

LANDING PAGE

****Apresentação do grupo e introdução do tema****

[Isabella]

Dentre as ferramentas da Governança de TI, temos o framework, uma estrutura de controles que determina os melhores métodos para que os resultados sejam alinhados com a estratégia empresarial, os frameworks tratam de assuntos específicos, em alguns casos complementares, se interligando em certos momentos para melhorar a capacidade nos resultados dos serviços que lhes são estabelecidos. Entre os frameworks mais utilizados todos possuem boas práticas, contudo eles se diferenciam em relação ao seu foco, algumas se destacam mais pela agilidade obtida em sua metodologia, o gerenciamento na qualidade do serviço que a empresa almeja, processos de planejamentos e implantação de seus projetos organizacionais, pela prática no gerenciamento de TI, promover uma gestão direcionada a entrega eficiente dos serviços ao cliente, entre outros objetivos organizacionais. A utilização dos frameworks dentro do ambiente de trabalho trás diversos benefícios, principalmente o auxílio na implementação e controle do TI. Ele simplifica o gerenciamento e trás uma organização maior no desenvolvimento, estabelecendo padrões a serem seguidos.

- ****Explicação sobre o que é um framework back-end e suas principais funções. - Victoria****

Uma explicação sobre o que é um framework back-end e suas principais funções.

[Victoria]

O QUE SÃO OS FRAMEWORKS?

Frameworks de backend são conjuntos de ferramentas, bibliotecas e padrões que auxiliam no desenvolvimento de aplicativos e sistemas web. Eles fornecem uma estrutura pré-definida que permite aos desenvolvedores criar e organizar o código de forma mais eficiente. Esses frameworks fornecem uma base sólida para o

desenvolvimento de aplicativos backend, lidando com tarefas comuns, como gerenciamento de banco de dados, autenticação de usuários e roteamento de URLs.

Para entender o que é framework, de forma simples, podemos dizer que é uma espécie de template que conta com diversas funcionalidades que podem ser utilizados pelo desenvolvedor em seus projetos.

Ele conta com ferramentas, sistemas, componentes e guias que agilizam o processo de criação de soluções, sendo, portanto, um artifício essencial na vida de um profissional de TI.

É importante destacar que framework não é a mesma coisa que biblioteca. Em outras palavras, uma biblioteca é feita para trabalhar de forma independente de onde ela será implementada. No entanto, um framework, ao ser adotado, irá reger, conduzir e guiar todo o desenvolvimento da aplicação que está sendo construída com ele. Portanto, dentro de um framework existem várias bibliotecas.

O desenvolvimento está por de trás (a parte não visível pelos usuários) do site ou aplicações desenvolvidas e faz a interação com bancos de dados e a comunicação com o frontend

PARA QUE SERVE OS FRAMEWORKS?

O principal objetivo dos frameworks é resolver problemas recorrentes com uma abordagem genérica. Com isso, o desenvolvedor não precisa ficar reescrevendo softwares, podendo focar seus esforços em resolver os problemas em si.

Eles também ajudam no desenvolvimento rápido e seguro de aplicações. Suas funcionalidades permitem atender qualquer necessidade do profissional, desde que o framework selecionado seja direcionado a função desejada.

Nesse cenário, é importante conhecer as tecnologias a fundo. Afinal, não adianta se aventurar em um framework de linguagem de programação sem conhecer os aspectos básicos do JavaScript, por exemplo.

FUNCIONALIDADES:

Os back end frameworks oferecem diversas funcionalidades essenciais, como roteamento de URLs, gerenciamento de sessões, autenticação de usuários e manipulação de dados. Essas características são fundamentais para a construção de aplicações robustas e escaláveis. Além disso, muitos frameworks incluem suporte para APIs RESTful, permitindo que diferentes sistemas se comuniquem de

forma eficiente. Essa integração é crucial em um mundo onde a conectividade entre serviços é cada vez mais necessária.

