

UNIVERSIDAD COMPLUTENSE DE MADRID

DESPLIEGUE DE LA APLICACIÓN EN AWS

ESTUDIANTE

ALEJANDRO TUY FAJARDO

PASAPORTE

PA0845133

ASIGNATURA

TRABAJO FIN DE MÁSTER

PROFESORES

CARLOS ORTEGA Y SANTIAGO MOTA

MAESTRÍA

MÁSTER BIG DATA Y DATA SCIENCE

PERIODO

2021-2022



UNIVERSIDAD
COMPLUTENSE
MADRID

CONTENIDO

INTRODUCCIÓN

El presente documento es un anexo del TFM, con el fin de no aportar contenido excesivo, pero sin dejar de presentar la evidencia e información necesaria para el trabajo elaborado. Implementé una aplicación en AWS utilizando el Framework Open Source Streamlit. e

CÓDIGO

```
1  import streamlit as st
2  import os
3  import cv2
4  import numpy as np
5  from keras.models import load_model
6  from PIL import Image
7
8  process_dir = os.getcwd()
9
10 def load_image(image_file):
11     img = Image.open(image_file)
12     return img
13
14 def chest_sickness_predict(image):
15     if "png" in str(image):
16         im = Image.open(image)
17         rgb_im = im.convert('RGB')
18         img2 = os.path.join(process_dir, "test.jpeg")
19         rgb_im.save(img2)
20
21     else:
22         img1 = Image.open(image)
23         img2 = os.path.join(process_dir, "test.jpeg")
24         img1.save(img2)
25
26     img = cv2.imread(img2)
27     chest_sickness_model = load_model('medical_trial_model.h5')
28     print('X-Ray Model->', chest_sickness_model)
29     test_image = cv2.resize(img,(224,224))
30     test_image = np.array(test_image)
31     test_image = np.expand_dims(test_image, axis= 0)
32     test_image = test_image/255
33     result = chest_sickness_model.predict(test_image)
34
35     if result[0][0] > 0.8:
36         print(result[0][0])
37         print(result)
38         print(result[0])
39         prediction = "COVID"
40         confidence_score = result[0][0] * 100
41         print(f'confidence score de covid -> {confidence_score}')
42     elif result[0][1] > 0.8:
43         print(result[0][1])
44         print(result)
45         print(result[0])
46         prediction = "NORMAL"
47         confidence_score = result[0][1] * 100
