

**Proyecto final Emtech:**  
**Creación de código para análisis de**  
**inventario de “Life Store”.**

**Juan Omar Reyes Sánchez**

**A: 6 de septiembre de 2020**

## Índice

Índice .....	2
Introducción .....	2
Definición de código.....	3
Log in .....	3
Ventas por producto .....	4
Búsquedas por producto.....	5
Ingresos por mes.....	5
Solución .....	5
Conclusión .....	6
Bibliografía .....	6
Código .....	6

## Introducción

El concepto de pensamiento computacional fue planteado por primera vez en 2006, y para 2012 Tilling y Fadel mencionaban que la resolución de problemas es la nueva base de aprendizaje para el S. XXI. La implementación del pensamiento lógico en la resolución de problemas, es la base del pensamiento computacional, base fundamental de los lenguajes computacionales (1).

EL lenguaje computacional Python, se crea en los 90's por Guido van Rossum, tomando como inspiración para el nombre al grupo de cómicos ingleses "Monty Python". Siendo un programa similar a uno pre existente, Pearl, con la diferencia de presentar una sintaxis más amplia y legible. El dinamismo que posee, junto con la sencillez de su sintaxis, convierte a Python en una de las mejores alternativas para que cualquiera se incorpore al mundo de la programación. No obstante, Python es utilizado por grandes compañías y organizaciones como Google, Yahoo!, NASA, y el SO Linux (2).

Es por esto que la implementación de Python en pequeñas y medianas empresas, mas que una buena alternativa, representa la mejor y más viable.

## Definición de código

### Log in

```
127
128     "#Inicio de programa#"
129     Usuarios= [{"Pepe123", "3detripacontodo"},
130     ["Juan456", "500demaiz"],
131     ["Linda789", "2cubanas"],
132     ["Maria910", "a10labolsita"],
133     ["x", "i"]
134     ]
135     Inicio= print("Bienvenido al inventario de 'Life
Store'.")
136     Usuario_entrada= input("Ingresa tu nombre de usuario: ")
137
138     Contraseña_entrada= input("Ingresa contraseña: ")
139
140     counter=0
141     for Usuario in Usuarios :
142         if Usuario [0] == Usuario_entrada and Usuario [1]==
Contraseña_entrada :
143             print ("Bienvenido "+ Usuario_entrada )
144             counter=1
145     if counter==1:
146         print("\n1. Productos más vendidos en la tienda\n",
"\n2. Productos más buscados en la tienda\n", "\n3.
Productos mejor calificados en la tienda\n", "\n4.
Productos menos vendidos en la tienda\n", "\n5.
Productos menos buscados en la tienda\n" )
147         inicio= int(input("Escoge la acción que deseas
realizar: "))
148         v=1
149     +
```

La primera parte de este código es la lista de usuarios administradores preestablecida, dividida en usuario y contraseña ([Usuario, contraseña]), ambos elementos en formato de cadena (str). Lo siguiente es lo primero que el usuario administrador verá al correr el código, donde se le pide que ingrese su nombre de usuario y contraseña. Al ingresar los datos correctos se mostrará un menú donde podrá escoger entre 4 tipos de listas: productos mas vendidos, mas buscados, menos buscados, menos vendidos y mejor calificados.

## Ventas por producto

```
152 while inicio:
153     if inicio == 1:
154         for Na in Ho:
155             #if Na[3] >= 5:
156             #print("\nEste producto:\n",Na[1], "\nSe ha
vendido:",Na[3],"veces." ) #Esta linea imprime solo
los productos con mas de 5 ventas en la base.
157             if Na[4] > 1: #Mas vendido
158                 print("\nEste producto:\n",Na[1], "\nSe ha
vendido:",Na[4],"veces." )
159                 counter+=1
160             break
```

Cuando el usuario seleccione la primera opción, podrá observar la lista de los productos mas vendidos en la tienda. En esta parte del código, la función esta resumida para mejor funcionamiento. Mas arriba se realizan las operaciones que permitieron hacer este listado y se hizo de la siguiente forma:

```
23
24 Ho = []
25
26 while totales_events:
27     acomodo = totales_events[0][4]
28     matt = totales_events[0]
29     for He in totales_events:
30         if He[4] > acomodo:
31             acomodo = He[4]
32             matt = He
33     Ho.append(matt)
34     totales_events.remove(matt)
35 #print(Ho)
36
37 print("##### Productos mas vendidos
#####")
38 d=1
39 ##### Primera consigna #####
40 #Productos mas vendidos
41 for Na in Ho:
42     if Na[3] >= 5:
43         fantasma=1
44         #print("\nEste producto:\n",Na[1], "\nSe ha vendido:",
Na[3],"veces." ) #Esta linea imprime solo los
productos con mas de 5 ventas en la base.
45         if Na[4] > 1:
46             fantasma =1
47             #print("\nEste producto:\n",Na[1], "\nSe ha
vendido:",Na[4],"veces." )
48             #print("\nEste producto:\n",Na[1], "\nSe ha
vendido:",Na[4],"veces." )
```

