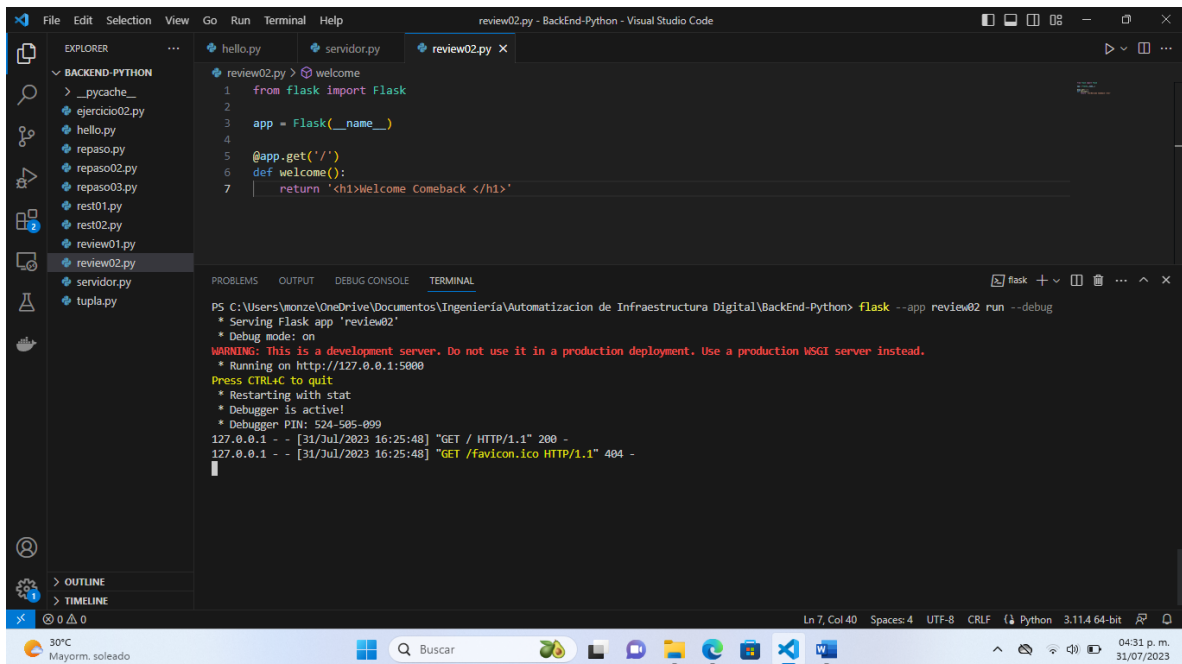


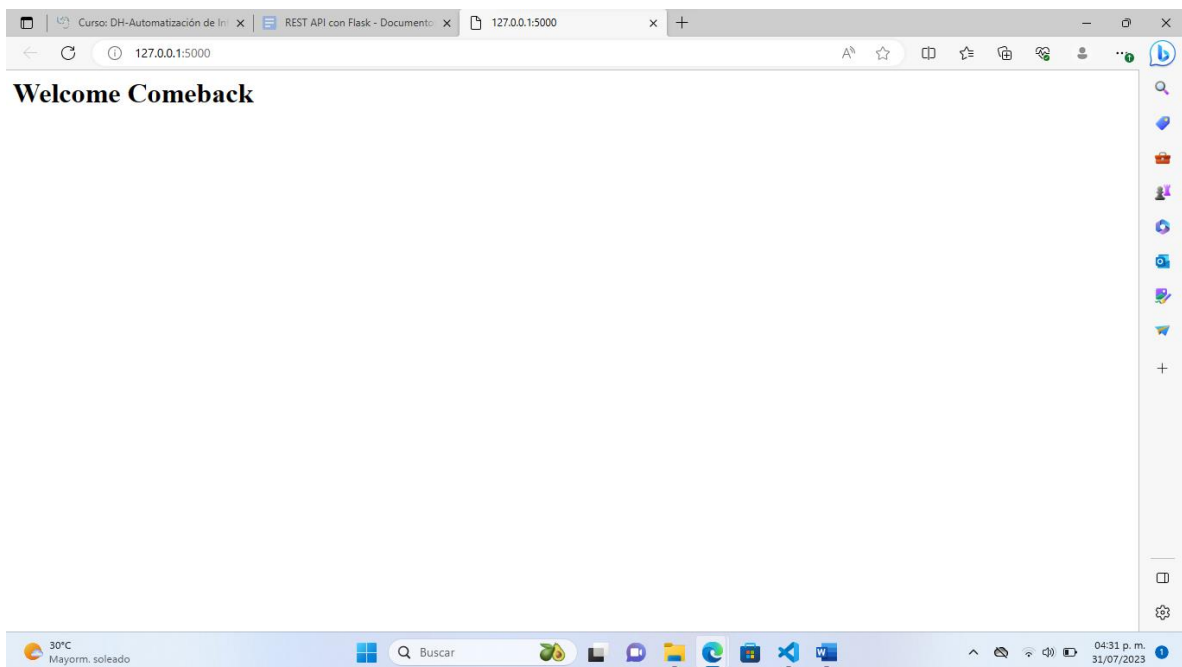
## Repaso



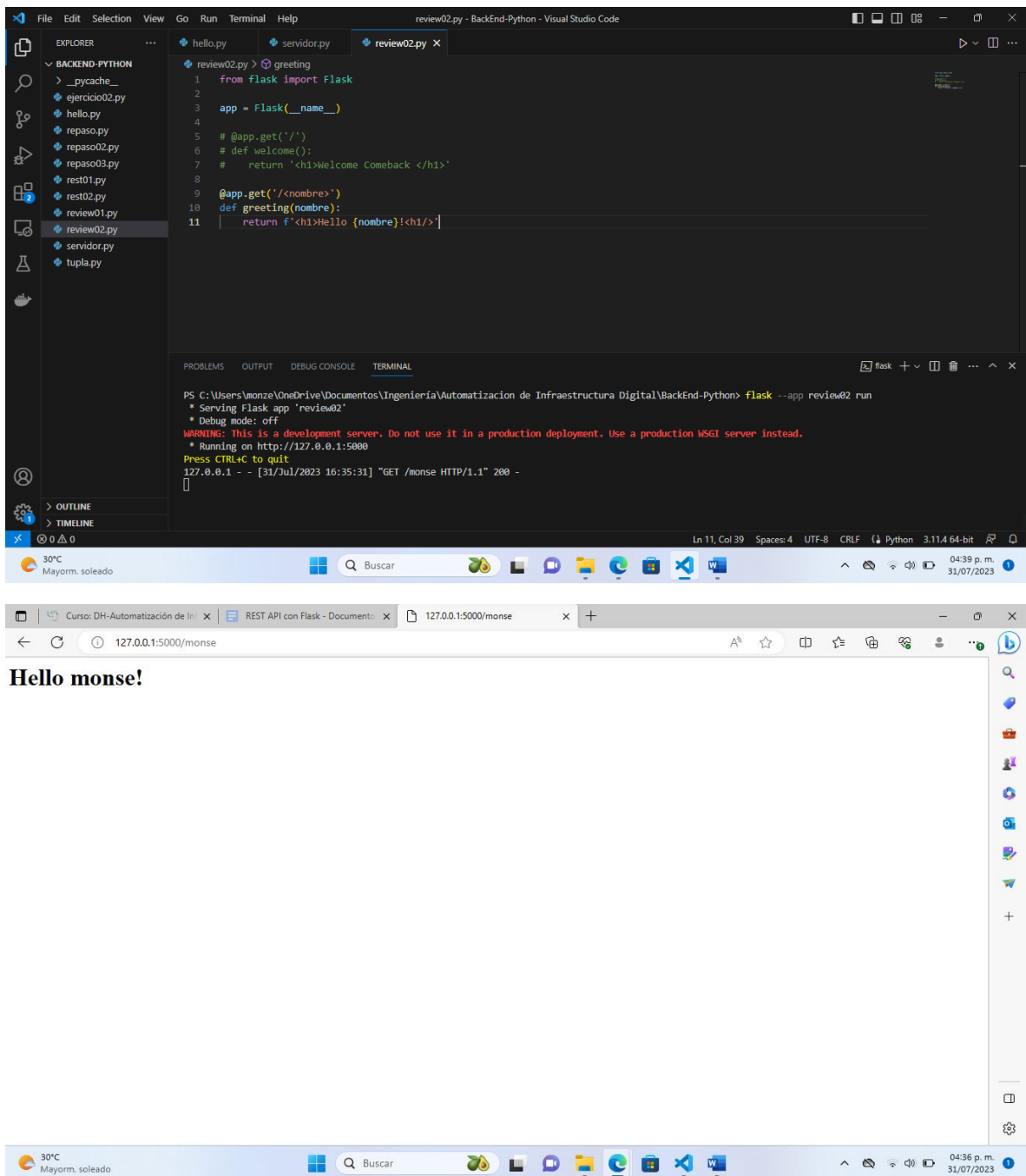
The screenshot shows the Visual Studio Code interface with a Python file named `review02.py` open. The code defines a Flask application with a single route `/'/` that returns the string `<h1>Welcome Comeback </h1>`. The terminal at the bottom shows the command `flask --app review02 run --debug` being executed, which starts the development server on `http://127.0.0.1:5000`. The terminal output includes a warning about using a development server and shows two successful GET requests: one for `/` returning 200 and another for `/favicon.ico` returning 404.

