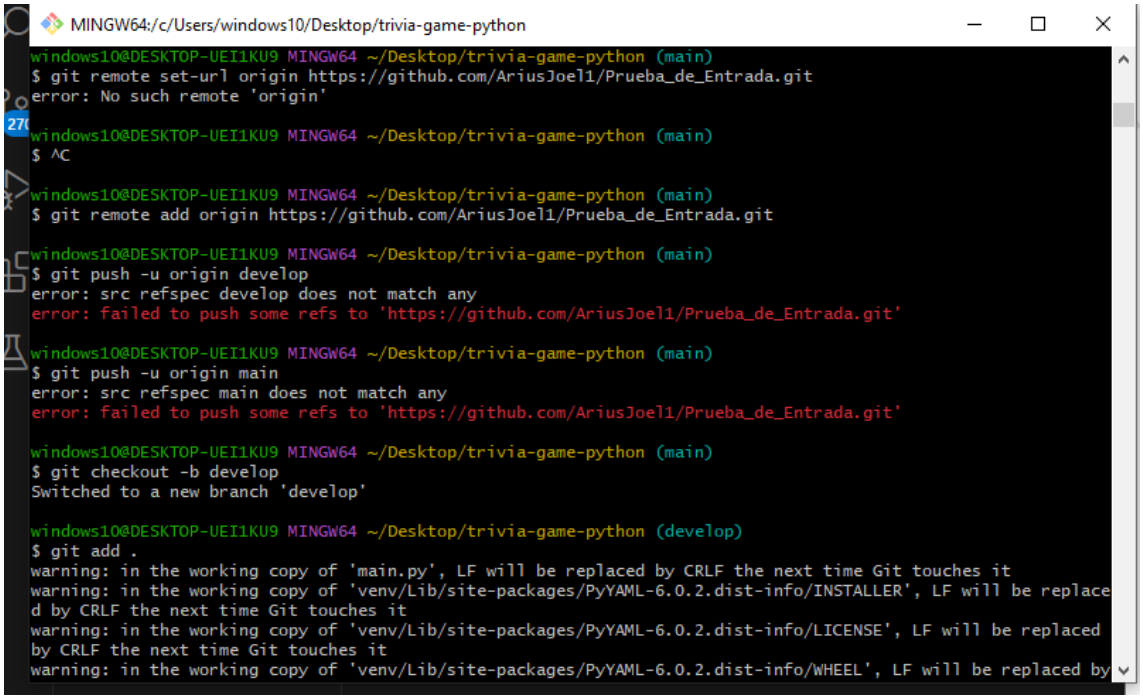
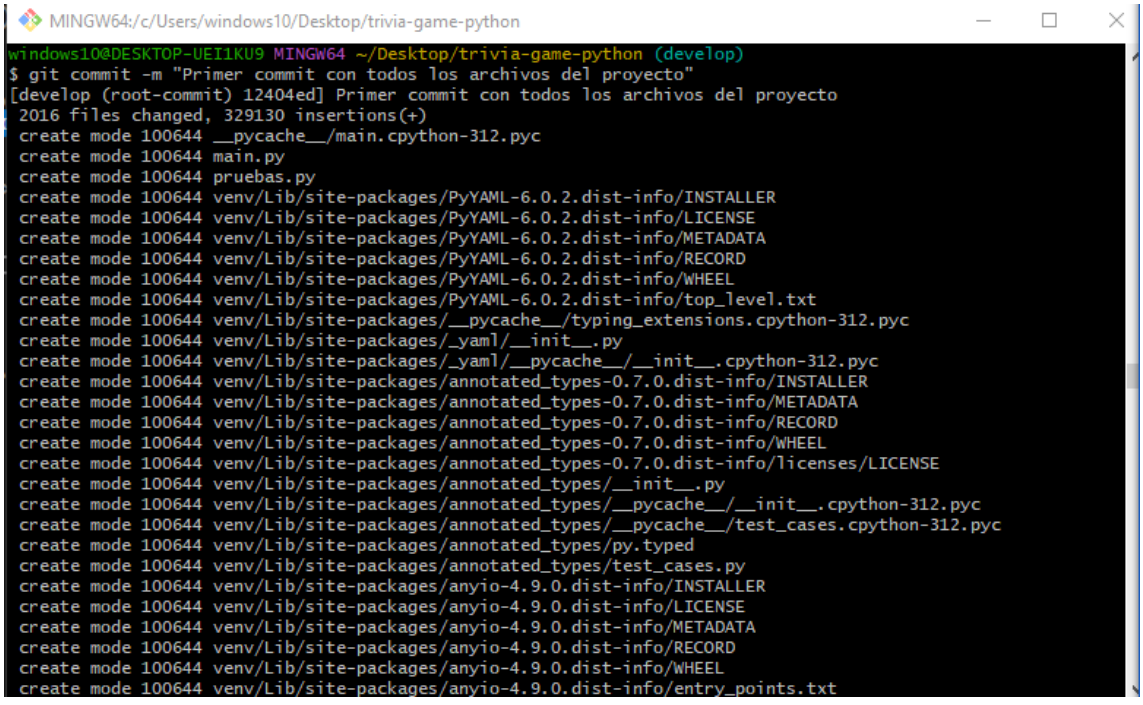
DIA 1:

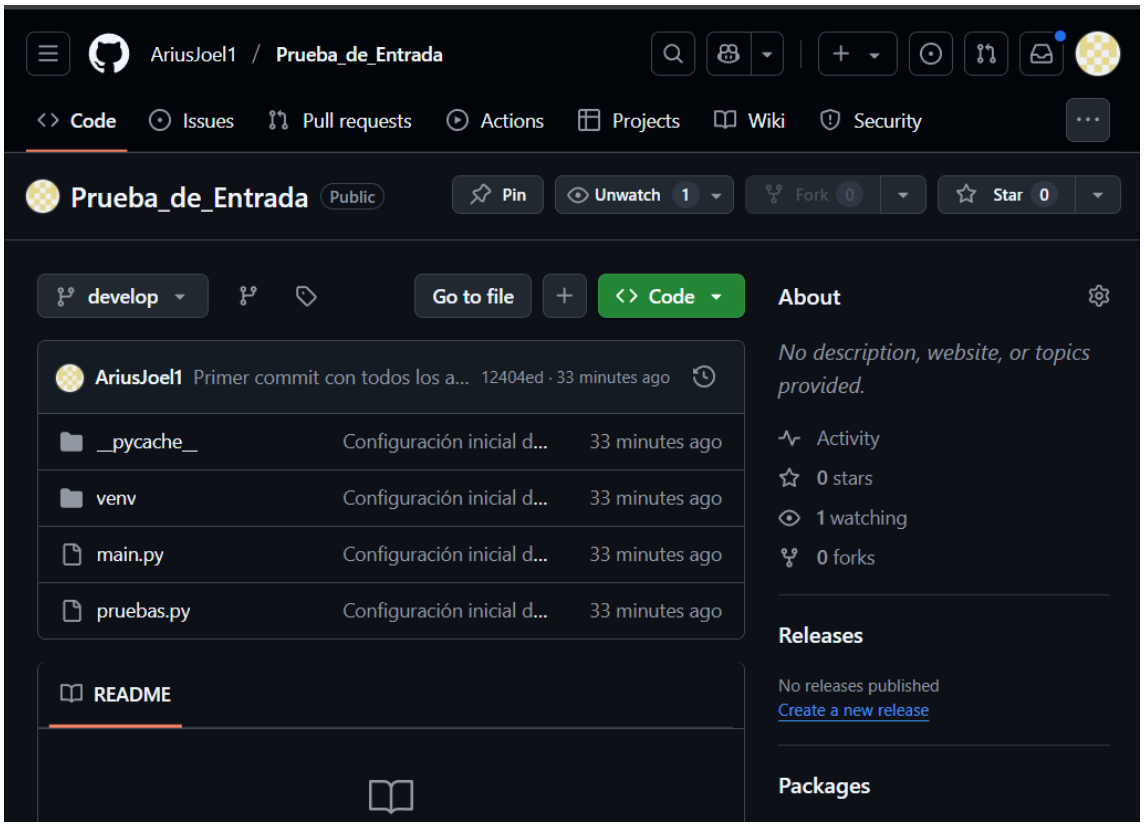
Rama develop y git add .



