

Pesquisa...

Tecnologias comuns utilizadas em conjunto com o CAP Node do SAP.

1. **SAP Business Technology Platform (BTP):** Plataforma onde o CAP é frequentemente utilizado, proporcionando serviços de banco de dados, integração e extensibilidade.
 - **O que faz:** Plataforma de nuvem que oferece serviços de banco de dados, análise e desenvolvimento de aplicações.
 - **Linguagens:** JavaScript, Node.js, Java.

2. **SAP HANA:** Banco de dados in-memory que pode ser utilizado como backend para persistência de dados.
 - **O que faz:** Banco de dados in-memory que oferece alto desempenho para processamento e análise de dados.
 - **Linguagens:** SQL, SQLScript, JavaScript (para XSJS (é um arquivo que roda no servidor do Hana)), Python (com SAP HANA Machine Learning Library).

3. **Node.js:** Usado com CAP (pacotes do npm)
 - **O que faz:** Executa JavaScript no servidor. CAP Node é baseado em Node.js.
 - **Linguagens:** JavaScript, TypeScript.

4. **OData:** Um protocolo frequentemente usado para criar APIs que interagem com serviços CAP.
 - **O que faz:** Protocolo de padrão aberto para criação e consumo de APIs. Facilita a interação com serviços CAP e permite operações CRUD (Create, Read, Update, Delete) em recursos.
 - **Linguagens:** Utilizado através de configurações e consumo em várias linguagens, incluindo JavaScript, C#, e Java.

5. **SAP Fiori/UI5:** Para o frontend, é comum usar SAP Fiori ou SAPUI5 para criar interfaces de usuário que consomem APIs CAP.

- **O que faz:** SAP Fiori é um design system para aplicações empresariais modernas e responsivas. SAPUI5 é uma biblioteca de desenvolvimento front-end baseada em JavaScript que implementa as diretrizes de design do SAP Fiori.

- **Linguagens:** JavaScript, HTML, CSS.

6. **VS Code :** (Linguagens) Tem foco especial em JavaScript, TypeScript, e Python para desenvolvimento CAP.

7. **CI/CD Pipelines:** Ferramentas como Jenkins, GitHub Actions, ou GitLab CI/CD são usadas para automação de build e deploy de aplicações CAP.

- **O que faz:** Ferramentas e processos que automatizam a integração contínua (CI) e entrega contínua (CD) de código, permitindo builds, testes e deploy automáticos.

- **Ferramentas:** Jenkins, GitHub Actions, GitLab CI/CD.

- **Linguagens:** Scripts de automação podem ser escritos em várias linguagens como Shell, Python, JavaScript, Groovy (para Jenkins).

8. **Cloud Foundry:** Plataforma de nuvem onde as aplicações CAP são frequentemente implantadas.

- **O que faz:** Plataforma de nuvem que permite a implantação, gerenciamento e escalabilidade de aplicações. Suporta múltiplas linguagens e é utilizada para hospedar aplicações CAP.

- **Linguagens:** Suporta várias linguagens incluindo Java, Node.js, Python, Ruby, Go.

9. **GraphQL:** Utilizado em alguns casos para consumir APIs em vez de OData.

- **O que faz:** Linguagem de consulta para APIs que permite solicitar exatamente os dados necessários, facilitando a interação eficiente com serviços backend.

- **Linguagens:** Utilizado principalmente em JavaScript/TypeScript para implementação de servidores e clientes GraphQL.

10. **REST APIs:** Além de OData, RESTful APIs podem ser utilizadas para comunicação entre serviços.

- **O que faz:** Estilo arquitetônico para criar serviços web que permitem a comunicação entre sistemas através de operações HTTP.
- **Linguagens:** Pode ser implementado e consumido em várias linguagens, incluindo JavaScript, Java, Python, Ruby.

11. **Jest/Mocha:** Frameworks de testes utilizados para testar aplicações Node.js/CAP.

- **O que faz:** Frameworks de testes utilizados para escrever e executar testes automatizados de código JavaScript. Jest é frequentemente usado para testes de unidade e integração, enquanto Mocha é uma estrutura flexível para testes de JavaScript com suporte a plugins como Chai para asserções.
- **Linguagens:** JavaScript, TypeScript.