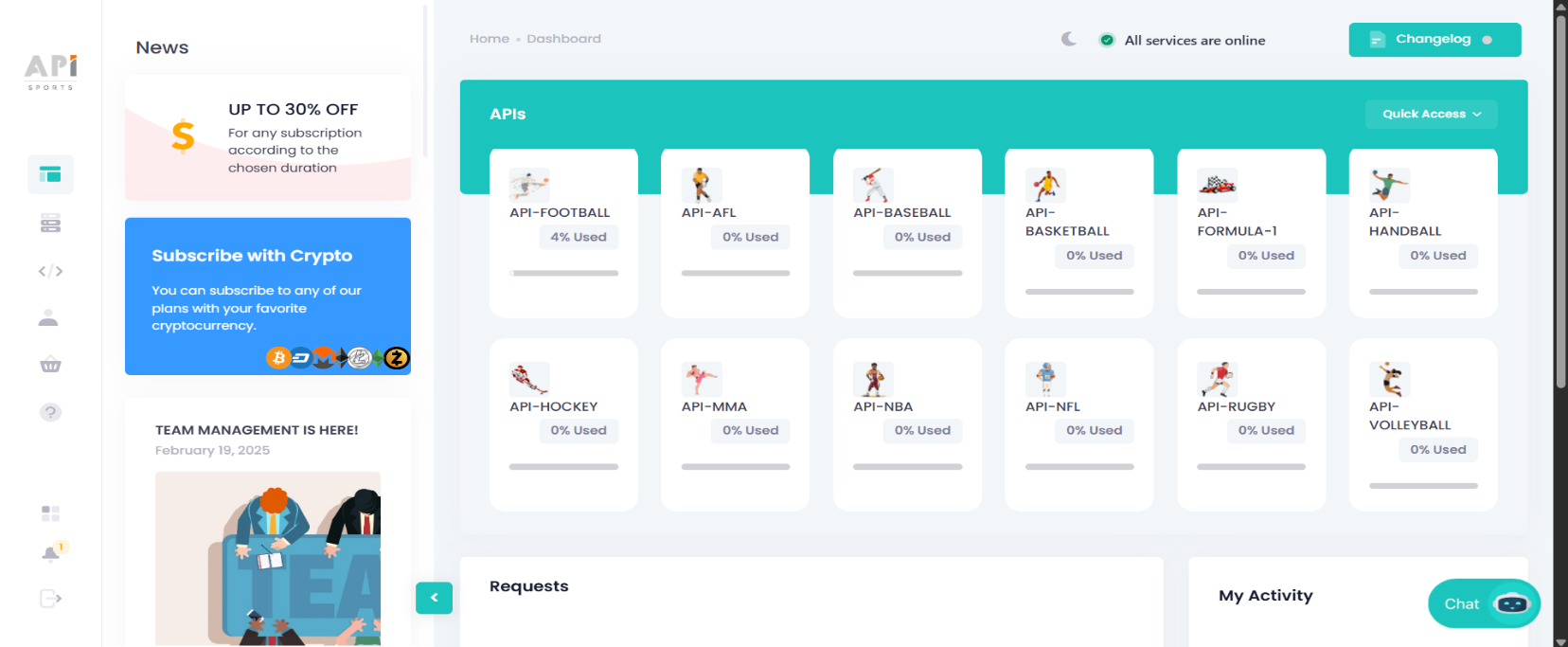
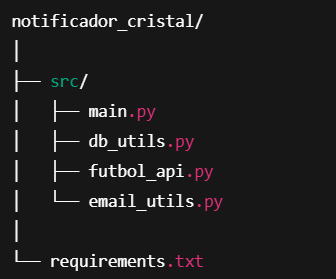
**Guía paso a paso: Notificador de Partidos Sporting Cristal con Python, API y PythonAnywhere**

**1. Conseguir la API Key de API-FOOTBALL**

* Regístrate en https://dashboard.api-football.com/
* Entra a “QUICK ACCESS”, luego a “API\_FOOTBALL”
* Luego selecciona “LIVE DEMO” y copia la “API KEY”
* Interfaz de usuario gráfica, Aplicación

  El contenido generado por IA puede ser incorrecto.La api key tiene un formato como este(ej:c383c8e816cd9c314d1dd2e11f8ee8d0).