commit

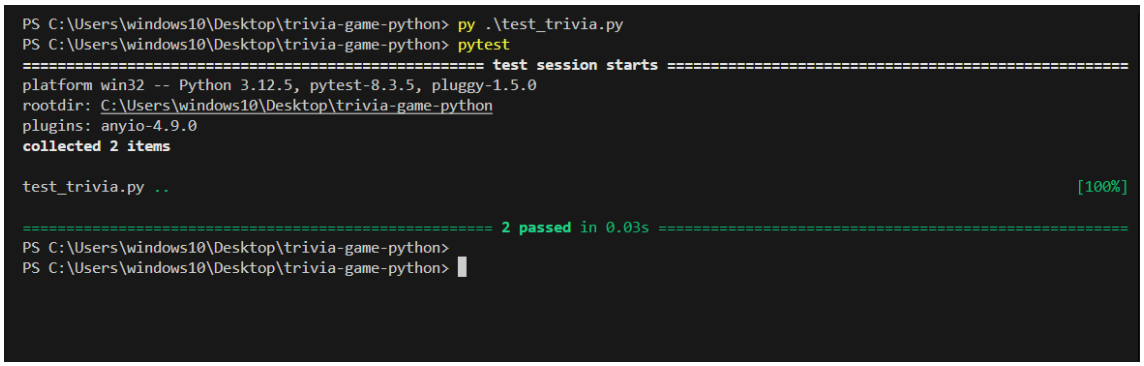


Archivos subidos al github

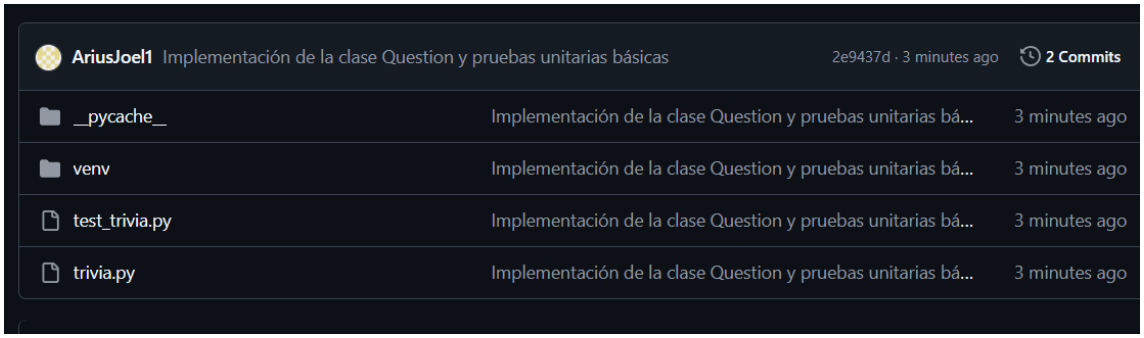


DIA 2

Tests:



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



DIA 3:

Agregando quiz

trivia.py M | test_trivia.py

trivia.py > Quiz > get_next_question

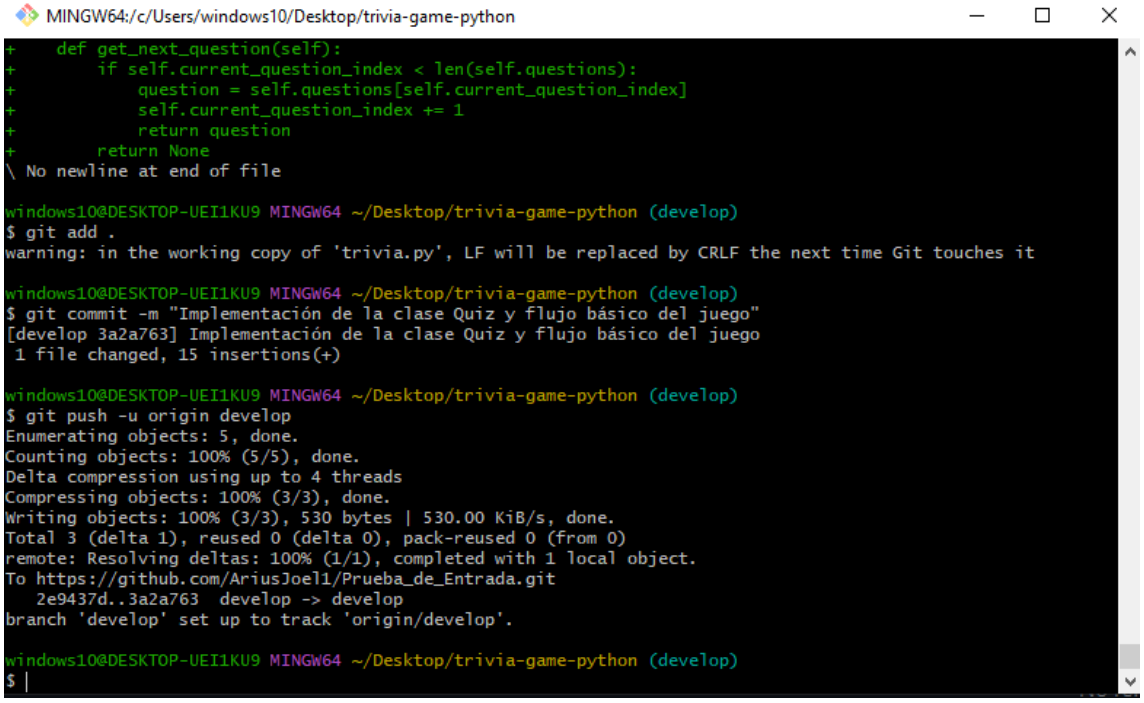
```python
class Question:
    def __init__(self, description, options, correct_answer):
        self.description = description
        self.options = options
        self.correct_answer = correct_answer

    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
```

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



DIA 4:

Modificamos la clase quiz

```python
class Quiz:
    def __init__(self):
        self.questions = []
        self.current_question_index = 0
        self.correct_answers = 0
        self.incorrect_answers = 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

    def answer_question(self, question, answer):
        if question.is_correct(answer):
            self.correct_answers += 1
            return True
        else:
            self.incorrect_answers += 1
            return False
```

Y la prueba de testeo

trivia.py M test_trivia.py M

test_trivia.py > ...

```python
import pytest
from trivia import Question, Quiz



def (variable) question: Question
    question = Question("What is 2 + 2?", ["1", "2", "3", "4"], "4")
    assert question.is_correct("4")

def test_question_incorrect_answer():
    question = Question("What is 2 + 2?", ["1", "2", "3", "4"], "4")
    assert not question.is_correct("2")


def test_quiz_scoring():
    quiz = Quiz()
    question = Question("What is 2 + 2?", ["1", "2", "3", "4"], "4")
    quiz.add_question(question)
    assert quiz.answer_question(question, "4") == True
    assert quiz.correct_answers == 1
```

Implementamos la función run_quiz()

```python
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.description)
        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 nex
t 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 fin
alización del juego"
[develop e268f91] Implementación de sistema de puntuación, manejo de rondas y fi
nalización 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()

