



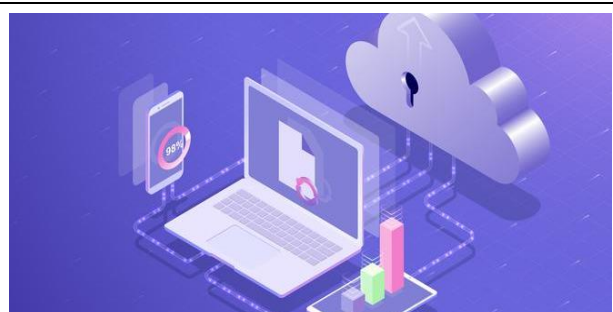
Universidad de Guadalajara

CUCEI – Centro Universitario de Ciencias Exactas e Ingenierías

Mtro. Michel Emanuel López Franco

Computación tolerante a fallas

Reporte 9



**Velasco Hernandez Victor
Manuel**

Código: 216598879

| D06 | 2022A |

| 07/Marzo/2022 |

Workflow





Universidad de Guadalajara
CUCEI
Computación tolerante a fallas

Victor Manuel Velasco Hernández
|Código: 216598879 | | 2022A |
|07/ Marzo /2022 | |D06|



Reporte 9- WorkFlow

Objetivo:

- - Parte 1.
Seguir el tutorial siguiente y ejecutarlo en tu maquina.
Getting Started with Prefect (PyData Denver)
<https://www.youtube.com/watch?v=FETN0iivZps&t=2545s>
- - Parte 2.
Modificarlo y Generar un ejemplo.

¿Que es Prefect?

Prefect es un sistema de gestión de flujo de trabajo(Workflow manager) basado en Python, facilita la adición de registros, reintentos, asignación dinámica, almacenamiento en caché, notificaciones de errores y más a sus canalizaciones de datos. [1]

Introducción:

El manejo de flujos de trabajo en un sistema de computo resultan ser una herramienta muy útil al momento de querer ejecutar una secuencia de procesos de manera programada, este tipo de herramientas hace posible que se corran procesos en segundo plano de una manera correcta, esto le brinda a los programadores potencialidad al momento de querer planificar procesos repetitivos, todo esto nos permite el estudio de estrategias para su implementación, lo cuál puede ayudar a mejorar la funcionalidad de los programas.

Desarrollo:

En esta actividad se va a crear un flujo de trabajo para generar spam en el correo electronico, esto será posible gracias a un flujo de trabajo y un script(código) que ejecutará.

Para comenzar, se desarrolla el código para el flujo de trabajo, en este caso, se le agregara al código del tutorial mencionado en los objetivos, solo para demostrar que se puede realizar muchas veces 1 o más tareas según lo agendado.



Se agrego la tarea llamada "spam()", la cuál hace que se envíe un mensaje de cierto destinatario a un establecido remitente.

```
schedule=IntervalSchedule(interval=datetime.timedelta(minutes=1))

with Flow("Mi Workflow-Victor Velasco", schedule) as f:
    db_table = create_table()
    raw = get_complaint_data()
    parsed = parse_complaint_data(raw)
    populated_table=store_complaints(parsed)
    populated_table.set_upstream(db_table)
    store_complaints(parsed)
    #subprocess.call("spamGenerator.py", shell=True)
    #exec(open("spamGenerator.py").read())
    spam()
```

La tarea spam tiene como objetivo ejecutar un script.

```
@task()
def spam():
    exec(open("spamGenerator.py").read())
```

El script ejecutado genera una conexión con el servicio SMTP, para poder enviar un mensaje por Gmail(en la imagen y repositorio, están censuradas las cuentas de correo y contraseña de aplicación, esto por motivos de seguridad).

```
1 import smtplib
2 import os
3
4 #Variable de entorno
5 PASSWORD_APP=os.environ.get('-')
6 EMAIL=os.environ.get('-')
7
8 #Establecer conexión al servidor de correos SMTP
9 conexion=smtplib.SMTP(host='smtp.gmail.com', port=587)
10 conexion.ehlo()
11
12 #Encriptacion TLS
13 conexion.starttls()
14
15 #Iniciar sesión en el servidor SMTP
16 conexion.login(user="-", password='-')
17
18 #Enviar correo
19 mensaje='Subject:Correo de Prueba\nEsto es una prueba'
20 conexion.sendmail(from_addr="-", to_addrs="-", msg=mensaje)
21
22 #Desconectar del servidor SMTP
23 conexion.quit()
```



Universidad de Guadalajara CUCEI Computación tolerante a fallas

Victor Manuel Velasco Hernández
|Código: 216598879 | | 2022A |
|07/ Marzo /2022 | |D06|

Una vez sabido lo anterior, solo queda ejecutar los comandos correspondientes para iniciar con el Workflow de Prefect.

