1. O que são templating engines e qual sua utilidade?

Templating engines são ferramentas que permitem gerar HTML dinâmico com dados do servidor.
São úteis para exibir informações personalizadas em páginas web, como nome de usuários ou listas de produtos.

2. Explique as diferenças entre EJS e Pug.

EJS usa **sintaxe semelhante ao HTML**, com tags <%= %> para inserir dados.
Pug usa **sintaxe própria e indentação**, sem necessidade de escrever HTML puro.

EJS é mais fácil para quem já conhece HTML.
Pug é mais conciso, mas tem curva de aprendizado maior.

3. Como configurar o EJS em um projeto Express?

npm install ejs

```
const express = require('express');
const app = express();


app.set('view engine', 'ejs');
app.set('views', './views');
```

Crie arquivos .ejs na pasta views.
Use res.render('arquivo', { dados }) para renderizar.

4. O que são partials e como eles auxiliam na reutilização de código?

Partials são **trechos de código reutilizáveis**, como cabeçalhos, rodapés ou menus.
Eles evitam repetição, permitindo incluir o mesmo conteúdo em várias páginas com um único arquivo.
No EJS, usa-se <%- include('nome-do-parcial') %>.

5. Como os dados do MySQL podem ser renderizados em um template?

Os dados são buscados com uma consulta SQL e passados ao template com res.render().

Exemplo:

```
connection.query('SELECT * FROM usuarios', (err, results) => {
  res.render('pagina', { usuarios: results });
});
```

No template (EJS), usa-se <%= usuarios[0].nome %> para exibir os dados.

6. Explique o papel do body-parser em aplicações Express.

O **body-parser** interpreta os dados enviados no corpo das requisições (como formulários ou JSON).
Ele transforma esses dados em um objeto acessível via req.body.
Hoje, o Express já inclui o body-parser internamente com express.json() e express.urlencoded().

7. Quais são as boas práticas ao organizar templates em um projeto?

- Usar uma pasta views para armazenar os templates.
- Separar **partials** em uma subpasta (views/partials).
- Nomear arquivos de forma clara e descritiva.
- Evitar lógica complexa nos templates.
- Reutilizar partials para manter o código limpo e DRY (Don’t Repeat Yourself).

8. Como funciona a inclusão de placeholders em layouts do EJS?

No EJS, não há suporte nativo para layouts com placeholders como em outros engines, mas é possível simular isso com **partials**.

Você cria um template base e usa <%- include('partials/header') %> e <%- include('partials/footer') %>, colocando o conteúdo principal entre eles em cada página.

Para estrutura com placeholders mais avançada, pode-se usar pacotes como express-ejs-layouts.

9. Três aplicações práticas para o uso de formulários no Express:

1. **Cadastro de usuários** – Enviar nome, e-mail e senha para salvar no banco de dados.
2. **Login** – Enviar credenciais para autenticar o usuário.
3. **Envio de comentários** – Permitir que usuários enviem mensagens ou feedbacks para exibir no site.

10. Por que é importante validar dados recebidos de formulários?

Para garantir que os dados sejam **corretos, seguros e completos**.
Isso evita erros no sistema, protege contra ataques (como SQL Injection) e melhora a qualidade das informações salvas no banco.