ACTIVIDAD COPILOT

Presentado por:

Nicolas Quezada Mora - *nquezada@unal.edu.co*

Profesor:

Oscar Eduardo Alvarez Rodriguez - *oalvarezr@unal.edu.co*

Sabado 01 de Febrero

Universidad Nacional de Colombia

Facultad de Ingeniería

Departamento de Industrial y Sistemas

2025
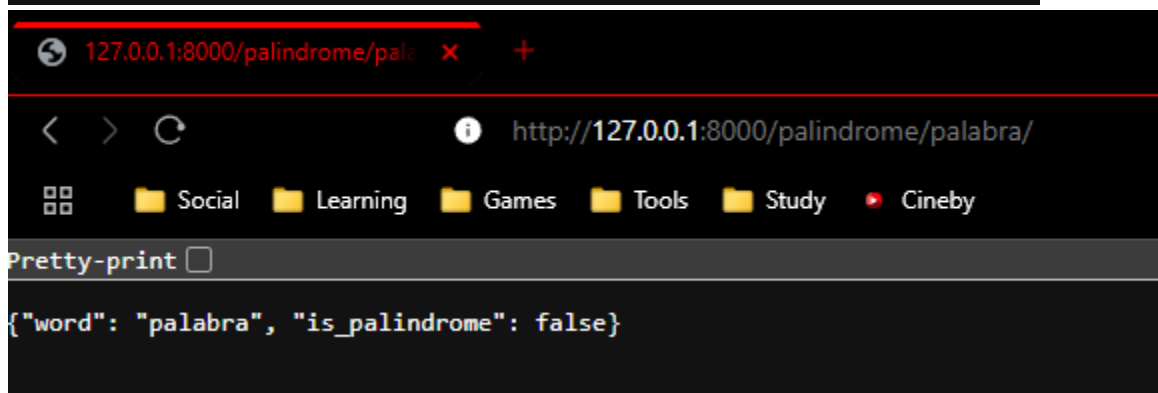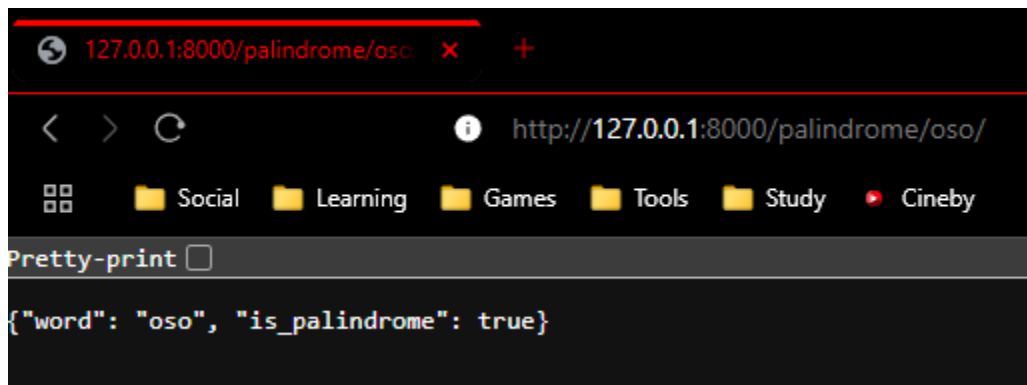
- Una URL que verifique si una cadena es un palíndromo o no, el método debe ser GET (no hay restricción de cómo pasa la palabra)

```python
from django.http import JsonResponse

def check_palindrome(request, word):
    is_palindrome = word == word[::-1]
    response_data = {
        'word': word,
        'is_palindrome': is_palindrome
    }
    return JsonResponse(response_data)
```

```python
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from . import views

urlpatterns = [
    path('admin/', admin.site.urls),
]
urlpatterns += [
    path('palindrome/<str:word>/', views.check_palindrome, name='check_palindrome'),
]
```