Vantagens

Produtividade aumentada

Reutilização de código

Comunidade ativa

Segurança aprimorada

otimizar o tempo de desenvolvimento

Redução na incidência de bugs

Padronização de códigos

Maior consistência das aplicações

Desvantagens

Curva de aprendizado

Restrições de personalização

Sobrecarga de recursos

Dependências de um só framework específico pode limitar a flexibilidade do projeto

- **Apresentação sobre os diferentes tipos de framework (Full Stack, Microframework e API frameworks) e as diferentes arquiteturas (MVC, MVVM). - Nat**

[Nat]

Há diferentes áreas em que frameworks são utilizados, sendo as principais:

- Full Stack Framework, que são Frameworks que fornecem uma solução completa, cobrindo tanto o front-end quanto o back-end do desenvolvimento.

Exemplos: Django (Python), Ruby on Rails (Ruby), Spring Boot (Java).

Vantagens: Integração completa entre front-end e back-end; ferramentas e bibliotecas integradas.

Desvantagens: Pode ser complexo e pesado; curva de aprendizado íngreme.

- Microramework, Frameworks minimalistas que fornecem apenas componentes essenciais, deixando a maior parte da funcionalidade a ser construída ou integrada.

Exemplos: Flask (Python), Sinatra (Ruby), Slim (PHP).

Vantagens: Leve e rápido; flexibilidade para escolher componentes adicionais.

Desvantagens: Menos funcionalidades out-of-the-box; requer mais configuração e integração manual.

- API framework, que são Frameworks especializados em construir APIs (Application Programming Interfaces).

Exemplos: Express.js (Node.js), FastAPI (Python), LoopBack (Node.js).

Vantagens: Facilita a criação e gerenciamento de APIs; ferramentas integradas para documentação e teste de APIs.

Desvantagens: Focado apenas em APIs, pode precisar de frameworks adicionais para outras funcionalidades.

- Arquiteturas

- MVC (Model-View-Controller), é um padrão de design que separa a aplicação em três componentes principais: Modelo, Visão e Controlador.

Componentes:

- Model: Representa a lógica de dados e negócios;

- View: Interface do usuário;
- Controller: Gerencia a comunicação entre Model e View.

Vantagens: Separação clara de responsabilidades; facilita a manutenção e teste.

Desvantagens: Pode ser mais complexo de implementar; overhead de comunicação entre componentes.

- MVVM (Model-View-ViewModel), um padrão de design que separa a interface do usuário (View) da lógica de negócios e apresentação (ViewModel), mantendo a lógica de dados (Model).

Componentes:

- Model: Representa os dados e a lógica de negócios.
- View: Interface do usuário.
- ViewModel: Intermediário que converte dados do Model para a View.

Vantagens: Facilita o binding de dados entre View e ViewModel; melhor separação de preocupações em relação ao MVC.

Desvantagens: Pode ser mais complicado de entender e implementar; curva de aprendizado íngreme.

[Marco]

O **Spring Boot** é um framework Java que facilita o desenvolvimento de aplicações autônomas e prontas para produção, reduzindo configurações complexas e oferecendo um servidor embutido. Ele faz parte do **ecossistema Spring**, sendo amplamente utilizado para criar sistemas empresariais.

Seus principais recursos incluem configuração automática, gestão simplificada de dependências, suporte a microsserviços, integração com bancos de dados via

Spring Data JPA, segurança com Spring Security e criação de APIs RESTful com Spring Web. Além disso, o Spring Cloud permite a comunicação eficiente entre serviços.

O Spring Boot é usado em APIs RESTful, microserviços, sistemas corporativos, mensageria financeira e aplicações web full stack. Ele é adotado por empresas que precisam de sistemas escaláveis, como plataformas de streaming e e-commerce.

Dessa forma, o Spring Boot se mostra uma solução eficiente e amplamente adotada para o desenvolvimento de aplicações Java modernas. Sua capacidade de simplificar configurações e oferecer suporte a diversas arquiteturas faz dele uma excelente escolha para projetos escaláveis e seguros.

****Encerramento do vídeo****