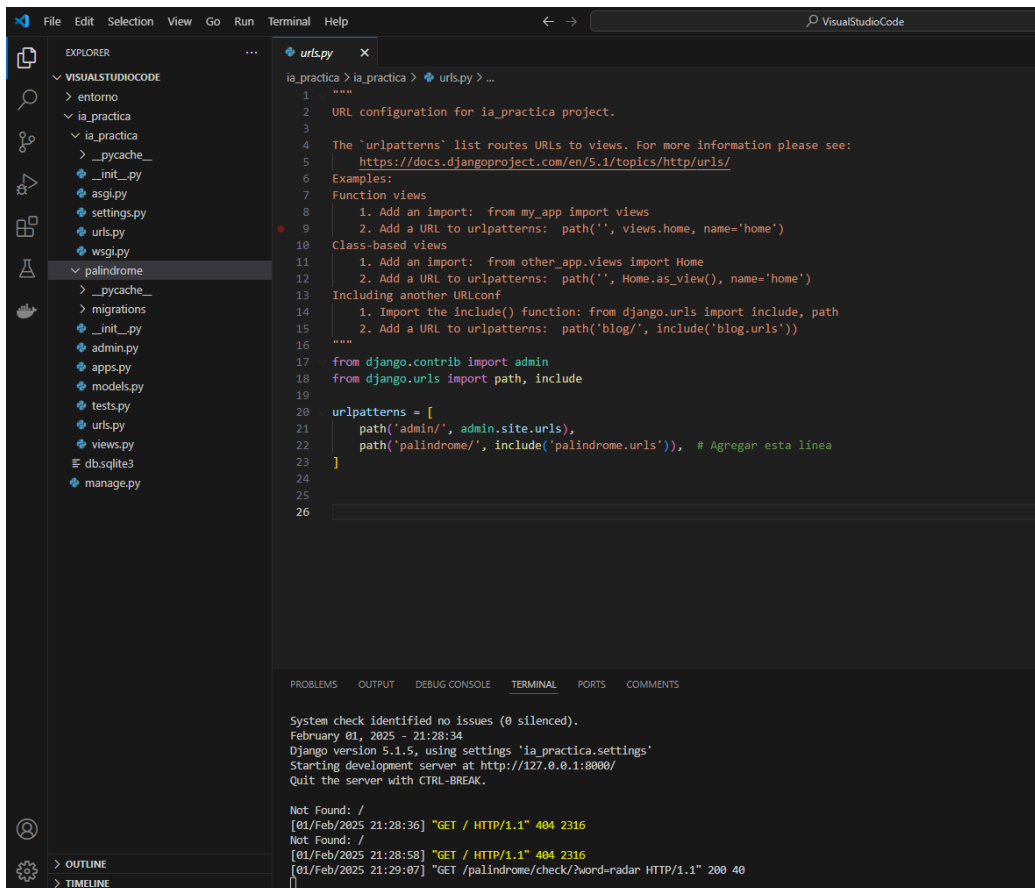


# EJERCICIO GITHUB COPILOT – DJANGO

## Ejercicio #1:



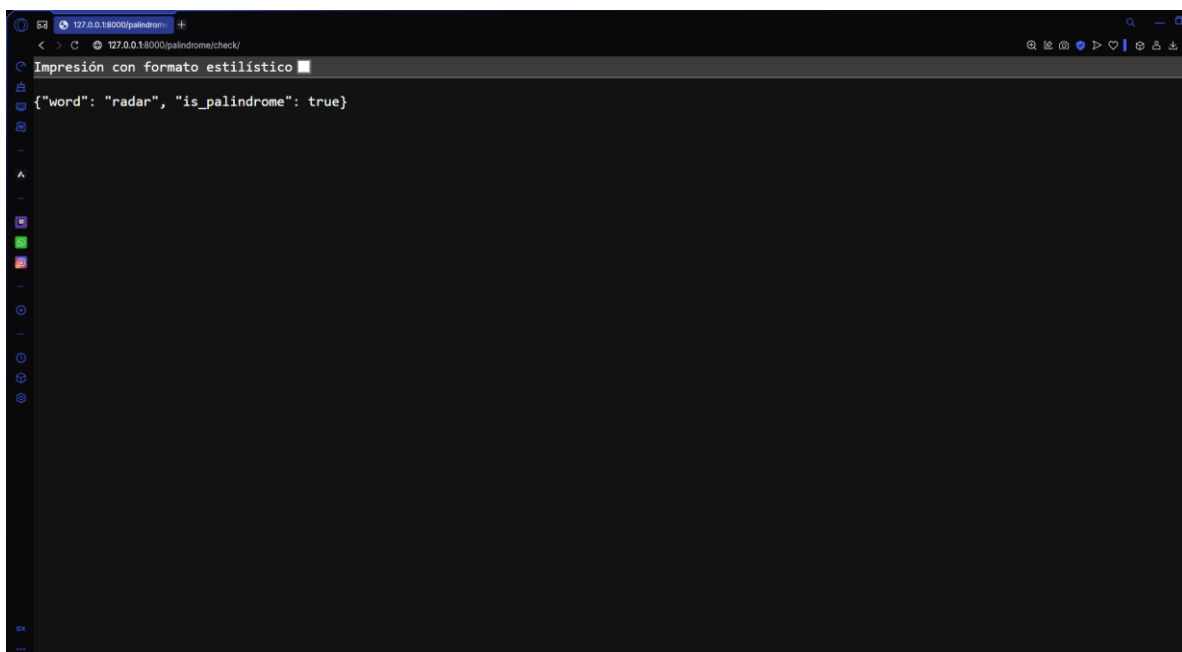
The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the project structure. The main editor shows the `urls.py` file for the `ia_practica` project. The file contains comments and code for configuring URL patterns, including function views and class-based views. The terminal at the bottom shows the output of the Django development server, indicating that the server is running successfully at `http://127.0.0.1:8000/`.

```
ia_practica > ia_practica > urls.py > ...
1  """
2  URL configuration for ia_practica project.
3
4  The 'urlpatterns' list routes URLs to views. For more information please see:
5  https://docs.djangoproject.com/en/5.1/topics/http/urls/
6  Examples:
7  Function views
8      1. Add an import: from my_app import views
9      2. Add a URL to urlpatterns: path('', views.home, name='home')
10 Class-based views
11     1. Add an import: from other_app.views import Home
12     2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 Including another URLconf
