**Conversa com Zeni**

**🧑‍💻 1. Front-End**

**📌 O que é:**

É a **interface visual** com a qual o usuário interage. Tudo o que você vê em um site ou app (botões, menus, formulários) é front-end.

**🧰 Tecnologias:**

* **TypeScript**: versão tipada do JavaScript. Ajuda a evitar erros e melhora a manutenção do código.
* **Angular**: framework front-end da Google. Usa TypeScript e é ideal para aplicações grandes e complexas.

**🧠 Papel no sistema:**

* Comunicação com APIs do back-end.
* Validação de dados no navegador.
* Experiência do usuário (UX/UI).

**🧑‍🔧 2. Back-End**

**📌 O que é:**

É a **lógica por trás da aplicação**. Processa dados, acessa bancos, aplica regras de negócio e responde ao front-end.

**🧰 Tecnologias:**

* **Java**: linguagem robusta, muito usada em sistemas bancários e corporativos.
* **Quarkus**: framework moderno para Java, otimizado para cloud e microsserviços.
* **IDEs**:
  + **IntelliJ**: muito usado para Java, com recursos avançados.
  + **VSCode**: mais leve, usado também para front-end e scripts.

**🧠 Papel no sistema:**

* Processamento de requisições.
* Integração com banco de dados.
* Segurança, autenticação, autorização.

**🗃️ 3. Banco de Dados**

**📌 O que é:**

Onde os dados são **armazenados, consultados e manipulados**.

**🧰 Tecnologias:**

* **MongoDB**: banco NoSQL, ideal para dados flexíveis (JSON).
* **Oracle, MySQL, DB2**: bancos relacionais (SQL), com estrutura fixa e forte integridade.

**🧠 Papel no sistema:**

* Armazenamento persistente.
* Consultas otimizadas (SQL).
* Relacionamentos entre dados (em bancos relacionais).

**🧩 4. Modelagem de Dados**

**📌 O que é:**

É o processo de **planejar como os dados serão organizados** no banco.

**🧰 Ferramentas:**

* **Erwin**, **Oracle Data Modeler**: criam diagramas ER, definem chaves primárias, relacionamentos, normalização.

**🧠 Papel no sistema:**

* Evita redundância e inconsistência.
* Facilita manutenção e escalabilidade.
* Serve como documentação técnica.

**☁️ 5. Cloud Computing**

**📌 O que é:**

Infraestrutura e serviços **hospedados na internet**, em vez de servidores locais.

**🧰 Provedores:**

* **Interno**: nuvem privada da empresa.
* **Externo**:
  + **Azure** (Microsoft)
  + **AWS** (Amazon)

**🧠 Papel no sistema:**

* Hospedagem de APIs, bancos, front-end.
* Escalabilidade automática.
* Segurança e backups.

**📱 6. Mobile**

**📌 O que é:**

Desenvolvimento de **aplicativos para smartphones**.

**🧰 Tecnologias:**

* **Kodiak**: plataforma low-code usada no Banco do Brasil para criar telas e fluxos rapidamente.
* **React Native**: framework JavaScript para criar apps nativos com uma única base de código.

**🧠 Papel no sistema:**

* Interface móvel para o usuário final.
* Comunicação com APIs e serviços em nuvem.
* Integração com recursos do celular (GPS, câmera, notificações).
* Inteligência Artificial:
  + Finalidade: Categorização do extrato.

* Desing:
  + Figma

* Git Hub:
  + GitLab
  + GitFlow

* Low Code/ No code;
* Pengwin;
* Linux;
* Docker;
* Open Shift | Red hat;
* Gaw;
* Mosaik;
* WSL

Front End

WEB:

**1. HTML (estrutura)**

* Aprenda a criar páginas com títulos, parágrafos, imagens, listas, links, formulários, etc.
* Entenda a estrutura básica de um documento HTML.

**🟡 2. CSS (estilo)**

* Aprenda a estilizar suas páginas: cores, fontes, espaçamentos, posicionamento.
* Estude Flexbox e Grid para criar layouts responsivos.
  + - Seletores Básicos:
    - Propriedades:
    - Box Model
    - Unidades de Medida
    - Flexbox
    - Seletores avançados:
      * Descendentes: div p
      * Filhos diretos: div > p
      * Pseudo-classes: :hover, :first-child, :nth-child()
      * Pseudo-elementos: ::before, ::after
    - Display e posicionamento

**🟠 3. JavaScript (comportamento)**

* Aprenda lógica de programação, manipulação do DOM, eventos, funções, arrays, objetos.
* Pratique criando interações simples como botões, sliders, validação de formulários.
  + Seletores:
    - document.getElementById()
    - document.querySelector()
  + document.querySelectorAll()
  + 📌 Manipulação:
    - innerHTML, textContent, value
    - classList.add(), remove(), toggle()
    - style.property

**🔵 4. TypeScript (versão tipada do JS)**

* Depois de entender bem JS, aprenda TS para ganhar mais segurança e produtividade.
* Comece a usar TS em projetos pequenos para se acostumar com a tipagem.

**🟣 5. Frameworks (como Angular)**

* Quando estiver confortável com HTML, CSS, JS e TS, aí sim mergulhe no Angular.
* Você já vai entender os fundamentos e vai aproveitar melhor o que o framework oferece.