Es en esta parte del código donde la operación se hace: En la variable Ho=[] se colocaran todos los datos que surjan de las siguientes operaciones, total\_events (te) es una matriz que contiene los datos de las listas Lifestore\_sales y Lifesotre\_products. Bucle while lo que hará es recorrer la matriz hasta que se le indique lo contrario. Se crean dos variables nuevas con los elementos que de "te" que deseamos comparar, en este caso es acomodo posee nombre y numero de ventas y matt solo posee el nombre del producto. Posteriormente en la sentencia for se crea una nueva variable que recorrerá la matriz de ts y dentro de esta indicación compararemos mediante la sentencia if, lo que se comparara será el numero de ventas de He (donde solo existe ese valor) contra el número de ventas de ts y se irán acomodando y eliminando de la lista, por lo que se deben guardar en otro sitio que es Ho.

El mismo procedimiento ocurre para realizar la búsqueda de los productos menos vendidos.

## Búsquedas por producto

Al igual que en la opción de mas ventas, al seleccionar cualquiera de las opciones, el usuario solo vera el resultado de las operaciones que llevaron a tal resultado.

```
if inicio ==2:#Mas buscado
    for Na_1 in Ho_1:
        if Na_1[3] >=20:
            print("\nEste producto:\n",Na_1[1], "\nSe ha
buscado:",Na_1[3],"veces." )
            counter+=0
        break
```

En este proceso se sigue exactamente el mismo procedimiento que se mencionó anteriormente, creando una matriz en blanco, empleando los datos de otra matriz, en este caso `totales_search` (ts).

```
99 .....
100 Ho_1 = [] #Lista donde estarán los valores ordenados
101
102 while totales_search:
103     bus = totales_search[0][3]
104     mat = totales_search[0]
105     for He_1 in totales_search:
106         if He_1[3] > bus:
107             bus = He_1[3]
108             mat = He_1
109     Ho_1.append(mat)
110     totales_search.remove(mat)
111     #print(He_1)
112
113 #Productos mas buscados.
114
115 for Na_1 in Ho_1:
116     if Na_1[3] >=20:
117         fantasma=2
118         #print("\nEste producto:\n",Na_1[1], "\nSe ha
buscado:",Na_1[3],"veces." )
```

## Ingresos por mes.

## Solución

Las ventas de la tienda son bajas, incluso las búsquedas de los productos tienen un marcado sesgo por solo algunos de los que conforman el inventario total de negocio. Considero que esto puede presentarse por varios factores: 1. El inventario solo contempla 2020, considerando la contingencia generada por la pandemia de SARS-CoV-2 (COVID-19), es muy probable que, al no ser un establecimiento de primera necesidad, las ventas y búsquedas bajaran considerablemente en comparación a otros años. Por lo tanto, la acumulación de inventario es inevitable. 2. Si tomar en cuenta la pandemia 2020, la posibilidad de que el establecimiento sea nuevo y carezca de un estudio de mercado para su zona es elevado debido a que las compras y búsquedas presentan el sesgo a unos elementos elegidos por consumidores comunes de productos informáticos.

Propongo un estudio de mercado que permita conocer las necesidades de los consumidores de la zona, para enfrentar el rezago de mercancías, producto de la contingencia, es necesario implementar una estrategia de ventas por internet. Dada la precariedad de la

situación, el uso de las redes sociales puede servir en primera instancia en lo que se establece una página web.

## Conclusión

El ejercicio resulta en una muy buena practica para poner a prueba lo aprendido durante las sesiones disponibles en la plataforma. A lo largo del curso, consideró que los materiales son muy buenos y nuestro tuto Javier, es muy paciente y dedicado. No obstante, se cae en el mismo problema que se presentan en las escuelas públicas. Muchos alumnos para un solo profesor y tiempo limitado.

La combinación de estos factores resultan en un aprovechamiento menor de los elementos que se proporcionan, pues, aunque el ser autodidáctico es un fundamenta, llega un momento en el que es necesaria la guía y la practica mediante ejercicio que cumplan con la finalidad del curso, ya que al buscar ejercicios o prácticas en línea, los alumnos con mucho menos manejo de la programación nos encontramos con ejercicios que no siguen la misma lógica que se nos presento en las clases.

