

DIA 1:

Rama develop y git add .

```
MINGW64; c:/Users/windows10/Desktop/trivia-game-python
windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (main)
$ git remote set-url origin https://github.com/AriusJoell/Prueba_de_Entrada.git
error: No such remote 'origin'

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (main)
$ AC

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (main)
$ git remote add origin https://github.com/AriusJoell/Prueba_de_Entrada.git

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (main)
$ git push -u origin develop
error: src refspec develop does not match any
error: failed to push some refs to 'https://github.com/AriusJoell/Prueba_de_Entrada.git'

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (main)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/AriusJoell/Prueba_de_Entrada.git'

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (main)
$ git checkout -b develop
Switched to a new branch 'develop'

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ git add .
warning: in the working copy of 'main.py', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'venv/Lib/site-packages/PyYAML-6.0.2.dist-info/INSTALLER', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'venv/Lib/site-packages/PyYAML-6.0.2.dist-info/LICENSE', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'venv/Lib/site-packages/PyYAML-6.0.2.dist-info/WHEEL', LF will be replaced by CRLF the next time Git touches it
```

commit

```
MINGW64; c:/Users/windows10/Desktop/trivia-game-python
windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ git commit -m "Primer commit con todos los archivos del proyecto"
[develop (root-commit) 12404ed] Primer commit con todos los archivos del proyecto
2016 files changed, 329130 insertions(+)
create mode 100644 __pycache__/main.cpython-312.pyc
create mode 100644 main.py
create mode 100644 pruebas.py
create mode 100644 venv/Lib/site-packages/PyYAML-6.0.2.dist-info/INSTALLER
create mode 100644 venv/Lib/site-packages/PyYAML-6.0.2.dist-info/LICENSE
create mode 100644 venv/Lib/site-packages/PyYAML-6.0.2.dist-info/METADATA
create mode 100644 venv/Lib/site-packages/PyYAML-6.0.2.dist-info/RECORD
create mode 100644 venv/Lib/site-packages/PyYAML-6.0.2.dist-info/WHEEL
create mode 100644 venv/Lib/site-packages/PyYAML-6.0.2.dist-info/top_level.txt
create mode 100644 venv/Lib/site-packages/__pycache__/typing_extensions.cpython-312.pyc
create mode 100644 venv/Lib/site-packages/_yaml/_init_.py
create mode 100644 venv/Lib/site-packages/_yaml/__pycache__/__init__.cpython-312.pyc
create mode 100644 venv/Lib/site-packages/annotated_types-0.7.0.dist-info/INSTALLER
create mode 100644 venv/Lib/site-packages/annotated_types-0.7.0.dist-info/METADATA
create mode 100644 venv/Lib/site-packages/annotated_types-0.7.0.dist-info/RECORD
create mode 100644 venv/Lib/site-packages/annotated_types-0.7.0.dist-info/WHEEL
create mode 100644 venv/Lib/site-packages/annotated_types-0.7.0.dist-info/licenses/LICENSE
create mode 100644 venv/Lib/site-packages/annotated_types/__init__.py
create mode 100644 venv/Lib/site-packages/annotated_types/__pycache__/__init__.cpython-312.pyc
create mode 100644 venv/Lib/site-packages/annotated_types/__pycache__/test_cases.cpython-312.pyc
create mode 100644 venv/Lib/site-packages/annotated_types/py.typed
create mode 100644 venv/Lib/site-packages/annotated_types/test_cases.py
create mode 100644 venv/Lib/site-packages/anyio-4.9.0.dist-info/INSTALLER
create mode 100644 venv/Lib/site-packages/anyio-4.9.0.dist-info/LICENSE
create mode 100644 venv/Lib/site-packages/anyio-4.9.0.dist-info/METADATA
create mode 100644 venv/Lib/site-packages/anyio-4.9.0.dist-info/RECORD
create mode 100644 venv/Lib/site-packages/anyio-4.9.0.dist-info/WHEEL
create mode 100644 venv/Lib/site-packages/anyio-4.9.0.dist-info/entry_points.txt
```

Archivos subidos al github

AriusJoel1

Prueba_de_Entrada

🔍

🔖

⌵

+

🕒

🔗

📧

👤

<> Code

🕒 Issues

🔗 Pull requests

🕒 Actions

📁 Projects

📖 Wiki

🛡 Security

⋮

👤 Prueba_de_Entrada

Public

📌 Pin

👁 Unwatch 1

🔗 Fork 0

★ Star 0

🔗 develop

🔗

📄

Go to file

+

<> Code

About

👤 AriusJoel1

Primer commit con todos los a...