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.get('/')
6 def welcome():
7     return '<h1>Welcome Comeback </h1>'
```

PS C:\Users\monze\OneDrive\Documentos\Ingeniería\Automatización de Infraestructura Digital\BackEnd-Python> flask --app review02 run --debug  
\* Serving Flask app 'review02'  
\* Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
\* Running on http://127.0.0.1:5000  
Press CTRL+C to quit  
\* Restarting with stat  
\* Debugger is active!  
\* Debugger PIN: 524-505-099  
127.0.0.1 - - [31/Jul/2023 16:25:48] "GET / HTTP/1.1" 200 -  
127.0.0.1 - - [31/Jul/2023 16:25:48] "GET /favicon.ico HTTP/1.1" 404 -



## Configuración del puerto 5000/monse



## Configuración del puerto 5000 con nombre y apellido

The image shows a Windows desktop environment. The top half displays the Visual Studio Code editor with a Python file named `review02.py` open. The code is a Flask application that listens on port 5000 and responds to GET requests with a personalized greeting. The terminal window shows the server running and receiving multiple requests from `127.0.0.1`, all returning a 200 status code.

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 # @app.get('/')
6 # def welcome():
7 #     return '<h1>Welcome Comeback </h1>'
8
9 @app.get('/<nombre, apellido>')
10 def greeting(nombre, apellido):
11     return f'<h1>Hello {nombre}{apellido}!</h1>'
```

The bottom half of the image shows a web browser window with the address `127.0.0.1:5000/MonseMartinez`. The browser displays the response: **Hello Monse Martinez!**

# Estudiantes

```
File Edit Selection View Go Run Terminal Help
estudiantes.py - BackEnd-Python - Visual Studio Code

EXPLORER
  BACKEND-PYTHON
    __pycache__
    ejercicio02.py
    estudiantes.py
    hello.py
    repaso.py
    repaso02.py
    repaso03.py
    rest01.py
    rest02.py
    review01.py
    review02.py
    servidor.py
    tupla.py

estudiantes.py
1 from flask import Flask, jsonify, request
2
3 app = Flask(__name__)
4
5 estudiantes = [
6     {'numero_control': '1220100603', 'nombre': 'ANZO AVALOS MARIA CITLALLI'},
7     {'numero_control': '1219100524', 'nombre': 'ARREDONDO GONZALEZ DANIEL ENRIQUE'},
8     {'numero_control': '1220100317', 'nombre': 'DUARTE VELAZQUEZ DANIEL'},
9     {'numero_control': '1220100629', 'nombre': 'GUTIERREZ MARTINEZ VALERIA IVONNE'},
10    {'numero_control': '1220100632', 'nombre': 'LUNA CANTERO ANGEL IVAN'},
11    {'numero_control': '1220100209', 'nombre': 'MARTINEZ RAMIREZ GUADALUPE MONSERRAT'},
12    {'numero_control': '1220100053', 'nombre': 'REVES MORALES SALVADOR'},
13    {'numero_control': '1220100597', 'nombre': 'SALAZAR LEON MARIA GUADALUPE'},
14    {'numero_control': '1220100596', 'nombre': 'TADEO MARTINEZ ALEJANDRO'},
15    {'numero_control': '1220100075', 'nombre': 'TORRES GARCIA JOSE ROGELIO'},
16    {'numero_control': '1220100595', 'nombre': 'TRANQUEÑO HERNANDEZ OSCAR ARMANDO'}
17
18 ]
19
20 @app.get('/estudiantes')
21 def get_estudiantes():
22     return jsonify(estudiantes)
23
24 # @app.get('/estudiantes/<numero_control>')
25 # def get_estudiantes(numero_control):
26 #     pass
27
28 @app.post('/estudiantes')
29 def add_estudiantes():
30     datos = request.get_json()
31     estudiantes.append(datos)
32     return {'success': 'Registro agregado con éxito'}, 201
33
34 if __name__ == '__main__':
35     app.run(debug=True)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

\* Serving Flask app 'estudiantes'  
\* Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
\* Running on http://127.0.0.1:5000  
Press CTRL+C to quit

127.0.0.1 - - [31/Jul/2023 17:26:44] "GET / HTTP/1.1" 404 -  
127.0.0.1 - - [31/Jul/2023 17:26:52] "GET /estudiantes HTTP/1.1" 200 -  
127.0.0.1 - - [31/Jul/2023 17:37:57] "GET /estudiantes HTTP/1.1" 200 -  
127.0.0.1 - - [31/Jul/2023 17:41:13] "POST /estudiantes HTTP/1.1" 201 -

Ln 26, Col 3 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit

```
File Edit Selection View Go Run Terminal Help
estudiantes.py - BackEnd-Python - Visual Studio Code

EXPLORER
  BACKEND-PYTHON
    __pycache__
    ejercicio02.py
    estudiantes.py
    hello.py
    repaso.py
    repaso02.py
    repaso03.py
    rest01.py
    rest02.py
    review01.py
    review02.py
    servidor.py
    tupla.py

estudiantes.py
19
20 @app.get('/estudiantes')
21 def get_estudiantes():
22     return jsonify(estudiantes), 200
23
24 # @app.get('/estudiantes/<numero_control>')
25 # def get_estudiantes(numero_control):
26 #     pass
27
28 @app.post('/estudiantes')
29 def add_estudiantes():
30     datos = request.get_json()
31     estudiantes.append(datos)
32     return {'success': 'Registro agregado con éxito'}, 201
33
34 if __name__ == '__main__':
35     app.run(debug=True)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\wconza\OneDrive\Documents\Ingeniería\Automatización de Infraestructura Digital\BackEnd-Python> flask --app estudiantes run

\* Serving Flask app 'estudiantes'  
\* Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
\* Running on http://127.0.0.1:5000  
Press CTRL+C to quit

127.0.0.1 - - [31/Jul/2023 17:26:44] "GET / HTTP/1.1" 404 -  
127.0.0.1 - - [31/Jul/2023 17:26:52] "GET /estudiantes HTTP/1.1" 200 -  
127.0.0.1 - - [31/Jul/2023 17:37:57] "GET /estudiantes HTTP/1.1" 200 -  
127.0.0.1 - - [31/Jul/2023 17:41:13] "POST /estudiantes HTTP/1.1" 201 -

Ln 26, Col 3 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit

A screenshot of a web browser window displaying a JSON array of student data. The browser's address bar shows the URL `127.0.0.1:5000/estudiantes`. The JSON data is as follows:

```
[
  {
    "nombre": "ANZO AVALOS MARIA CITLALLI",
    "numero_control": 1220100603
  },
  {
    "nombre": "ARREDONDO GONZALEZ DANIEL ENRIQUE",
    "numero_control": 1219100524
  },
  {
    "nombre": "DUARTE VELAZQUEZ DANIEL",
    "numero_control": 1220100317
  },
  {
    "nombre": "GUTIERREZ MARTINEZ VALERIA IVONNE",
    "numero_control": 1220100629
  },
  {
    "nombre": "LUNA CANTERO ANGEL IVAN",
    "numero_control": 1220100632
  },
  {
    "nombre": "MARTINEZ RAMIREZ GUADALUPE MONSERRAT",
    "numero_control": 1220100209
  },
  {
    "nombre": "REYES MORALES SALVADOR",
    "numero_control": 1220100053
  },
  {
    "nombre": "SALAZAR LEON MARIA GUADALUPE",
    "numero_control": 1220100597
  },
  {
    "nombre": "TADEO MARTINEZ ALEJANDRO",
    "numero_control": 1220100596
  },
  {
    "nombre": "TORRES GARCIA JOSE ROGELIO",
    "numero_control": 1220100075
  },
  {
    "nombre": "TRANQUEÑO HERNANDEZ OSCAR ARMANDO",
    "numero_control": 1220100595
  }
]
```