```python
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"),
    ]
```

```python
    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

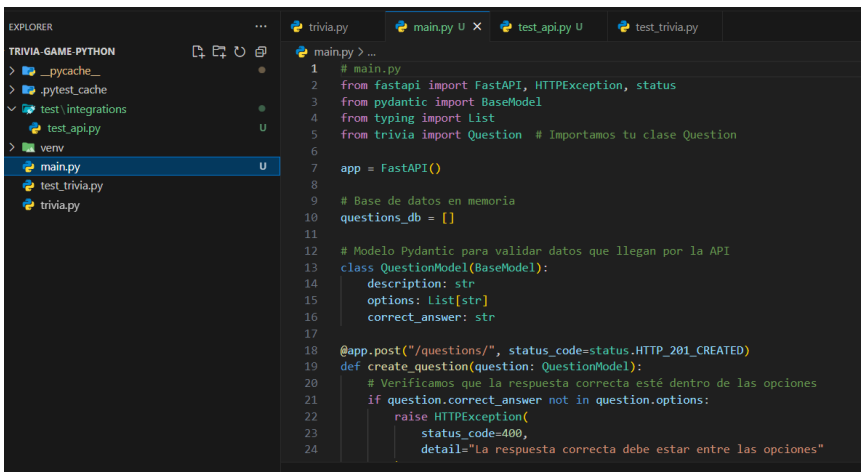
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

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



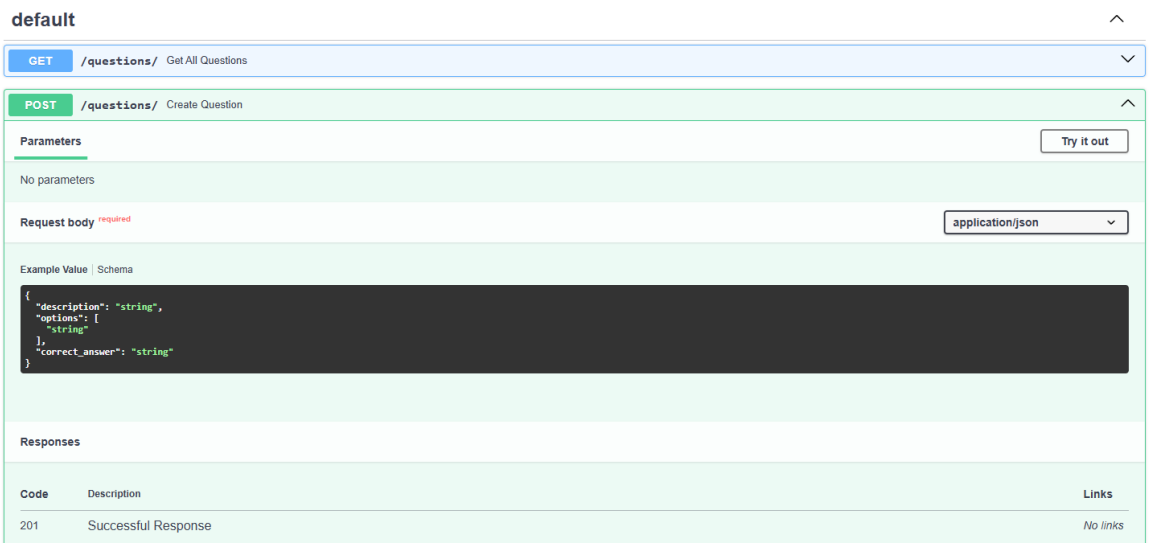
Instalamos las librerias necesarias

```
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
```

