

# Copiloto

Natalia Carolina Bautista Sanchez

[nabautista@unal.edu.co](mailto:nabautista@unal.edu.co)

## Ejercicios

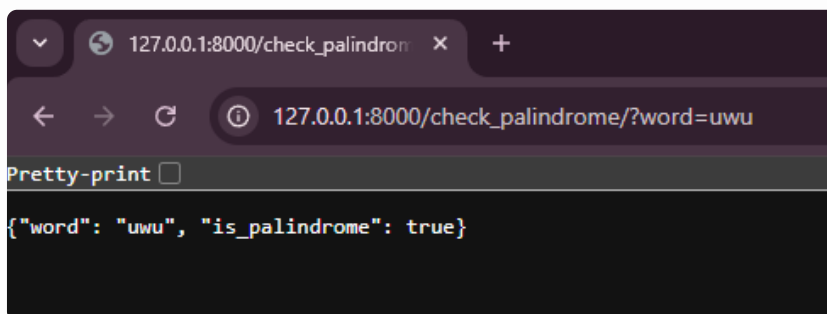
- Una URL que verifique si una cadena es un palíndromo o no, el método debe ser GET (no hay restricción de cómo pasa la palabra)
- Una URL que implemente la búsqueda binaria, el método debe ser POST y debe recibir el array y el dato a buscar por medio del Body. (casi todos usaron números)

## Palíndromo

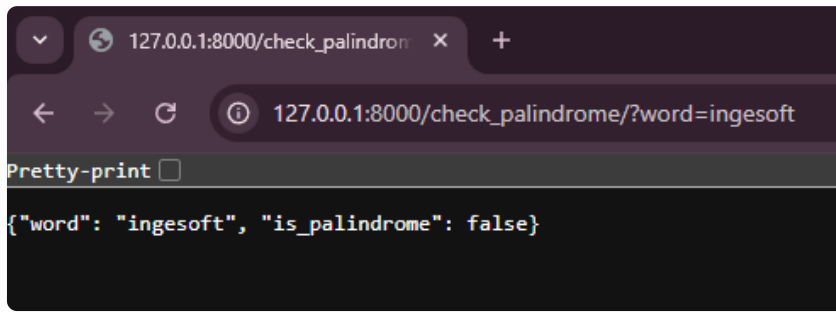
Para el punto del palíndromo este fue el código generado por Copilot.

```
ia_practica > ia_practica > views.py > BinarySearchView > post
1 from django.http import JsonResponse, HttpResponse
2 from django.views import View
3 from django.utils.decorators import method_decorator
4 from django.views.decorators.csrf import csrf_exempt
5 import json
6
7 @method_decorator(csrf_exempt, name='dispatch')
8 class PalindromeCheckView(View):
9     def get(self, request):
10         word = request.GET.get('word', '')
11         is_palindrome = word == word[::-1]
12         return JsonResponse({'word': word, 'is_palindrome': is_palindrome})
```

Así se ve la respuesta de una palabra palíndroma



Y así se ve la respuesta de una no palíndroma



## Binary Search

Para la binary search este fue el código generado por Copilot.

```
ia_practica > ia_practica > views.py > BinarySearchView > post
14 @method_decorator(csrf_exempt, name='dispatch')
15 class BinarySearchView(View):
16     def post(self, request):
17         try:
18             data = json.loads(request.body)
19             array = data.get('array', [])
20             target = data.get('target', None)
21
22             if target is None or not isinstance(array, list):
23                 return JsonResponse({'error': 'Invalid input'}, status=400)
24
25             array.sort()
26             left, right = 0, len(array) - 1
27             found = False
28
29             while left <= right:
30                 mid = (left + right) // 2
31                 if array[mid] == target:
32                     found = True
33                     break
34                 elif array[mid] < target:
35                     left = mid + 1
36                 else:
37                     right = mid - 1
38
39             return JsonResponse({'array': array, 'target': target, 'found': found})
40         except json.JSONDecodeError:
41             return JsonResponse({'error': 'Invalid JSON'}, status=400)
```

Como se requería enviar las peticiones por post, usé postman.

Así se ve la petición y la respuesta

POST http://127.0.0.1:8000/b/ +

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

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

Params Authorization Headers (9) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {  
2   "array": [1, 2, 3, 4, 5],  
3   "target": 3  
4 }
```

Body Cookies Headers (8) Test Results

{ } JSON Preview Visualize

```
1 {  
2   "array": [  
3     1,  
4     2,  
5     3,  
6     4,  
7     5  
8   ],  
9   "target": 3,  
10  "found": true  
11 }
```