Respostas - Estrutura de Projeto ASP.NET Core

# 1. Qual a função da pasta Pages?

A pasta 'Pages' contém as páginas da aplicação Razor Pages. Cada página é composta por dois arquivos:  
- `NomeDaPagina.cshtml`: contém o HTML e marcação Razor.  
- `NomeDaPagina.cshtml.cs`: o PageModel, responsável pela lógica de backend (dados, eventos, ações etc.).  
Ela cumpre o papel de 'Views' e 'Controllers' juntos, tornando o desenvolvimento mais simples e direto.

# 2. O que faz o arquivo Program.cs?

O arquivo 'Program.cs' é o ponto de entrada da aplicação. Ele:  
- Cria e configura o builder da aplicação (`WebApplicationBuilder`);  
- Adiciona os serviços necessários (como Razor Pages);  
- Define o pipeline de requisição HTTP, com middlewares como:  
 \* HTTPS redirection  
 \* Static files  
 \* Routing  
 \* Authorization  
- Por fim, chama `app.Run()` para iniciar o servidor web.

# 3. Onde são configurados os serviços da aplicação?

Os serviços são configurados no 'Program.cs', usando o método:  
`builder.Services.AddRazorPages();`  
Outros exemplos comuns incluem `AddDbContext`, `AddAuthentication`, `AddControllers`, etc.  
Esses serviços são injetados por meio de Dependency Injection durante a execução da aplicação.

# 4. Como é feito o roteamento de URLs?

O roteamento em Razor Pages é feito automaticamente com base na estrutura de arquivos da pasta Pages.  
Por exemplo:  
- '/Pages/Index.cshtml' => Rota '/'  
- '/Pages/Contact.cshtml' => Rota '/Contact'  
Além disso, o `app.MapRazorPages();` no 'Program.cs' habilita esse roteamento padrão.  
Se desejar rotas personalizadas, você pode usar `[Route]` ou configurar no arquivo 'Startup' (ou via `AddRazorPages().AddRazorPagesOptions(...)`).