48         print(f'confidence score de normal -> {confidence_score}')
```

```

49 elif result[0][2] > 0.8:
50     print(result[0][0])
51     prediction = "VIRUS RESPIRATORIO"
52     confidence_score = result[0][2] * 100
53     print(f'confidence score de virus respiratorio ->
54           {confidence_score}')
55     return prediction, confidence_score
56
57 def main():
58     st.title('**CT X-Rays Predictions - Alejandro Tuy**')
59     html_temp = """
60     <div style="background-color:red;padding:10px; text-align:center">
61     <h2>Predicción de Pulmon</h2>
62     """
63     st.markdown(html_temp, unsafe_allow_html = True)
64     filename = st.file_uploader("SUBE LA IMAGEN CT",
65                                type=['jpg', 'png', 'jpeg', 'gif'])
66
67     if ((filename is not None) and ('jpg' in str(filename))):
68         st.image(load_image(filename))
69         prediction, confidence_score =
70             chest_sickness_predict(filename)
71         confidence_score = str(confidence_score)
72     elif ((filename is not None) and ('png' in str(filename))):
73         st.image(load_image(filename))
74         prediction, confidence_score =
75             chest_sickness_predict(filename)
76         confidence_score = str(confidence_score)
77     elif ((filename is not None) and ('jpeg' in str(filename))):
78         st.image(load_image(filename))
79         prediction, confidence_score =
80             chest_sickness_predict(filename)
81         confidence_score = str(confidence_score)
82     elif ((filename is not None) and ('gif' in str(filename))):
83         st.image(load_image(filename))
84         prediction, confidence_score =
85             chest_sickness_predict(filename)
86         confidence_score = str(confidence_score)
87     else:
88         pass
89     result=""
90     if st.button("Predicción"):
91         st.success(f'Predicción: {prediction}')
92         st.success(f'Probabilidad del Arrojada por el Modelo:
93                   {confidence_score} %')
94 if __name__ == '__main__':
95     main()

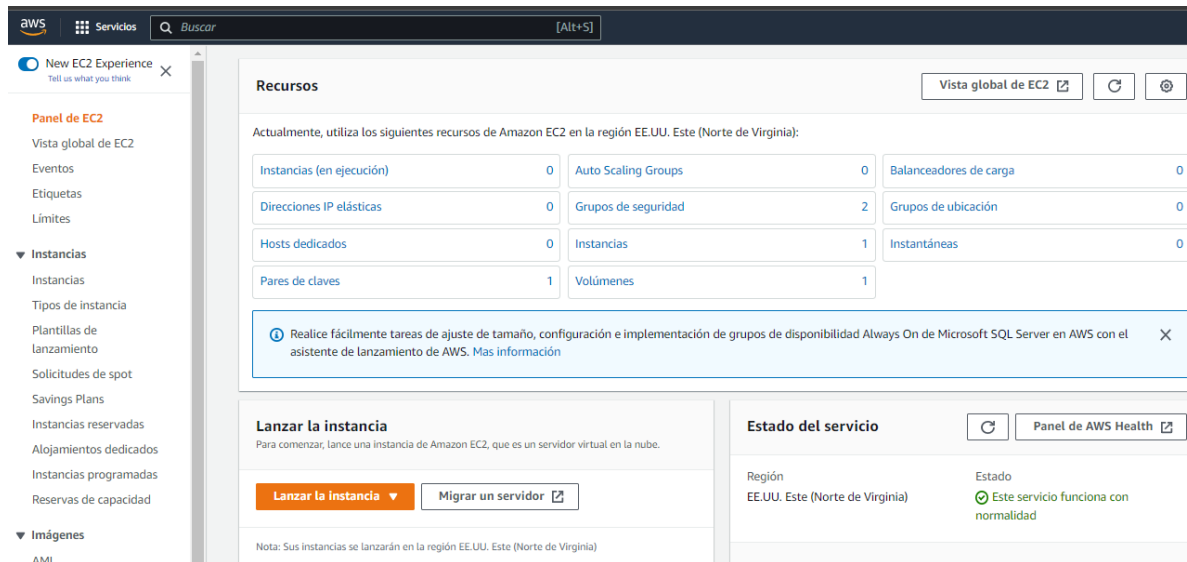
```