Primero ejecutamos el servidor web, el cuál funciona gracias a Docker, esto con el comando: “prefect backend server” y “prefect server start --postgres-port 5430”

```
PS C:\Users\Victo\Desktop\Portafolio 2 Semestre\Sexto Semestre\Computacion tolerante\Computacion\ComputacionTolerante\4-Workflow> prefect agent local start
[2022-03-25 05:35:15,947] INFO - agent | Registering agent...
[2022-03-25 05:35:20,473] INFO - agent | Registration successful!

[2022-03-25 05:35:22,837] INFO - agent | Starting LocalAgent with labels ['DESKTOP-AG40UEU']
[2022-03-25 05:35:22,839] INFO - agent | Agent documentation can be found at https://docs.prefect.io/orchestration/
[2022-03-25 05:35:22,840] INFO - agent | Waiting for flow runs...
[2022-03-25 05:37:00,048] INFO - agent | Deploying flow run 3a23f0cc-0e5f-401d-a053-b1126c40cb8d to execution environment...
[2022-03-25 05:37:02,594] INFO - agent | Completed deployment of flow run 3a23f0cc-0e5f-401d-a053-b1126c40cb8d
[2022-03-25 05:38:00,010] INFO - agent | Deploying flow run 077f7292-3bfe-40fa-87de-1fd78d5934f to execution environment...
[2022-03-25 05:38:02,192] INFO - agent | Completed deployment of flow run 077f7292-3bfe-40fa-87de-1fd78d5934f
[2022-03-25 05:39:01,870] INFO - agent | Deploying flow run cadf545c-afc2-4cf4-bb74-8f3837705717 to execution environment...
[2022-03-25 05:39:04,835] INFO - agent | Completed deployment of flow run cadf545c-afc2-4cf4-bb74-8f3837705717
[2022-03-25 05:40:00,805] INFO - agent | Deploying flow run ba74b5e-375d-4334-ac7b-5839f2eb8cf5 to execution environment...
[2022-03-25 05:40:02,273] INFO - agent | Completed deployment of flow run ba74b5e-375d-4334-ac7b-5839f2eb8cf5
[2022-03-25 05:41:04,282] INFO - agent | Deploying flow run 4174fd03-28d9-49d2-b96b-cc0e808d90a to execution environment...
[2022-03-25 05:41:06,424] INFO - agent | Completed deployment of flow run 4174fd03-28d9-49d2-b96b-cc0e808d90a
[2022-03-25 05:42:00,011] INFO - agent | Deploying flow run 3aacfb9f-cf59-4aef-ace1-e2311102f814 to execution environment...
```

Después ejecutamos el comando “prefect create project 'Introduction’”, para crear el proyecto con el que planeamos trabajar.

```
PS C:\Users\Victo\Desktop\Portafolio 2 Semestre\Sexto Semestre\Computacion tolerante\Computacion\ComputacionTolerante\4-Workflow> prefect create project 'Introduction'
Introduction created
PS C:\Users\Victo\Desktop\Portafolio 2 Semestre\Sexto Semestre\Computacion tolerante\Computacion\ComputacionTolerante\4-Workflow> prefect create project 'Introduction'
Introduction created
PS C:\Users\Victo\Desktop\Portafolio 2 Semestre\Sexto Semestre\Computacion tolerante\Computacion\ComputacionTolerante\4-Workflow>
```

Y después ejecutamos el código del Workflow, el cuál será cargado al proyecto que creamos.

