# Parte 4 – Como o Django resolve URLs (Exemplo real do projeto Pr. Albino Marks)

Imagine que um usuário acessa a URL:

https://localhost:8000/artigos/

## 1. O Django começa pelo urls.py do projeto

Arquivo: pralbinomarks/urls.py  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('', views.home, name='home'),  
 path('artigos/', include(('A\_Lei\_no\_NT.urls', 'A\_Lei\_no\_NT'), namespace='A\_Lei\_no\_NT')),  
]  
  
→ O Django detecta que a URL começa com 'artigos/'  
→ Ele repassa o restante do caminho (que neste caso é vazio) para o arquivo A\_Lei\_no\_NT/urls.py

## 2. O Django avalia o urls.py do app

Arquivo: A\_Lei\_no\_NT/urls.py  
  
urlpatterns = [  
 path('', views.lista\_artigos, name='lista\_artigos'),  
 path('<slug:slug>/', views.detalhe\_artigo, name='artigo\_detalhe'),  
]  
  
→ Como o caminho restante é '', ele casa com o path('')  
→ O Django chama a função views.lista\_artigos(request)

## 3. O template usa o namespace corretamente

Dentro de um arquivo HTML:  
  
<a href="{% url 'A\_Lei\_no\_NT:lista\_artigos' %}">Ver artigos</a>  
  
→ O Django procura no namespace A\_Lei\_no\_NT  
→ Acha a view com name='lista\_artigos'  
→ Gera o link final como /artigos/

## 4. Quando acessamos um artigo específico

URL: https://localhost:8000/artigos/o-amor-e-a-lei/  
  
→ O Django envia o slug 'o-amor-e-a-lei' para o path:  
 path('<slug:slug>/', views.detalhe\_artigo, name='artigo\_detalhe')  
  
→ A função chamada será:  
 views.detalhe\_artigo(request, slug='o-amor-e-a-lei')

## Resumo visual do fluxo

URL: /artigos/  
↓  
urls.py do projeto detecta 'artigos/' e faz include()  
↓  
urls.py do app A\_Lei\_no\_NT detecta path('')  
↓  
views.lista\_artigos() é chamada