TRABAJO REALIZADO

1. Creé una imagen en el servicio de AWS EC2



2. Lancé una Instancia t2.micro que incluye 750 horas de uso al mes, y se encuentra dentro de la capa gratuita.



▼

Resumen

Número de instancias

información

1

Imagen de software (AMI)

Amazon Linux 2 Kernel 5.10 AMI...más información

ami-0aa7d40eeae50c9a9

Tipo de servidor virtual (tipo de instancia)

t2.micro

Firewall (grupo de seguridad)

Nuevo grupo de seguridad

Almacenamiento (volumenes)

1 volumen(es): 8 GiB

?

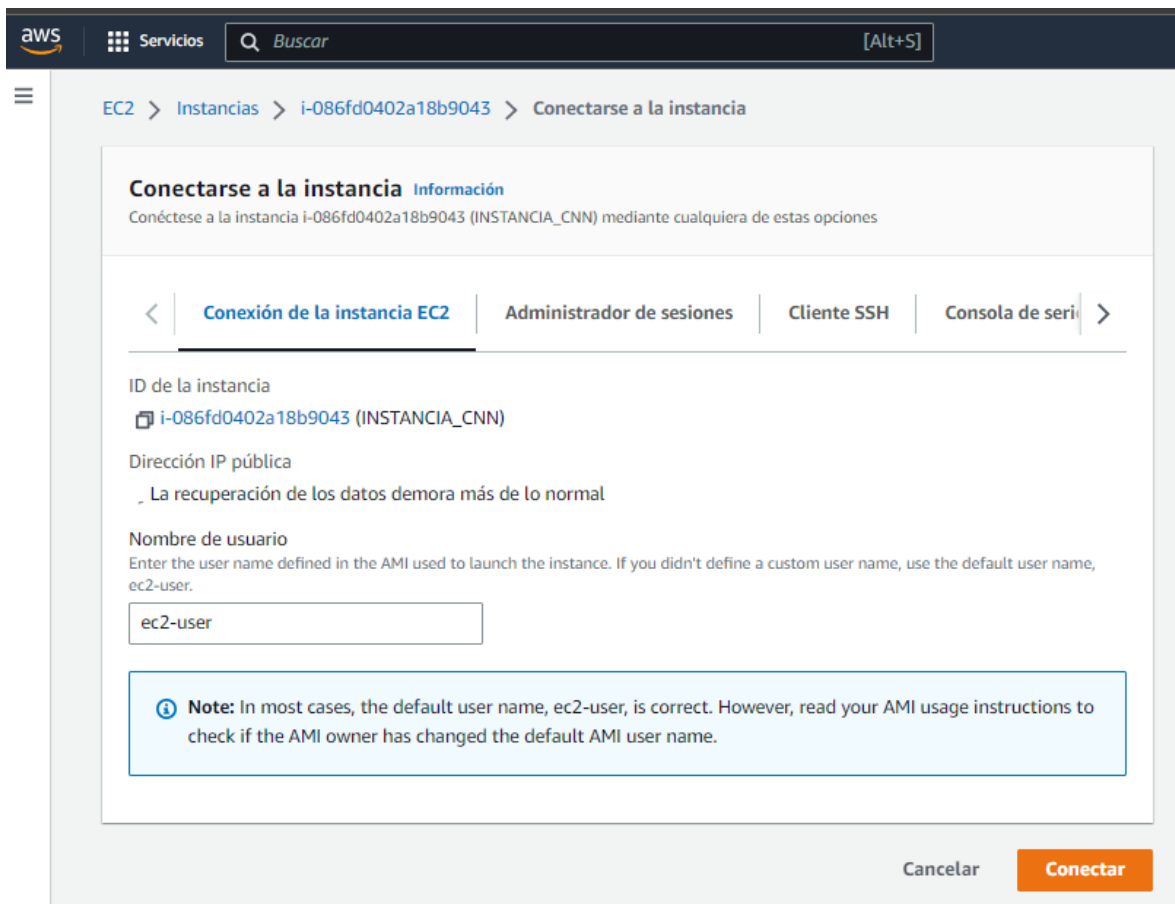
Nivel gratuito: El primer año incluye 750 horas de uso de instancias t2.micro (o t3.micro en las regiones en las que t2.micro no esté disponible) en las AMI del nivel gratuito al mes, 30 GiB de almacenamiento de EBS, 2 millones de E/S, 1 GB de instantáneas y 100 GB de ancho de banda a Internet.

×

Cancelar

Lanzar instancia

3. Me conecté a la instancia mediante Terminal en “modo directo” gestionarlo de manera remota.



La gestión de la instancia es a través de una línea de comando de Linux

```
Last login: Mon Feb 13 08:13:04 2023 from ec2-18-206-107-29.compute-1.amazonaws.com

  _ | _ | _ |
 _ | ( _ | /   Amazon Linux 2 AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-52-92 ~]$
```

4. Elevo los privilegios como **usuario root**, para tener todos los permisos e instalar todo lo que necesito, con el comando `sudo -i`

```
aws Servicios Buscar
[ec2-user@ip-172-31-52-92 ~]$ sudo -i
[root@ip-172-31-52-92 ~]#
```

5. Instalao git para poder descargar el repositorio de GitHub donde tengo el código fuente de la aplicación con el comando **yum install git**