En resumen, me parece que la problemática radica principalmente en la dificultad de encontrar ejercicios o practicas que puedan seguir la misma lógica o finalidad del curso. Pues debido al numero de integrantes y tutores, junto con la disponibilidad de horarios, vuelve imposible solucionar la problemática de numero de alumnos por número de tutores.

## Bibliografía

1. Pérez Angulo Jesús Alberto. 2019. El pensamiento computacional en la vida cotidiana. Disponible en:  
[https://www.researchgate.net/publication/335149706\\_El\\_pensamiento\\_computacional\\_en\\_la\\_vida\\_cotidiana?enrichId=rgreq-7441c509ad812522278b8801e797e21e-XXX&enrichSource=Y292ZXJQYWdlOzMzNTE0OTcwNjtBUzo3OTE1NDcwNTQ0MDM1ODRAMTU2NTczMDkzMjEyMg%3D%3D&el=1\\_x\\_3&esc=publicationCoverPdf](https://www.researchgate.net/publication/335149706_El_pensamiento_computacional_en_la_vida_cotidiana?enrichId=rgreq-7441c509ad812522278b8801e797e21e-XXX&enrichSource=Y292ZXJQYWdlOzMzNTE0OTcwNjtBUzo3OTE1NDcwNTQ0MDM1ODRAMTU2NTczMDkzMjEyMg%3D%3D&el=1_x_3&esc=publicationCoverPdf)
2. González Duque Raúl. Consultado el 6 de septiembre de 2020. Disponible en:  
<http://mundogeek.net/tutorial-python/>

## Código

```
from lifestore_file import lifestore_products
from lifestore_file import lifestore_sales
from lifestore_file import lifestore_searches

#Número de veces que se ha vendido un producto.
contador = 0
totales_events = []
for product in lifestore_products:
    for venta in lifestore_sales:
        if product[0] == venta[1]:
```

```

        contador += 1

    #Matriz = [product[0], product[1], product[3], contador]
    Matriz = [product[0], product[1], product[3], venta[2], contador]
    totales_events.append(Matriz)

    contador = 0
#Numero de ventas por producto, sin orden.
for total in totales_events:
    #print("\n El producto: \n", total[1], "\n Se ha vendido: ", total[3],
        # " veces.")

Ho = []

while totales_events:
    acomodo = totales_events[0][4]
    matt = totales_events[0]
    for He in totales_events:
        if He[4] > acomodo:
            acomodo = He[4]
            matt = He
    Ho.append(matt)
    totales_events.remove(matt)
#print(Ho)

print("##### Productos mas vendidos #####")
d=1
##### Primera consigna #####
#Productos mas vendidos
for Na in Ho:
    if Na[3] >=5:
        fantasma=7
        #print("\nEste producto:\n",Na[1], "\nSe ha vendido:",Na[3],"veces." ) #Esta linea
        imprime solo los productos con mas de 5 ventas en la base.
    if Na[4] >1:
        fantasma =1
        #print("\nEste producto:\n",Na[1], "\nSe ha vendido:",Na[4],"veces." )
        #print("\nEste producto:\n",Na[1], "\nSe ha vendido:",Na[4],"veces." )
        #print (Na)
print("#####\n")

print("##### Productos rezagados #####")
print("#####\n")

