1. O que são **Service Providers** em Laravel e para que servem?

São os organizadores do seu projeto, ele é uma classe especial que registra e configura serviços (services providers) que sua aplicação irá utilizar para configurar coisas como:

Rotas, middlewares, configurações de pacotes, etc.

1. Qual a diferença entre **hasOne** e **hasMany** no Eloquent ORM?

A relação hasOne representa uma **relação de um para um (1:1)** entre dois modelos. Ou seja, **um registro de uma tabela está associado a exatamente um registro de outra tabela.**  
  
Exemplo pratico:   
  
**Tabelas:**

* usuario  
  **Campos:** id, nomeUser, credCard\_id -> recebendo uma foreinkey\_id do cartao da outra tabela
* credCard  
  **Campos:** id, numero, codigoSeguranca, dataVencimento

---------------------------------------------------------------------------------------------------

A relação hasMany representa uma **relação de um para muitos (1:N)**. Isso significa que **um único registro de uma tabela está associado a vários registros em outra**.

Exemplo prático:

**Tabelas:**

* **usuarios  
  Campos:** id, nomeUsuario
* **enderecos  
  Campos:** id, endereco, numero, usuario\_id

Um usuário pode ter vários endereços, e cada endereço pertence a um único usuário.

1. O que é **Dependency Injection** e como ela é usada no Laravel?

Dependency Injection (DI) é um jeito de fornecer objetos que uma classe precisa, ao invés da própria classe criá-los.

Ex abstrato: Uma impressora que precisa de tinta, ao invés dela mesma fabricar a tinta, o sistema entrega a tinta pronta para uso.

Ex em código:

use App\Services\ReportService;

class ReportController extends Controller

{

protected $report;

public function \_\_construct(ReportService $report)

{

$this->report = $report;

}

public function index()

{

return $this->report->generate();

}

}

1. Explique o conceito de **middleware** e dê um exemplo de uso.

Middleware em Laravel é uma camada intermediária que intercepta as requisições HTTP antes ou depois de elas chegarem aos controladores. Ele é usado para filtrar, modificar ou proteger rotas, como por exemplo: verificar se o usuário está autenticado antes de acessar determinada página.

Ex abstrato: A porta de segurança de um banco só deixa você passar se você não estiver com algum tipo de metal. Seguindo a lógica, digamos que o middleware é uma porta e só deixa você passar se tiver tudo OK

Ex código:

Route::get('/dashboard', function () {

return view('dashboard');

})->middleware('auth'); -> verifica se o usuário está autenticado

1. Como funcionam **migrations** e quais suas vantagens?

As migrations são arquivos PHP que são usados para criar, modificar ou excluir tabelas e colunas do seu BD de forma controlada, vensionada e automatizada.

Cada migration representa uma **mudança na estrutura do banco**, como se fosse um "controle de versão" para o banco de dados. (como um git para o banco de dados)

**Principais vantagens:**

**Vencionamento:** Permite o histórico de mudanças no BD.

**Automação**: Permite a automação pois não é necessário criar as tabelas de forma manual no BD

**Reversibilidade:** Permite corrigir possíveis falhas com o rollback (como se fosse um CRTL + Z no word)

1. O que é **Queue** no Laravel e quando usá-la?

Queue (Fila) no Laravel é um recurso usado para adiar a execução de tarefas demoradas, como envio de e-mails, processamento de imagens ou integração com APIs externas, para que elas não travem o carregamento da aplicação.

Deve ser usada quando: uma ação leva tempo demais para ser executada, não precisa acontecer imediatamente, pode ser feita em “segundo plano”

1. Explique a diferença entre **API Resource** e um Controller tradicional.

Controller tradicional manipula requisições HTTP diretamente (GET,POST,PUT,DELETE), contém lógica de retorno e pode retornar views (HTML) ou JSON, dependendo da sua aplicação

Já a API Resource foca em formatar e padronizar os dados retornados na API, separa a lógica de apresentação dos dados do controller, retorna apenas o necessário, de forma limpa e mais organizada.