**2. ARMAR LAS CARPETAS DEL PROYECTO**

* **Con la siguiente estructura:**

**3. CÓDIGO BASE DE CADA ARCHIVO**

* **db\_utils.py**

def obtener\_credenciales():

# Retorna el correo y la contraseña de aplicación de Gmail directamente (NO uses tu contraseña normal)

return 'cruzmaster18full@gmail.com', 'yddkppskupcsmqrz'

**CAMBIAR POR TU CORREO Y CONTRASEÑA DE APLICACIÓN DE GMAIL**

* **futbol\_api.py**

import requests

API\_KEY = 'c383c8e816cd9c314d1dd2e11f8ee8d0' # Tu API Key

def obtener\_partidos():

url = "https://v3.football.api-sports.io/fixtures"

params = {

"team": "76", # ID Sporting Cristal

"next": 5

**NO OLVIDAR PONER TU API KEY GENERADO AQUI**

}

headers = {

"x-apisports-key": API\_KEY

}

response = requests.get(url, headers=headers, params=params)

data = response.json()

partidos = []

for item in data["response"]:

fixture = item["fixture"]

teams = item["teams"]

league = item["league"]

partidos.append({

"fecha": fixture["date"][:10],

"hora": fixture["date"][11:16],

"local": teams["home"]["name"],

"visitante": teams["away"]["name"],

"liga": league["name"],

})

return partidos

* **email\_utils.py**

**AQUÍ DEBES CAMBIAR EL CORREO AL CUAL QUIERES QUE TE LLEGUE LA NOTIFICACIÓN**

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

EMAIL\_DESTINO = 'jeferjorge164@gmail.com' # Cambia por tu correo de destino

def formatear\_html(partidos):

html = f"""

<h2 style="color:#1A3AC2;">Próximos partidos de Sporting Cristal</h2>

<table border="1" cellpadding="5" style="border-collapse:collapse;font-family:Arial;">

<tr style="background-color:#1A3AC2; color:white;">

<th>Fecha</th>

<th>Hora</th>

<th>Local</th>

<th>Visitante</th>

<th>Liga</th>

</tr>

"""

if partidos:

for p in partidos:

html += f"""

<tr>

<td>{p['fecha']}</td>

<td>{p['hora']}</td>

<td>{p['local']}</td>

<td>{p['visitante']}</td>

<td>{p['liga']}</td>

</tr>

"""

else:

html += """

<tr>

<td colspan="5" style="text-align:center; color:gray; background:#e8eaf6;">

No hay partidos próximos registrados.

</td>

</tr>

"""

html += "</table>"

return html

def enviar\_correo(html, email\_origen, email\_contrasena):

msg = MIMEMultipart('alternative')

msg['Subject'] = "Próximos partidos de Sporting Cristal"

msg['From'] = email\_origen

msg['To'] = EMAIL\_DESTINO

part = MIMEText(html, 'html')

msg.attach(part)

with smtplib.SMTP\_SSL('smtp.gmail.com', 465) as smtp:

smtp.login(email\_origen, email\_contrasena)

smtp.sendmail(email\_origen, EMAIL\_DESTINO.split(','), msg.as\_string())

* **main.py**

from db\_utils import obtener\_credenciales

from futbol\_api import obtener\_partidos

from email\_utils import formatear\_html, enviar\_correo

if \_\_name\_\_ == "\_\_main\_\_":

# Obtén el correo y la contraseña de aplicación desde db\_utils.py

email\_origen, email\_contrasena = obtener\_credenciales()

# Consulta los próximos partidos desde futbol\_api.py

partidos = obtener\_partidos()

# Formatea la información en HTML

html = formatear\_html(partidos)

# Envía el correo con la información

enviar\_correo(html, email\_origen, email\_contrasena)

print("Correo enviado con éxito.")

