



UNIVERSIDAD DE GUADALAJARA  
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS  
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

# Docker

Oscar Evanilson Gutiérrez Pérez

Carrera: Ingeniería en Computación

Código de estudiante: 219748308

21 de marzo de 2022

En este programa, utilicé el lenguaje de Python para hacer un ejemplo de un programa Python en Docker, es un ejemplo sencillo donde se lee un archivo y se escoge una letra, y devuelve cuantas veces aparece esa letra, todo esto siendo ejecutado en Docker.

```
1 import os
2 import argparse
3
4 name = os.getenv('BOT_NAME') or 'Generic Bot'
5
6 parser=argparse.ArgumentParser()
7 parser.add_argument("input", help = "Input File Path")
8 parser.add_argument("character", help = "Character to count")
9 args = parser.parse_args()
10
11 input = (open(args.input, encoding='utf8')).read()
12
13 occurences = input.count(args.character)
14 print('Hello I am {name}!'.format(name=name))
15 print('Your specified character appears {occurences} time(s)!'.format(occurences=occurences))
```

En esta primera parte del ejemplo vemos el programa de Python que se va a correr con Docker, lo que hace este programa es solicitar al momento de correr en Docker, el nombre del archivo a leer y la letra que se quiere buscar en este mismo.

Además, al final de este se devuelve un pequeño texto que indica el total de veces que aparece dicha letra en el archivo indicado anteriormente

```
# specify start image
FROM python

# all commands start from this directory
WORKDIR /count-bot

# copy all files from this folder to working directory (ignores files in .docker
ignore)
COPY . .

# set a default environment variable for the name of your bot
ENV BOT_NAME='Docker Bot'
```

```
# set the start command  
CMD [ "python3", "count-bot.py" , "./test.txt", "n"]
```

En esta parte del ejemplo, vemos las líneas de código con las que se creará el archivo Docker y se va a correr para ejecutar el ejemplo de Python anterior.

Vemos que se crea una imagen para obtener el archivo de Python que ya creamos, después se crea un bot de Docker que se va a encargar de ejecutar correctamente el archivo de Python y finalmente se incluyen los parámetros que requiere este archivo, que es el nombre del archivo a leer y la letra.

Link de supositorio en GitHub para revisar el código:

<https://github.com/oscarevanilson/Computacion-Tolerante-a-Fallas>

## Conclusiones

Este programa me resultó un poco complicado, ya que al inicio de la investigación que realicé para entender esta actividad no conocía nada sobre Docker, ni como funcionaba ni nada acerca de él, lo bueno es que gracias a algunos vídeos y de algunos artículos que me puse a leer pude comprender lo necesario para realizar un pequeño ejemplo.

Creo que no hay aún mucha información sobre Docker pero es más simple de entender y de implementar que los ejercicios y librerías anteriores ya que estos se me hicieron demasiado complicados y no termine de comprenderlos.

Espero que con el paso de tareas y de temas que veamos vayamos comprendiendo poco a poco, más acerca de todos estos temas que estamos revisando en las últimas semanas de clase.

Por último, espero que se avance el desarrollo de todas estas herramientas para que en un futuro sea más fácil de utilizar y de comprender, además de que se puedan hacer más cosas con estas y que pueda ayudar a muchos programadores

en proyectos importantes o incluso a estudiantes como nosotros en sus tareas y proyectos.