

Buscando integrar com APIs públicas ou privadas específicas das farmácias **Drogasil**, **Pague Menos** e **Farmácias Globo** para obter informações de preços e estoque. Contudo, no momento, não há APIs públicas documentadas dessas farmácias disponíveis na internet para acesso direto.

## Próximos Passos:

### 1. Verificação de APIs ou Parcerias:

- Como não há APIs públicas disponíveis de forma explícita dessas farmácias, o próximo passo seria **entrar em contato diretamente com essas farmácias** para verificar se elas oferecem APIs privadas ou soluções de integração.
- Algumas empresas, especialmente grandes redes de farmácias, podem fornecer APIs privadas ou até mesmo soluções de **web scraping** ou **parcerias comerciais** que permitem acessar esses dados.

### 2. Desenvolver Sistema com Integração via Web Scraping (caso não haja API):

- Se as APIs não estiverem disponíveis, uma alternativa seria **fazer scraping** dos sites dessas farmácias para extrair dados de preços e estoque (lembrando que é importante verificar as questões legais de uso dessa prática).

Para isso, você pode usar pacotes como o **Goutte** ou **Symfony Panther** para interagir com páginas web e extrair as informações necessárias.

### 3. Exemplo de Integração com Web Scraping (Laravel)

Caso você opte por fazer scraping, aqui está um exemplo básico utilizando o pacote **Goutte** para Laravel:

## Passos:

### 1. Instalar Pacote de Web Scraping (Goutte):

```
composer require fabpot/goutte
```

### 2. Criar Serviço de Scraping em Laravel:

```
<?php

namespace App\Services;

use Goutte\Client;

class FarmaciaService
{
    // Função para buscar informações da Drogasil
    public function consultarDrogasil($produtoId)
    {
        $client = new Client();
        $crawler = $client->request('GET', 'https://www.drogasil.com.br/busca?Ntt=' . $produtoId);

        // Aqui, você vai precisar ajustar os seletores de acordo com a estrutura do site da Drogasil
```

```

    $produto = $crawler->filter('.product-title')->text(); // Exemplo de seletor
    $preco = $crawler->filter('.product-price')->text(); // Exemplo de seletor

    return [
        'produto' => $produto,
        'preco' => $preco
    ];
}

// Função para buscar informações da Pague Menos
public function consultarPagueMenos($produtold)
{
    $client = new Client();
    $crawler = $client->request('GET', 'https://www.paguemenos.com.br/busca?Ntt=' . $produtold);

    // Ajuste de acordo com o site da Pague Menos
    $produto = $crawler->filter('.product-name')->text();
    $preco = $crawler->filter('.product-price')->text();

    return [
        'produto' => $produto,
        'preco' => $preco
    ];
}

// Função para buscar informações das Farmácias Globo
public function consultarFarmaciasGlobo($produtold)
{
    $client = new Client();
    $crawler = $client->request('GET', 'https://www.drogariaglobos.com.br/busca?Ntt=' . $produtold);

    // Ajuste de acordo com o site da Farmácias Globo
    $produto = $crawler->filter('.product-title')->text();
    $preco = $crawler->filter('.product-price')->text();

    return [
        'produto' => $produto,
        'preco' => $preco
    ];
}
}

```

### 3. Criar Rota e Controlador:

```

<?php

namespace App\Http\Controllers;

use App\Services\FarmaciaService;

class FarmaciaController extends Controller
{
    protected $farmaciaService;

    public function __construct(FarmaciaService $farmaciaService)
    {
        $this->farmaciaService = $farmaciaService;
    }

    public function mostrarProduto($produtold)

```

```

{
    // Chama os serviços para consultar os produtos nas farmácias
    $drogasil = $this->farmaciaService->consultarDrogasil($produtoid);
    $pagueMenos = $this->farmaciaService->consultarPagueMenos($produtoid);
    $farmaciasGlobo = $this->farmaciaService->consultarFarmaciasGlobo($produtoid);

    return view('produto', compact('drogasil', 'pagueMenos', 'farmaciasGlobo'));
}
}

```

#### 4. Exibir na View:

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Detalhes do Produto</title>
</head>
<body>
    <h1>Produto: {{ $drogasil['produto'] }}</h1>
    <p><strong>Preço na Drogasil:</strong> {{ $drogasil['preco'] }}</p>

    <h2>Preço na Pague Menos:</h2>
    <p>{{ $pagueMenos['produto'] }}: {{ $pagueMenos['preco'] }}</p>

    <h2>Preço na Farmácias Globo:</h2>
    <p>{{ $farmaciasGlobo['produto'] }}: {{ $farmaciasGlobo['preco'] }}</p>
</body>
</html>

```

### Considerações Importantes:

1. **Ajustar Seletores:** As classes e IDs dos elementos HTML (como `.product-title`, `.product-price`) que você usa para scraping podem mudar dependendo da estrutura do site de cada farmácia. Você deve inspecionar o código fonte do site e ajustar os seletores CSS no código de scraping.
2. **Legalidade e Permissão:** Certifique-se de que está cumprindo as regras de uso e a legalidade do scraping. Algumas empresas proíbem o scraping em seus Termos de Serviço.
3. **Restrições:** Se os sites tiverem **anti-scraping** ou proteções como CAPTCHAs, você precisará de soluções adicionais, como APIs de contorno de CAPTCHA ou técnicas avançadas de scraping.

### Conclusão:

Como não há APIs públicas de fácil acesso para Drogasil, Pague Menos e Farmácias Globo, a integração por **web scraping** é uma alternativa viável. No entanto, entre em contato com as farmácias para verificar a possibilidade de acordos comerciais ou APIs privadas que poderiam fornecer os dados de forma mais eficiente e legal.