* **requirements.txt**

requests

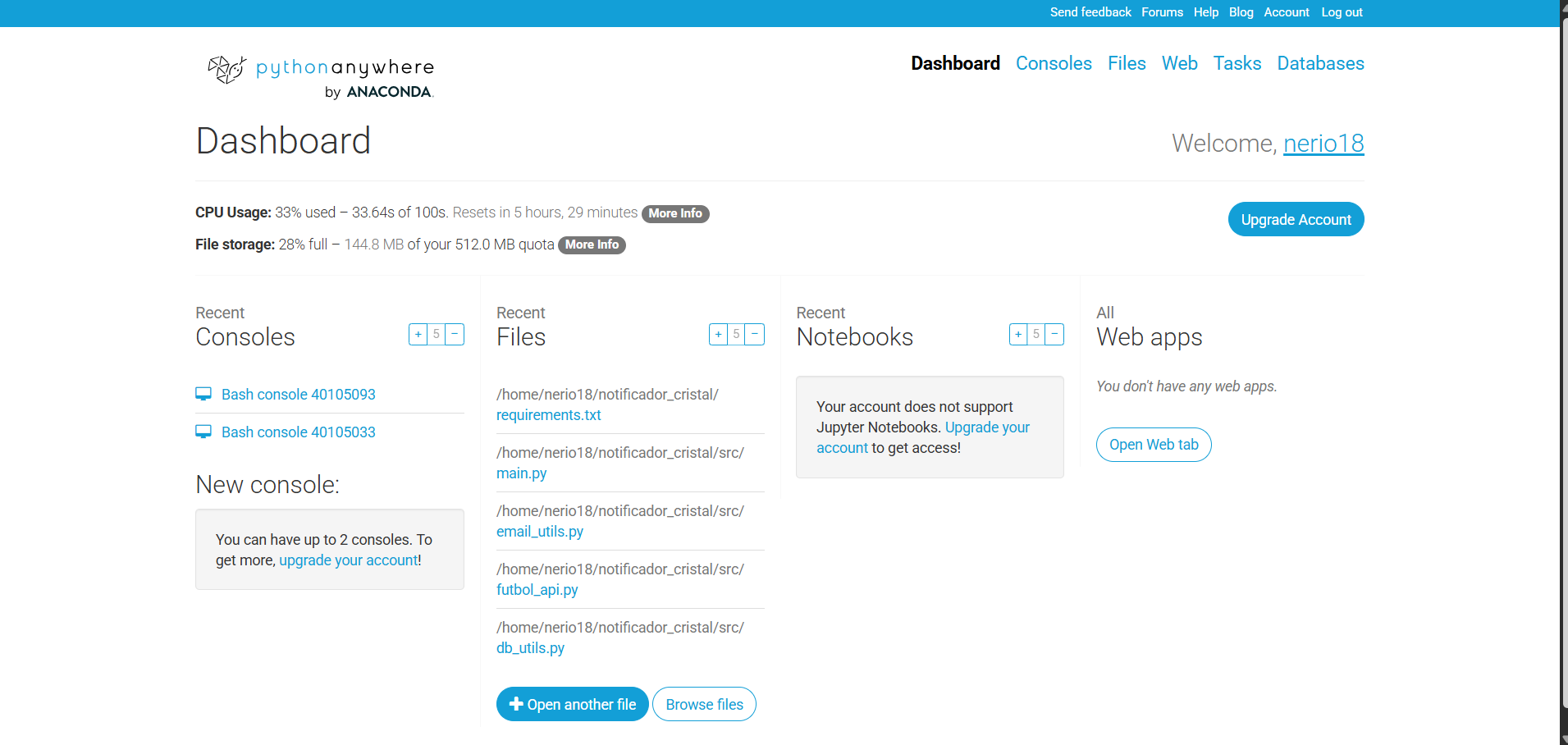
mysql-connector-python

**4. SUBIR Y CONFIGURAR EN PYTHON ANYWHERE**

**A. Empaqueta tu proyecto:**

* comprime tu proyecto: Haz un ZIP de la carpeta **notificador\_cristal/**.