12404ed · 33 minutes ago

🕒

📁 __pycache__

Configuración inicial d...

33 minutes ago

📁 venv

Configuración inicial d...

33 minutes ago

📄 main.py

Configuración inicial d...

33 minutes ago

📄 pruebas.py

Configuración inicial d...

33 minutes ago

📖 README

No description, website, or topics provided.

🏠 Activity

★ 0 stars

👁 1 watching

🔗 0 forks

Releases

No releases published

[Create a new release](#)

Packages

DIA 2

Tests:

```
PS C:\Users\windows10\Desktop\trivia-game-python> py .\test_trivia.py
PS C:\Users\windows10\Desktop\trivia-game-python> pytest
===== test session starts =====
platform win32 -- Python 3.12.5, pytest-8.3.5, pluggy-1.5.0
rootdir: C:\Users\windows10\Desktop\trivia-game-python
plugins: anyio-4.9.0
collected 2 items

test_trivia.py .. [100%]

===== 2 passed in 0.03s =====
PS C:\Users\windows10\Desktop\trivia-game-python>
PS C:\Users\windows10\Desktop\trivia-game-python>
```

Commit “Implementación de la clase Question y pruebas unitarias básicas”

AriusJoel1

Implementación de la clase Question y pruebas unitarias básicas

2e9437d · 3 minutes ago

🕒 2 Commits

📁 __pycache__

Implementación de la clase Question y pruebas unitarias bá...

3 minutes ago

📁 venv

Implementación de la clase Question y pruebas unitarias bá...

3 minutes ago

📄 test_trivia.py

Implementación de la clase Question y pruebas unitarias bá...

3 minutes ago

📄 trivia.py

Implementación de la clase Question y pruebas unitarias bá...

3 minutes ago

DIA 3:

Agregando quiz

```
trivia.py M X test_trivia.py
trivia.py > Quiz > get_next_question
1 class Question:
2     def __init__(self, description, options, correct_answer):
3         self.description = description
4         self.options = options
5         self.correct_answer = correct_answer
6
7     def is_correct(self, answer):
8         return self.correct_answer == answer
9
10 class Quiz:
11     def __init__(self):
12         self.questions = []
13         self.current_question_index = 0
14
15     def add_question(self, question):
16         self.questions.append(question)
17
18     def get_next_question(self):
19         if self.current_question_index < len(self.questions):
20             question = self.questions[self.current_question_index]
21             self.current_question_index += 1
22             return question
23         return None
```

Visualizando cambios

```
MINGW64:/c/Users/windows10/Desktop/trivia-game-python
branch 'develop' set up to track 'origin/develop'.

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ git diff
warning: in the working copy of 'trivia.py', LF will be replaced by CRLF the next time Git touches it
diff --git a/trivia.py b/trivia.py
index a3d9f69..eef2f26 100644
--- a/trivia.py
+++ b/trivia.py
@@ -6,3 +6,18 @@ class Question:
     def is_correct(self, answer):
         return self.correct_answer == answer
+
+class Quiz:
+    def __init__(self):
+        self.questions = []
+        self.current_question_index = 0
+
+    def add_question(self, question):
+        self.questions.append(question)
+
+    def get_next_question(self):
+        if self.current_question_index < len(self.questions):
+            question = self.questions[self.current_question_index]
+            self.current_question_index += 1
+            return question
+        return None
\ No newline at end of file

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$
```

