



7-9-2020

# Proyecto 1.

Introducción a Python

Carlos Jesús Bermúdez Bonilla

## ÍNDICE

<b>Introducción.....</b>	<b>II</b>
<b>Código .....</b>	<b>1</b>
<b>Solución al problema .....</b>	<b>10</b>
<b>Conclusión.....</b>	<b>15</b>

# Introducción

LifeStore es una tienda virtual que maneja una amplia gama de artículos, recientemente, la Gerencia de ventas, se percató que la empresa tiene una importante acumulación de inventario. Asimismo, se ha identificado una reducción en las búsquedas de un grupo importante de productos, lo que ha redundado en una disminución sustancial de sus ventas del último trimestre.

Se requiere hacer un análisis de los datos de rotación de productos, este análisis se hará con la ayuda de Python, se analizará los productos más vendidos, menos vendidos, más buscados, menos buscados y la reseña por parte del consumidor. A continuación, se explicará el código empelado para el análisis de LifeStore.

```

"""
Created on Tue Aug 25 14:03:03 2020

@author: Carlos Bermudez
"""
#Se importan las Listas que contienen los datos para nuestro análisis
from lifestore_file import lifestore_products
from lifestore_file import lifestore_searches
from lifestore_file import lifestore_sales

"""
Para empezar se recomienda iniciar sesión con alguno de los usuarios en lista admin_system,
así podrá registrar su usuario, dando la opción 1, luego pedirá datos como: nombre, apellido,
contraseña y si está autorizado o no, con un usuario autorizado podrá acceder al contenido de ventas.
"""

admin_system = [
    ["Carlos Bermudez", "Passw0rd25"],
    ["Sandra Palma", "Passw0rd20"]]

#Si el usuario se equivoca mas de 3 veces con su contraseña se eliminará el usuario
intentos = 0
while intentos <= 3 :
    usuario = input("Usuario: ")
    password = input("Contraseña: ")
    for admin in admin_system:
        if usuario == admin[0] and password == admin[1]:
            autorizacion = "maybe"
            intentos = 4
            #Se despliega las opciones que se pueden hacer como admin_system
            print ("\nHola", usuario.split()[0], ", es un gusto verte de nuevo.\n"
                  "Si deseas agregar un usuario escribe [1].\n"
                  "Si deseas eliminar un usuario escribe [2].\n"
                  "Si deseas ver los usuarios escribe [3].\n"
                  "Si deseas salir escribe [4]\n"
                  "Recuerda que sólo debes de escribir el número dentro de [].")
            respuesta = input("Opción: ")
            count = 0
            while count < 1:
                """
                Con la opción 1 se pueden agregar los usuarios que deseen, la opción 2 sirve para eliminar usuarios,
                la opción 3 imprime los usuarios que están registrados, y la 4 es para salir del sistema.
                Se usan archivos de texto para guardar los datos, se usan las funciones de open(), close(), readlines(),
                write() para la manipulación de los archivos de texto
                """
                if "1" in respuesta:
                    rectificacion = input("¿Desea agregar un usuario? Sí[1]/ No[0]: ")
                    if "1" in rectificacion:
                        repetir = 0
                        while repetir < 1:
                            archivo = open("usuarios.txt", "a")
                            nombre = input("Nombre: ")
                            apellido = input("Apellido: ")
                            contraseña = input("Contraseña: ")

```

```

acceso = input("¿Tiene acceso a la información? Sí[1]/ No[0]: ")
if acceso == "1":
    acceso = "autorizado"
else:
    acceso = "no autorizado"
#Con ayuda de format() Le damos orden para guardar de mejor manera los datos
archivo.write("\n{ }{},{},{ }".format(nombre, apellido, contraseña, acceso))
archivo.close()
repetir_accion = input("¿Desea agregar a alguien más? Sí[1]/ No[0]: ")
if repetir_accion != "1":
    repetir = 1
    nueva_accion = input("Requiere hacer otra acción. Sí[1]/ No[0]: ")
    if nueva_accion == "1":
        respuesta = input("Volver a elegir opción: ")
    else:
        print("Hasta la vista Baby")
        count = 1
else:
    respuesta = input("Volver a elegir opción: ")
elif "2" in respuesta:
    rectificacion = input("¿Desea eliminar un usuario? Sí[1]/ No[0]: ")
    if "1" in rectificacion:
        repetir = 0
        while repetir < 1:
            usuario_eliminado = input("Usuario a eliminar: ")
            archivo = open("usuarios.txt", "r")
            lines = archivo.readlines()
            archivo.close()
            nuevo_archivo = open("usuarios.txt", "w")
            for line in lines:
                if not usuario_eliminado in line:
                    nuevo_archivo.write(line)
            nuevo_archivo.close()
            repetir_accion = input("¿Desea eliminar a alguien más? Sí[1]/ No[0]: ")
            if repetir_accion != "1":
                repetir = 1
                nueva_accion = input("Requiere hacer otra acción. Sí[1]/ No[0]: ")
                if nueva_accion == "1":
                    respuesta = input("Volver a elegir opción: ")
                else:
                    print("MAY THE FORCE BE WITH YOU")
                    count = 1
            else:
                respuesta = input("Volver a elegir opción: ")
elif "3" in respuesta:
    rectificacion = input("¿Desea ver los usuarios? Sí[1]/ No[0]: ")
    if "1" in rectificacion:
        files = open("usuarios.txt", "r")
        for linea in files:
            lista = linea.strip().split(",")
            print(lista[0])

