

# Laboratorio: Validación de Contraseñas con Pandas en Python

Tu Nombre  
Universidad Tecnológica de Pereira

## I. INTRODUCCIÓN

En este laboratorio, exploraremos cómo utilizar la biblioteca Pandas en Python para validar contraseñas. Utilizaremos un conjunto de reglas específicas para determinar si una contraseña es válida o no. Este problema es común en el desarrollo de aplicaciones web y sistemas de autenticación.

## II. OBJETIVOS

Al finalizar este laboratorio, los estudiantes deberán ser capaces de:

- Crear y manipular DataFrames con Pandas.
- Aplicar funciones de validación en Pandas para analizar contraseñas.
- Comprender y resolver problemas de validación de datos.

## III. PROBLEMA: VALIDACIÓN DE CONTRASEÑAS

Escriba un programa que valide una nueva contraseña, teniendo en cuenta las siguientes reglas:

- La contraseña debe tener al menos 8 caracteres.
- La contraseña debe tener al menos una letra mayúscula y una minúscula.
- La contraseña debe tener al menos un número.

El programa debe validar las contraseñas del conjunto de datos establecidos.

### III-A. Paso 1: Crear un DataFrame

Cree un DataFrame con contraseñas para su análisis. Puede utilizar una lista de contraseñas de ejemplo.

#### Algoritmo 1.

```
1 import pandas as pd
2
3 data = {'Contraseña': ['Pass123', 'SecurePwd', 'weak', 'StrongPwd1']}
4 df = pd.DataFrame(data)
5 print(df)
```

### III-B. Paso 2: Implementar Validación en el DataFrame

Implemente la validación directamente en el DataFrame para cada regla especificada en el problema. Por ejemplo:

### III-C. Paso 3: Análisis de Datos

Realiza un análisis exploratorio de datos utilizando Pandas. Calcula la cantidad de contraseñas válidas e inválidas, identifica patrones y presenta conclusiones.

#### Algoritmo 2.

```
1 # Paso 2: Implementar Validación en el DataFrame
2 df['Longitud válida'] = df['Contraseña'].apply(lambda x:
3     len(x) >= 8)
4 df['Mayúscula'] = df['Contraseña'].apply(lambda x: any(
5     char.isupper() for char in x))
6 df['Minúscula'] = df['Contraseña'].apply(lambda x: any(
7     char.islower() for char in x))
8 df['Número'] = df['Contraseña'].apply(lambda x: any(char
9     .isdigit() for char in x))
10 df['Carácter Especial'] = df['Contraseña'].apply(lambda
11     x: any(not char.isalnum() for char in x))
```

#### Algoritmo 3.

```
1
2 # Paso 3: Análisis de Datos
3 # Imprime estadísticas para cada regla de validación
4 print("Estadísticas de Validación de Contraseñas:")
5 print("Cantidad de contraseñas válidas (longitud >= 8):",
6     df['Longitud válida'].sum())
7 print("Cantidad de contraseñas con al menos una mayú
8     scula:", df['Mayúscula'].sum())
9 print("Cantidad de contraseñas con al menos una minú
10     scula:", df['Minúscula'].sum())
11 print("Cantidad de contraseñas con al menos un número:",
12     df['Número'].sum())
13 print("Cantidad de contraseñas con al menos un carácter
14     especial:", df['Carácter Especial'].sum())
```

### III-D. Visualización de Datos

Visualice con un diagrama de barras la cantidad de contraseñas vs. la validación.

#### Algoritmo 4.

```
1 # Paso 4: Visualización
2 import matplotlib.pyplot as plt
3
4 validaciones = ['Longitud válida', 'Mayúscula', 'Minú
5     scula', 'Número', 'Carácter Especial']
6 cantidad_validaciones = [df[val].sum() for val in
7     validaciones]
8
9 plt.bar(validaciones, cantidad_validaciones, color=['
10     blue', 'green', 'orange', 'red', 'purple'])
11 plt.title('Estadísticas de Validación de Contraseñas')
12 plt.xlabel('Reglas de Validación')
13 plt.ylabel('Cantidad de Contraseñas')
14 plt.show()
```

## IV. RESULTADOS

```
1 Estadísticas de Validación de Contraseñas:
2 Cantidad de contraseñas válidas (longitud >= 8): 2
3 Cantidad de contraseñas con al menos una mayúscula: 3
4 Cantidad de contraseñas con al menos una minúscula: 4
5 Cantidad de contraseñas con al menos un número: 2
```

6 | Cantidad de contraseñas con al menos un carácter especial: 0

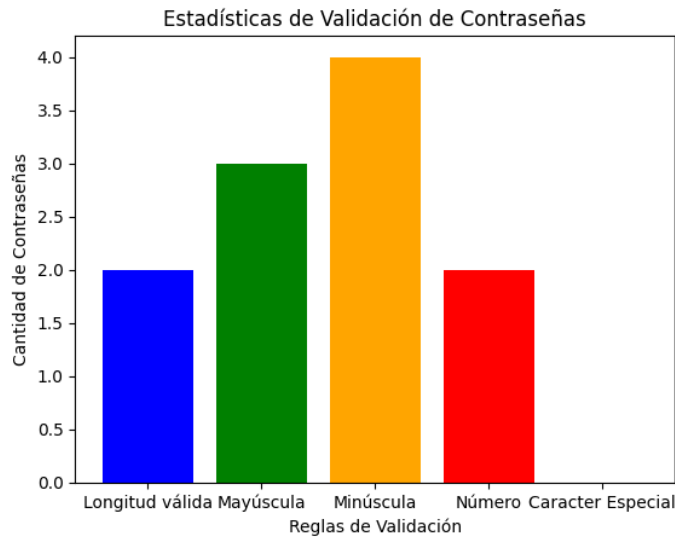


Figura 1. Diagrama de barras del proceso

## V. CONCLUSIONES

En este laboratorio, hemos utilizado Pandas para analizar y validar contraseñas directamente en el DataFrame. Se ha demostrado cómo trabajar con DataFrames, aplicar validaciones y realizar un análisis exploratorio de datos. Este conocimiento es fundamental en el desarrollo de sistemas de autenticación robustos.