Agregando el quiz al git hub

```
MINGW64~/c/Users/windows10/Desktop/trivia-game-python
+ def get_next_question(self):
+     if self.current_question_index < len(self.questions):
+         question = self.questions[self.current_question_index]
+         self.current_question_index += 1
+         return question
+     return None
\ No newline at end of file

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ git add .
warning: in the working copy of 'trivia.py', LF will be replaced by CRLF the next time Git touches it

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ git commit -m "Implementación de la clase Quiz y flujo básico del juego"
[develop 3a2a763] Implementación de la clase Quiz y flujo básico del juego
1 file changed, 15 insertions(+)

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ git push -u origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), 530 bytes | 530.00 KiB/s, done.
Writing objects: 100% (3/3), 530 bytes | 530.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/AriusJoel1/Prueba_de_Entrada.git
   2e9437d..3a2a763  develop -> develop
branch 'develop' set up to track 'origin/develop'.

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ |
```

DIA 4:

Modificamos la clase quiz

```
9
10 class Quiz:
11     def __init__(self):
12         self.questions = []
13         self.current_question_index = 0
14         self.correct_answers = 0
15         self.incorrect_answers = 0
16
17     def add_question(self, question):
18         self.questions.append(question)
19
20     def get_next_question(self):
21         if self.current_question_index < len(self.questions):
22             question = self.questions[self.current_question_index]
23             self.current_question_index += 1
24             return question
25         return None
26
27     def answer_question(self, question, answer):
28         if question.is_correct(answer):
29             self.correct_answers += 1
30             return True
31         else:
32             self.incorrect_answers += 1
33             return False
34
35
```

Y la prueba de testeo

```
trivia.py M test_trivia.py M X
test_trivia.py > ...
1 import pytest
2 from trivia import Question, Quiz
3
4
5
6 def (variable) question: Question
7     question = Question("What is 2 + 2?", ["1", "2", "3", "4"], "4")
8     assert question.is_correct("4")
9
10 def test_question_incorrect_answer():
11     question = Question("What is 2 + 2?", ["1", "2", "3", "4"], "4")
12     assert not question.is_correct("2")
13
14
15 def test_quiz_scoring():
16     quiz = Quiz()
17     question = Question("What is 2 + 2?", ["1", "2", "3", "4"], "4")
18     quiz.add_question(question)
19     assert quiz.answer_question(question, "4") == True
20     assert quiz.correct_answers == 1
```

Implementamos la función run_quiz()

```
def run_quiz():
    quiz = Quiz()

    # Lista de 10 preguntas
    preguntas = [
        ("¿Cuánto es 2 + 2?", ["1", "2", "3", "4"], "4"),
        ("¿Cuál es la capital de Francia?", ["Londres", "Berlín", "París", "Madrid"], "París"),
        ("¿Qué planeta es conocido como el planeta rojo?", ["Tierra", "Marte", "Venus", "Júpiter"], "Marte"),
        ("¿Quién escribió 'Hamlet'?", ["Shakespeare", "Cervantes", "Tolstói", "Homero"], "Shakespeare"),
        ("¿Cuánto es 5 * 6?", ["30", "20", "25", "40"], "30"),
        ("¿Cuál es el punto de ebullición del agua (°C)?", ["50", "100", "80", "120"], "100"),
        ("¿Qué gas absorben las plantas?", ["Oxígeno", "Hidrógeno", "Dióxido de carbono", "Nitrógeno"], "Dióxido de carbono"),
        ("¿Cuál es el océano más grande?", ["Atlántico", "Índico", "Ártico", "Pacífico"], "Pacífico"),
        ("¿Cuál es la raíz cuadrada de 81?", ["9", "8", "7", "6"], "9"),
        ("¿Quién pintó la Mona Lisa?", ["Picasso", "Van Gogh", "Da Vinci", "Rembrandt"], "Da Vinci"),
    ]

    # Agregar las preguntas al quiz
    for descripcion, opciones, respuesta_correcta in preguntas:
        quiz.add_question(Question(descripcion, opciones, respuesta_correcta))

    # Ejecutar el quiz
    for _ in range(10):
        pregunta = quiz.get_next_question()
        print("\n" + pregunta.descripcion)
        for i, opcion in enumerate(pregunta.options):
            print(f"{i + 1}. {opcion}")

        try:
            eleccion = int(input("Ingresa el número de la opción: ")) - 1
            if 0 <= eleccion < len(pregunta.options):
                respuesta_usuario = pregunta.options[eleccion]
                if quiz.answer_question(pregunta, respuesta_usuario):
                    print("¡Correcto!")
                else:
                    print(f"¡Incorrecto! La respuesta correcta era: {pregunta.correct_answer}")
            else:
                print("Opción inválida. Pregunta omitida.")
        except ValueError:
            print("Entrada inválida. Debes ingresar un número. Pregunta omitida.")

    print("\n;Quiz finalizado!")
    print(f"Respuestas correctas: {quiz.correct_answers}")
    print(f"Respuestas incorrectas: {quiz.incorrect_answers}")
```