```

```

files.close()
nueva_accion = input("Requiere hacer otra acción. Sí[1]/ No[0]: ")
if nueva_accion == "1":
    respuesta = input("Volver a elegir opción: ")
else:
    print("Hasta el infinito y mas allá")
    count = 1
else:
    respuesta = input("Volver a elegir opción: ")
elif "4" in respuesta:
    print()
    rectificacion = input("¿Desea salir? Sí[1]/ No[0]: ")
    if "1" in rectificacion:
        print("No me quiero ir señor Stark")
        break
    else:
        respuesta = input("Volver a elegir opción: ")
#Los administradores no tienen limite de fallos en la contraseña
elif usuario == admin[0] and password != admin[1]:
    print("Contraseña incorrecta intente de nuevo.")
#En esta parte se analiza la entrada del usuario por si no es un administrador
archivo = open("usuarios.txt", "r")
for linea in archivo.readlines():
    if usuario in linea:
        #Se usa strip() para quitar los saltos de linea a la hora de recuperar los datos
        #Se usa split() para crear una lista con los datos obtenidos del archivo usuarios.txt
        if usuario == linea.strip().split(",")[0] and password == linea.strip().split(",")[1]:
            intentos = 4
            if linea.strip().split(",")[2] == "autorizado":
                autorizacion = "sí"
            else:
                autorizacion = "no"
        elif intentos == 2:
            print("Contraseña incorrecta, último intento.")
            intentos += 1
        elif intentos == 3: #Cuando se llega a 3 intentos fallidos se elimina el usuario
            print("Por seguridad su usuario será eliminado")
            archivo = open("usuarios.txt", "r")
            lines = archivo.readlines()
            archivo.close()
            nuevo_archivo = open("usuarios.txt", "w")
            for line in lines:
                if not usuario in line:
                    nuevo_archivo.write(line)
            nuevo_archivo.close()
            intentos += 1
        else:
            intentos += 1
            print("Contraseña incorrecta intente de nuevo")
    archivo.close()
total_products = len(lifestore_products) #número de productos que se venden

```

```

#Se crean Listas para in contando Las busquedas y puntuación.
count_searches = []
count_score = []
for n in range(total_products):
    count_searches.append([n+1,0,lifestore_products[n][1]])
    count_score.append([n+1,0,0,lifestore_products[n][1]])
    for search in lifestore_searches:
        if search[1] == count_searches[n][0]:
            count_searches[n][1] += 1
    for search in lifestore_sales:
        if search[1] == count_score[n][0]:
            count_score[n][1] += search[2]
            count_score[n][2] += 1
#Se seapan Los productos de Los que sí tienen busquedas de Los que no.
without_searches = []
with_searches = []
with_searches_copy = []
for count in count_searches:
    if count[1] != 0:
        with_searches.append(count)
        with_searches_copy.append(count)
    else:
        without_searches.append(count)
searches_ordenado_menos = []
searches_ordenado_mas = []
#Se ordenan Los valores de menor busqueda a mayor busqueda
while with_searches:
    minimo = with_searches[0][1]
    searches_actual = with_searches[0]
    for search in with_searches:
        if search[1] < minimo:
            minimo = search[1]
            searches_actual = search
    searches_ordenado_menos.append(searches_actual)
    with_searches.remove(searches_actual)
#Se ordenan Los valores de mayor busqueda a menor
while with_searches_copy:
    maximo = with_searches_copy[0][1]
    searches_actual = with_searches_copy[0]
    for search in with_searches_copy:
        if search[1] > maximo:
            maximo = search[1]
            searches_actual = search
    searches_ordenado_mas.append(searches_actual)
    with_searches_copy.remove(searches_actual)
vendidos = []
vendidos_copy = []
promedio = []
promedio_copy = []
sin_registro = []
for score in count_score:

```

```

if score[2] != 0:
    vendidos.append([score[0],score[2],score[3]])
    vendidos_copy.append([score[0],score[2],score[3]])
    promedio.append([score[0], round(score[1]/score[2],2),score[3]])
    promedio_copy.append([score[0], round(score[1]/score[2],2),score[3]])
else:
    sin_registro.append([score[0], "Sin registro de venta"])
#Se ordenan productos mas a vendidos, puntuacionde mayor a menor y de menor a mayor
vendidos_mas = []
vendidos_menos = []
promedio_mas = []
promedio_menos = []
while vendidos:
    minimo = vendidos[0][1]
    ventas_actual = vendidos[0]
    for ventas in vendidos:
        if ventas[1] < minimo:
            minimo = ventas[1]
            ventas_actual = ventas
    vendidos_menos.append(ventas_actual)
    vendidos.remove(ventas_actual)
while promedio_copy:
    maximo =promedio_copy[0][1]
    promedio_actual = promedio_copy[0]
    for puntuacion in promedio_copy:
        if puntuacion[1] > maximo:
            maximo = puntuacion[1]
            promedio_actual = puntuacion
    promedio_mas.append(promedio_actual)
    promedio_copy.remove(promedio_actual)
while promedio:
    minimo = promedio[0][1]
    promedio_actual = promedio[0]
    for puntuacion in promedio:
        if puntuacion[1] < minimo:
            minimo = puntuacion[1]
            promedio_actual = puntuacion
    promedio_menos.append(promedio_actual)
    promedio.remove(promedio_actual)
while vendidos_copy:
    maximo = vendidos_copy[0][1]
    vendidos_actual = vendidos_copy[0]
    for ventas in vendidos_copy:
        if ventas[1] > maximo:
            maximo = ventas[1]
            vendidos_actual = ventas
    vendidos_mas.append(vendidos_actual)
    vendidos_copy.remove(vendidos_actual)
#Se crean listas para el calculo de ventas por mes y anual, tambien se ordenan de mayor a menor
by_date = []
for sale in lifestore_sales:

```



```

    by_date.append([lifestore_products[sale[1]-1][2],sale[3][3:],sale[4]])
month_list = []
year_list= []
for month in by_date:
    if not month[1] in month_list:
        month_list.append(month[1])
    if not month[1][-4:] in year_list:
        year_list.append(month[1][-4:])
contador_ventas_mensual = []
for month in month_list:
    contador_ventas_mensual.append([month, 0])
for sale in by_date:
    for month in contador_ventas_mensual:
        if sale[1] == month[0]:
            month[1] += sale[0]
contador_ventas_anual = []
for year in year_list:
    contador_ventas_anual.append([year,0,0])
for sale in contador_ventas_mensual:
    for year in contador_ventas_anual:
        if sale[0][-4:] == year[0]:
            year[1] += sale[1]
            year[2] += 1
meses_mayor_ventas = []
while contador_ventas_mensual:
    maximo = contador_ventas_mensual[0][1]
    vendidos_actual = contador_ventas_mensual[0]
    for ventas in contador_ventas_mensual:
        if ventas[1] > maximo:
            maximo = ventas[1]
            vendidos_actual = ventas
    meses_mayor_ventas.append(vendidos_actual)
    contador_ventas_mensual.remove(vendidos_actual)
meses = ["Enero", "Febrero","Marzo","Abril","Mayo","Junio",
        "Julio", "Agosto","Septiembre","Octubre","Noviembre", "Diciembre"]
"""
Si el usuario está autorizado podrá acceder a datos de productos más vendidos, menos vendidos,
mas buscados, menos buscados, sin buscar, sin ventas, con las mejores reseñas o las peores,
tambien se puede acceder a la venta total anual, o mensuales.
Se puede escoger la cantidad de productos a desplegar, pueden ser 1 hasta los que quiera el usuario,
si el usuario quiere mas de los que hay disponibles, no hay problema el sistema solo desplegara los que hay
"""
if autorizacion == "sí":
    print("Acceso autorizado")
    print ("Hola", usuario, ", es un gusto verte de nuevo.\n")
    "Si deseas ver top productos más vendidos escribe [1].\n"
    "Si deseas ver top productos menos vendidos escribe [2].\n"
    "Si deseas ver top productos más buscados escribe [3].\n"
    "Si deseas ver top productos menos buscados escribe [4]\n"
    "Si deseas ver top productos sin busquedas [5]\n"
    "Si deseas ver top productos mejores reseñas escribe [6]\n"