- 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)

```python
1   from django.http import JsonResponse
2   from django.views.decorators.http import require_POST
3   import json
4
5   @require_POST
6   def binary_search(request):
7       try:
8           data = json.loads(request.body)
9           array = data.get('array', [])
10          target = data.get('target')
11
12          if not isinstance(array, list) or not isinstance(target, (int, float)):
13              return JsonResponse({'error': 'Invalid input'}, status=400)
14
15          original_array = array.copy()
16          array.sort()
17          left, right = 0, len(array) - 1
18          found = False
19          position = -1
20          iterations = 0
21
22          while left <= right:
23              iterations += 1
24              mid = (left + right) // 2
25              if array[mid] == target:
26                  found = True
27                  position = mid
28                  break
29              elif array[mid] < target:
30                  left = mid + 1
```

```python
def binary_search(request):
            else:
                right = mid - 1

        original_position = original_array.index(target) if found else -1

        response_data = {
            'original_array': original_array,
            'sorted_array': array,
            'target': target,
            'found': found,
            'position_in_sorted_array': position,
            'position_in_original_array': original_position,
            'iterations': iterations
        }
        return JsonResponse(response_data)

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

ia_practiva > ia_practiva > 🐍 urls.py > ...
```python
1    """
2    URL configuration for ia_practiva 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
19   from . import views
20
21
22   urlpatterns = [
23       path('admin/', admin.site.urls),
24       path('binary_search/', views.binary_search, name='binary_search'),
25   ]
```

```
Django version 5.1.5, using settings 'ia_practiva.settings'
Django version 5.1.5, using settings 'ia_practiva.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
Quit the server with CTRL-BREAK.

[01/Feb/2025 08:35:45] "POST /binary_search/ HTTP/1.1" 200 171
```

POST http://localhost:8000/l ●   +   No environment

HTTP   http://localhost:8000/binary_search/   💾 Save ⌄   Share

POST ⌄   http://localhost:8000/binary_search/   Send ⌄

Params   Authorization   Headers (9)   Body ●   Scripts   Settings   Cookies

○ none   ○ form-data   ○ x-www-form-urlencoded   ● raw   ○ binary   ○ GraphQL   JSON ⌄   Beautify

```json
1  {
2      "array": [5, 2, 9, 1],
3      "target": 2
4  }
```

Body   Cookies   Headers (8)   Test Results   ⟳   200 OK · 15 ms · 446 B · ⊕   ⚬⚬⚬

{ } JSON   ▷ Preview   ⟡ Visualize ⌄   🔗

> sorted_array [4]

target	2
found	true
position_in_sorted_array	1
position_in_original_array	1
iterations	1

Request 1 (top):

```
POST  http://localhost:8000/binary_search/          Send

Params  Authorization  Headers (9)  Body  Scripts  Settings          Cookies
none  form-data  x-www-form-urlencoded  raw  binary  GraphQL  JSON          Beautify

1  {
2      "array": [5, 2, 10, 9, 1,11,15,3,4],
3      "target": 3
```

Body Cookies Headers (8) Test Results   200 OK · 5 ms · 482 B

{} JSON ▷ Preview Visualize

Root / **sorted_array**

0	1
1	2
2	3
3	4
4	5
5	9
6	10
7	11
8	15

Request 2 (bottom):

```
POST  http://localhost:8000/binary_search/          Send

Params  Authorization  Headers (9)  Body  Scripts  Settings          Cookies
none  form-data  x-www-form-urlencoded  raw  binary  GraphQL  JSON          Beautify

1  {
2      "array": [5, 2, 10, 9, 1,11,15,3,4],
3      "target": 3
```

Body Cookies Headers (8) Test Results   200 OK · 5 ms · 482 B

{} JSON ▷ Preview Visualize

> **original_array** [9]
> **sorted_array** [9]

target	3
found	true
position_in_sorted_array	2
position_in_original_array	7
iterations	3