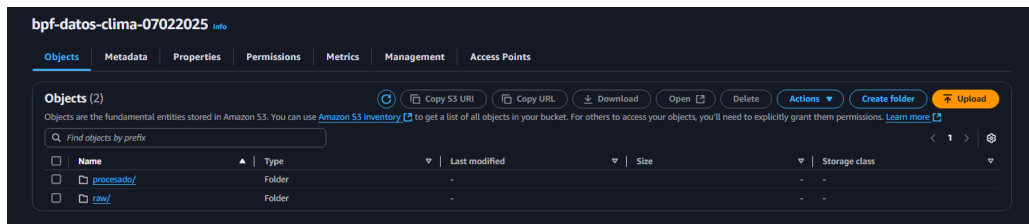


Proceso ETL usando AWS

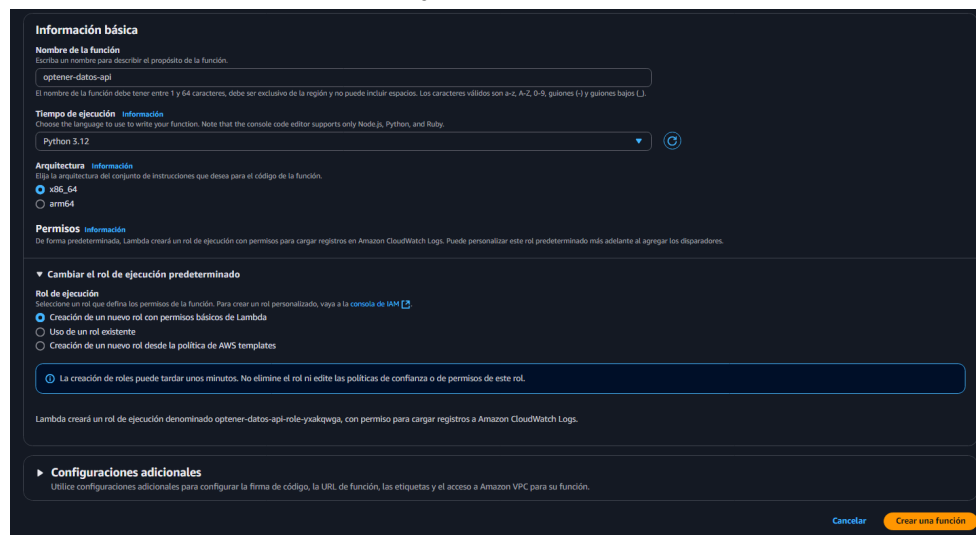
S3

1. Iniciamos sesión en la consola de aws.
2. Abrimos el servicio de S3.
3. Creamos el bucket con el nombre bpf-datos-clima-070222052.
 - a. Dejamos las configuraciones por defecto y damos en crear.
4. Dentro de nuestro bucket creamos dos carpetas con los nombres “raw/” y “procesado/”.



LAMBDA

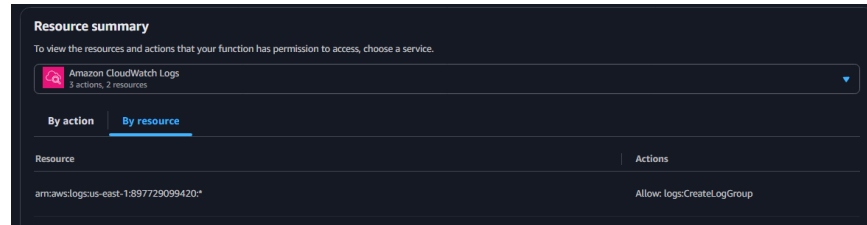
1. Abrimos nuestro servicio de lambda y damos en crear funcion
 - a. Damos el nombre “obtener-datos-api”
 - b. Seleccionamos un entorno de ejecución python 3.12
 - c. Creamos un nuevo rol de ejecución



