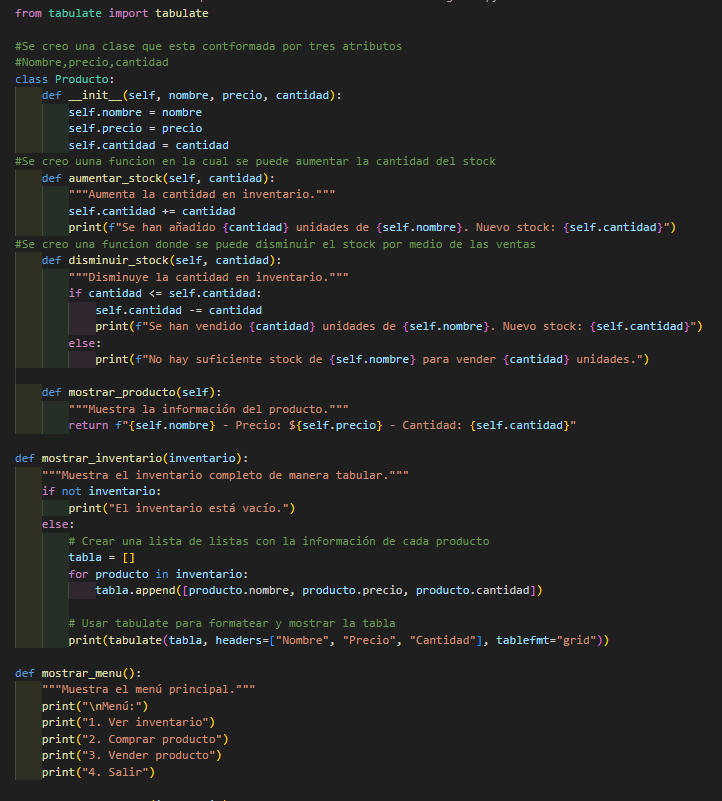
 Para la codificación del algoritmo utilizamos los siguientes programas:

* PSeInt.
* Visual Studio Code.
* Python
* Git hub.

1. ***Instrucciones del código fuente***

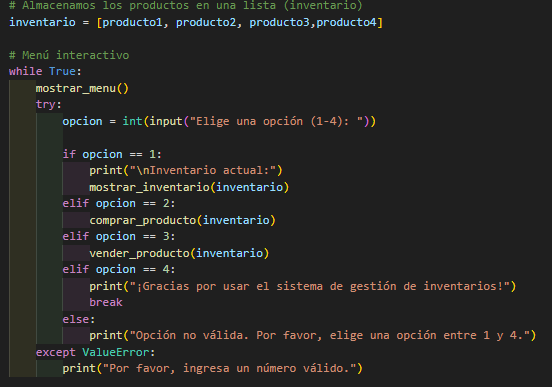
* **Programación orientada a objetos**
  + Clase=Producto
    - Atributos=nombre, precio, cantidad
    - Métodos=aumentar stock, disminuir stock y mostrar producto.
* **Funciones**
  + Mostar inventario se muestra el inventario utilizando la librería tabulate.
  + Mostrar menú Muestra el menu principal de opciones que el usuario puede elegir.
  + Comprar producto permite al usuario aumentar el stock del producto.
  + Vender producto: permite al usuario reducir el stock del producto.

1. ***Código fuente***

****

Texto

Descripción generada automáticamente



***VII. EJECUCIÓN DEL PROGRAMA.***

En este caso comenzamos con la apertura de la terminal de Git Bash. Si ya se creó el repositorio remoto y se sincronizan las cuentas de los desarrolladores con el repositorio local, se envía el archivo al Staging área con el comando git add.  Luego ejecutamos el git commit -m “Comentario” para confirmar los cambios, donde, posteriormente usamos git push origin o pb master o main dependiendo la rama que se esté usando con el fin de almacenar la información en el repositorio remoto de GitHub.