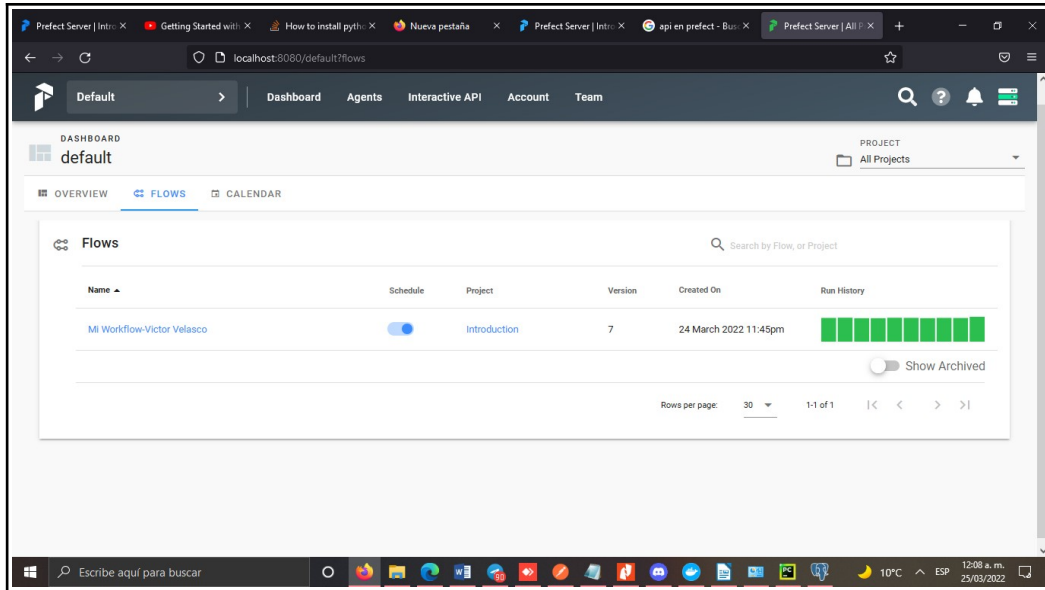
```
Run: pydata_denver_demo
C:\Users\Victo\AppData\Local\Programs\Python\Python38\python.exe "C:\Users\Victo\Desktop\Portafolio 2 Semestre\Sexto Semestre\Computacion tolerante\Computacion\ComputacionTolerante\4-Workflow\pydata_denver_demo.py"
Flow URL: http://localhost:8080/default/flow/a8fd7a90-9e87-4f21-95da-c7d70c0ee2ad0
ID: 65a4b4d0-53f9-42ca-aa85-9e4ecf5b2b7a
Project: Introduction
Labels: ['DESKTOP-AG40UEU']
Process finished with exit code 0
```



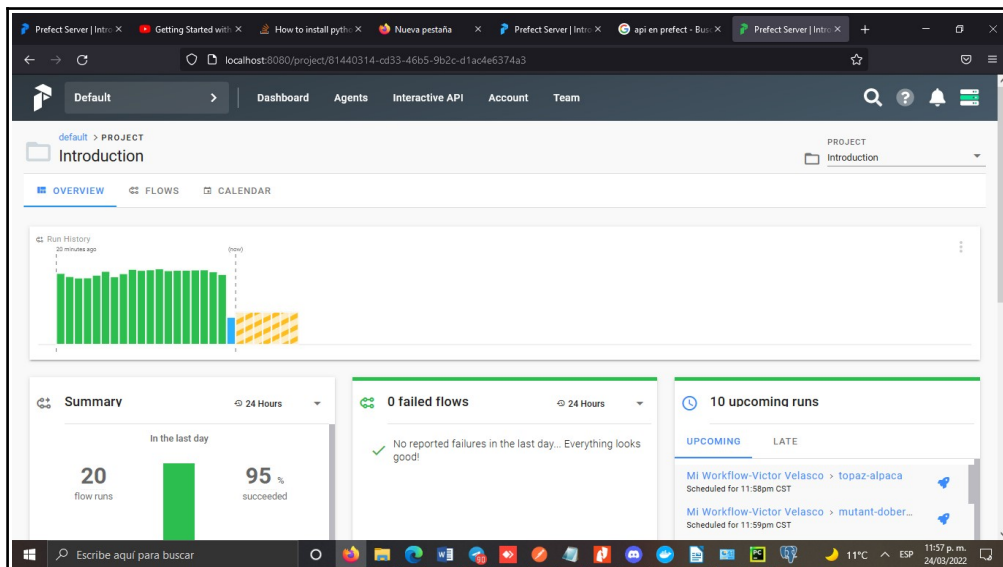
Universidad de Guadalajara
CUCEI
Computación tolerante a fallas

Victor Manuel Velasco Hernández
|Código: 216598879 | | 2022A |
|07/ Marzo /2022 | |D06|

Después de esto, nuestro flujo de trabajo ha sido cargado al proyecto, el cuál será mostrado en la pagina desplegada.



Como se podrá apreciar en la siguiente imagen, el proceso se está realizando correctamente, además de estar planeado y uno ejecutándose.