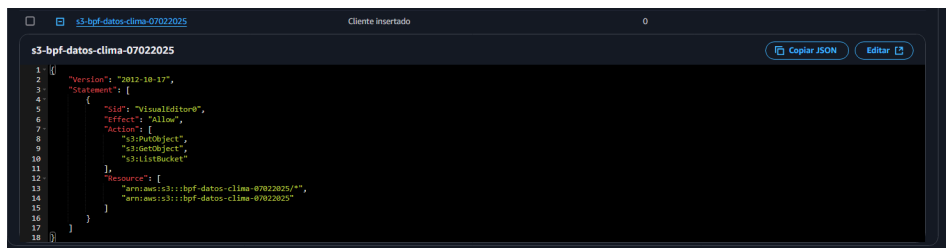
- d. Seleccionamos crear funcion
2. Agregamos capas:
 - a. En la pestaña código hasta la parte de abajo seleccionamos agregar capa
 - b. Obtenemos los arn de request , pandas y boto3 desde la pagina [:https://github.com/keithroza](https://github.com/keithroza)
 - c. Agregamos los arn de cada capa , nos aseguramos que sean de el entorno 3.12

Layers info						Edit	Add a layer
Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN		
1	Klayers-p312-boto3	15	python3.12	x86_64	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p312-boto3:15		
2	Klayers-p312-requests	12	python3.12	x86_64	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p312-requests:12		
3	Klayers-p312-pandas	15	python3.12	x86_64	arn:aws:lambda:us-east-1:770693421928:layer:Klayers-p312-pandas:15		

3. Modificamos lo permisos del Rol

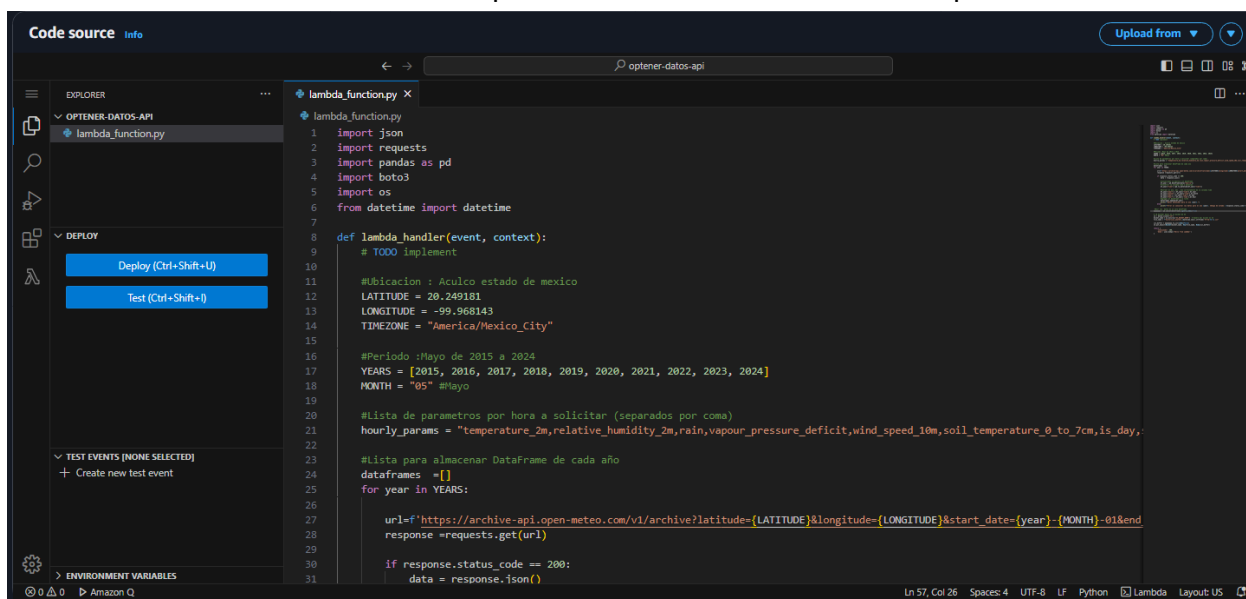


- En la pestaña configuración seleccionamos permisos en nombre del rol nos abrirá IAM
- Creamos una nueva política para darle acceso de **put get y list** a todos los objetos de nuestro bucket "bpf-datos-clima-070222052".



- Adjuntamos la política al rol
- Damos en guardar.
- Regresamos a lambda y actualizamos para ver que ya contamos con los permisos.

