

ESTUDIANTE: Nicolás Bernal

RETO: reto 6

METODO IDEAL:

1) IDENTIFICAR EL PROBLEMA.

- **Interesados:** Cliente: Transporte del Norte / usuarios: Departamento de Logística
- **Problema:** desarrollar una interfaz que permita mostrar los indicadores de desempeño de cada vehículo, para cada cliente.
- **Objetivo:** la interfaz debe permitir:
 1. Seleccionar el punto de distribución al que se le calcularán los indicadores de desempeño
 2. Desplegar los siguientes indicadores de desempeño para cualquier punto de distribución (LAS PRUEBAS AUTOMÁTICAS EN REPLIT SE APLICAN SOLO AQUÍ):
 - Eficiencia en tiempos de despacho (%) = $100 * (\text{Tiempo total de despacho asignado} - \text{Tiempo total de despacho registrado}) / \text{Tiempo de total despacho asignado}$
 - Tasa de entrega (cajas/min) = $\text{Cantidad total de cajas despachadas} / \text{Tiempo total de despacho}$
 - Nivel de cumplimiento de los despachos (%) = $100 * (\text{Total de cajas despachadas} / \text{Total de cajas asignadas})$
 - Entregas a tiempo (%) = $100 * (\text{No. de entregas a tiempo} / \text{No. Total de entregas realizadas})$. Nota: este es un resultado que es constante, es decir, es independiente del punto de distribución porque toma en cuenta todos los puntos de distribución.
 3. Registrar (escribir) en un archivo CSV un informe con los indicadores de desempeño anteriores (todos excepto el último), para cada punto de distribución (ESTO ES UN COMPLEMENTO SIN PRUEBA AUTOMÁTICA), que luzca como sigue:
EficienciaTiemposDespacho_%,TasaEntrega_cajas/min,
NivelCumplimientoDespachos_% 7.3, 22.3, 83.3
 4. Realizar 6 graficas usando matplotlib: 3 con los datos del archivo 'ValoresAsignados.csv', y 3 con los datos del archivo 'Valores Registrados.csv' (ESTO ES OTRO COMPLEMENTO SIN PRUEBA AUTOMÁTICA):
 - Carga_cajas (Eje Y) vs PuntoDistribucion (Eje X)
 - Tiempo_minutos (Eje Y) vs PuntoDistribucion (Eje X)
 - Unica graáfica que muestre i y ii
- **Restricciones:**

1. Los datos serán suministrados en dos archivos
2. archivo "ValoresAsignados.csv", que cuenta con la **programación asignada** por la empresa:
 - a) PuntoDistribución
 - b) CargaEntregada_cajas
 - c) TiempoDespacho_minutos
3. archivo "ValoresRegistrados.csv", **operación registrada** por los conductores.
 - a) PuntoDistribución
 - b) CargaEntregada_cajas
 - c) TiempoDespacho_minutos

2) DEFINIR EL PROBLEMA:

- **Necesidad:**
- **Recomendaciones:**
 - El ingreso de las planillas será mediante el empleo de archivos
 - Mantener el formato de la "salida esperada" (orden, palabras y signos). Para ello se recomienda copiar y pegar del ejemplo.
 - Redondear los números reales a 1 dígito decimal.

3) ESTRATEGIAS/ALGORITMO:

Tareas o funciones / dividir problema:

- main:
 - i. Llamar las funciones
 - ii. imprimir los valores solicitados
 - iii. para las graficas utilizar import matplotlib.pyplot as plt
- función 1 Control:
 - i. Crear una matriz para cada uno de los archivos. Con un FOR y para pasar por todas las líneas y un append para ingresar en la nueva matriz
 - ii. Crear valores con la columna fija y variable la fila según el punto registrado por el usuario
 - iii. Con estos valores se hacen los cálculos solicitados
 - iv. Para la eficiencia total, generar un contador para los valores que sean de la matriz registrado, respecto a la asignada revisando el mismo valor de la fila.
- función 2 indicadores de desempeño:

- i. Utilizando las funciones de archivos, open, write y close, llamar a la función control, generar los indicadores en variables, anidarlas en una sola variable incluyendo las comas y el salto de línea.
- función 3 graficas:
 - i. Crear una lista con los valores del eje x, que son constantes.
 - ii. Para los del eje x, se genera una lista nueva y se utiliza un FOR para ingresar los valores solicitados de cada una de las lines y se entrega con un return.