```

```
#Productos menos vendidos"
```

```
for Ne in Ho:
```

```
    if Ne[4] <5:
```

```
        fantasma=5
```

```
        #print("\nEste producto:\n",Ne[1], "\nSe ha vendido:",Ne[4],"veces." )
```

```
print("##### Reseñas de productos #####")
```

```
print("#####\n")
```

```
print("##### Mejores Reseñas de productos #####")
```

```
print("#####\n")
```

```
for Ne in Ho:
```

```
    if Ne [3] >3:
```

```
        fantasma=3
```

```
        #print("\nEste producto:\n",Ne[1], "\nTiene una calificación de: ",Ne[3],"estrellas." )
```

```
print("#####\n")
```

```
print("##### Peores Reseñas de productos #####")
```

```
print("#####\n")
```

```
##### Segunda consigna #####
```

```
contador_1 = 0
```

```
totales_search = [] #[[id][cont],...n]
```

```
for product_1 in lifestore_products:
```

```
    for busq in lifestore_searches:
```

```
        if product_1[0] == busq[1]:
```

```
            contador += 1
```

```
        Matriz_1 = [product_1[0], product_1[1], product_1[3], contador]
```

```
        totales_search.append(Matriz_1)
```

```
        contador = 0
```

```
#print(totales_search) #hasta aqui, todo bien.
```

```
"""
```

```
print("##### Busquedas #####")
```

```
print("#####\n")
```

```
#Numero de busquedas por producto
```

```
#for total_1 in totales_search:
```

```
    #print("\n El producto: \n", total_1[1], "\n Se ha buscado: ", total_1[3],
```

```
        # " veces.") #Hasta aqui tambien bien.
```

```
#d=0
```

```
#d=4
```

```
#Productos ordenados por busquedas
```



```

print("##### Más buscados #####")
print("====")

Ho_1 = [] #Lista donde estarán los valores ordenados

while totales_search:
    bus = totales_search[0][3]
    mat = totales_search[0]
    for He_1 in totales_search:
        if He_1[3] > bus:
            bus = He_1[3]
            mat = He_1
    Ho_1.append(mat)
    totales_search.remove(mat)
#print(He_1)

#Productos mas buscados.

for Na_1 in Ho_1:
    if Na_1[3] >=20:
        fantasma=2
        #print("\nEste producto:\n",Na_1[1], "\nSe ha buscado:",Na_1[3],"veces." )
        print("#####\n")

print("##### Menos buscados #####")
for Ne_1 in Ho_1:
    if Ne_1[3] <20:
        fantasma=6
        print("\nEste producto:\n",Ne_1[1], "\nSe ha buscado:",Ne_1[3],"veces." )
        print("##### Menos buscados #####")

"#Inicio de programa#"
Usuarios= [
    ["Pepe123", "3detripacontodo"],
    ["Juan456", "500demaiz"],
    ["Linda789", "2cubanas"],
    ["María910", "a10labolsita"],
    ["x", "1"]
]

Inicio= print("Bienvenido al inventario de 'Life Store'.")
Usuario_entrada= input("Ingresa tu nombre de usuario: ")

Contraseña_entrada= input("Ingresa contraseña: ")

counter=0

```

```

for Usuarios in Usuarios :
    if Usuario [0] == Usuario_entrada and Usuario [1]== Contraseña_entrada :
        print ("Bienvenido "+ Usuario_entrada )
        counter=1
if counter==1:
    print("\n1. Productos más vendidos en la tienda\n", "\n2. Productos más buscados en la tienda\n", "\n3. Productos mejor calificados en la tienda\n","\n4. Productos menos vendidos en la tienda\n","\n5. Productos menos buscados en la tienda\n" )
    inicio= int(input("Escoge la acción que deseas realizar: "))
    v=1
else :
    print("Usuario incorrecto, iniciando protocolo 'SKYNET'...\n1. Iniciar Mod. T-800 \n2. Activar ojivas. \n3. Liberar Ergo Proxy" )
    x= input("Indique la acción que desea realizar:")
while inicio:
    if inicio ==1:
        for Na in Ho:
            #if Na[3] >=5:
            #print("\nEste producto:\n",Na[1], "\nSe ha vendido:",Na[3],"veces." ) #Esta linea imprime solo los productos con mas de 5 ventas en la base.
            if Na[4] >1:#Mas vendido
                print("\nEste producto:\n",Na[1], "\nSe ha vendido:",Na[4],"veces." )
                counter=0
            break
    if inicio ==2:#Mas buscado
        for Na_1 in Ho_1:
            if Na_1[3] >=20:
                print("\nEste producto:\n",Na_1[1], "\nSe ha buscado:",Na_1[3],"veces." )
                counter=0
            break
    if inicio ==3:
        for Ne in Ho:
            if Ne [3] >3:
                print("\nEste producto:\n",Ne[1], "\nTiene una calificación de: ",Ne[3],"estrellas." )
            break
    if inicio ==4:
        for Na in Ho:
            #if Na[3] >=5:
            #print("\nEste producto:\n",Na[1], "\nSe ha vendido:",Na[3],"veces." ) #Esta linea imprime solo los productos con mas de 5 ventas en la base.
            if Na[4] <1:#Mas vendido
                print("\nEste producto:\n",Na[1], "\nSe ha vendido:",Na[4],"veces." )
                #counter=0
            break

```

```
if inicio ==5:#Mas buscado
for Na_1 in Ho_1:
    if Na_1[3] <20:
        print("\nEste producto:\n",Na_1[1], "\nSe ha buscado:",Na_1[3],"veces." )
        #counter=0
    break
```