4. Agregamos el código que descarga los datos por medio de la api y los guardará en un archivo csv en el bucket bpf-datos-clima-070222052 en la carpeta raw



```
def lambda_handler(event, context):
    for year in YEARS:
        if response.status_code == 200:
            data = response.json()

            #convertimos la hourly a un dataframe
            df_year = pd.DataFrame(data["hourly"])
            #convertir la columna time a datetime
            df_year["time"] = pd.to_datetime(df_year["time"])

            #Extraer el año, mes, día correctamente de la columna time
            df_year["year"] = df_year["time"].dt.year
            df_year["month"] = df_year["time"].dt.month
            df_year["day"] = df_year["time"].dt.day
            df_year["hour"] = df_year["time"].dt.hour
            #agregar el dataframe a la lista
            dataframes.append(df_year)
            print(f"Datos obtenidos para el año {year}.")
        else:
            print(f"Error al consultar los datos para el año {year}. Código de estado: {response.status_code}")

    #Unir los datos en un solo dataframe
    df_clima = pd.concat(dataframes, ignore_index=True)

    # Guardar datos en un bucket de S3
    s3 = boto3.client("s3")
    bucket_name = os.environ["S3_BUCKET_NAME"] # Nombre del bucket en S3
    file_name = f"historical_weather_{datetime.now().strftime('%Y-%m-%d')}.csv"
    csv_buffer = df_clima.to_csv(index=False)
```

```
def lambda_handler(event, context):
    df_clima = pd.concat(dataframes, ignore_index=True)

    # Guardar datos en un bucket de S3
    s3 = boto3.client("s3")
    bucket_name = os.environ["S3_BUCKET_NAME"] # Nombre del bucket en S3
    file_name = f"historical_weather_{datetime.now().strftime('%Y-%m-%d')}.csv"
    csv_buffer = df_clima.to_csv(index=False)
    s3.put_object(Bucket=bucket_name, Key=file_name, Body=csv_buffer)

    return {
        "statusCode": 200,
        "body": json.dumps("El archivo historical wheather se guardo en la carpeta raw")
    }
```

Execution Results

Status: Succeeded

Test Event Name: (unsaved) test event

Response:

```
{
  "statusCode": 200,
  "body": "El archivo historical wheather se guardo en la carpeta raw"
}
```

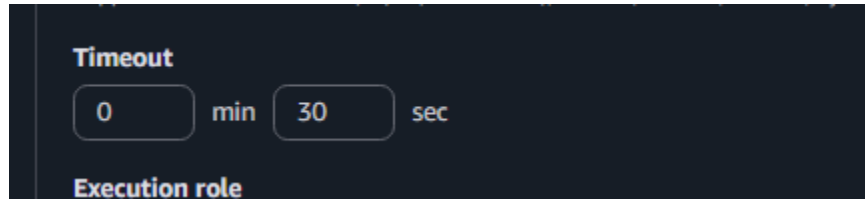
Deployment successful

5. Agregamos nuestra variable de entorno que en este caso es el nombre de nuestro Bucket “bpf-datos-clima-070222052”. Este valor se asignará para la variable S3_BUCKET_NAME.

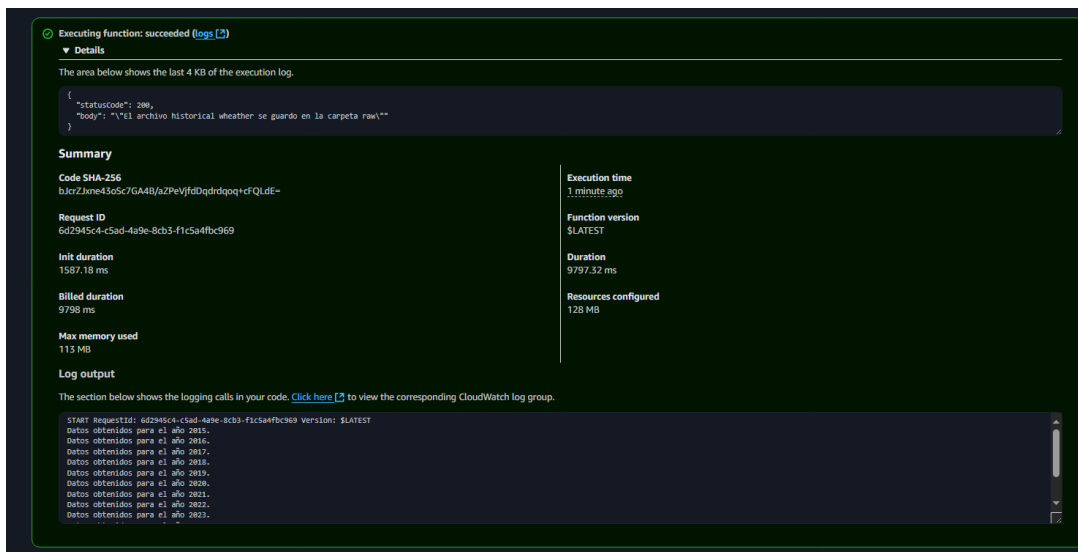
Key	Value
S3_BUCKET_NAME	bpf-datos-clima-070222052

