



UNIVERSIDAD AUTÓNOMA DE  
**CHIHUAHUA**

UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA  
Facultad de Ingeniería



Ingeniería en Ciencias de la Computación

## **COMPUTO PARALELO Y DISTRIBUÍDO**

### **1.18 Actividad 5: Threads y Callbacks**

*Trabajo de:* ADRIAN A. GONZÁLEZ DOMÍNGUEZ [359834]  
*Asesora:* JOSE SAUL DE LIRA MIRAMONTES

*6 de septiembre de 2024*

```

import logging
import requests
import threading
import random
import time

logging.basicConfig(level=logging.INFO, format='%(message)s')

def get_pokemon_params():
    pokemon_id = random.randint(1, 1000)
    return f'{pokemon_id}'

def get_pokemon_name(response_json):
    name = response_json.get('forms')[0].get('name')
    logging.info(f'El nombre del pokemon es : {name}')

def get_name_random(response_json):
    name = response_json.get('results')[0].get('name').get('first')
    logging.info(f'El nombre del usuario es : {name}')

def error(status_code):
    logging.error(f'No es posible completar la operación. Error: {status_code}')

def generate_request(url, success_callback, error_callback,
get_params=lambda: ''):
    params=get_params()
    url_with_params=f'{url}/{params}'
    # logging.info(url_with_params)
    response = requests.get(url_with_params)
    if response.status_code == 200:
        success_callback(response.json())
    else:
        error_callback(response.status_code)

def thread_request():
    threads = []
    for _ in range(2): # 5 requests simultáneas
        thread1 = threading.Thread(
            target=generate_request,
            kwargs={'url': 'https://pokeapi.co/api/v2/pokemon',
                    'get_params': get_pokemon_params,
                    'success_callback': get_pokemon_name,
                    'error_callback': error
            })

```

```

    )
    threads.append(thread1)
    thread1.start()
    thread2 = threading.Thread(
        target=generate_request,
        kwargs={'url': 'https://randomuser.me/api',
                'success_callback': get_name_random,
                'error_callback': error
               }
    )
    threads.append(thread2)
    thread2.start()

for thread in threads:
    thread.join()

if __name__ == '__main__':
    for _ in range(5):
        thread_request()
        time.sleep(1)

```

```

thead@adrigondo MINGW64 ~/Documents/UACH/Seventh Semester/Parallel and Distributing Computing/code
$ python PDC.September\ 6\, \ 2024.py
El nombre del pokemon es : pawmi
El nombre del pokemon es : lillipup
El nombre del usuario es : Nellie
El nombre del usuario es : Enzo
El nombre del pokemon es : runerigus
El nombre del pokemon es : maushold-family-of-four
El nombre del usuario es : Eileen
El nombre del usuario es : Soan
El nombre del pokemon es : drifloon
El nombre del pokemon es : snorlax
El nombre del usuario es : Arlo
El nombre del usuario es : Pascale
El nombre del pokemon es : applin
El nombre del pokemon es : sheldler
El nombre del usuario es : Kurt
El nombre del usuario es : Christian
El nombre del pokemon es : sliggoo
El nombre del pokemon es : articuno
El nombre del usuario es : Else
El nombre del usuario es : Magnus

```