14     1. Import the include() function: from django.urls import include, path
15     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path, include
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('palindrome/', include('palindrome.urls')), # Agregar esta línea
23 ]
24
25
26
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

System check identified no issues (0 silenced).  
February 01, 2025 - 21:28:34  
Django version 5.1.5, using settings 'ia\_practica.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.

Not Found: /  
[01/Feb/2025 21:28:36] "GET / HTTP/1.1" 404 2316  
Not Found: /  
[01/Feb/2025 21:28:58] "GET / HTTP/1.1" 404 2316  
[01/Feb/2025 21:29:07] "GET /palindrome/check/?word=radar HTTP/1.1" 200 40



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/palindrome/check/`. The page content shows the result of a GET request to the `palindrome/check/` endpoint, which returns a JSON response: `{"word": "radar", "is_palindrome": true}`.

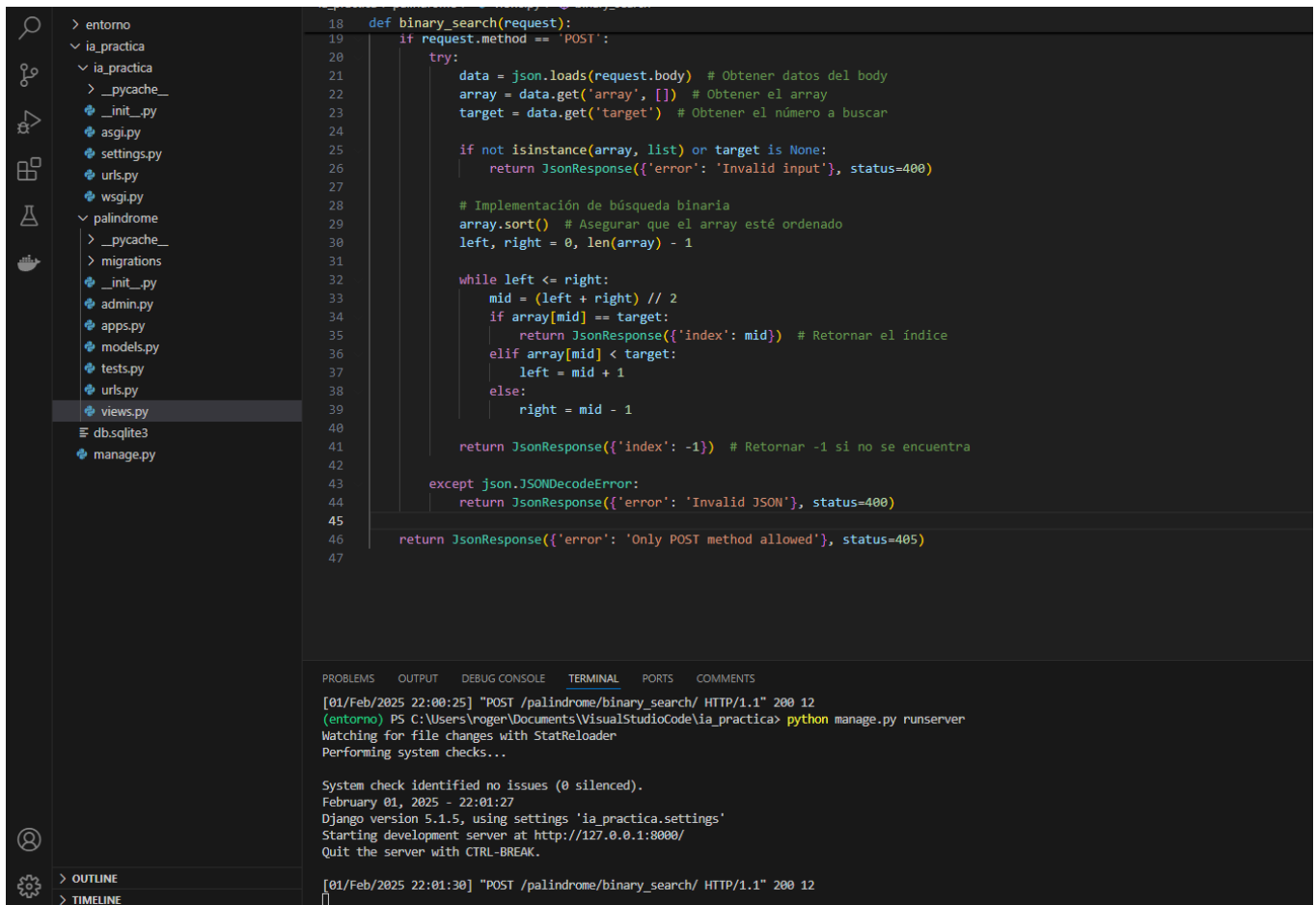
127.0.0.1:8000/palindrome/

Impresión con formato estilístico

