

REPORTE DE TÉCNICO DE DESARROLLO DE PROYECTO PROGRAMACIÓN

Juan Alberto Estrada García

**Desarrollo de herramienta de análisis
de ventas en tienda de productos**

Descripción del caso - Introducción
Desarrollo de la herramienta - Definición del código
Interfaz usuario-administrador
Desarrollo de las consignas - Solución al problema
Discusión - Conclusiones

DESCRIPCIÓN DEL CASO - INTRODUCCIÓN

Repositorio de Github:

<https://github.com/shift-Jaegger/Proyecto1-EMTECH-JuanAlbertoEstradaGarcia/tree/master>

El presente trabajo detalla el desarrollo de una herramienta para el análisis descriptivo de datos a través de la programación básica en lenguaje Python.

La premisa del proyecto es poder hacer una exploración del registro de ventas de una tienda de artículos electrónicos: *Life Store* a partir del registro hipotético de ventas del año 2020 completo.

Para la gerencia de la tienda, es importante obtener 3 tipos de resultados del desempeño de venta de los productos, así como las reseñas de los mismos y el reporte de ventas generales a nivel mensual y anual. a través de la programación de la herramienta se podrá obtener el resultado que requiere *Life Store*, no sólo de manera actual, sino que se podrá emplear iterativamente en el futuro para los mismos propósitos.

DESARROLLO DE LA HERRAMIENTA - DEFINICIÓN DEL CÓDIGO

La estructura del programa de Python sigue una estructura simple que sigue un comportamiento fractal para hacer de la obtención los diversos resultados requeridos algo factible a través de programación fundamental.

INTERFAZ USUARIO-ADMINISTRADOR

Se diseñó una interfaz sencilla donde se debe ingresar el usuario y después la contraseña. Los administradores registrados son:

- Usuario: presionar Enter, Contraseña: presionar Enter
- Usuario: Admin, Contraseña: Admin
- Usuario: Daniela, Contraseña: 2323

Los usuarios cliente de demostración (no cuentan con el nivel de acceso a esta herramienta) existentes son:

- Usuario: User, Contraseña: User
- Usuario: Lorenzo, Contraseña: 578

Este código demuestra el uso de dos variables de entrada con usuario y contraseña, y después variables bandera para conceder acceso y también detectar si existe el usuario pero no coincide la contraseña. Cada condicional dependerá de qué condición se satisface.

```
user = input('Teclee su usuario: ')
contra = input('Teclee su contraseña: ')

usr_exist = 0
log = 0
acceso_admin = 0
while acceso_admin == 0:
    for admin in usuarios_admin:
        if admin[0] == user:
            usr_exist = 1
        if admin[0] == user and admin[1] == contra:
            print(f'Validación exitosa \nNivel de acceso: Administrador\
\nBienvenido {admin[0]} al sitio.')
            log = 1
            acceso_admin = 1

    for cliente in usuarios_cliente:
        if cliente[0] == user:
```

```

    usr_exist = 1
    if cliente[0] == user and cliente[1] == contra:
        print(f'Validación exitosa \nNivel de acceso: Cliente \n{cliente[0]},\nno tiene acceso al sitio.\nIntente con una cuenta válida')
        log = 1
        acceso_admin = 0
        user = input('Teclee su usuario: ')
        contra = input('Teclee su contraseña: ')

    if log == 0 and usr_exist == 1:
        print('Usuario existe, contraseña no coincide. \nIntente de nuevo')
        user = input('Teclee su usuario: ')
        contra = input('Teclee su contraseña: ')

    elif log == 0 and usr_exist == 0:
        print('Credenciales no coinciden, ingrese credenciales correctas')
        user = input('Teclee su usuario: ')
        contra = input('Teclee su contraseña: ')