Agregamos los cambios

```
windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$
git add .
warning: in the working copy of 'trivia.py', LF will be replaced by CRLF the next time Git touches it

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ git commit -m "Implementación de sistema de puntuación, manejo de rondas y finalización del juego"
[develop e268f91] Implementación de sistema de puntuación, manejo de rondas y finalización del juego
 2 files changed, 70 insertions(+), 3 deletions(-)

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$
```

DIA 5:

Mejoramos la función run_quiz()

```
def run_quiz():
    print("¡Bienvenido al juego de Trivia!")
    print("Responde las siguientes preguntas seleccionando el número de la opción correcta.")

    quiz = Quiz()

    # Cargar 10 preguntas en español
    preguntas = [
        ("¿Cuánto es 2 + 2?", ["1", "2", "3", "4"], "4"),
        ("¿Cuál es la capital de Francia?", ["Londres", "Berlín", "París", "Madrid"], "París"),
        ("¿Qué planeta es conocido como el planeta rojo?", ["Tierra", "Marte", "Venus", "Júpiter"], "Marte"),
        ("¿Quién escribió 'Hamlet'?", ["Shakespeare", "Cervantes", "Tolstói", "Homero"], "Shakespeare"),
        ("¿Cuánto es 5 * 6?", ["30", "20", "25", "40"], "30"),
        ("¿Cuál es el punto de ebullición del agua (°C)?", ["50", "100", "80", "120"], "100"),
        ("¿Qué gas absorben las plantas?", ["Oxígeno", "Hidrógeno", "Dióxido de carbono", "Nitrógeno"], "Dióxido de carbono"),
        ("¿Cuál es el océano más grande?", ["Atlántico", "Índico", "Ártico", "Pacífico"], "Pacífico"),
        ("¿Cuál es la raíz cuadrada de 81?", ["9", "8", "7", "6"], "9"),
        ("¿Quién pintó la Mona Lisa?", ["Picasso", "Van Gogh", "Da Vinci", "Rembrandt"], "Da Vinci"),
    ]

    for desc, opciones, correcta in preguntas:
        quiz.add_question(Question(desc, opciones, correcta))

    while quiz.current_question_index < 10:
        question = quiz.get_next_question()
        if question:
            print(f"\nPregunta {quiz.current_question_index}: {question.description}")
            for idx, option in enumerate(question.options):
                print(f"{idx + 1}) {option}")
            answer_input = input("Tu respuesta (número): ")

            try:
                answer_index = int(answer_input) - 1
                if 0 <= answer_index < len(question.options):
                    answer_text = question.options[answer_index]
                    if quiz.answer_question(question, answer_text):
                        print("¡Correcto!")
                    else:
                        print(f"Incorrecto. La respuesta correcta era: {question.correct_answer}")
                else:
                    print("Número fuera de rango. Pregunta omitida.")
            except ValueError:
                print("Entrada inválida. Debes ingresar un número.")
            else:
                break

    print("\nJuego terminado.")
    print(f"Preguntas contestadas: {quiz.current_question_index}")
    print(f"Respuestas correctas: {quiz.correct_answers}")
    print(f"Respuestas incorrectas: {quiz.incorrect_answers}")
```