**B. Sube el ZIP a python anywhere:**

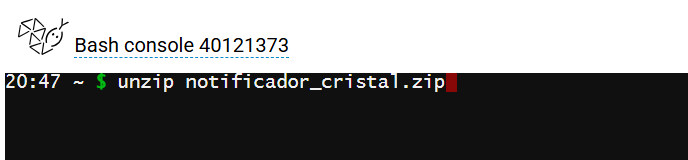
* ****Ve a **“FILES”** en PythonAnywhere y usa **“UPLOAD A FILE”.**

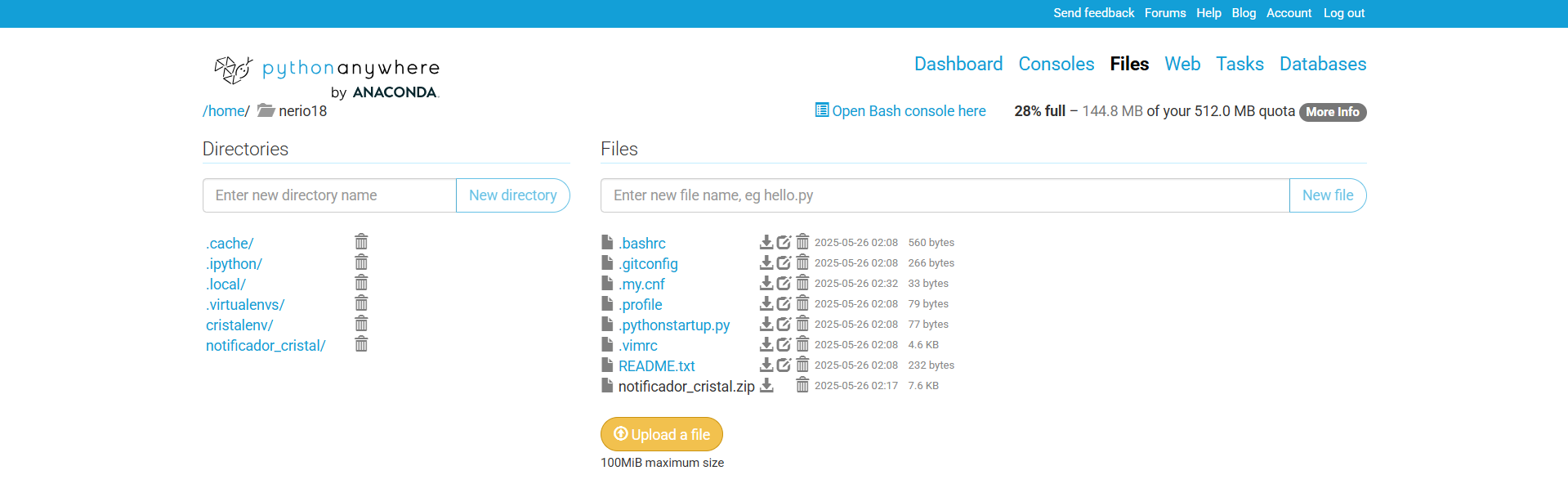
**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**

**C. Descomprime el ZIP:**

* Abre la **consola Bash** en python anywhere y ejecuta el siguiente codigo**:**

**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.unzip notificador\_cristal.zip**

**Debe salirte algo así después de presionar “ENTER”**

**D. Crea y activa un entorno virtual:**

* **En la misma consola de bash ejecuta este código:**

python3 -m venv cristalenv

source cristalenv/bin/activate

pip install --upgrade pip

Texto

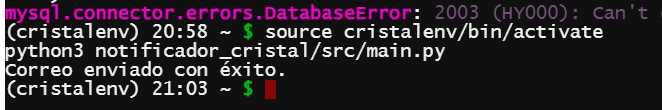
El contenido generado por IA puede ser incorrecto.pip install -r notificador\_cristal/requirements.txt