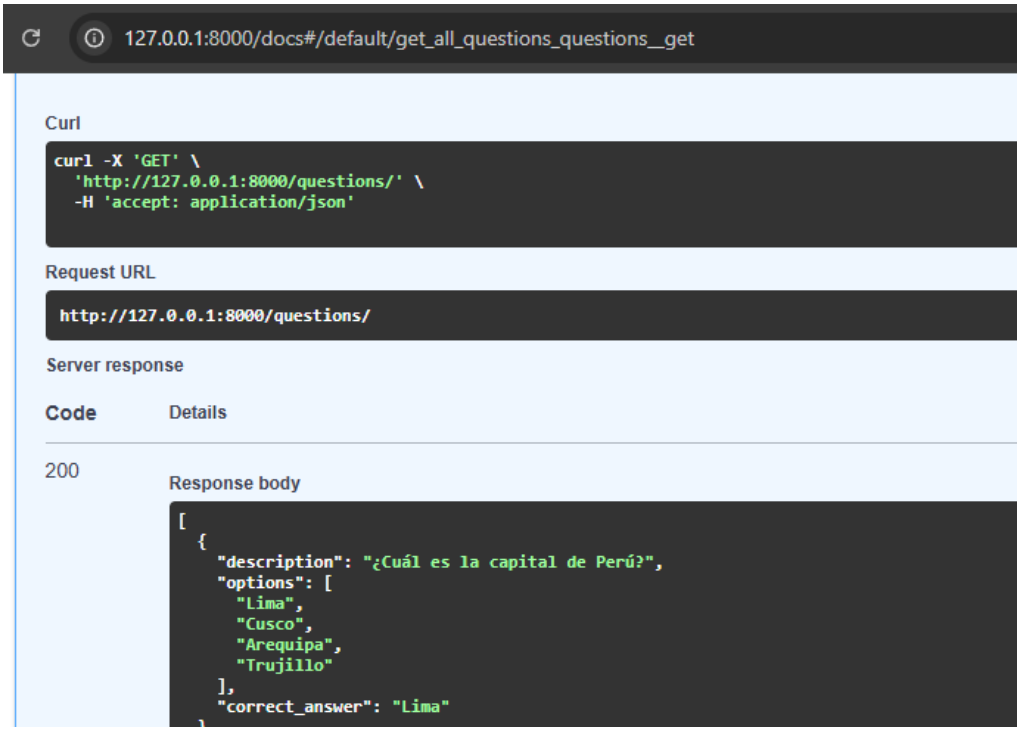
Y ejecutamos

```
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
```

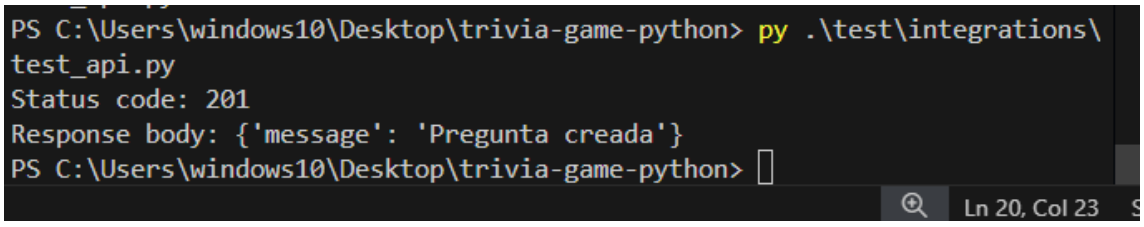
Hacemos los cambios respectivos en el curl



