

Endpoint en Django

Presentado por:

Sergio Ruiz – seruizh@unal.edu.co

Profesor:

Oscar Eduardo Alvarez Rodriguez

01/02/2024



Universidad Nacional de Colombia

Facultad de Ingeniería

Ingeniería de Software

Departamento de Ingeniería de Sistemas

Endpoint en Django

1. 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)

Creación

```
ia_practica > api > views.py > busqueda_binaria
1  from django.http import JsonResponse
2
3  def es_palindromo(request):
4      palabra = request.GET.get('palabra', '') # Obtiene la palabra desde la URL
5      es_palindromo = palabra == palabra[::-1] # Verifica si es un palíndromo
6      return JsonResponse({'palabra': palabra, 'es_palindromo': es_palindromo})
7
1  from django.urls import path
2  from .views import es_palindromo, busqueda_binaria
3
4  urlpatterns = [
5      path('palindromo/', es_palindromo, name='es_palindromo'),
6      path('busqueda-binaria/', busqueda_binaria, name='busqueda_binaria'),
7  ]
```

Uso en el navegador

← → ↻ ⓘ 127.0.0.1:8000/palindromo/?palabra=radar

Impresión con formato estilístico ☒

```
{
  "palabra": "radar",
  "es_palindromo": true
}
```

← → ↻ ⓘ 127.0.0.1:8000/palindromo/?palabra=entonces

Impresión con formato estilístico ☒

```
{
  "palabra": "entonces",
  "es_palindromo": false
}
```

← → ↻ ⓘ 127.0.0.1:8000/palindromo/?palabra=anatana

Impresión con formato estilístico ☒

```
{
  "palabra": "anatana",
  "es_palindromo": true
}
```

Respuesta en consola

```
[01/Feb/2025 18:34:20] "GET /palindromo/?palabra=radar HTTP/1.1" 200 43
[01/Feb/2025 18:35:19] "GET /palindromo/?palabra=entonces HTTP/1.1" 200 47
[01/Feb/2025 18:36:24] "GET /palindromo/?palabra=anatana HTTP/1.1" 200 45
```

