

# Codigo de almacenamiento para repuesto de motor de equipos de workover

Nicolas Parra Salgado

PROM-M






# Contexto

Un equipo de Workover es una unidad de perforación y mantenimiento utilizada en la industria petrolera y de gas para llevar a cabo operaciones de mantenimiento, reparación, reacondicionamiento o intervención en pozos de petróleo y gas que ya han sido perforados y producen hidrocarburos.

Es fundamental que el motor de la unidad básica opere en óptimas condiciones pues de este depende todo su funcionamiento, por ende, es indispensable realizar sus mantenimientos respectivos y con ello contar con los repuestos necesarios.



# Pregunta realizada a chat gpt.



¿Podrías darme un ejemplo de un código en Python que realice el inventario de repuestos para realizar mantenimiento al motor de la unidad básica de un equipo de workover el cual es un Detroit diésel serie 60 de 475 hp, ten en cuenta que los repuestos utilizados son: aceite, filtro de aire, filtro de aceite y correa de transmisión , se necesita que por lo menos exista en inventario 3 unidades para cada repuesto y se muestre en una grafica de barras el nombre de la pieza y la cantidad disponible, se debe tener en cuenta que la empresa cuenta con 28 equipos de workover?

# Complemento a la pregunta realizada a chat gpt.

N

puesto que los 28 equipos están divididos en 3 zonas necesito que también realices esa división, donde los equipos se encuentran divididos así: 13 en zona Meta, 5 en zona Arauca y 10 en zona Huila.

N

debes mostrarme una grafica de barras por zona para la el nombre y cantidad de repuestos



# Codigo generado

python

Copy code

```
import pandas as pd
import matplotlib.pyplot as plt

# Crear un DataFrame para gestionar el inventario de repuestos
inventario = pd.DataFrame(columns=['Zona', 'Repuesto', 'Cantidad'])

# Agregar repuestos iniciales al inventario para cada zona
repuestos_iniciales = [
    {'Zona': 'Meta', 'Repuesto': 'Aceite', 'Cantidad': 20},
    {'Zona': 'Meta', 'Repuesto': 'Filtro de Aire', 'Cantidad': 25},
    {'Zona': 'Meta', 'Repuesto': 'Filtro de Aceite', 'Cantidad': 30},
    {'Zona': 'Meta', 'Repuesto': 'Correa de Transmisión', 'Cantidad': 15},
    {'Zona': 'Arauca', 'Repuesto': 'Aceite', 'Cantidad': 20},
    {'Zona': 'Arauca', 'Repuesto': 'Filtro de Aire', 'Cantidad': 25},
    {'Zona': 'Arauca', 'Repuesto': 'Filtro de Aceite', 'Cantidad': 30},
    {'Zona': 'Arauca', 'Repuesto': 'Correa de Transmisión', 'Cantidad': 15},
    {'Zona': 'Huila', 'Repuesto': 'Aceite', 'Cantidad': 20},
    {'Zona': 'Huila', 'Repuesto': 'Filtro de Aire', 'Cantidad': 25},
    {'Zona': 'Huila', 'Repuesto': 'Filtro de Aceite', 'Cantidad': 30},
    {'Zona': 'Huila', 'Repuesto': 'Correa de Transmisión', 'Cantidad': 15},
]
```

# Código generado

```
# Función para registrar la utilización de repuestos y verificar el inventario
def registrar_repuestos_utilizados(zona, repuesto, cantidad_utilizada):
    if repuesto in inventario[(inventario['Zona'] == zona)][['Repuesto']].values:
        inventario.loc[(inventario['Zona'] == zona) & (inventario['Repuesto'] == repuesto)]['cantidad_disponible'] -= cantidad_utilizada
        if cantidad_disponible < 3:
            print(f"Alerta en {zona}: Quedan menos de 3 unidades de {repuesto}")

# Función para mostrar el estado actual del inventario por zona
def mostrar_inventario_por_zona(zona):
    zona_inventario = inventario[inventario['Zona'] == zona]
    print(f"Estado actual del inventario en la zona {zona}:")
    print(zona_inventario)

# Registramos la utilización de repuestos en las zonas (simulación)
zona = 'Meta' # Cambia la zona según corresponda
registrar_repuestos_utilizados(zona, 'Aceite', 5)
registrar_repuestos_utilizados(zona, 'Filtro de Aire', 10)
```

# Código generado

```
# Mostrar el estado actual del inventario por zona después de registrar la u
zonas = ['Meta', 'Arauca', 'Huila']
for zona in zonas:
    mostrar_inventario_por_zona(zona)

# Crear gráficas de barras para visualizar el inventario en cada zona
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(15, 5))

for i, zona in enumerate(zonas):
    zona_inventario = inventario[inventario['Zona'] == zona]
    zona_inventario.set_index('Repuesto', inplace=True)
    zona_inventario.plot(kind='bar', ax=axes[i], y='Cantidad', legend=False)
    axes[i].set_title(f'Inventario en {zona}')
    axes[i].set_ylabel('Cantidad')
    axes[i].set_ylim(0, max(zona_inventario['Cantidad']) + 5)

plt.tight_layout()
plt.show()
```

# Resultado

Estado actual del inventario en la zona Meta:

	Zona	Repuesto	Cantidad
0	Meta	Aceite	15
1	Meta	Filtro de Aire	15
2	Meta	Filtro de Aceite	30
3	Meta	Correa de Transmisión	15

Estado actual del inventario en la zona Arauca:

	Zona	Repuesto	Cantidad
4	Arauca	Aceite	20
5	Arauca	Filtro de Aire	25
6	Arauca	Filtro de Aceite	30
7	Arauca	Correa de Transmisión	15

Estado actual del inventario en la zona Huila:

	Zona	Repuesto	Cantidad
8	Huila	Aceite	20
9	Huila	Filtro de Aire	25
10	Huila	Filtro de Aceite	30
11	Huila	Correa de Transmisión	15