Realizamos las pruebas

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Tu respuesta (número): 2
Incorrecto. La respuesta correcta era: Pacífico

Pregunta 9: ¿Cuál es la raíz cuadrada de 81?
1) 9
2) 8
3) 7
4) 6
Tu respuesta (número): 3
Incorrecto. La respuesta correcta era: 9

Pregunta 10: ¿Quién pintó la Mona Lisa?
1) Picasso
2) Van Gogh
3) Da Vinci
4) Rembrandt
Tu respuesta (número): 4
Incorrecto. La respuesta correcta era: Da Vinci

Juego terminado.
Preguntas contestadas: 10
Respuestas correctas: 1
Respuestas incorrectas: 9
PS C:\Users\windows10\Desktop\trivia-game-python>
```

Agregamos los commits desde la rama

```
windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ git add .

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ git commit -m "Mejoras en la interfaz de usuario y resumen final detallado"
[develop b5f2311] Mejoras en la interfaz de usuario y resumen final detallado
 2 files changed, 32 insertions(+), 23 deletions(-)

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$
```

DIA 6:

Implementamos el archivo main.py

```
EXPLORER
TRIVIA GAME PYTHON
> .__pycache__
> .pytest_cache
> test\integrations
> test_api.py
> venv
> main.py
test_trivia.py
trivia.py

main.py
1 # main.py
2 from fastapi import FastAPI, HTTPException, status
3 from pydantic import BaseModel
4 from typing import List
5 from trivia import Question # Importamos tu clase Question
6
7 app = FastAPI()
8
9 # Base de datos en memoria
10 questions_db = []
11
12 # Modelo Pydantic para validar datos que llegan por la API
13 class QuestionModel(BaseModel):
14     description: str
15     options: List[str]
16     correct_answer: str
17
18 @app.post("/questions/", status_code=status.HTTP_201_CREATED)
19 def create_question(question: QuestionModel):
20     # Verificamos que la respuesta correcta esté dentro de las opciones
21     if question.correct_answer not in question.options:
22         raise HTTPException(
23             status_code=400,
24             detail="La respuesta correcta debe estar entre las opciones"
```


Instalamos las librerías necesarias

```
windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ pip install fastapi uvicorn
Requirement already satisfied: fastapi in c:\users\windows10\desktop\trivia-
ages (0.115.12)
Requirement already satisfied: uvicorn in c:\users\windows10\desktop\trivia-
ages (0.34.0)
Requirement already satisfied: starlette<0.47.0,>=0.40.0 in c:\users\windows
venv\lib\site-packages (from fastapi) (0.46.1)
Requirement already satisfied: pydantic!=1.8,!1.8.1,!2.0.0,!2.0.1,!2.1.0
```

Y ejecutamos

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[notice] To update, run: python.exe -m pip install --upgrade pip

windows10@DESKTOP-UEI1KU9 MINGW64 ~/Desktop/trivia-game-python (develop)
$ uvicorn main:app --reload
INFO:      Will watch for changes in these directories: ['C:\\Users\\windows10\\D
']
INFO:      Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:      Started reloader process [4956] using WatchFiles
INFO:      Started server process [15868]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
INFO:      Application startup complete.
INFO:      127.0.0.1:51219 - "GET /docs HTTP/1.1" 200 OK
INFO:      Application startup complete.
INFO:      127.0.0.1:51219 - "GET /docs HTTP/1.1" 200 OK
INFO:      127.0.0.1:51219 - "GET /openapi.json HTTP/1.1" 200 OK
INFO:      127.0.0.1:51267 - "POST /questions/ HTTP/1.1" 201 Created
INFO:      127.0.0.1:51267 - "POST /questions/ HTTP/1.1" 201 Created
INFO:      127.0.0.1:51267 - "POST /questions/ HTTP/1.1" 201 Created
INFO:      127.0.0.1:51272 - "POST /questions/ HTTP/1.1" 201 Created
INFO:      127.0.0.1:51279 - "GET /questions/ HTTP/1.1" 200 OK
INFO:      127.0.0.1:51279 - "GET /questions/ HTTP/1.1" 200 OK
```

