

Ejercicio palíndromo:

```
← → ↻ ⓘ 127.0.0.1:8000/palindromo/?palabra=radar
Dar formato al texto ☐
{"palabra": "radar", "es_palindromo": true}
```

Ejercicio búsqueda binaria:

```
PS C:\Users\Estudiante> Invoke-WebRequest -Uri "http://127.0.0.1:8000/busqueda-binaria/" `
>> -Method POST `
>> -Headers @{ "Content-Type" = "application/json" } `
>> -Body '{"arreglo": [10,3,5,8,2,7], "objetivo": 5}' `
>> -UseBasicParsing

StatusCode      : 200
StatusDescription : OK
Content         : {"arreglo": [2, 3, 5, 7, 8, 10], "objetivo": 5, "encontrado": true, "posicion": 2}
RawContent      : HTTP/1.1 200 OK
                  X-Frame-Options: DENY
                  X-Content-Type-Options: nosniff
                  Referrer-Policy: same-origin
                  Cross-Origin-Opener-Policy: same-origin
                  Content-Length: 82
                  Content-Type: application/json
                  Date...
Forms           :
Headers         : {[X-Frame-Options, DENY], [X-Content-Type-Options, nosniff], [Referrer-Policy, same-origin],
                  [Cross-Origin-Opener-Policy, same-origin]...}
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      :
RawContentLength : 82
```


Codigo usado

Funciones:

```
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
import json

@csrf_exempt # Evita problemas con CSRF en pruebas (en producción,
manejar de otra formac

def es_palindromo(request):
    palabra = request.GET.get('palabra', '')

    if not palabra:
        return JsonResponse({'error': 'No se proporcionó ninguna
palabra'}, status=400)

    palabra = palabra.lower().replace(" ", "") # Convertir a
minúsculas y quitar espacios
    es_palindromo = palabra == palabra[::-1]

    return JsonResponse({'palabra': palabra, 'es_palindromo':
es_palindromo})

def busqueda_binaria(request):
    if request.method != 'POST':
        return JsonResponse({'error': 'Método no permitido. Usa POST'},
status=405)

    try:
        data = json.loads(request.body) # Convertir el JSON recibido a
un diccionario
        arreglo = data.get('arreglo', [])
        objetivo = data.get('objetivo', None)

        if not isinstance(arreglo, list) or objetivo is None:
            return JsonResponse({'error': 'Datos inválidos. Se requiere
un arreglo y un número objetivo'}, status=400)

        # Asegurar que el arreglo esté ordenado para la búsqueda
binaria
        arreglo.sort()
```

```

# Implementación de búsqueda binaria
izquierda, derecha = 0, len(arreglo) - 1
encontrado = False
posicion = -1

while izquierda <= derecha:
    medio = (izquierda + derecha) // 2
    if arreglo[medio] == objetivo:
        encontrado = True
        posicion = medio
        break
    elif arreglo[medio] < objetivo:
        izquierda = medio + 1
    else:
        derecha = medio - 1

    return JsonResponse({'arreglo': arreglo, 'objetivo': objetivo,
'encontrado': encontrado, 'posicion': posicion})

except json.JSONDecodeError:
    return JsonResponse({'error': 'Formato JSON inválido'},
status=400)

```

Urls:

```

from django.contrib import admin
from django.urls import path
from .View import es_palindromo, busqueda_binaria

urlpatterns = [
    path('admin/', admin.site.urls),
    path('busqueda-binaria/', busqueda_binaria,
name='busqueda_binaria'),
    path('palindromo/', es_palindromo, name='verificar_palindromo')
]

```