6. Al finalizar este proceso regresamos a la pestaña de código y en el entorno de desarrollo damos en deploy para guardar todos los cambios hechos en el código
7. Vamos a la pestaña configuración y en configuración general:

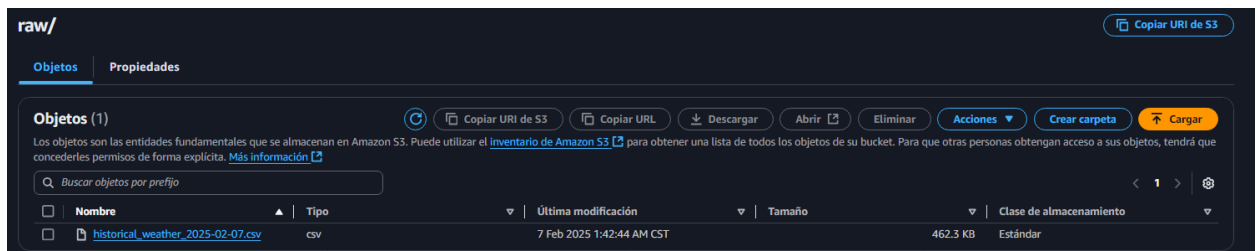
- a. Editamos y en el timeout aumentamos el tiempo un valor adecuado para que se haga la descarga de datos desde la API.



8. Testeamos el código:
 - a. Vamos a la carpeta test y usamos el test que viene por defecto , ya que no tenemos un disparador de la función , está ejecutara y descarga los archivos cuando presionemos test
 - b. Damos click en Test



9. Revisamos nuestra carpeta raw en nuestro bucket “bpf-datos-clima-070222052”, para confirmar que nuestro archivo se generó.



AWS GLUE

Como nuestros archivos no tienen un id y queremos cambiar los nombres de las columnas , realizamos la transformación en Glue

1. Abrimos el servicio de aws Glue
2. Seleccionamos la opción Crear Crawler y le damos un nombre

Set crawler properties

Crawler details [Info](#)

Name

Name can be up to 255 characters long. Some character set including control characters are prohibited.

Description - optional

Descriptions can be up to 2048 characters long.

Tags - optional
Use tags to organize and identify your resources.

[Cancel](#) [Next](#)

3. Añadimos una fuente de datos , en este caso nuestra carpeta en el bucket de s3

Add data source ✕

Data source
Choose the source of data to be crawled.

S3 ▾

Network connection - optional
Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any other S3 targets will also use the same connection (or none, if left blank).

▾ ↻

[Clear selection](#) [Add new connection](#) ↗

Location of S3 data
☒ In this account
☐ In a different account

S3 path
Browse for or enter an existing S3 path.

✕ [View](#) ↗ [Browse S3](#)

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Subsequent crawler runs
This field is a global field that affects all S3 data sources.
☒ **Crawl all sub-folders**
Crawl all folders again with every subsequent crawl.
☐ **Crawl new sub-folders only**
Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.
☐ **Crawl based on events**
Rely on Amazon S3 events to control what folders to crawl.

☐ Sample only a subset of files
☐ Exclude files matching pattern

[Cancel](#) [Add an S3 data source](#)

Choose data sources and classifiers

Data source configuration
 Is your data already mapped to Glue tables?
☒ Not yet
 Select one or more data sources to be crawled.
☐ Yes
 Select existing tables from your Glue Data Catalog.

Data sources (1) [Info](#)
 The list of data sources to be scanned by the crawler.

Edit Remove Add a data source

Type	Data source	Parameters
<input type="radio"/> S3	s3://bpf-datos-clima-07022025	Recrawl all

Custom classifiers - optional
 A classifier checks whether a given file is in a format the crawler can handle. If it is, the classifier creates a schema in the form of a StructType object that matches that data format.