Universidad de Guadalajara
CUCEI
Computación tolerante a fallas

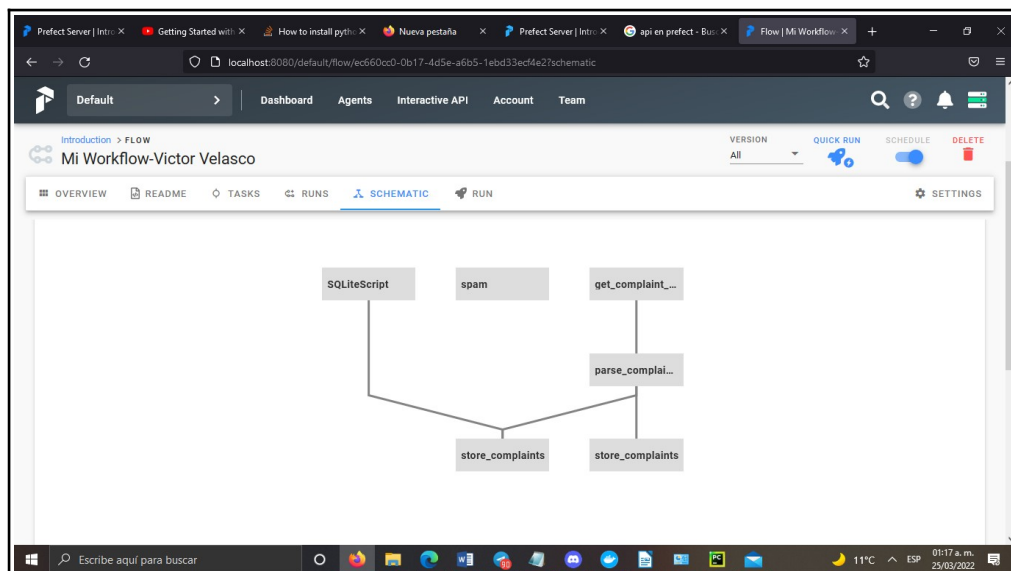
Victor Manuel Velasco Hernández
|Código: 216598879 | | 2022A |
|07/ Marzo /2022 | |D06|

Al seleccionar uno de los tantos procesos agendados, nos podemos dar cuenta de que las tareas que se realizan están listadas:

The screenshot shows the Prefect Server interface for a workflow named 'alluring-koala'. The 'Task Runs' section displays a table of executed tasks.

Task	Start Time	End Time	Duration	State
get_complaint_data	24 Mar 2022...	24 Mar 2022...	...	Completed
store_complaints (Parent)	24 Mar 2022...	24 Mar 2022...	2 seconds	Completed
store_complaints (Parent)	24 Mar 2022...	24 Mar 2022...	2 seconds	Completed
parse_complaint_data	24 Mar 2022...	24 Mar 2022...	2 seconds	Completed
SQLiteScript	24 Mar 2022...	24 Mar 2022...	2 seconds	Completed
spam	24 Mar 2022...	24 Mar 2022...	3 seconds	Completed

A continuación el grafo que describe el flujo de trabajo, como se podrán dar cuenta, “spam” se encuentra independiente de los otros procesos, lo cuál cumple con el propósito establecido.

