

# DOCUMENTACION SERVERLESS / BACKEND

## PRACTICA I



### PROYECTO DE COMPUTACION III

ANGEL ESCOBAR ANCHUELO


En este proyecto de Google Cloud, subiremos una “function” que permita descargar noticias de categorías desconocidas


Mas adelante entrenaremos un modelo que permita clasificarlos

Observamos que el profesor este agregado como propietario tanto en GC como en el repo

sukuzhanay@gmail.com

Propietario 





Manage access 


Add people

☐ Select all

Type ▾

 Find a collaborator...

☐  **Christian Vladimir Sucuzhanay Arévalo**  
Awaiting sukuzhanay's response

Pending Invite 

Remove

## CODIGO .PY

```
def hello(requests):
    listaCategorias = ['https://www.elmundo.es/ciencia-y-salud/salud.html',
                       'https://www.elmundo.es/ciencia-y-salud/ciencia.html',
                       'https://elpais.com/noticias/delitos-odio/']
    nombreCategoria = ['salud', 'ciencia', 'odio']
    contador = 0
```

Primero declaramos links para extraer noticias

```
for link in listaCategorias:
    rMundo = req.get(link)
    soup_Mundo = BeautifulSoup(rMundo.text, 'html.parser')
    Enlaces_Noticias = []
    if contador == 2:
        ElMundoBloqueEntero = soup_Mundo.find('div', 'b-b b-au_b')
        ElMundo_odio_pagina = ElMundoBloqueEntero.find_all('article')
        for i in ElMundo_odio_pagina:
            enlace = "https://elpais.com" + (((i.find('header')).find('h2', 'c_t')).find('a'))["href"]
            print(enlace)
            print("\n")
            Enlaces_Noticias.append(enlace)
            pasar_datos_ficheros_Odio(Enlaces_Noticias, './webScraping/' + nombreCategoria[contador] + '/',
                                      nombreCategoria[contador])
        else:
            ElMundo_noticias = soup_Mundo.find_all('div', 'ue-l-cover-grid__unit ue-l-cover-grid__unit--no-grow')

            for noticia in ElMundo_noticias:
                atag = noticia.find("a")
                atag["href"] = atag["href"]
                print(atag["href"])
                Enlaces_Noticias.append(atag["href"])

            pasar_datos_ficheros(Enlaces_Noticias, './webScraping/NoOdio/', nombreCategoria[contador])
            contador = contador + 1
```

Hacemos cortes en el HTML para extraer los links de cada noticia

```
def crear_ficheros_noticiasMundoOdio(enlace, contador, ruta, catego):
    parrafos = ""
    fecha = datetime.today().strftime('%Y-%m-%d')
    url_enlaces = req.get(enlace)
    soup_enlaces = BeautifulSoup(url_enlaces.text, 'html')

    try:
        Titulo = (soup_enlaces.find('div', 'a_e_txt_df')).find('h1', 'a_t').getText()
    except:
        Titulo = ""

    try:
        Entradilla = (soup_enlaces.find('div', 'a_e_txt_df')).find('h2', 'a_st').getText()
    except:
        Entradilla = ""

    try:
        Body = (soup_enlaces.find('div', 'a_c_clearfix')).find_all('p')
        for p in Body:
            parrafos = parrafos + p.getText()
    except:
        parrafos = ""

    try:
        LugarYFecha = (soup_enlaces.find('article', 'a_g_g-lg_g-o')).find('div', 'a_md_txt')).find('span').getText()
    except:
        LugarYFecha = ""
```

Y esta imagen genera los txt con los apartados

## En el Google Cloud creamos una Cloud Function

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Entorno	Nombre ↑	Región	Activador	Tiempo de ejecución	Memoria asignada	Función ejecutada	Última implementación	Authentication ⓘ	Acciones
<input type="checkbox"/>	<input checked="" type="checkbox"/>	1st gen	scraper3	europe-west1	HTTP	Python 3.9	256 MB	hello	23 mar. 2022 21:02:30		⋮

La configuración del cloud function es

✓ **scraper3** 1st gen  
europe-west1

### Activador

**⌚ HTTP**

Tipo de activador

HTTP

**URL**

https://europe-west1-practica1pc3-21841972.cloudfunctions.net/scraper3

☒ El HTTPS es obligatorio ⓘ

**GUARDAR** **CANCELAR**

Y el código que está ejecutando es el scraper.py del repositorio

Entorno de ejecución  
Python 3.9 ⓘ

Punto de entrada \*  
hello ⓘ

Código fuente

Editor directo

+

main.py ...


requirements.txt


```
1  ## IMPORTS
2  #import firebase_admin
3  #import functions_framework
4  import requests as req
5  from bs4 import BeautifulSoup
6  from datetime import datetime
7
8  ##VARIABLES
9
10 #functions_framework.http
11 def hello(requests):
```

Creamos un scheduler que ejecute la function viernes 8:30

<input type="checkbox"/>	Nombre ↑	Región	Estado	Descripción	Frecuencia	Destino	Última ejecución	Resultado de la última ejecución	Próxima publicaci
<input type="checkbox"/>	Scrapper	europa-west1	Habilitado	Ejecuta el código y extrae noticias	30 8 * * 5 (Europe/Madrid)	URL : https://europe-west1-practica1pc3-21841972.cloudfunctions.net/scrapper3	23 mar. 2022 21:20:16	Aún no se ha ejecutado	25 mar. 2022 08

La configuracion del scheduler

 Cloud Scheduler

 Scrapper

- Define el programa

Región

europa-west1

Descripción

Ejecuta el código y extrae noticias

Frecuencia \*

30 8 \* \* 5

Las programaciones se especifican con el formato cron de Unix. P. ej., cada minuto: `* * * * *`, cada 3 horas: `"0 */3 * * *`, todos los lunes a las 9:00 a.m.: `"0 9 * * 1"`. [Más información](#)

Zona horaria \*

hora estándar de Europa central (ØØØ) ▼

Los trabajos establecidos en zonas horarias afectadas por el horario de verano pueden ejecutarse fuera de la cadencia durante el cambio a DST. Si usas una zona horaria UTC, puedes evitar el problema. [Más información](#)

CONTINUAR

Link de url:

<https://europe-west1-practica1pc3-21841972.cloudfunctions.net/scrapper3>

Link del repositorio

<https://github.com/SecretoAngel/PC3>