Cancel Previous Next

4. Creamos o seleccionamos un Rol que tenga acceso de creación y descargar de objetos de S3.
5. Añadimos un nueva base de datos destino , le damos nombre y la creamos
6. Revisamos que todo esté bien y creamos el Crawler

Crawler successfully starting
 The following crawler is now starting: "brandon-api-clima"

brandon-api-clima

Last updated (UTC)
 February 7, 2025 at 21:33:35
 Run crawler Edit Delete

Crawler properties			
Name brandon-api-clima	IAM role AWSGlueServiceRole-apiclima	Database clima-catalog-proyecto	State RUNNING
Description -	Security configuration -	Lake Formation configuration -	Table prefix -
Maximum table threshold -			
Advanced settings			

7. Una vez creando el crawler seleccionamos el crawler y damos en Run Crawler
 - a. Esperamos que se ejecute y una vez terminado , revisamos los resultados

Crawler successfully starting
 The following crawler is now starting: "brandon-api-clima"

brandon-api-clima

Last updated (UTC)
 February 7, 2025 at 21:44:18
 Run crawler Edit Delete

Crawler properties			
Name brandon-api-clima	IAM role AWSGlueServiceRole-apiclima	Database clima-catalog-proyecto	State READY
Description -	Security configuration -	Lake Formation configuration -	Table prefix -
Maximum table threshold -			
Advanced settings			

Crawler runs (2)
 The list of crawler runs for this crawler.
 Stop run View CloudWatch logs View run details

Filter data Filter by a date and time range

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
February 7, 2025 at 21:46:31	February 7, 2025 at 21:48:26	01 min 55 s	Completed	-	-

- b.
8. En la pestaña base de datos , seleccionamos Tablas y verificamos que se haya carga los metadatos de nuestro archivo csv que se proceso en nuestro crawler

Schema (13) [Edit schema as JSON](#) [Edit schema](#)

View and manage the table schema.

Q Filter schemas

#	Column name	Data type	Partition key	Comment
1	time	string	-	-
2	temperature_2m	double	-	-
3	relative_humidity_2m	bigint	-	-
4	rain	double	-	-
5	vapour_pressure_deficit	double	-	-
6	wind_speed_10m	double	-	-
7	soil_temperature_0_to_7cm	double	-	-
8	is_day	bigint	-	-
9	shortwave_radiation	double	-	-
10	year	bigint	-	-
11	month	bigint	-	-
12	day	bigint	-	-
13	hour	bigint	-	-

GLUE VISUAL JOB

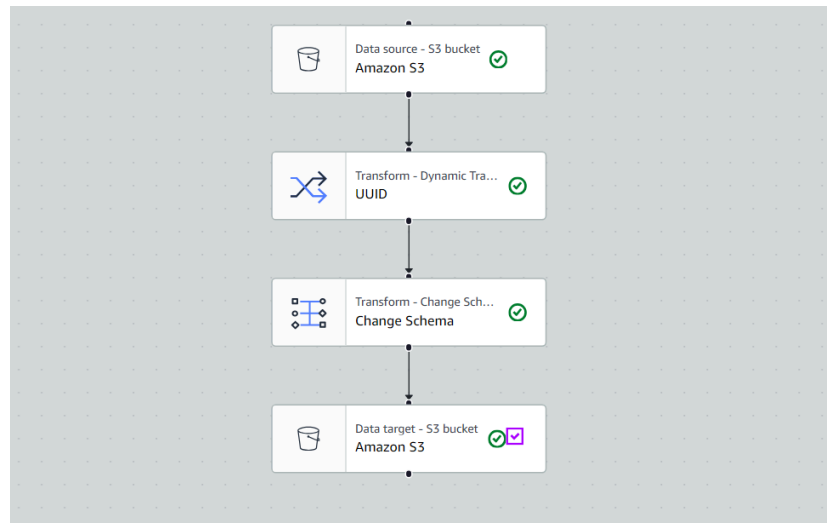
1. Dentro de nuestro entorno para crear el trabajo seleccionamos Fuente para añadir nuestro tabla que se encuentra en el data Catalog
2. En la pestaña job Details colocamos un nombre y añadimos el rol que puede acceder escritura de S3
3. Añadimos un bloque de transformación UUID para añadirle un id a nuestro schema
4. Añadimos un bloque más de transformación Change Schema y cambiamos los nombres de nuestras filas por los deseados, así como podemos observar que se ha añadido la columna de id que agregamos en paso anterior , también eliminamos la columna de time que contiene un formato dateTime