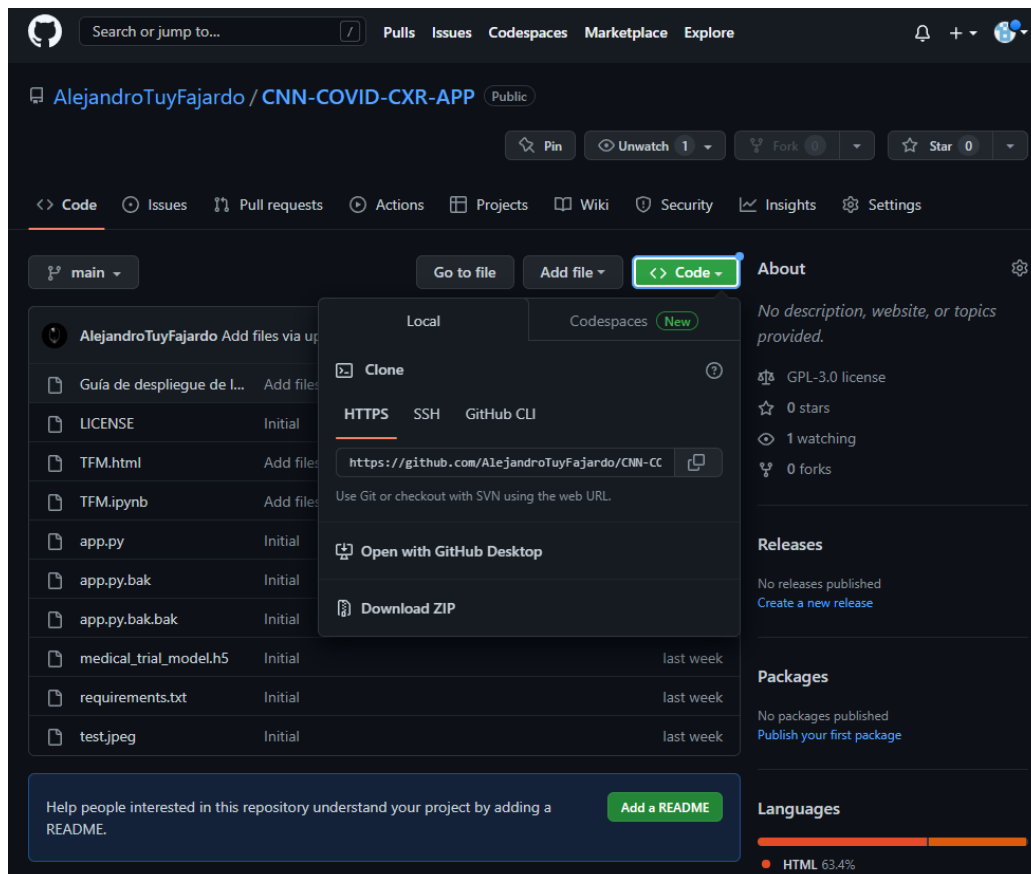
```
[root@ip-10-251-0-242 ~]# yum install git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.32.0-1.amzn2.0.1 will be installed
--> Processing Dependency: perl-Git = 2.32.0-1.amzn2.0.1 for package: git-2.32.0-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core-doc = 2.32.0-1.amzn2.0.1 for package: git-2.32.0-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core = 2.32.0-1.amzn2.0.1 for package: git-2.32.0-1.amzn2.0.1.x86_64
--> Processing Dependency: emacsfilesystem >= 27.1 for package: git-2.32.0-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.32.0-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git::118N) for package: git-2.32.0-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git) for package: git-2.32.0-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package emacsfilesystem.noarch 1:27.2-4.amzn2.0.1 will be installed
--> Package git-core.x86_64 0:2.32.0-1.amzn2.0.1 will be installed
--> Package git-core-doc.noarch 0:2.32.0-1.amzn2.0.1 will be installed
--> Package perl-Git.noarch 0:2.32.0-1.amzn2.0.1 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.32.0-1.amzn2.0.1.noarch
--> Package perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-Error.noarch 1:0.17020-2.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

Package Arch Version Repository Size
Installing:
git x86_64 2.32.0-1.amzn2.0.1 amzn2-core 126 k
Installing for dependencies:
emacsfilesystem noarch 1:27.2-4.amzn2.0.1 amzn2-core 67 k
git-core x86_64 2.32.0-1.amzn2.0.1 amzn2-core 4.8 M
git-core-doc noarch 2.32.0-1.amzn2.0.1 amzn2-core 2.7 M
perl-Error noarch 1:0.17020-2.amzn2 amzn2-core 32 k
perl-Git noarch 2.32.0-1.amzn2.0.1 amzn2-core 43 k
perl-TermReadKey x86_64 2.30-20.amzn2.0.2 amzn2-core 31 k

Transaction Summary
Install 1 Package (+6 Dependent packages)
```