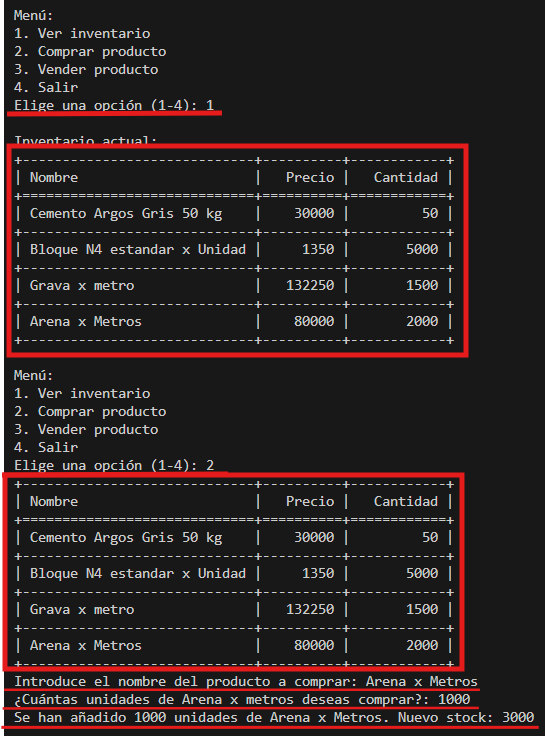


Ilustración 3.El usuario elige la opción 1 donde se le muestra el menu luego quiere agregar productos al inventario por lo que selecciona la opción 2 por lo que se le solicita al usuario que ingrese el nombre del producto, la cantidad y la información.

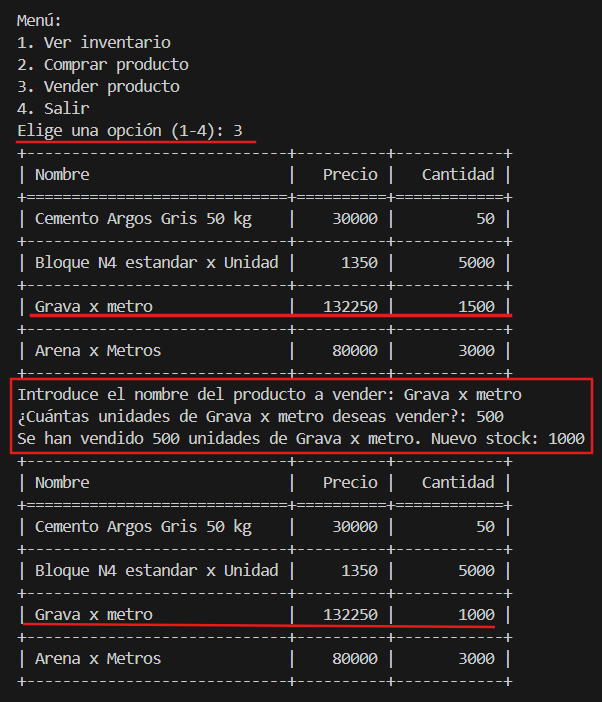


Ilustración 4.El usuario elige la opción 3 se le muestra el menu actualizado y le solicita que ingrese el producto que desea vender.

**VII. VERIFICACIÓN Y DEPURACIÓN.**

* 1. **Sintaxis:** se identificó que la sintaxis del código esta correctamente efectuada en el código.
  2. Se le muestra un mensaje al usuario en caso de que el inventario este vacío.
  3. Si el usuario ingresa un producto que no existe el menu le envía un mensaje indicándole que está equivocado.
  4. Si el usuario ingresa valores negativos el programa no se ejecuta y le solicita que ingrese nuevamente el valor
  5. El stock se actualiza correctamente cada ves que el usuario hace compra o ventas.
  6. Si el usuario ingresar las palabras en mayúscula o minúscula el programa las reconoce.
  7. El usuario puede ingresar la cantidad de productos que deseé sin un limité en específico.
  8. Si el usuario anexa algún dato en una parte del menu en especifico la tabla se actualiza correctamente.

1. ***Manual de usuario***

*Este manual va dirigido para que los usuarios puedan gestionar los productos dentro del inventario*

***Descripción del funcionamiento del programa:***

Después que el usuario ejecute el código se le desplegara un manual.