```

```

"Si deseas ver top productos peores reseñas escribe [7]\n"
"Si deseas ver los productos sin ventas escribe [8]\n"
"Si deseas ver total ventas escribe [9]\n"
"Recuerda que sólo debes de escribir el número dentro de []."
opcion = input("Elegir opción: ")
continuar = 0

while continuar < 1:
    #Top de mas vendidos
    if "1" in opcion:
        print("Máximo de datos a ver {}".format(str(len(vendidos_mas))))
        top = int(input("Número de datos a ver: "))
        if top > len(vendidos_mas):
            top = len(vendidos_mas)
        print("{} más vendidos:".format(top))
        for n in range(top):
            print("{:>2s}. {:>3s} piezas se vendieron de {}".format(str(n+1),str(vendidos_mas[n][1]),vendidos_mas[n][2].split(",")[0]))
        eleccion = input("¿Desea ver otro dato? Sí[1]/ No[0]: ")
        if eleccion == "1":
            opcion = input("Elegir opción: ")
        else:
            continuar = 1
    #Top de menos vendidos
    elif "2" in opcion:
        print("Máximo de datos a ver {}".format(str(len(vendidos_menos))))
        top = int(input("Número de datos a ver: "))
        if top > len(vendidos_menos):
            top = len(vendidos_menos)
        print("{} menos vendidos:".format(top))
        for n in range(top):
            print("{:>2s}. {:>3s} piezas se vendieron de {}".format(str(n+1),str(vendidos_menos[n][1]),vendidos_menos[n][2].split(",")[0]))
        eleccion = input("¿Desea ver otro dato? Sí[1]/ No[0]: ")
        if eleccion == "1":
            opcion = input("Elegir opción: ")
        else:
            continuar = 1
    #top de mas buscados
    elif "3" in opcion:
        print("Máximo de datos a ver {}".format(str(len(searches_ordenado_mas))))
        top = int(input("Número de datos a ver: "))
        if top > len(searches_ordenado_mas):
            top = len(searches_ordenado_mas)
        print("{} más buscados:".format(top))
        for n in range(top):
            print("{:>2s}. {:>3s} busquedas de {}".format(str(n+1),str(searches_ordenado_mas[n][1]),searches_ordenado_mas[n][2].split(",")[0]))
        eleccion = input("¿Desea ver otro dato? Sí[1]/ No[0]: ")
        if eleccion == "1":
            opcion = input("Elegir opción: ")
        else:
            continuar = 1
    #top de menos buscados

```

```

elif "4" in opcion:
    print("Máximo de datos a ver {}".format(str(len(searches_ordenado_menos))))
    top = int(input("Número de datos a ver: "))
    if top > len(searches_ordenado_menos):
        top = len(searches_ordenado_menos)
    print("{} menos buscados:".format(top))
    for n in range(top):
        print("{:>2s}. {:>3s} búsquedas de {}".format(str(n+1),str(searches_ordenado_menos[n][1]),searches_ordenado_menos[n][2].split(",")[0]))
    eleccion = input("¿Desea ver otro dato? Sí[1]/ No[0]: ")
    if eleccion == "1":
        opcion = input("Elegir opción: ")
    else:
        continuar = 1
#Lista de productos sin búsquedas
elif "5" in opcion:
    print("Productos sin búsquedas: ")
    indice = 0
    for product in lifestore_products:
        for busqueda in without_searches:
            if busqueda[0] == product[0]:
                indice += 1
                print("{:>2s}. {}".format(str(indice), product[1].split(",")[0]))
    eleccion = input("¿Desea ver otro dato? Sí[1]/ No[0]: ")
    if eleccion == "1":
        opcion = input("Elegir opción: ")
    else:
        continuar = 1
#Mejores puntuados
elif "6" in opcion:
    print("Máximo de datos a ver {}".format(str(len(promedio_mas))))
    top = int(input("Número de datos a ver: "))
    if top > len(promedio_mas):
        top = len(promedio_mas)
    print("{} mejores puntuados:".format(top))
    for n in range(top):
        print("{:>2s}. Calificación {:>2f}/5 de {}".format(str(n+1),str(promedio_mas[n][1]),promedio_mas[n][2].split(",")[0]))
    eleccion = input("¿Desea ver otro dato? Sí[1]/ No[0]: ")
    if eleccion == "1":
        opcion = input("Elegir opción: ")
    else:
        continuar = 1
#Peores puntuados
elif "7" in opcion:
    print("Máximo de datos a ver {}".format(str(len(promedio_menos))))
    top = int(input("Número de datos a ver: "))
    if top > len(promedio_menos):
        top = len(promedio_menos)
    print("{} peores puntuados:".format(top))
    for n in range(top):
        print("{:>2s}. Calificación {:>2f}/5 de {}".format(str(n+1),str(promedio_menos[n][1]),promedio_menos[n][2].split(",")[0]))
    eleccion = input("¿Desea ver otro dato? Sí[1]/ No[0]: ")