```
{"word": "radar", "is_palindrome": true}
```

## Ejercicio #2:

El código que se realizó ordena el array y retorna el índice del número en el array.



The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a folder named 'ia\_practica' containing several files. The code editor shows a Python function named 'binary\_search' that takes a 'request' object as input. The function checks if the request method is 'POST'. If it is, it tries to parse the request body as JSON, extracts the 'array' and 'target' values, and performs a binary search. If the target is found, it returns the index. If not, it returns -1. If the request method is not 'POST', it returns a 405 status code. The code also includes error handling for JSON decoding errors and ensures the array is sorted before searching.

```
18 def binary_search(request):
19     if request.method == 'POST':
20         try:
21             data = json.loads(request.body) # Obtener datos del body
22             array = data.get('array', []) # Obtener el array
23             target = data.get('target') # Obtener el número a buscar
24
25             if not isinstance(array, list) or target is None:
26                 return JsonResponse({'error': 'Invalid input'}, status=400)
27
28             # Implementación de búsqueda binaria
29             array.sort() # Asegurar que el array esté ordenado
30             left, right = 0, len(array) - 1
31
32             while left <= right:
33                 mid = (left + right) // 2
34                 if array[mid] == target:
35                     return JsonResponse({'index': mid}) # Retornar el índice
36                 elif array[mid] < target:
37                     left = mid + 1
38                 else:
39                     right = mid - 1
40
41             return JsonResponse({'index': -1}) # Retornar -1 si no se encuentra
42
43         except json.JSONDecodeError:
44             return JsonResponse({'error': 'Invalid JSON'}, status=400)
45
46     return JsonResponse({'error': 'Only POST method allowed'}, status=405)
47
```

The terminal output shows the following messages:

```
[01/Feb/2025 22:00:25] "POST /palindrome/binary_search/ HTTP/1.1" 200 12
(entorno) PS C:\Users\roger\Documents\VisualStudioCode\ia_practica> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 01, 2025 - 22:01:27
Django version 5.1.5, using settings 'ia_practica.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[01/Feb/2025 22:01:30] "POST /palindrome/binary_search/ HTTP/1.1" 200 12
```

## HOPPSCOTCH



POST Untitled



POST



http://127.0.0.1:8000/palindrome/binary\_search/



Parámetros

Cuerpo

Encabezados

Autorización

Script previo a la solicitud



Tipo de contenido

application/json



Anular

cuerpo sin procesar

```
1 {  
2   "array": [10, 3, 7, 20, 15],  
3   "target": 7  
4 }
```

Estado: 200 • OK

Tiempo: 15 ms

Tamaño: 12 B

JSON

Sin procesar

Encabezados 8

Resultados de las pruebas

Cuerpo de respuesta

```
1 {  
2   "index": 1  
3 }
```