Change Schema (Apply mapping)

Source key	Target key	Data type	Drop
ime			<input checked="" type="checkbox"/>
temperature_2m	temperatura	double	<input type="checkbox"/>
relative_humidity_2m	humedad_relativa	long	<input type="checkbox"/>
rain	precipitacion	double	<input type="checkbox"/>
vapour_pressure_deficit	deficit_presion_vapo	double	<input type="checkbox"/>
wind_speed_10m	velocidad_viento	double	<input type="checkbox"/>
soil_temperature_0_to_7cm	temperatura_suelo	double	<input type="checkbox"/>
is_day	es_dia	int	<input type="checkbox"/>
shortwave_radiation	radiacion	double	<input type="checkbox"/>
year	year	long	<input type="checkbox"/>
month	month	long	<input type="checkbox"/>
day	day	long	<input type="checkbox"/>
hour	hour	long	<input type="checkbox"/>
id	id	string	<input type="checkbox"/>

- Por último añadimos un bloque de destino hacia nuestra carpeta procesado de nuestro bucket s3 en que le indicaremos el que formato de salida será un archivo csv sin ningún tipo de compresión

Tu diagrama deberá verse como este :



- Una vez que se finalice , da en guardar y ejecuta tu job
 - Espera a que se termine de ejecutar.

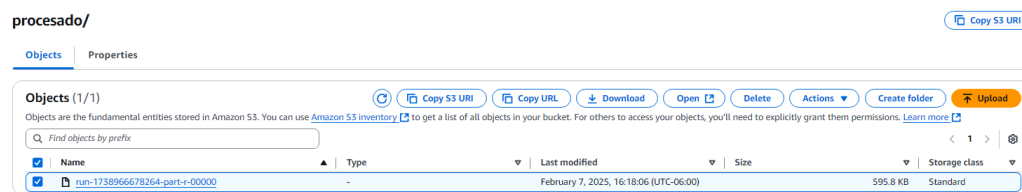
Job runs (1/2) info

Filter job runs by property

Last updated: 6/7/2025 February 7, 2025 at 22:18:57 [View details](#) [Stop job run](#) [Troubleshoot with AI](#) [Table View](#) [Card View](#)

Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity (DPUs)	Worker type	Glue version
<input checked="" type="radio"/> Succeeded	0	02/07/2025 16:16:40	02/07/2025 16:18:17	1 m 29 s	10 DPUs	G.1X	5.0
<input type="radio"/> Stopped	0	02/07/2025 16:14:46	02/07/2025 16:15:06	0 s	10 DPUs	G.1X	5.0

- Revisa tu bucket en la carpeta de procesado para verificar que tu documento se encuentra ahí.

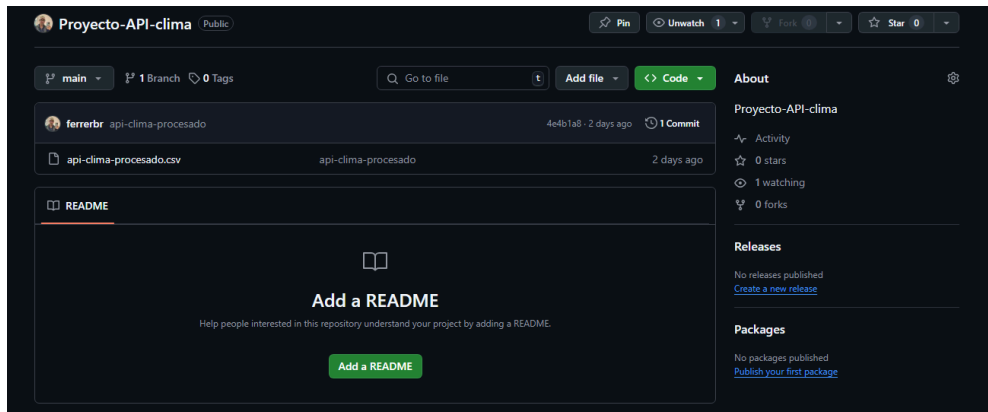


- Descargamos el archivo y en el explorador editamos y colocamos el .csv que faltaba



GitHub

1. Crea un repositorio en GitHub, sube tu archivo .



Listo ya puedes consumir tu archivo para continuar en colab.