5.1 Copio el enlace HTTPS del repositorio de GitHub



5.2 Corro el comando Git clone + la url del repositorio y de descarga el repositorio en la EC2

```
aws
Servicios
Q Buscar [Alt+S]

last login: Thu Feb 16 06:23:48 2023 from ec2-18-206-107-27.compute-1.amazonaws.com

_ _ _ _ _
| | | | |
_ _ _ _ _ Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-52-92 ~]$ git clone https://github.com/AlejandroTuyPajardo/CNN-COVID-CXR-APP.git
```

```
aws
Servicios
Q Buscar [Alt+S]

[root@ip-172-31-52-92 ~]# ls
Convolutional-Neural-Network prueba
[root@ip-172-31-52-92 ~]# cd Convolutional-Neural-Network/
[root@ip-172-31-52-92 Convolutional-Neural-Network]# ls
app.py app.py.save app.py.save.1 LICENSE medical_trial_model.h5 model.h5 __pycache__ requirements.txt test.jpeg
[root@ip-172-31-52-92 Convolutional-Neural-Network]#
```

6. Instalé las dependencias de Python necesarias con el comando `python3 -m pip install [Libreria]`

```
aws
Servicios
Q Buscar [Alt+S]

[root@ip-172-31-52-92 Convolutional-Neural-Network]# python3 -m pip install pysentimiento
WARNING: Running pip install with root privileges is generally not a good idea. Try 'python3 -m pip install --user' instead.
Collecting pysentimiento
  Downloading pysentimiento-0.4.2-py3-none-any.whl (30 kB)
Collecting transformers==4.13
  Downloading transformers-4.13.0-py3-none-any.whl (3.3 MB)
    | 3.3 MB 22.8 MB/s
Collecting datasets<2.0.0,>=1.13.3
  Downloading datasets-1.18.4-py3-none-any.whl (312 kB)
    | 312 kB 34.3 MB/s
Collecting emoji<2.0.0,>=1.6.1
  Downloading emoji-1.7.0.tar.gz (175 kB)
    | 175 kB 37.0 MB/s
Collecting sklearn<0.1,>=0.0
  Downloading sklearn-0.0.post1.tar.gz (3.6 kB)
Collecting torch<2.0.0,>=1.9.0
  Downloading torch-1.13.1-cp37m-manylinux1_x86_64.whl (887.5 MB)
    | 824.9 MB 691 bytes/s eta 1 day, 1:09:37
```

7. Inicializo la aplicación corriendo el comando `python3 -m streamlit run app.py` para que ejecutar el archivo principal de la aplicación streamlit.

```
aws
Servicios
Q Buscar [Alt+S]

[root@ip-172-31-52-92 ~]# cd Convolutional-Neural-Network/
[root@ip-172-31-52-92 Convolutional-Neural-Network]# ls
app.py app.py.save app.py.save.1 LICENSE medical_trial_model.h5 model.h5 __pycache__ requirements.txt test.jpeg
[root@ip-172-31-52-92 Convolutional-Neural-Network]# python3 -m streamlit run app.py

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

You can now view your Streamlit app in your browser.

Network URL: http://172.31.52.92:8501
External URL: http://18.207.3.236:8501
```