CODIGO EN REPLIT

MAIN

```
import control as ctr
import matplotlib.pyplot as plt

#ingresar el punto que se requiere conocer eficiencia
punto_distribucion=int(input("punto_distribucion="))

#la funcion control entrega los 4 puntos de eficiencia solicitados en el reto
eficiencia, cajas_tasa,
cumplimiento_cajas,eficiencia_total=ctr.control(punto_distribucion)

#se imprimen los 4 valores entregados por la función control
print("Eficiencia en tiempos de despacho =",eficiencia,"%")
print("Tasa de entrega =", cajas_tasa,"cajas/min")
print("Nivel de cumplimiento de los despachos =",cumplimiento_cajas,"%")
print("Entregas a tiempo =",eficiencia_total,"%")

#función que genera el el texto para el informe de indicadores de desempeño
desempeño=ctr.indicadores_desempeño()

ejex, ejey1, ejey2=ctr.graficar_asignados()
print("=====GRAFICA VALORES ASIGNADOS=====")
print("-----ASIGNADOS - Carga_cajas vs PuntoDistribucion-----")
plt.plot(ejex,ejey1,"r*")
plt.ylabel("Carga_cajas asignadas")
plt.xlabel("PuntoDistribucion")
plt.savefig("gafica1")
plt.show()
print("-----ASIGNADOS - Tiempo_minutos vs PuntoDistribucion-----")
```

```

plt.plot(ejex,ejey2,"r*")
plt.ylabel("Tiempo_minutos asignados")
plt.xlabel("PuntoDistribucion")
plt.savefig("gafica2")
plt.show()
print("-----ASIGNADOS - Carga_cajas Y Tiempo_minutos vs PuntoDistribucion---
-----")
plt.plot(ejex,ejey1,"r*",ejex,ejey2,"b+")
plt.ylabel("Tiempo_minutos y Carga_cajas asignados")
plt.xlabel("PuntoDistribucion")
plt.savefig("gafica3")
plt.show()

ejex2, ejey3, ejey4=ctr.graficar_registrados()
print("=====GRAFICAS VALORES REGISTRADOS=====")
print("-----REGISTRADOS Carga_cajas vs PuntoDistribucion-----")
plt.plot(ejex,ejey3,"b--")
plt.ylabel("Carga_cajas registrados")
plt.xlabel("PuntoDistribucion")
plt.savefig("gafica4")
plt.show()
print("-----REGISTRADO - Tiempo_minutos vs PuntoDistribucion-----")
plt.plot(ejex,ejey4,"b--")
plt.ylabel("Tiempo_minutos registrados")
plt.xlabel("PuntoDistribucion")
plt.savefig("gafica5")
plt.show()
print("-----REGISTRADO - Carga_cajas Y Tiempo_minutos vs PuntoDistribucion--
-----")
plt.plot(ejex,ejey3,"b--",ejex,ejey4,"r*")
plt.ylabel("Tiempo_minutos y Carga_cajas registrados")
plt.xlabel("PuntoDistribucion")
plt.savefig("gafica6")
plt.show()

```

CONTROL

```

# Funci n que genera los 4 puntos de eficiencia del reto
def control(punto_distribucion):
    f=punto_distribucion
    # este es el codigo para la lectura de los archivos

```

```

fh1= open("ValoresAsignados.csv")
MA = []
for linea in fh1:
    valores1= linea.rstrip().split(',')
    MA.append(valores1)
fh2= open("ValoresRegistrados.csv")
MR = []
for linea in fh2:
    valores2= linea.rstrip().split(',')
    MR.append(valores2)
#este es el tado para generar los valores que se engredsaran a los calculos
tiempo_asignado=float(MA[f][2])
tiempo_registrado=float(MR[f][2])
cajas_asignado=float(MA[f][1])
cajas_registrado=float(MR[f][1])
#RESPUESTA 1, formula matematica
eficiencia=round(100*(tiempo_asignado-tiempo_registrado)/tiempo_asignado,1)
#RESPUESTA 2, formula matematica
cajas_tasa=round(cajas_registrado/tiempo_registrado,1)
#RESPUESTA 3, formula matematica
cumplimiento_cajas=round(100*(cajas_registrado/cajas_asignado),1)
#RESPUESTA 4, formula matematica
cont=0
cumplidos=0
for f in range(1,31):
    c=2
    if MR[f][c]<=MA[f][c]:
        cumplidos+=1
    cont+=1
eficiencia_total=100*(cumplidos/cont)
return eficiencia, cajas_tasa, cumplimiento_cajas,eficiencia_total

```

```

def indicadores_desempeño():
    f = open ("IndicadoresDesempeño.csv","wb")
    desempeño="EficienciaTiemposDespacho_%,TasaEntrega_cajas/min,
NivelCumplimientoDespachos_%\n"
    ingreso=desempeño.encode("ascii")
    f.write(ingreso)
    cont=1
    while cont<=30:
        a,b,c,d=control(cont)
        desempeño=str(a)+","+str(b)+","+str(c)+"\n"
        ingreso=desempeño.encode("ascii")

```

```
f.write(ingreso)
cont+=1
f.close()
return f
```

```
def graficar_asignados():
```

```
puntosX=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30]
fh1= open("ValoresAsignados.csv")
lista_cajas=[]
lista_tiempos=[]
for linea in fh1:
    valores= linea.rstrip().split(',')
    lista=list(valores)
    lista_cajas.append(lista[1])
    lista_tiempos.append(lista[2])
del lista_cajas[0]
del lista_tiempos[0]
return puntosX, lista_cajas, lista_tiempos
```

```
def graficar_registrados():
```

```
puntosX=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30]
fh2= open("ValoresRegistrados.csv")
lista_cajas=[]
lista_tiempos=[]
for linea in fh2:
    valores= linea.rstrip().split(',')
    lista=list(valores)
    lista_cajas.append(lista[1])
    lista_tiempos.append(lista[2])
del lista_cajas[0]
del lista_tiempos[0]
return puntosX, lista_cajas, lista_tiempos
```