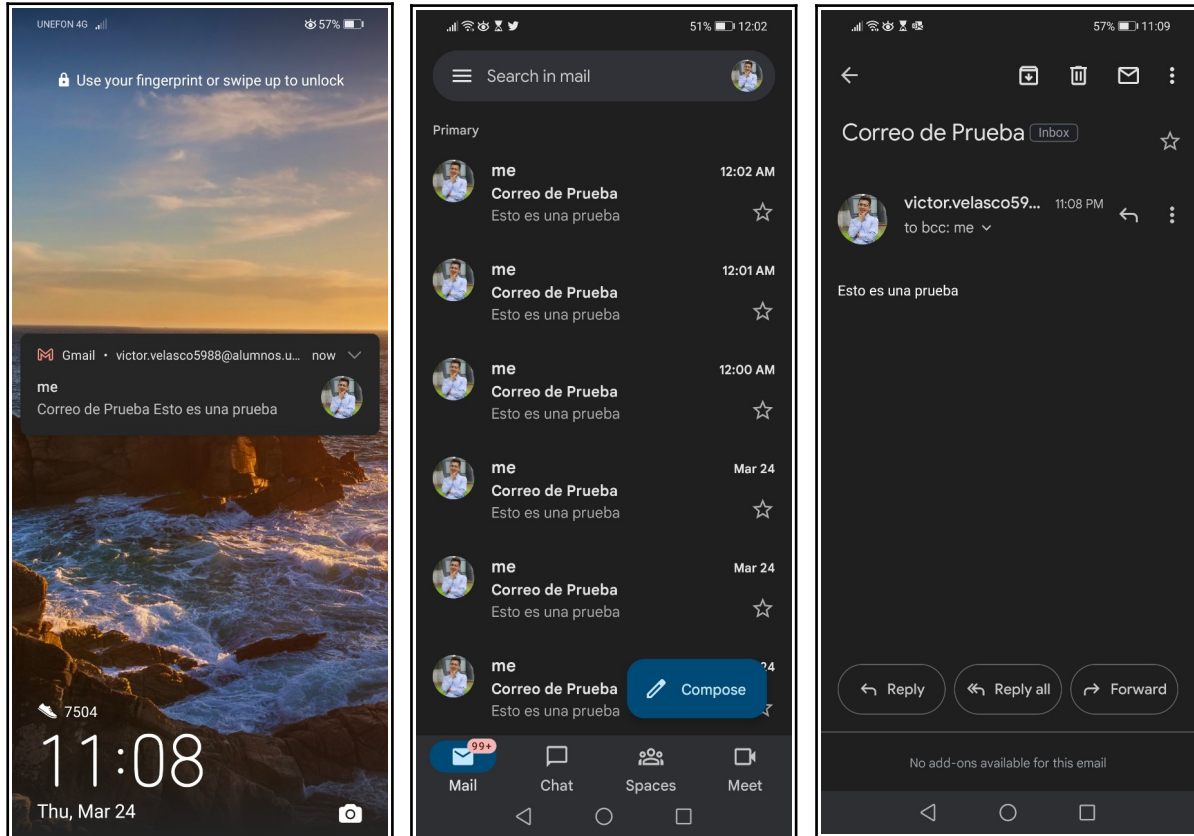




Universidad de Guadalajara
CUCEI
Computación tolerante a fallas

Victor Manuel Velasco Hernández
|Código: 216598879 | | 2022A |
|07/ Marzo /2022 | |D06|

A continuación, las evidencias de que los correos fueron enviados:



Conclusiones:

El uso de flujos de trabajo para hacer tareas repetitivas tiene un gran potencial como lo es el spam, realizar todo esto en segundo plano resulta ser muy útil para realizar procesos sin preocuparnos por estarlos ejecutando cada vez que queramos, con los Workflows, basta con que agendemos el tiempo que queramos, en caso de que se presenten fallas, se reportara en nuestra interfaz, en este caso, no se reporto nada porque el ejemplo, lo cuál demuestra la programación negativa en práctica, lo que es esencial para hacer un sistema tolerante a fallos, esto resulta ser fundamental para conocer la naturaleza de este tipo de herramienta para desarrollar cada vez, mejores flujos de trabajo que garanticen la calendarización y ejecución de los procesos, lo cuál ayuda a tener una mejor comprensión de la lógica



Universidad de Guadalajara
CUCEI
Computación tolerante a fallas

Victor Manuel Velasco Hernández
|Código: 216598879 | | 2022A |
|07/ Marzo /2022 | |D06|

del Workflow que se ejecuta en nuestras computadoras o clusters, esto hace que pensemos en ideas de aprovechamiento para el futuro.

Link de repositorio:

<https://github.com/Victor012396/ComputacionTolerante.git>

Bibliografía:

- [1] D. Radečić, "Prefect: How to Write and Schedule Your First ETL Pipeline with Python", Medium, el 22 de julio de 2021. <https://towardsdatascience.com/prefect-how-to-write-and-schedule-your-first-etl-pipeline-with-python-54005a34f10b> (consultado el 18 de marzo de 2022).
- [2] PyData, Task Failed Successfully - Jeremiah Lowin, (2019). Consultado: el 18 de marzo de 2022. [En línea]. Disponible en: https://www.youtube.com/watch?v=TlawR_gi8-Y
- [3] Prefect, Getting Started with Prefect (PyData Denver), (2020). Consultado: el 18 de marzo de 2022. [En línea]. Disponible en: <https://www.youtube.com/watch?v=FETN0iivZps>
- [4] Tiempo, Enviar correos con Python usando smtplib, (el 14 de noviembre de 2020). Consultado: el 25 de marzo de 2022. [En línea]. Disponible en: <https://www.youtube.com/watch?v=kVKagJrZCyU>