

Ejercicio Práctico: Análisis de Datos Biosanitarios con Docker, NumPy, Pandas y Matplotlib

Descripción del ejercicio

En este ejercicio, los alumnos crearán un entorno de Data Science usando **Docker**, donde analizarán datos de pacientes ficticios mediante **NumPy**, **Pandas** y **Matplotlib**.

El flujo de trabajo incluye:

- ✓ Contenerización de Jupyter con las bibliotecas necesarias.
 - ✓ Generación de datos de pacientes con NumPy.
 - ✓ Análisis de datos biosanitarios con Pandas.
 - ✓ Visualización de resultados con Matplotlib.
 - ✓ Publicación del contenedor en **Docker Hub** para compartir su trabajo.
-

1. Preparación del entorno

1. Crea el directorio del proyecto con la siguiente estructura:

```
mi_proyecto_biosanitario/  
├── notebooks/      # Para los notebooks de Jupyter  
├── data/           # Para guardar datasets  
├── results/        # Para almacenar gráficos y análisis  
├── docker-compose.yml  
├── Dockerfile  
└── requirements.txt
```

2. Edita **requirements.txt** para incluir las bibliotecas necesarias:
-

2. Creación del Dockerfile

Dentro del directorio, crea un `Dockerfile` con el siguiente contenido:

3. Configuración de `docker-compose.yml`

Modifica el archivo `docker-compose.yml` para incluir el uso del `Dockerfile` creado:

4. Creación del Notebook "análisis_pacientes.ipynb"

Dentro de la carpeta `notebooks`, los alumnos deben crear un archivo Jupyter que incluya el siguiente análisis:

Generación de datos biosanitarios con NumPy

Los alumnos deben generar un dataset de `pacientes ficticios` con NumPy:

```
data = {
    "ID_Paciente": ,
    "Edad":(18, 90,),
    "Peso (kg)":(70, 15,), ),
    "Altura (cm)":((170, 10,), 2),
    "Presión Sistólica":(90, 180,),
    "Presión Diastólica":(60, 110,),
```

```
"Frecuencia Cardíaca":(50, 120,)
}

# Convertir a DataFrame de Pandas

# Guardar el dataset en un archivo CSV dentro del volumen persistente

print(f"Datos de pacientes guardados en {ruta_csv}")
df_pacientes.head()
```

5. Análisis de datos con Pandas

Los alumnos deben leer los datos desde el CSV y analizar la distribución de variables relevantes:

```
# Cargar Los datos

# Ver primeras filas

# Estadísticas descriptivas

# Identificar posibles hipertensos (Presión Sistólica > 140 o Diastólica > 90)

# Guardar Los resultados
```

6. Visualización de datos con Matplotlib

Los alumnos deben crear gráficos para interpretar los datos:

```
# Histograma de edades
plt.figure(figsize=(8,5))
plt.hist()
plt.xlabel()
plt.ylabel("Número de Pacientes")
plt.title("Distribución de Edades de los Pacientes")
plt.grid()
plt.savefig()
plt.show()

# Relación entre peso y altura
plt.figure(figsize=(8,5))
plt.scatter()
plt.xlabel("Altura (cm)")
plt.ylabel("Peso (kg)")
plt.title("Relación entre Altura y Peso")
plt.grid()
plt.savefig()
plt.show()
```

7. Construcción de la imagen Docker

Desde la terminal, dentro del directorio del proyecto, ejecutar:

```
docker-compose build
```

Esto generará la imagen con todas las dependencias necesarias.

8. Probar el contenedor

Ejecuta el contenedor con:

```
docker-compose up
```

Accede a Jupyter Lab en <http://localhost:8000> y verifica que los datos se generan y se analizan correctamente.

9. Publicar la imagen en Docker Hub

1. Iniciar sesión en Docker:

```
docker login
```

2. Etiquetar la imagen:

```
docker tag mi_proyecto_biosanitario:latest <TU_USUARIO>/mi_proyecto_biosanitario:v1
```

3. Subir la imagen:

```
docker push <TU_USUARIO>/mi_proyecto_biosanitario:v1
```

10. Probar la imagen en otra máquina

Para verificar que la imagen funciona en otro entorno:

```
docker run -p 8000:8888 -v $(pwd)/notebooks:/home/jovyan/notebooks -v  
$(pwd)/data:/home/jovyan/data -v $(pwd)/results:/home/jovyan/results  
<TU_USUARIO>/mi_proyecto_biosanitario:v1
```

Esto descargará la imagen desde **Docker Hub** y ejecutará el entorno con persistencia de datos.

Entrega del ejercicio

Los alumnos deben entregar:

- ✓ Un enlace a su imagen en Docker Hub.
 - ✓ Capturas de pantalla mostrando Jupyter, los análisis y gráficos generados.
 - ✓ Sus archivos `Dockerfile`, `docker-compose.yml` y `requirements.txt`.
 - ✓ Un breve informe explicando su proceso y problemas encontrados.
-