```

```

if eleccion == "1":
    opcion = input("Elegir opción: ")
else:
    continuar = 1
#Lista de productos sin ventas
elif "8" in opcion:
    print("Productos sin ventas: ")
    indice = 0
    for product in lifestore_products:
        for registro in sin_registro:
            if registro[0] == product[0]:
                indice += 1
                print("{:>2s}. {}".format(str(indice), product[1].split(",")[0]))
    eleccion = input("¿Desea ver otro dato? Sí[1]/ No[0]: ")
    if eleccion == "1":
        opcion = input("Elegir opción: ")
    else:
        continuar = 1
#Datos de ventas, anuales y mensuales
elif "9" in opcion:
    print("Ventas anuales")
    for year in contador_ventas_anual:
        print("{:}: $ {},{}".format(year[0],str(year[1])[:-3],str(year[1])[-3:]))
    print("Promedio mensual")
    for year in contador_ventas_anual:
        promedio_mensual = str(round(year[1]/year[2],2))
        print("{:}: $ {}".format(year[0],promedio_mensual))
    print("Ventas por mes")
    indice = 0
    for mes in meses_mayor_ventas:
        indice += 1
        a = int(mes[0][:2])-1
        print("{:>2s}. {:>10s} {:}: ${},{:.00}".format(str(indice),meses[a],mes[0][-4:],str(mes[1])[:-3],str(mes[1])[-3:]))
    eleccion = input("¿Desea ver otro dato? Sí[1]/ No[0]: ")
    if eleccion == "1":
        opcion = input("Elegir opción: ")
    else:
        continuar = 1
elif autorizacion == "maybe": #Esto es para evitar que salga el mensaje de no autorizado al admin_system
    print("")
else:
    print("Acceso no autorizado, adiós. >.<")#si Los usuarios no estan autorizados sale este mensaje

```

# Solución al problema

20 más vendidos:

1. 50 piezas se vendieron de SSD Kingston A400
2. 42 piezas se vendieron de Procesador AMD Ryzen 5 2600
3. 20 piezas se vendieron de Procesador Intel Core i3-9100F
4. 18 piezas se vendieron de Tarjeta Madre ASRock Micro ATX B450M Steel Legend
5. 15 piezas se vendieron de SSD Adata Ultimate SU800
6. 14 piezas se vendieron de Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING
7. 13 piezas se vendieron de Procesador AMD Ryzen 5 3600
8. 13 piezas se vendieron de Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8
9. 11 piezas se vendieron de SSD XPG SX8200 Pro
10. 9 piezas se vendieron de Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC
11. 9 piezas se vendieron de SSD Kingston A2000 NVMe
12. 7 piezas se vendieron de Procesador Intel Core i7-9700K
13. 6 piezas se vendieron de Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0)
14. 6 piezas se vendieron de Tarjeta Madre MSI ATX B450 TOMAHAWK MAX
15. 5 piezas se vendieron de Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030
16. 4 piezas se vendieron de Procesador Intel Core i5-9600K
17. 3 piezas se vendieron de Procesador Intel Core i9-9900K
18. 3 piezas se vendieron de Tarjeta de Video ASUS AMD Radeon RX 570
19. 3 piezas se vendieron de Kit SSD Kingston KC600
20. 3 piezas se vendieron de SSD Kingston UV500

20 menos vendidos:

1. 1 piezas se vendieron de MSI GeForce 210
2. 1 piezas se vendieron de Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix
3. 1 piezas se vendieron de Tarjeta de Video Gigabyte AMD Radeon R7 370 OC
4. 1 piezas se vendieron de Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC
5. 1 piezas se vendieron de Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti
6. 1 piezas se vendieron de Tarjeta Madre Gigabyte XL-ATX TRX40 Designare
7. 1 piezas se vendieron de Tarjeta Madre ASRock ATX H110 Pro BTC+
8. 1 piezas se vendieron de Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2
9. 1 piezas se vendieron de SSD Crucial MX500
10. 1 piezas se vendieron de Kit Memoria RAM Corsair Dominator Platinum DDR4
11. 1 piezas se vendieron de TCL Smart TV LED 55S425 54.6
12. 1 piezas se vendieron de TV Monitor LED 24TL520S-PU 24
13. 1 piezas se vendieron de Logitech Audífonos Gamer G332
14. 1 piezas se vendieron de Cougar Audífonos Gamer Phontum Essential
15. 1 piezas se vendieron de HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro
16. 2 piezas se vendieron de Procesador AMD Ryzen 3 3300X S-AM4
17. 2 piezas se vendieron de Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC
18. 2 piezas se vendieron de Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming
19. 2 piezas se vendieron de Tarjeta Madre ASUS ATX PRIME Z390-A
20. 2 piezas se vendieron de SSD Western Digital WD Blue 3D NAND

20 más buscados:

1. 263 búsquedas de SSD Kingston A400
2. 107 búsquedas de SSD Adata Ultimate SU800
3. 60 búsquedas de Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING
4. 55 búsquedas de Procesador AMD Ryzen 5 2600
5. 41 búsquedas de Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8
6. 35 búsquedas de Logitech Audífonos Gamer G635 7.1
7. 32 búsquedas de TV Monitor LED 24TL520S-PU 24
8. 31 búsquedas de Procesador Intel Core i7-9700K
9. 30 búsquedas de Procesador Intel Core i3-9100F
10. 30 búsquedas de SSD XPG SX8200 Pro
11. 27 búsquedas de SSD Kingston A2000 NVMe
12. 25 búsquedas de Tarjeta Madre MSI ATX B450 TOMAHAWK MAX
13. 24 búsquedas de Procesador AMD Ryzen 5 3600
14. 23 búsquedas de Tarjeta Madre ASRock Micro ATX B450M Steel Legend
15. 20 búsquedas de Procesador Intel Core i5-9600K
16. 15 búsquedas de Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC
17. 15 búsquedas de Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC
18. 15 búsquedas de TCL Smart TV LED 55S425 54.6
19. 11 búsquedas de Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030
20. 11 búsquedas de SSD Kingston UV500

20 menos buscados:

1. 1 búsquedas de Procesador Intel Core i3-8100
2. 1 búsquedas de MSI GeForce 210
3. 1 búsquedas de Tarjeta de Video VisionTek AMD Radeon HD5450
4. 1 búsquedas de Tarjeta Madre Gigabyte micro ATX Z390 M GAMING
5. 1 búsquedas de Tarjeta Madre ASRock ATX H110 Pro BTC+
6. 1 búsquedas de SSD Samsung 860 EVO
7. 1 búsquedas de Samsung Smart TV LED 43
8. 1 búsquedas de Ghia Bocina Portátil BX800
9. 1 búsquedas de Ginga Audífonos con Micrófono GI18ADJ01BT-RO
10. 2 búsquedas de Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix
11. 2 búsquedas de SSD para Servidor Lenovo Thinksystem S4500
12. 2 búsquedas de Acteck Bocina con Subwoofer AXF-290
13. 2 búsquedas de Genius GHP-400S Audífonos
14. 3 búsquedas de Tarjeta de Video Gigabyte AMD Radeon R7 370 OC
15. 3 búsquedas de ASUS T. Madre uATX M4A88T-M
16. 3 búsquedas de logear Audífonos Gamer GHG601
17. 4 búsquedas de Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming
18. 4 búsquedas de Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2
19. 4 búsquedas de Seiki TV LED SC-39HS950N 38.5
20. 4 búsquedas de Samsung Smart TV LED UN55TU7000FXZX 55