**Debe quedarte algo asi**

**F. Prueba el script manualmente en la consola:**

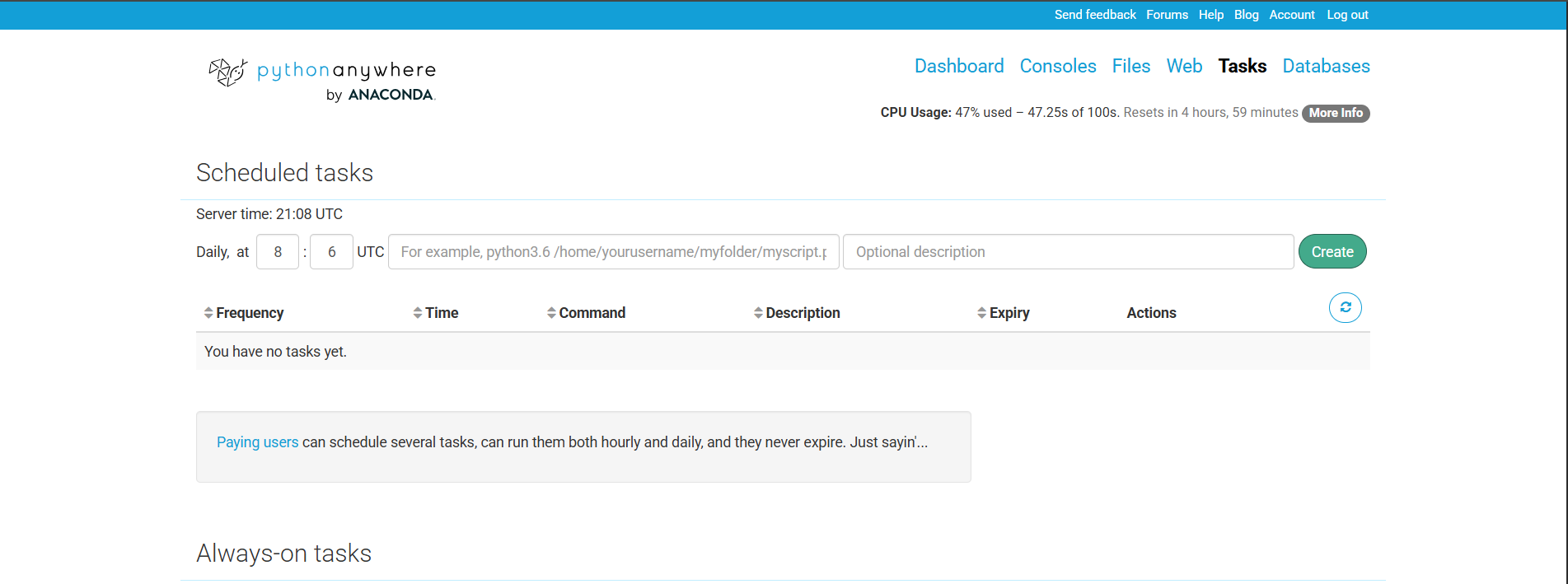
* **En la misma consola bash ejecuta este código:**