1. **Ver inventario**
2. **Comprar producto**
3. **Vender producto**
4. **Salir**

Donde puede seleccionar alguna de las siguientes opciones y se desplegara lo siguiente:

**Opción 1**

Se le mostrara el inventario actual

Inventario actual:

+-----------------------------+----------+-------------------+

| Nombre | Precio | Cantidad |

+=====================+=====+========+

| Cemento Argos Gris 50 kg | 30000 | 50 |

+------------------------------------+----------+------------+

| Bloque N4 estándar x Unidad | 1350 | 5000 |

+---------------------------------- -+----------+-----------+

| Grava x metro | 132250 | 1500 |

+-----------------------------------+-----------+------------+

| Arena x Metros | 80000 | 2000 |

+-----------------------------------+----------+------------ +

**Opción 2.**

El usuario ingresa la información de la comprar productos para almacenarlos en el inventario.

1. Ver inventario

2. Comprar producto

3. Vender producto

4. Salir

Elige una opción (1-4): 2

+-----------------------------------+----------+------------+

| Nombre | Precio | Cantidad |

+=====================+=====+========+

| Cemento Argos Gris 50 kg | 30000 | 50 |

+-----------------------------------+----------+------------+

| Bloque N4 estándar x Unidad | 1350 | 5000 |

+-----------------------------------+----------+------------+

| Grava x metro | 132250 | 1500 |

+-----------------------------------+----------+------------+

| Arena x Metros | 80000 | 2000 |

+-----------------------------------+----------+------------+

**Ejemplo:**

Introduce el nombre del producto a comprar: Arena x Metros

¿Cuántas unidades de Arena x metros deseas comprar?: 50

Se han añadido 50 unidades de Arena x Metros. Nuevo stock: 2050

**Opción 3**

El usuario ingresa la información de la venta de productos para actualizar el inventario.

Elige una opción (1-4): 3

+-----------------------------+----------+------------+

| Nombre | Precio | Cantidad |

+ ====================+======+=======+

| Cemento Argos Gris 50 kg | 30000 | 50 |

+-----------------------------------+----------+------------+

| Bloque N4 estándar x Unidad | 1350 | 5000 |

+-----------------------------------+----------+------------+

| Grava x metro | 132250 | 1500 |

+-----------------------------------+----------+------------+

| Arena x Metros | 80000 | 2050 |

+-----------------------------------+----------+------------+

**Ejemplo:**

Introduce el nombre del producto a vender: Bloque N4 estándar x Unidad

¿Cuántas unidades de Bloque n4 estándar x unidad deseas vender?: 1500

Se han vendido 1500 unidades de Bloque N4 estándar x Unidad. Nuevo stock: 3500

+-----------------------------------+----------+------------+

| Nombre | Precio | Cantidad |

+=====================+======+=======+

| Cemento Argos Gris 50 kg | 30000 | 50 |

+-----------------------------------+----------+------------+

| Bloque N4 estándar x Unidad | 1350 | 3500 |

+-----------------------------------+----------+------------+

| Grava x metro | 132250 | 1500 |

+-----------------------------------+----------+------------+

| Arena x Metros | 80000 | 2050 |

+-----------------------------------+----------+------------+

**Opción 4**

El usuario sale del menu:

Elige una opción (1-4): 4

¡Gracias por usar el sistema de gestión de inventarios!

**X MANTENIMIENTO.**

Por el momento esta sería la versión final de la calculadora porque a futuro se pueden plantear nuevas modificaciones y actualizaciones ya en un futuro se podría hacer un mantenimiento en base a la experiencia de los usuarios con el aplicativo

**XI. CONCLUSIONES.**

* En el presente trabajo logramos describir el diseño y codificación de un algoritmo el cual permite gestionar informaciones inventario.

* Implementamos un análisis situacional específico para captar con mayor detalle los requisitos, él cual pudimos utilizar para escribir el código de manera correcta
* Aprendimos que es necesario aplicar una metodología que hace más comprensible y efectivo el proceso a la hora de desarrollar un algoritmo.

* Empleamos la programación modular para hacer un análisis del problema abarcado y como resultado utilizarlo como guía para estructurar de una forma clara el algoritmo en Python.

* Para realizar la codificación del código utilizamos varios programas como Visual Studio Code y GitHub los cuales facilitan el trabajo en grupo y la gestión de información.

1. Documento elaborado septiembre 2024 [↑](#footnote-ref-1)