## Productos sin búsquedas:

1. Tarjeta de Video EVGA NVIDIA GeForce GT 710
2. Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming
3. Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile
4. Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC
5. Tarjeta de Video MSI Radeon X1550
6. Tarjeta de Video PNY NVIDIA GeForce RTX 2080
7. Tarjeta Madre AORUS ATX Z390 ELITE
8. Tarjeta Madre ASRock Z390 Phantom Gaming 4
9. Tarjeta Madre ASUS ATX PRIME Z390-A
10. Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI
11. Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0)
12. Tarjeta Madre ASRock ATX Z490 STEEL LEGEND
13. Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0
14. Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM
15. Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING
16. SSD Addlink Technology S70
17. SSD para Servidor Supermicro SSD-DM128-SMCMVN1
18. SSD para Servidor Lenovo Thinksystem S4510
19. Kit Memoria RAM Corsair Dominator Platinum DDR4
20. Kit Memoria RAM Corsair Vengeance LPX DDR4
21. Makena Smart TV LED 32S2 32"
22. Samsung TV LED LH43QMREBGCXGO 43
23. Samsung Smart TV LED UN70RU7100FXZX 70
24. Makena Smart TV LED 40S2 40"
25. Hisense Smart TV LED 40H5500F 39.5
26. Samsung Smart TV LED UN32J4290AF 32
27. Hisense Smart TV LED 50H8F 49.5
28. Lenovo Barra de Sonido
29. Verbatim Bocina Portátil Mini
30. Ghia Bocina Portátil BX300
31. Naceb Bocina Portátil NA-0301
32. Ghia Bocina Portátil BX900
33. Ghia Bocina Portátil BX400
34. Ghia Bocina Portátil BX500
35. ASUS Audífonos Gamer ROG Theta 7.1
36. Acer Audífonos Gamer Galea 300
37. Audífonos Gamer Balam Rush Orphix RGB 7.1
38. Energy Sistem Audífonos con Micrófono Headphones 1
39. Getttech Audífonos con Micrófono Sonority
40. Klip Xtreme Audífonos Blast

20 mejores puntuados:

1. Calificación 5.0/5 de Procesador AMD Ryzen 3 3300X S-AM4
2. Calificación 5.0/5 de Procesador Intel Core i9-9900K
3. Calificación 5.0/5 de Procesador Intel Core i7-9700K
4. Calificación 5.0/5 de Procesador Intel Core i5-9600K
5. Calificación 5.0/5 de Tarjeta de Video ASUS AMD Radeon RX 570
6. Calificación 5.0/5 de Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC
7. Calificación 5.0/5 de Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC
8. Calificación 5.0/5 de Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming
9. Calificación 5.0/5 de Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti
10. Calificación 5.0/5 de Tarjeta Madre Gigabyte XL-ATX TRX40 Designare
11. Calificación 5.0/5 de Kit SSD Kingston KC600
12. Calificación 5.0/5 de SSD Crucial MX500
13. Calificación 5.0/5 de SSD Western Digital WD Blue 3D NAND
14. Calificación 5.0/5 de Kit Memoria RAM Corsair Dominator Platinum DDR4
15. Calificación 5.0/5 de TCL Smart TV LED 55S425 54.6
16. Calificación 5.0/5 de TV Monitor LED 24TL520S-PU 24
17. Calificación 5.0/5 de Logitech Audífonos Gamer G332
18. Calificación 5.0/5 de Logitech Audífonos Gamer G635 7.1
19. Calificación 4.87/5 de SSD Adata Ultimate SU800
20. Calificación 4.81/5 de Procesador AMD Ryzen 5 2600

20 peores puntuados:

1. Calificación 1.0/5 de Tarjeta de Video Gigabyte AMD Radeon R7 370 OC
2. Calificación 1.0/5 de Tarjeta Madre ASRock ATX H110 Pro BTC+
3. Calificación 1.83/5 de Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0)
4. Calificación 2.0/5 de Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2
5. Calificación 3.0/5 de Cougar Audífonos Gamer Phontum Essential
6. Calificación 4.0/5 de MSI GeForce 210
7. Calificación 4.0/5 de Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix
8. Calificación 4.0/5 de HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro
9. Calificación 4.14/5 de Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING
10. Calificación 4.23/5 de Procesador AMD Ryzen 5 3600
11. Calificación 4.4/5 de Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030
12. Calificación 4.46/5 de Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8
13. Calificación 4.5/5 de Tarjeta Madre ASUS ATX PRIME Z390-A
14. Calificación 4.5/5 de Logitech Bocinas para Computadora con Subwoofer G560
15. Calificación 4.55/5 de SSD XPG SX8200 Pro
16. Calificación 4.56/5 de Tarjeta Madre ASRock Micro ATX B450M Steel Legend
17. Calificación 4.67/5 de Tarjeta Madre MSI ATX B450 TOMAHAWK MAX
18. Calificación 4.67/5 de SSD Kingston A2000 NVMe
19. Calificación 4.67/5 de SSD Kingston UV500
20. Calificación 4.7/5 de Procesador Intel Core i3-9100F