Hacemos los cambios respectivos en el curl

default

GET /questions/ Get All Questions

POST /questions/ Create Question

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "description": "string",
  "options": [
    "string"
  ],
  "correct_answer": "string"
}
```

Responses

Code	Description	Links
201	Successful Response	No links

Ejecutamos y subimos las preguntas

127.0.0.1:8000/docs#/default/get_all_questions_questions__get

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/questions/' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/questions/
```

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "description": "¿Cuál es la capital de Perú?", "options": ["Lima", "Cusco", "Arequipa", "Trujillo"], "correct_answer": "Lima" }]</pre>

Ejecutando las pruebas

```
PS C:\Users\windows10\Desktop\trivia-game-python> py .\test\integrations\
test_api.py
Status code: 201
Response body: {'message': 'Pregunta creada'}
PS C:\Users\windows10\Desktop\trivia-game-python>
```

DIA 7:

Conexión a postgresql y usando dotenv para cargar el url y la contraseña

```
based.py > ...
1  import psycopg2
2
3  from dotenv import load_dotenv
4  import os
5
6  load_dotenv()
7  DATABASE_URL = os.getenv("DATABASE_URL")
8  SECRET_KEY = os.getenv("SECRET_KEY")
9
10 db_host = DATABASE_URL
11 db_port = "5432"
12 db_user = "postgres"
13 db_password = SECRET_KEY
14 connect_db = "trivia_db"
15
16 print(f"🔗 Intentando conectar a PostgreSQL en {db_host}:{db_port} como usuario '{db_user}'...")
17
18 try:
19
20     with psycopg2.connect(
21         dbname=connect_db,
22         user=db_user,
23         password=db_password,
24         host=db_host,
25         port=db_port
26     ) as connection:
27
28         connection.autocommit = True
29         print(f"✅ Conectado a la base de datos '{connect_db}'.")
30
31         with connection.cursor() as cursor:
32             print("🔍 Obteniendo lista de bases de datos...")
33
34
35     list_databases_query = "SELECT datname FROM pg_database;"
36     cursor.execute(list_databases_query)
37
38
39     database_list = cursor.fetchall()
40
41     if database_list:
42         print("\n--- Bases de Datos Encontradas ---")
43         for db in database_list:
44             print(f"- {db[0]}")
45         print("-----")
46         print(f"✅ Operación completada. {len(database_list)} bases de datos listadas.")
47     else:
48
49         print("😞 No se encontraron bases de datos.")
50
51
52
53 except psycopg2.OperationalError as e:
54     print("\n❌ Error de Conexión a PostgreSQL:")
55     print(f"    {e}")
56     print("\n    Por favor, verifica:")
57     print(f"    1. Que el servidor PostgreSQL esté corriendo en '{db_host}:{db_port}'.")
58     print(f"    2. Que el nombre de usuario ('{db_user}') y la contraseña sean correctos.")
59     print(f"    3. Que el host ('{db_host}') sea accesible desde donde ejecutas el script.")
60     print(f"    4. Que la base de datos de conexión ('{connect_db}') exista (generalmente 'postgres' sí existe).")
61     print(f"    5. Que no haya un firewall bloqueando la conexión al puerto {db_port}.")
62
63
64 except Exception as e:
65     print(f"\n❌ Ocurrió un error inesperado: {e}")
66
67 finally:
68     print("\n🏁 Script finalizado.")
```

Conexcion a postgresql exitosa

```
🔄 Intentando conectar a PostgreSQL en localhost:5432
res'...
✅ Conectado a la base de datos 'postgres'.
🔍 Obteniendo lista de bases de datos...

--- Bases de Datos Encontradas ---
- postgres
- prueba_entrada
- template1
- template0
- trivia_db
-----

✅ Operación completada. 5 bases de datos listadas.
```