The browser's taskbar at the bottom shows the system clock as 31°C and the location as Soleado.

## Numero de control

A screenshot of the Visual Studio Code editor showing a Python Flask application. The file explorer on the left lists several files, including `estudiantes.py`. The main editor window displays the code for `estudiantes.py`, which defines a Flask application with two endpoints: `/estudiantes` and `/estudiantes/<int:numero_control>`. The code is as follows:

```
9  {
10  {'numero_control':1220100629, 'nombre':'GUTIERREZ MARTINEZ VALERIA IVONNE'},
11  {'numero_control':1220100632, 'nombre':'LUNA CANTERO ANGEL IVAN'},
12  {'numero_control':1220100209, 'nombre':'MARTINEZ RAMIREZ GUADALUPE MONSERRAT'},
13  {'numero_control':1220100053, 'nombre':'REYES MORALES SALVADOR'},
14  {'numero_control':1220100597, 'nombre':'SALAZAR LEON MARIA GUADALUPE'},
15  {'numero_control':1220100596, 'nombre':'TADEO MARTINEZ ALEJANDRO'},
16  {'numero_control':1220100075, 'nombre':'TORRES GARCIA JOSE ROGELIO'},
17  {'numero_control':1220100595, 'nombre':'TRANQUEÑO HERNANDEZ OSCAR ARMANDO'}
18  ]
19
20  @app.get('/estudiantes')
21  def get_estudiantes():
22      return jsonify(estudiantes), 200
23
24  @app.route('/estudiantes/<int:numero_control>', methods=['GET'])
25  def get_estudiante(numero_control):
26      estudiante = next((est for est in estudiantes if est['numero_control'] == numero_control), None)
27      if estudiante:
28          return jsonify(estudiante)
29      else:
30          return jsonify({"message": "Estudiante no encontrado"}), 404
31
```

The terminal window at the bottom shows the output of the application, including the status of the debugger and the results of HTTP requests:

```
* Restarting with stat
* Debugger is active!
* Debugger PIN: 635-427-452
127.0.0.1 - - [09/Aug/2023 16:17:02] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [09/Aug/2023 16:17:12] "GET /1220100209 HTTP/1.1" 404 -
127.0.0.1 - - [09/Aug/2023 16:17:50] "GET /estudiantes/1220100209 HTTP/1.1" 200 -
```

```
1 {  
2   "nombre": "MARTINEZ RAMIREZ GUADALUPE MONSERRAT",  
3   "numero_control": 1220100209  
4 }
```

## POSTMAN - GET

The screenshot shows the Postman application interface. The top bar includes navigation links (Home, Workspaces, API Network, Explore), a search bar, and user options (Invite, Upgrade). The left sidebar shows the 'My Workspace' section with a collection named 'Estudiantes' containing two requests: 'GET New Request' and 'POST New Request'. The main panel displays the 'GET New Request' configuration for the URL 'http://127.0.0.1:5000/estudiantes'. The 'Params' tab is active, showing a table for query parameters. The 'Body' tab is also active, showing the response in 'Pretty' format. The response is a JSON array of three student records.

Key	Value	Description
Key	Value	Description

Body: 200 OK 81 ms 947 B Save as Example

```
33 {  
34   {  
35     "nombre": "TADEO MARTINEZ ALEJANDRO",  
36     "numero_control": 1220100596  
37   },  
38   {  
39     "nombre": "TORRES GARCIA JOSE ROGELIO",  
40     "numero_control": 1220100075  
41   },  
42   {  
43     "nombre": "TRANQUEÑO HERNANDEZ OSCAR ARMANDO",  
44     "numero_control": 1220100595  
45   }  
46 }
```

# POSTMAN - POST

