1 Porque não é possível desenvolver um aplicativo moderno utilizando apenas uma linguagem de programação? Porque diferentes linguagens cumprem papéis distintos na estrutura do sistema?

R:

Porque existe vários modelos de linguagens como HTML, CSS, JAVASCRIPT

Cada uma tem suas funções como partes visuais e logicas

Chat:

Não é possível desenvolver um aplicativo moderno usando apenas uma linguagem de programação porque um sistema é formado por várias partes que fazem coisas diferentes. Cada parte tem necessidades específicas, como mostrar a interface para o usuário, processar dados ou guardar informações. Por isso, usamos linguagens diferentes, cada uma criada para resolver um tipo de problema.

Por exemplo, linguagens como **HTML, CSS e JavaScript** são usadas para criar a parte visual que o usuário vê e interage (front-end). Já linguagens como **Python, Java ou Node.js** cuidam da lógica que roda nos servidores (back-end), processando informações e comunicando com bancos de dados. E os bancos de dados usam suas próprias linguagens para organizar e armazenar dados.

Então, diferentes linguagens cumprem papéis distintos na estrutura do sistema porque cada uma é otimizada para uma função específica. Assim, o sistema funciona melhor, é mais seguro e fácil de manter.

2 O que diferencia a lógica da programação do back-end da estrutura visual front-end? Mostre como essas duas camadas se relacionam na prática e por que não são independentes?

R:

A fornt-end é a parte interativa do usuário com o sistema

O back\_end é onde funciona a mecânica do sistema

Eles não são independes pois um necessita do outro

Chat:

O **front-end** é a parte visual de um sistema, onde o usuário clica e interage, como botões e formulários. Já o **back-end** é a parte que faz os dados funcionarem por trás, como salvar informações, calcular preços ou fazer login.

Eles trabalham juntos: o front-end envia dados para o back-end, que processa e responde. Um depende do outro para que o sistema funcione corretamente.

3 Explique em suas palavras o que é uma API e por que ela pode ser considerada um contrato entre sistemas diferentes. Dê um exemplo de situação em que APIs são fundamentais, mesmo que invisíveis para o usuário?

R:

API são as regras que faz a comunicação entre sistemas diferentes

Um exemplo é o google que pede verificação para usuarios

Chat:

Uma API é um conjunto de regras que permite que sistemas diferentes conversem e troquem informações, funcionando como um contrato entre eles. Por exemplo, quando você usa um app de clima, ele usa uma API para buscar dados do tempo. Outro exemplo é quando você faz login em um site usando sua conta do Google: o site usa uma API do Google para confirmar sua identidade. Tudo isso acontece sem você perceber.

4 Imagine que uma empresa precisa construir um sistema de agendamento de consultas online. Que ferramentas (editores, linguagens, banco de dados) você sugeriria e por quê? Relacione sua escolha ao tipo de dados, segurança e volume de acessos esperados?

R:

Para criar um sistema de agendamento de consultas, eu usaria o Visual Studio Code para programar. No front-end, React.js para a tela do usuário. No back-end, Node.js com Express para controlar o sistema. Usaria PostgreSQL para guardar os dados com segurança.

Para proteger o sistema, usaria HTTPS, autenticação segura e criptografia de senhas. Também faria backups para evitar perdas.

Essas ferramentas aguentam muitos usuários ao mesmo tempo e permitem crescer conforme o sistema aumenta. Assim, o sistema fica seguro, rápido e fácil de usar.

Chat:

Para construir um sistema de agendamento de consultas online, eu sugeriria usar o Visual Studio Code como editor de código, pois é leve e possui muitas ferramentas que facilitam o desenvolvimento. Para o front-end, indicaria React.js porque permite criar uma interface rápida e fácil de usar, melhorando a experiência do usuário ao marcar consultas. No back-end, escolheria Node.js com Express, pois é eficiente e escalável para lidar com muitos usuários ao mesmo tempo, além de facilitar a criação da lógica do sistema, como autenticação e envio de notificações.

Como banco de dados, optaria pelo PostgreSQL, que é um banco relacional confiável, ideal para armazenar dados sensíveis, como informações pessoais e horários de consultas, garantindo integridade e segurança dos dados. Para segurança, usaria HTTPS para proteger a comunicação, autenticação com tokens JWT para sessões seguras e criptografia para senhas. Também faria backups regulares para evitar perda de informações.

Considerando o volume de acessos, especialmente em horários de pico, essas tecnologias são capazes de suportar muitos usuários simultâneos e permitem crescimento futuro, podendo ser hospedadas em serviços de nuvem com balanceamento de carga.

Essa combinação garante um sistema seguro, eficiente e fácil de usar, adequado para lidar com dados pessoais sensíveis e um número considerável de acessos.

5 O que um desenvolvedor web deve considerar ao escolher o banco de dados para sua aplicação? Considere desempenho, facilidade de integração, estrutura dos dados e escalabilidade?

R:

O desenvolvedor deve escolher um banco rápido, fácil de usar, que funcione bem com os dados da aplicação e que possa crescer quando precisar. Bancos relacionais são para dados organizados, NoSQL para dados mais livres. NoSQL cresce fácil, mas bancos relacionais são mais confiáveis.

Chat:

Ao escolher um banco de dados para uma aplicação web, o desenvolvedor deve considerar o desempenho, que impacta na rapidez das respostas às consultas, especialmente com muitos usuários. A facilidade de integração é essencial para garantir que o banco funcione bem com as linguagens e ferramentas usadas no projeto. A estrutura dos dados também importa: bancos relacionais são ideais para dados organizados e com relações claras, enquanto bancos NoSQL funcionam melhor com dados flexíveis ou não estruturados. Por fim, a escalabilidade deve ser avaliada para que o banco suporte o crescimento da aplicação, sendo que bancos NoSQL costumam escalar mais facilmente, enquanto bancos relacionais oferecem maior consistência.