```

Ahora bien, un segundo menú similar se genera, pero ahora para poder seleccionar el reporte al que se quiere acceder. Se vuelve a emplear el uso de variables bandera para poder permitir el acceso a alguno de los loops internos o para seguir pidiendo por una opción válida, o bien, dar por terminada la sesión y terminar con la ejecución del programa

```
opcion_valida = 0
```

```

while opcion_valida != 1:
    print('Haga una selección de la clase de reporte disponible al que desee acceder: \
\n1.- Productos más vendidos y productos rezagados. \
\n2.- Productos por reseña en el servicio. \
\n3.- Ingresos ventas \
\n4.- Finalizar sesión.\n')

    opcion_seleccionada = input('Teclee su selección: ')

    #Mas vendidos y rezagados
    if opcion_seleccionada == '1':
        print('Seleccionó los informes de productos más vendidos y productos\
rezagados. \nHaga una selección del reporte que desea consultar: \
\n1.- 50 productos con mayores ventas.\
\n2.- 50 productos con mayores búsquedas.\
\n3.- 10 productos con menores ventas por categoría. \
\n4.- 10 productos con menores búsquedas por categoría. \
\n5.- Regresar al menú anterior')
        opcion_valida = 1

        subopcion1_seleccionada = input('Teclee su selección: ')
        subopcion1_valida = 0

        while subopcion1_valida != 1:
            #lista 50 productos con mayores ventas
            if subopcion1_seleccionada == '1':

```

Ahora bien, se emplearon 3 rutinas básicas para el desarrollo de la herramienta, aquí se muestran los bucles for muy empleados. La premisa básica es el poder iterar sobre los productos y sobre alguna otra de las listas existentes para poder hacer un registro por producto ya sea del agregado de ventas o de búsquedas.

```

ventas_totales = 0
lista_ventas_ind = [] # [[id,ventas]]

```

```

for producto in lifestore_products:
    ventas_individual = 0
    for venta in lifestore_sales:
        if producto[0] == venta[1]:
            ventas_individual += 1
            ventas_totales += 1
    registro_individual = [producto[0], producto[1], ventas_individual]
    lista_ventas_ind.append(registro_individual)

lista_ventas_max = list(lista_ventas_ind)
ventas_ordenadas_max = []

while lista_ventas_max:
    maximo = lista_ventas_max[0][2]
    registro_actual = lista_ventas_max[0]
    for registro in lista_ventas_max:
        if registro[2] > maximo:
            maximo = registro[2]
            registro_actual = registro
    ventas_ordenadas_max.append(registro_actual)
    lista_ventas_max.remove(registro_actual)

for i in range(50):
    print(f'El producto número {i+1} en ventas: ', \
          ventas_ordenadas_max[i][1], 'se ha vendido en total\
          ', ventas_ordenadas_max[i][2], 'veces')
opcion_valida = 0
break
subopcion1_valida = 1

```

DESARROLLO DE LAS CONSIGNAS - SOLUCIÓN AL PROBLEMA

A través del uso de la herramienta desarrollada, se puede obtener el resultado de las consignas desarrolladas. Donde de las tres consignas se determina que:

- Entre los discos duros, procesadores y tarjetas madre están las mayores ventas, por lo que el sector específico del mercado es evidente que puede estar concentrado en el armado de computadoras de alto desempeño.
- Los procesadores y tarjetas de video son muy bien reseñados, por lo que deben impulsarse, sin embargo las tarjetas madre han resultado problemáticas para los clientes.
- Los meses cercanos al fin del año no registran muchas ventas y entre el primer y segundo trimestres del año se concentran las ventas generadas en su mayoría.

Como estrategia sugerida para la tienda se puede concluir que existe una necesidad por hacer una mejor revisión del método de cálculo del inventario necesario, debido a que no todos los productos se pueden vender de manera similar, ni tampoco la demanda es homogénea a lo largo del tiempo.

En general, un análisis de clasificación A,B,C puede ser benéfico, así, calcular no sólo la contribución individual de cada producto al ingreso total, sino también su varianza mensual, con lo que puede ser posible determinar la volatilidad de los productos para poder enfocarse en productos que tengan demanda constante y poder mantener un inventario justo del producto. Por el contrario, de los productos con alta volatilidad y con bajo aporte al ingreso total, evaluar la continuidad del producto, y definir si es benéfico seguir ofertándolo o sustituir su inventario por alguno otro que tenga una demanda más significativa.

DISCUSIÓN - CONCLUSIONES

El desarrollo de la presente herramienta, así como de los resultados que de ella surgen me permitieron el desarrollo de habilidades de programación fundamentales, así como de la realización de análisis descriptivos básicos sin el uso de paqueterías de software avanzadas y como una muestra de la capacidad de poder llevar a cabo analítica básica. Con este conocimiento, el uso de paqueterías y funcionalidades más avanzadas del lenguaje de programación Python se ven potenciadas ante su combinación con conceptos fundamentales de la programación.

Es claro que una visualización avanzada, así como la eficiencia de la compilación del código no fueron los objetivos de este proyecto, sino la demostración de las capacidades de desarrollo a partir de funciones mínimas y empleando condicionales.