

Computación Tolerante a Fallas

Sección D06

Workflow Managers



Fernández Venegas David Guadalupe

Código: 216437352

Profesor: López Franco Michel Emanuel

Introducción

Los Workflow Managers son herramientas fundamentales en el mundo del desarrollo de software y la automatización de procesos. Estas herramientas permiten a los equipos de desarrollo diseñar, ejecutar y monitorear flujos de trabajo complejos de manera eficiente y confiable. Los flujos de trabajo pueden abarcar desde tareas simples hasta procesos empresariales completos, coordinando la ejecución de múltiples tareas de manera secuencial o paralela.

Dentro de estas herramientas se encuentra la librería de Python Prefect, con la cual los desarrolladores pueden crear flujos de trabajo complejos con una sintaxis intuitiva y expresiva, aprovechando características como el control de dependencias, la gestión de errores y la escalabilidad. Además, Prefect ofrece una interfaz de usuario amigable para monitorear y gestionar flujos de trabajo en tiempo real, lo que facilita la depuración y optimización de procesos.

Programa

En este trabajo se utiliza la librería Prefect para definir dos flujos de trabajo distintos, cada uno centrado en una función específica.

Librerías:

- Prefect: para el manejo de flow y task
- Random: para generar números aleatorios.
- Datetime: para manejar intervalos de tiempo.

Task:

Se definen dos funciones de tareas utilizando el decorador `@task` de Prefect: `eliminar_pares` y `eliminar_impares` son funciones que eliminan números pares e impares, respectivamente, de una lista de datos.

Estas tareas además tienen un porcentaje de probabilidad de fallar, esto con el propósito de poder observar en la interfaz los resultados de las tareas y los flujos.

```
# Función para eliminar números pares de la lista
@task(cache_expiration=timedelta(minutes=2))
def eliminar_pares(datos_a_ordenar):
    # Simulación de fallo aleatorio
    if random.random() < 0.4: # Probabilidad de fallo del 40%
        raise Exception("Error: Tiempo de eliminación de números pares
excedido.")
    return [x for x in datos_a_ordenar if x % 2 != 0]
```

```
# Función para eliminar números impares de la lista
@task(cache_expiration=timedelta(minutes=2))
def eliminar_impares(datos_a_ordenar):
    # Simulación de fallo aleatorio
    if random.random() < 0.4: # Probabilidad de fallo del 40%
        raise Exception("Error: Tiempo de eliminación de números impares
excedido.")
    return [x for x in datos_a_ordenar if x % 2 == 0]
```

Flows:

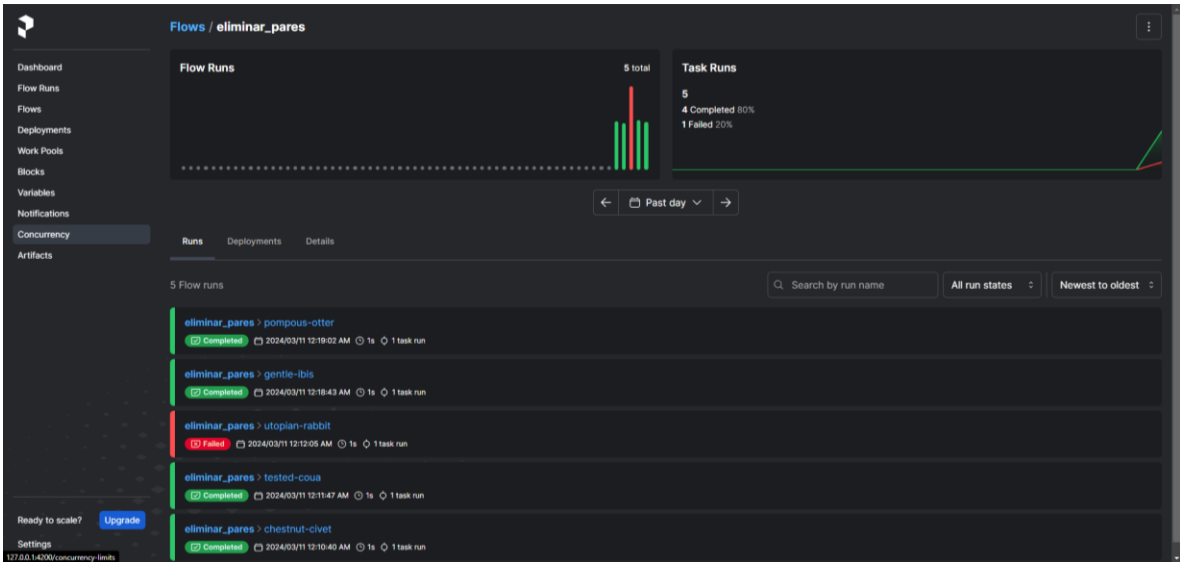
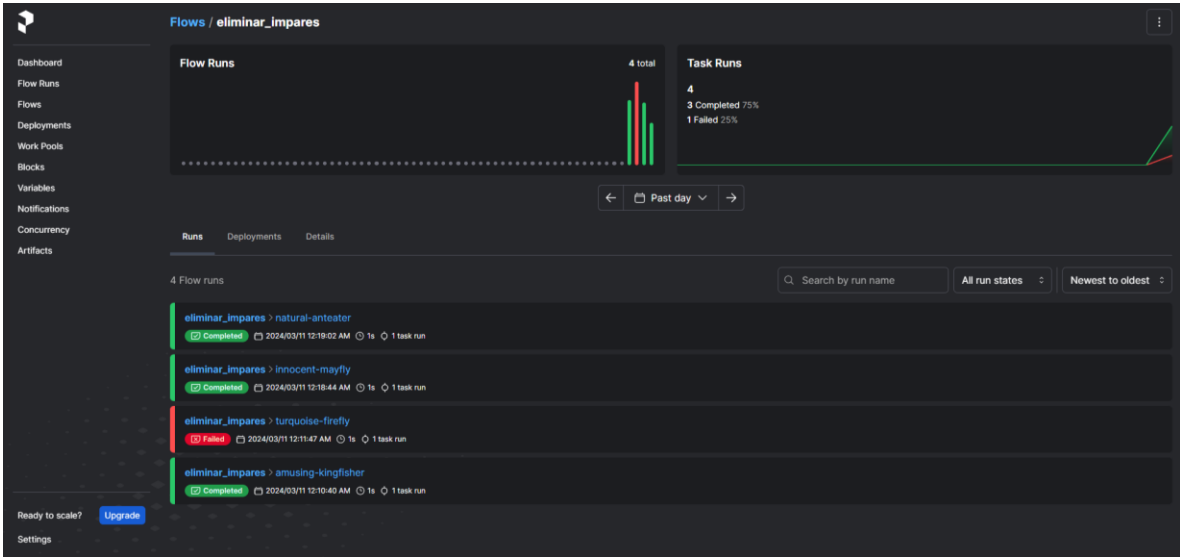
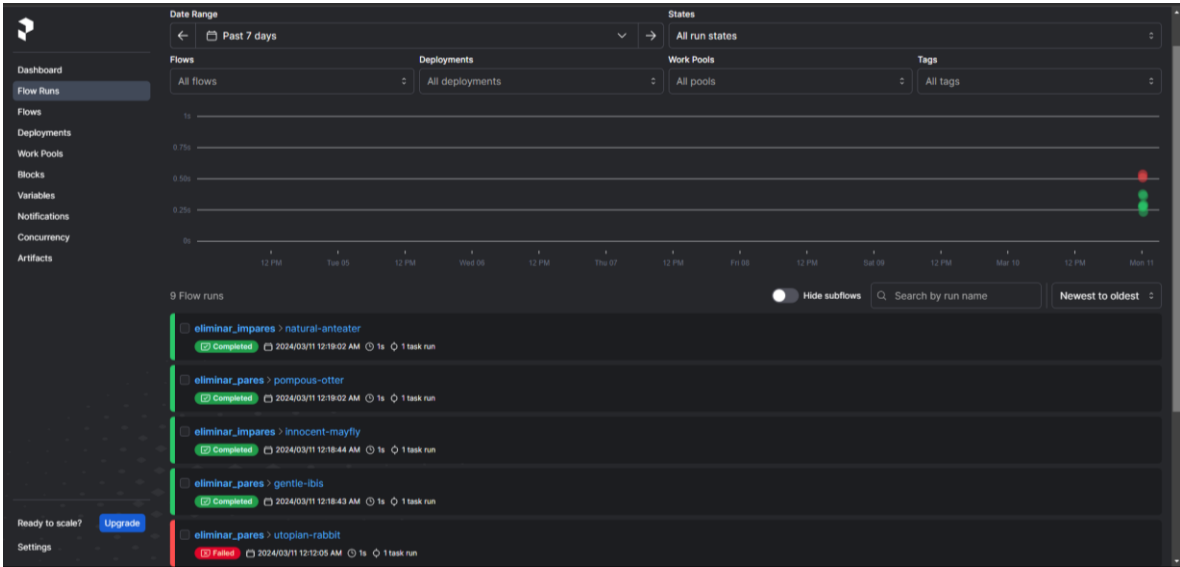
Se definen dos flujos de trabajo utilizando la etiqueta @flow de Prefect. Cada flujo genera una lista de datos aleatorios y luego aplica la función de tarea correspondiente (eliminar_pares o eliminar_impares) a esa lista.

```
# Flujo para eliminar números pares
@flow(name="eliminar_pares", retries=2, retry_delay_seconds=2,
log_prints=True)
def flujo_eliminar_pares():
    datos_a_ordenar = [random.randint(1, 500) for _ in range(1000)]
    return eliminar_pares(datos_a_ordenar)
```

```
# Flujo para eliminar números impares
@flow(name="eliminar_impares", retries=2, retry_delay_seconds=2,
log_prints=True)
def flujo_eliminar_impares():
    datos_a_ordenar = [random.randint(1, 500) for _ in range(1000)]
    return eliminar_impares(datos_a_ordenar)
```

Resultados

Utilizando la interfaz gráfica proporcionada por Prefect, podemos ver los resultados de nuestros flujos.



Conclusión

Con este trabajo hemos podido observar cómo la librería Prefect facilita la creación y gestión de flujos de trabajo en Python, permitiendo a los desarrolladores automatizar procesos de manera eficiente y confiable. Mediante el uso de funciones de tareas decoradas con `@task`, se pueden definir unidades de trabajo independientes que forman parte de flujos de trabajo más amplios, los cuales se definen utilizando el decorador `@flow`.

Prefect ofrece características poderosas, como la gestión de errores, la configuración de reintento y la caché de resultados, que mejoran la robustez y eficiencia de los flujos de trabajo. Además, la integración de Prefect con herramientas de monitoreo y gestión de flujos de trabajo proporciona una visión completa del estado y rendimiento de los procesos automatizados.

Al usar Prefect, los equipos de desarrollo pueden simplificar el diseño, implementación y mantenimiento de flujos de trabajo complejos, lo que resulta en una mejora significativa en la productividad y calidad del software.