Ejecutamos y subimos las preguntas

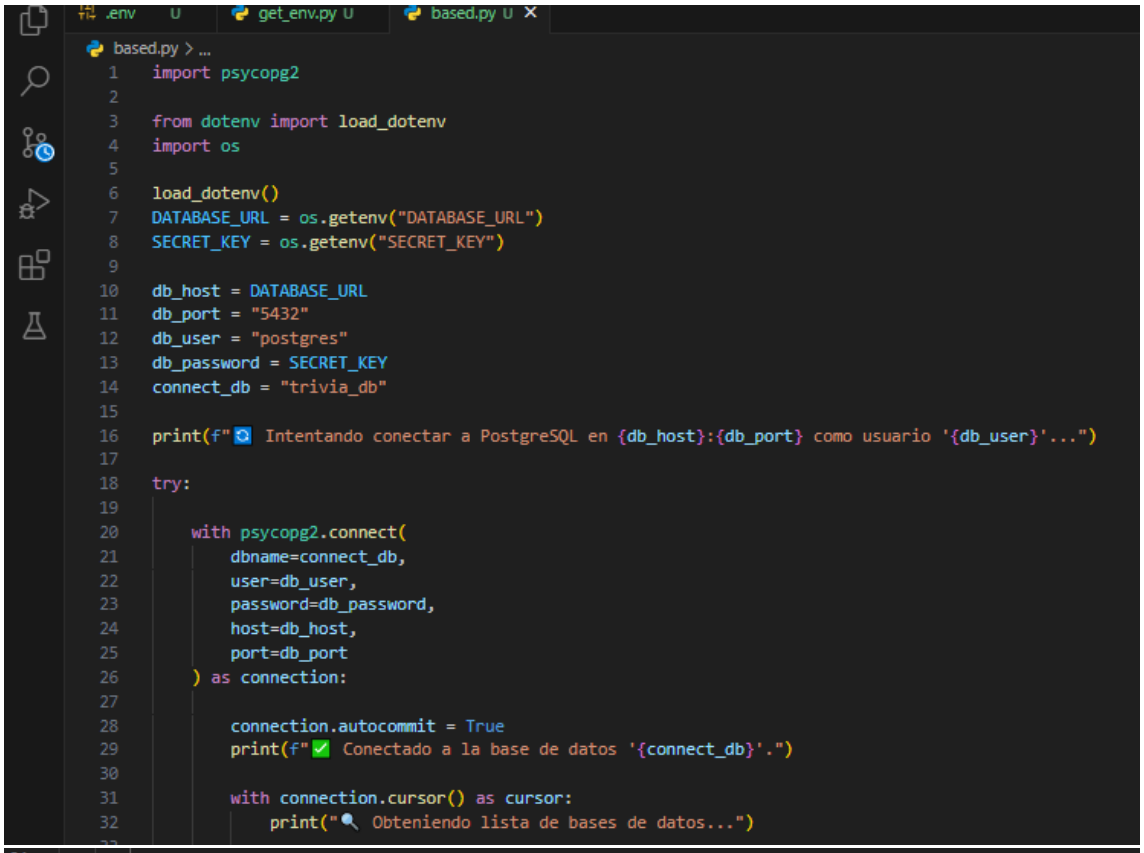


Ejecutando las pruebas



DIA 7:

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



```python
import psycopg2

from dotenv import load_dotenv
import os

load_dotenv()
DATABASE_URL = os.getenv("DATABASE_URL")
SECRET_KEY = os.getenv("SECRET_KEY")

db_host = DATABASE_URL
db_port = "5432"
db_user = "postgres"
db_password = SECRET_KEY
connect_db = "trivia_db"

print(f"🔄 Intentando conectar a PostgreSQL en {db_host}:{db_port} como usuario '{db_user}'...")

try:

    with psycopg2.connect(
        dbname=connect_db,
        user=db_user,
        password=db_password,
        host=db_host,
        port=db_port
    ) as connection:

        connection.autocommit = True
        print(f"✅ Conectado a la base de datos '{connect_db}'.")

        with connection.cursor() as cursor:
            print("🔍 Obteniendo lista de bases de datos...")
```

```python

            list_databases_query = "SELECT datname FROM pg_database;"
            cursor.execute(list_databases_query)


            database_list = cursor.fetchall()

            if database_list:
                print("\n--- Bases de Datos Encontradas ---")
                for db in database_list:

                    print(f"- {db[0]}")
                print("---------------------------------")
                print(f"\n✅ Operación completada. {len(database_list)} bases de datos listadas.")
            else:

                print("🤔 No se encontraron bases de datos.")


except psycopg2.OperationalError as e:
    print("\n❌ Error de Conexión a PostgreSQL:")
    print(f"   {e}")
    print("\n   Por favor, verifica:")
    print(f"   1. Que el servidor PostgreSQL esté corriendo en '{db_host}:{db_port}'.")
    print(f"   2. Que el nombre de usuario ('{db_user}') y la contraseña sean correctos.")
    print(f"   3. Que el host ('{db_host}') sea accesible desde donde ejecutas el script.")
    print(f"   4. Que la base de datos de conexión ('{connect_db}') exista (generalmente 'postgres' sí existe).")
    print(f"   5. Que no haya un firewall bloqueando la conexión al puerto {db_port}.")


except Exception as e:
    print(f"\n❌ Ocurrió un error inesperado: {e}")

finally:
    print("\n🏁 Script finalizado.")
```

Conexcion a postgresql exitosa