Productos sin ventas:

1. Procesador Intel Core i3-8100
2. Tarjeta de Video EVGA NVIDIA GeForce GT 710
3. Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming
4. Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming
5. Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile
6. Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC
7. Tarjeta de Video MSI Radeon X1550



8. Tarjeta de Video PNY NVIDIA GeForce RTX 2080
9. Tarjeta de Video VisionTek AMD Radeon HD 5450
10. Tarjeta de Video VisionTek AMD Radeon HD5450
11. Tarjeta Madre AORUS ATX Z390 ELITE
12. Tarjeta Madre ASRock Z390 Phantom Gaming 4
13. Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI
14. Tarjeta Madre Gigabyte micro ATX Z390 M GAMING
15. Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0)
16. Tarjeta Madre ASRock ATX Z490 STEEL LEGEND
17. Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0
18. ASUS T. Madre uATX M4A88T-M
19. Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM
20. Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING
21. SSD Addlink Technology S70
22. SSD para Servidor Supermicro SSD-DM128-SMCMVN1
23. SSD para Servidor Lenovo Thinksystem S4500
24. SSD para Servidor Lenovo Thinksystem S4510
25. SSD Samsung 860 EVO
26. Kit Memoria RAM Corsair Vengeance LPX DDR4
27. Makena Smart TV LED 32S2 32"
28. Seiki TV LED SC-39HS950N 38.5
29. Samsung TV LED LH43QMREBGCXGO 43
30. Samsung Smart TV LED UN70RU7100FXZX 70
31. Makena Smart TV LED 40S2 40"
32. Hisense Smart TV LED 40H5500F 39.5
33. Samsung Smart TV LED 43
34. Samsung Smart TV LED UN32J4290AF 32
35. Hisense Smart TV LED 50H8F 49.5
36. Samsung Smart TV LED UN55TU7000FXZX 55
37. Lenovo Barra de Sonido
38. Acteck Bocina con Subwoofer AXF-290
39. Verbatim Bocina Portátil Mini
40. Ghia Bocina Portátil BX300
41. Naceb Bocina Portátil NA-0301
42. Ghia Bocina Portátil BX800
43. Ghia Bocina Portátil BX900
44. Ghia Bocina Portátil BX400
45. Ghia Bocina Portátil BX500
46. ASUS Audífonos Gamer ROG Theta 7.1
47. Acer Audífonos Gamer Galea 300
48. Audífonos Gamer Balam Rush Orphix RGB 7.1
49. Energy Sistem Audífonos con Micrófono Headphones 1
50. Genius GHP-400S Audífonos
51. Getttech Audífonos con Micrófono Sonority
52. Ginga Audífonos con Micrófono GI18ADJ01BT-RO
53. Iogear Audífonos Gamer GHG601
54. Klip Xtreme Audífonos Blast

#### Ventas anuales

2020: \$ 755,968.00

2019: \$ 4,209.00

#### Promedio mensual

2020: \$ 83996.44

2019: \$ 4209.0

#### Ventas por mes

1. Abril 2020: \$193,295.00
2. Marzo 2020: \$164,729.00
3. Enero 2020: \$120,237.00
4. Febrero 2020: \$110,139.00
5. Mayo 2020: \$96,394.00
6. Junio 2020: \$36,949.00
7. Julio 2020: \$26,949.00
8. Noviembre 2019: \$4,209.00
9. Septiembre 2020: \$4,199.00
10. Agosto 2020: \$3,077.00

Se sugiere que la empresa deje de comprar Smart TV, así como tarjetas de video, ya que son de las categorías con menos búsquedas y menos ventas, se propone que se pongan en oferta los excedentes de inventario para disminuir el costo de este.

## Conclusión

LifeStore oferta 96 productos diferentes, de los cuales solo vende el 44%, de sus productos solo hay registro de búsqueda para el 58%, ten lo que va del 2020 tiene una venta anual de \$755,968.00 con un promedio mensual de \$89996.44. Abril fue el mes con mayor ventas registrando \$193,295.00

Enlace Github

<https://github.com/CarlosBerBo/Fundamentospython>