source cristalenv/bin/activate

****python3 notificador\_cristal/src/main.py

**Deberías ver ese mensaje y la notificación ya te debió llegar al correo**

**4. AUTOMATIZA EL ENVIO DIARIO**

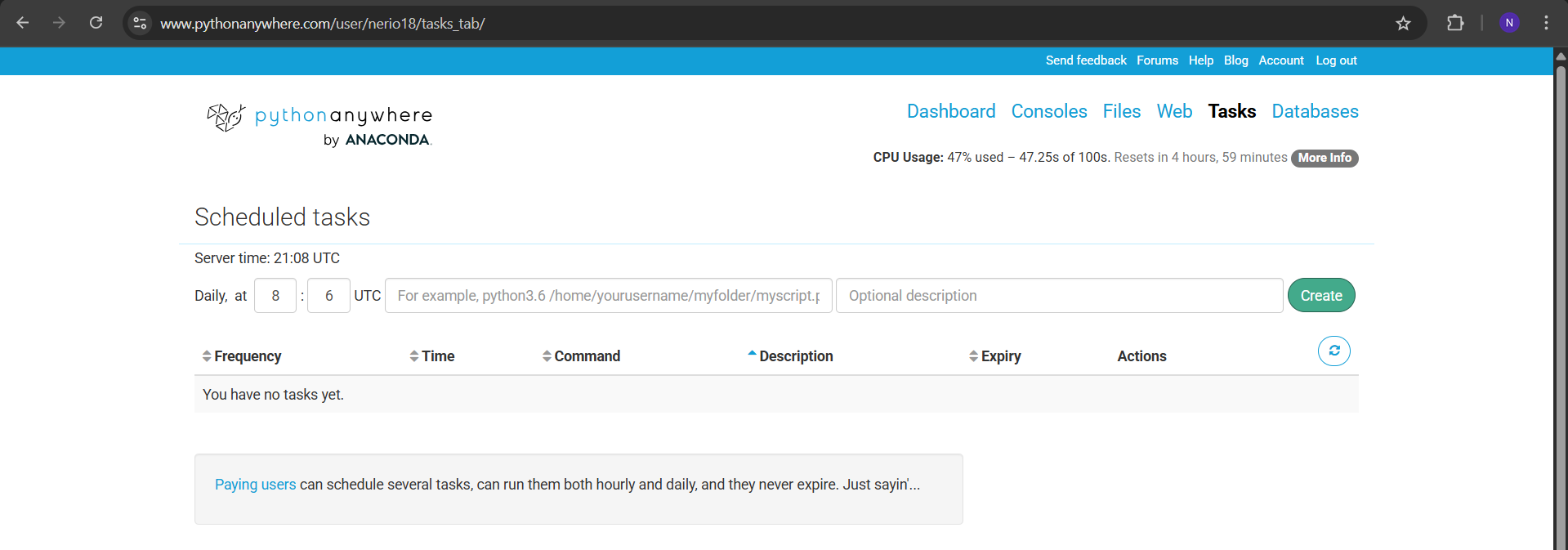
* Ve a la pestaña “**TASK**”
* Llena los campos con los siguientes datos:

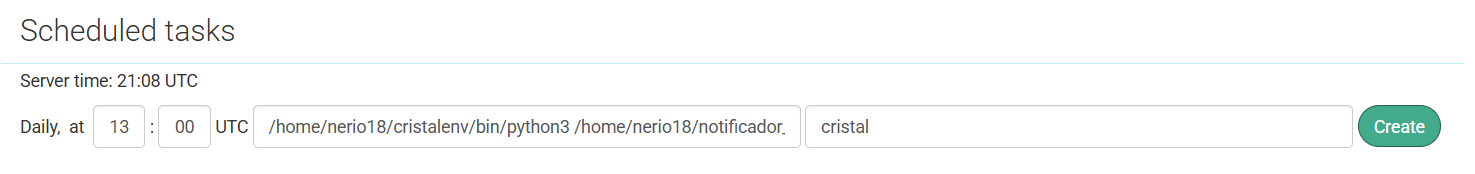
1. **en daily at:** **13:00** (para la notificación te llegue a las 8:00 am hora Perú todos los dias)
2. **después de UTC:**

**/home/nerio18/cristalenv/bin/python3 /home/nerio18/notificador\_cristal/src/main.py**

**IMPORTANTE: cambiar el nombre de usuario por el tuyo**

Tu nombre de **USUARIO** lo encuentras aquí:



1. en **OPTINAL DESCRIPTION** pon el nombre gustes.

-asi deberían quedar llenados los datos

-luego presiona **CREATE** y listo!!!