7.1 Acceso mediante la **External Url** (ip y puerto) para que cualquier persona pueda acceder desde internet.

Grupos de seguridad (1/1) [Información](#)

🔄

Acciones ▾

Exportar los grupos de seguridad a CSV ▾

Crear grupo de seguridad

🔍 Filtrar grupos de seguridad

< 1 > ⚙

Nombre del grupo de seguridad: launch-wizard-1 ✕

Quitar los filtros

<input checked="" type="checkbox"/>	Name ▾	ID del grupo de segu... ▾	Nombre del grupo ... ▾	ID de la VPC ▾	Descripción ▾	Propietario ▾	Número de reglas ... ▾	Número de reglas de salida ▾
<input checked="" type="checkbox"/>	-	sg-0c56155edfa0e3ea0	launch-wizard-1	vpc-0f805fe0d4b77eado 🔗	launch-wizard created ...	950220985811	3 Entradas de permisos	1 Entrada de permiso

Detalles

Reglas de entrada

Reglas de salida

Etiquetas

📘 Ahora puede comprobar la conectividad de red con Reachability Analyzer

Ejecutar Reachability Analyzer ✕

Reglas de entrada (3)

🔄

Administrar etiquetas

Editar reglas de entrada

🔍 Filtrar reglas de grupo de seguridad

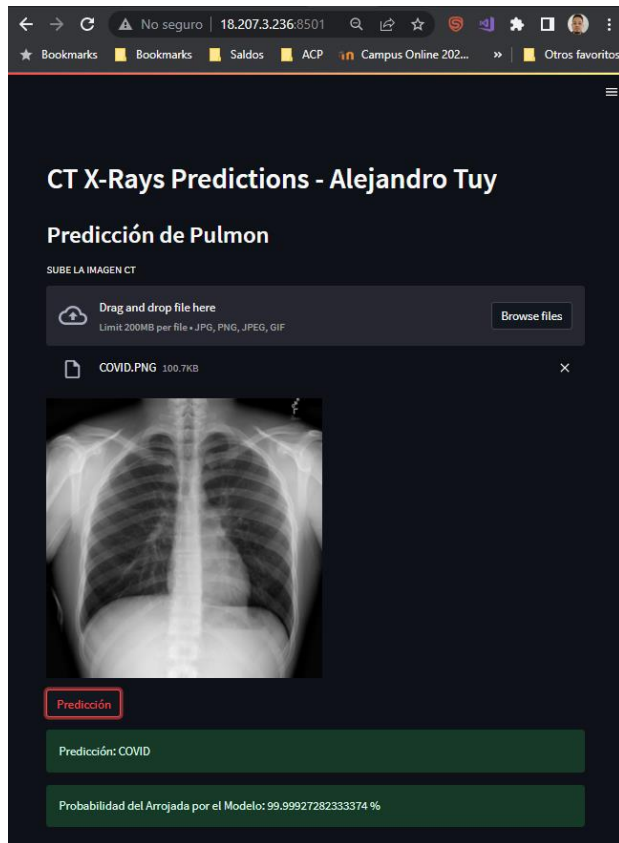
< 1 > ⚙

<input type="checkbox"/>	Name ▾	ID de la regla del g... ▾	Versión de IP ▾	Tipo ▾	Protocolo ▾	Intervalo de puertos ▾	Origen ▾	Descripción ▾
<input type="checkbox"/>	-	sgr-Q2482e90ef1071f38	IPv4	TCP personalizado	TCP	8501 ✓	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-09447e7885e890...	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0eeacbd4adf6d678	IPv4	HTTP	TCP	80	0.0.0.0/0	-

7.2 Se observa la aplicación en la ip asignada.

<http://18.207.3.236:8501>

COVID



NORMAL



VIRUS