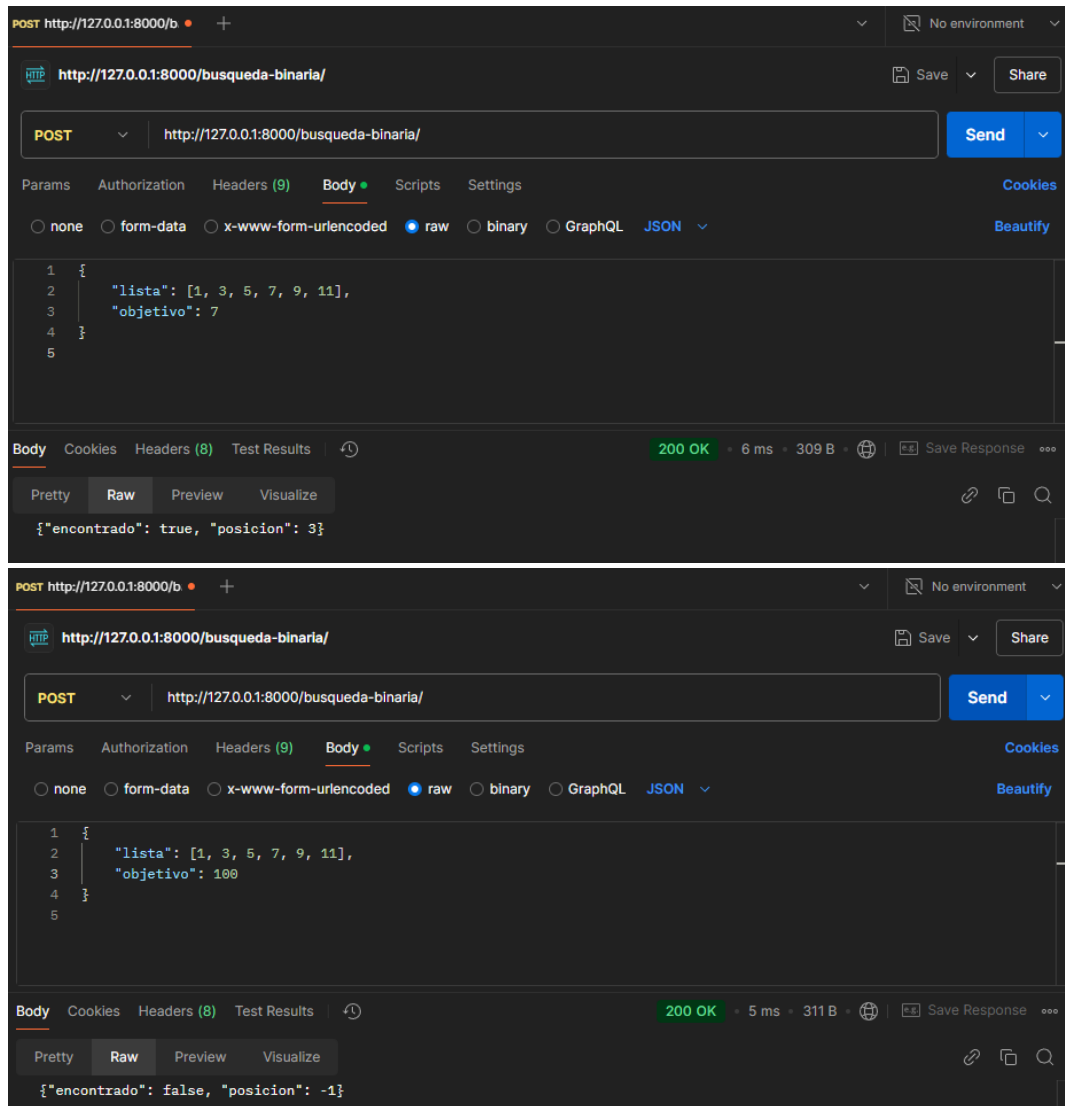
2. Una URL que implemente la búsqueda binaria, el método debe ser POST y debe recibir el array y el dato a buscar por medio del Body. (casi todos usaron números)

Creación

```
12 @csrf_exempt
13 def busqueda_binaria(request):
14     if request.method == "POST":
15         try:
16             data = json.loads(request.body) # Obtiene los datos del cuerpo de la petición
17             lista = data.get('lista', []) # Extrae el array
18             objetivo = data.get('objetivo', None) # Extrae el número a buscar
19
20             # Algoritmo de búsqueda binaria
21             lista.sort() # Asegurarse de que la lista esté ordenada
22             izquierda, derecha = 0, len(lista) - 1
23
24             while izquierda <= derecha:
25                 medio = (izquierda + derecha) // 2
26                 if lista[medio] == objetivo:
27                     return JsonResponse({'encontrado': True, 'posicion': medio})
28                 elif lista[medio] < objetivo:
29                     izquierda = medio + 1
30                 else:
31                     derecha = medio - 1
32
33             return JsonResponse({'encontrado': False, 'posicion': -1})
34
35         except json.JSONDecodeError:
36             return JsonResponse({'error': 'Formato JSON inválido'}, status=400)
37
38     return JsonResponse({'error': 'Método no permitido'}, status=405)
```

```
1 from django.urls import path
2 from .views import es_palindromo, busqueda_binaria
3
4 urlpatterns = [
5     path('palindromo/', es_palindromo, name='es_palindromo'),
6     path('busqueda-binaria/', busqueda_binaria, name='busqueda_binaria'),
7 ]
```

Uso en Postman



Respuesta en consola

```
[01/Feb/2025 18:40:56] "POST /busqueda-binaria/ HTTP/1.1" 200 35
[01/Feb/2025 18:42:18] "POST /busqueda-binaria/ HTTP/1.1" 200 37
```