



**UNIVERSIDAD DE
GUADALAJARA**

Red Universitaria e Institución Benemérita de Jalisco

CENTRO DE CIENCIAS EXACTAS E INGENIERIAS

DIVISION DE INGENIERIAS



Daniel Padilla Mora - 221350184

Computación Tolerante a Fallas

Segundo ejemplo práctico utilizando el orquestador de Prefect

Paso 1: Importar las Bibliotecas Necesarias

```
import prefect
from prefect import task, Flow
import random
import json
import matplotlib.pyplot as plt
import statistics
```

Paso 2: Definir las Tareas

Vamos a definir las tareas que compondrán nuestro flujo de trabajo. Cada tarea será una función decorada con `@task`.

```
@task
def generar_numeros_aleatorios():
    return [random.randint(1, 100) for _ in range(10)]

@task
def guardar_en_json(datos, nombre_archivo):
    with open(nombre_archivo, 'w') as archivo_json:
        json.dump(datos, archivo_json)

@task
def cargar_desde_json(nombre_archivo):
    try:
        with open(nombre_archivo, 'r') as archivo_json:
            return json.load(archivo_json)
    except FileNotFoundError:
        return None

@task
def calcular_estadisticas(numeros):
    return {
        "media": statistics.mean(numeros),
        "mediana": statistics.median(numeros),
        "desviacion_estandar": statistics.stdev(numeros),
        "minimo": min(numeros),
        "maximo": max(numeros)
    }

@task
```

```

def validar_datos():
    numeros_generados = generar_numeros_aleatorios.run()
    while any(num <= 40 for num in numeros_generados):
        numeros_generados = generar_numeros_aleatorios.run()
    return numeros_generados

@task
def visualizar_errores(errores_por_intentos, numeros_validos):
    intentos = range(1, len(errores_por_intentos) + 1)
    fig, ax1 = plt.subplots()

    color = 'tab:red'
    ax1.set_xlabel('Intento')
    ax1.set_ylabel('Número de Números No Válidos', color=color)
    ax1.plot(intentos, errores_por_intentos, marker='o', linestyle='-',
color=color)
    ax1.tick_params(axis='y', labelcolor=color)

    ax2 = ax1.twinx()
    color = 'tab:blue'
    ax2.set_ylabel('Números Válidos', color=color)
    ax2.plot(intentos, numeros_validos, marker='o', linestyle='-',
color=color)
    ax2.tick_params(axis='y', labelcolor=color)

    fig.tight_layout()
    plt.title('Errores en la Validación de Números Aleatorios por Intento')
    plt.show()

```

Paso 3: Definir el Flujo de Trabajo

Ahora definiremos el flujo de trabajo utilizando el contexto Flow. En este paso, estableceremos las dependencias entre las tareas y cualquier lógica adicional necesaria.

```
nombre_archivo = 'numeros_guardados.json'

with Flow("Procesamiento de Datos") as flow:
    numeros_guardados = cargar_desde_json(nombre_archivo)
    numeros_validos = generar_numeros_aleatorios()
    errores_validacion = validar_datos()
    visualizacion = visualizar_errores(errores_validacion, numeros_validos)
    estadisticas = calcular_estadisticas(numeros_validos)

    if numeros_guardados is not None:
        guardar_en_json(numeros_validos, nombre_archivo)

    visualizacion.set_upstream(errores_validacion)
    estadisticas.set_upstream(numeros_validos)
```

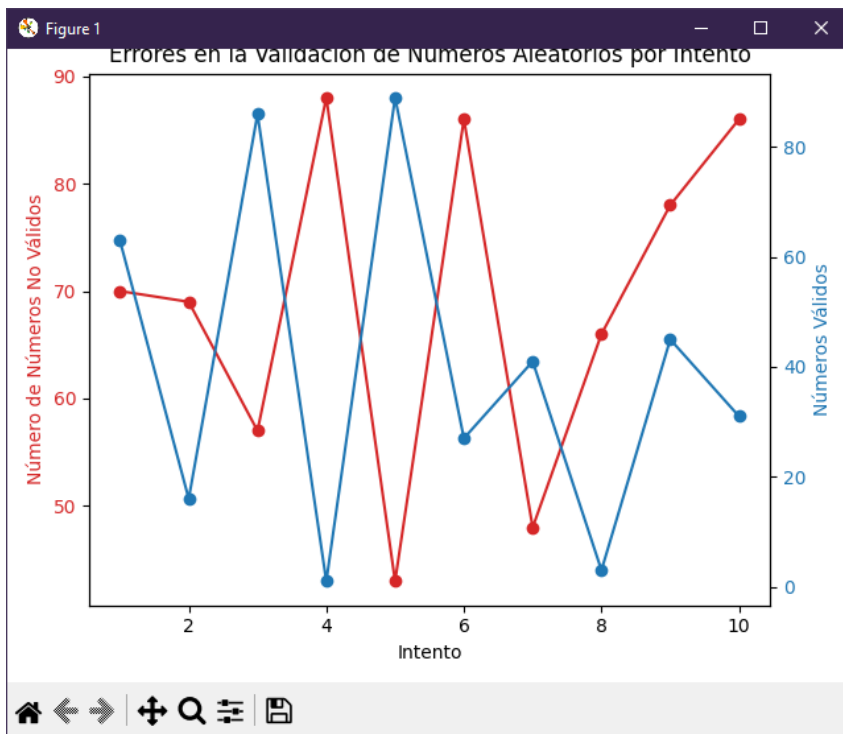
Paso 4: Ejecutar el Flujo de Trabajo

Finalmente, ejecutaremos nuestro flujo de trabajo utilizando el método run() del flujo que hemos definido.

```
flow.run()
```

Paso 5: Capturas de Pantalla de la Ejecución

```
orquestadorPrefect.py {} numeros_guardados.json X
C: > Users > padil > OneDrive > Documentos > School > Semestr
1 [[63, 16, 86, 1, 89, 27, 41, 3, 45, 31]]
```



```
[2024-03-17 00:21:35-0600] INFO - prefect.FlowRunner | Beginning Flow run for 'Procesamiento de Datos'
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'cargar_desde_json': Starting task run...
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'cargar_desde_json': Finished task run for task with final state: 'Success'
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'validar_datos': Starting task run...
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'validar_datos': Finished task run for task with final state: 'Success'
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'generar_numeros_aleatorios': Starting task run...
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'generar_numeros_aleatorios': Finished task run for task with final state: 'Success'
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'guardar_en_json': Starting task run...
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'guardar_en_json': Finished task run for task with final state: 'Success'
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'calcular_estadisticas': Starting task run...
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'calcular_estadisticas': Finished task run for task with final state: 'Success'
[2024-03-17 00:21:35-0600] INFO - prefect.TaskRunner | Task 'visualizar_errores': Starting task run...
[2024-03-17 00:21:48-0600] INFO - prefect.TaskRunner | Task 'visualizar_errores': Finished task run for task with final state: 'Success'
[2024-03-17 00:21:48-0600] INFO - prefect.FlowRunner | Flow run SUCCESS: all reference tasks succeeded
```