

Programação WEB III

Construção de sistemas web

Prof. Dr. Danilo Barbosa



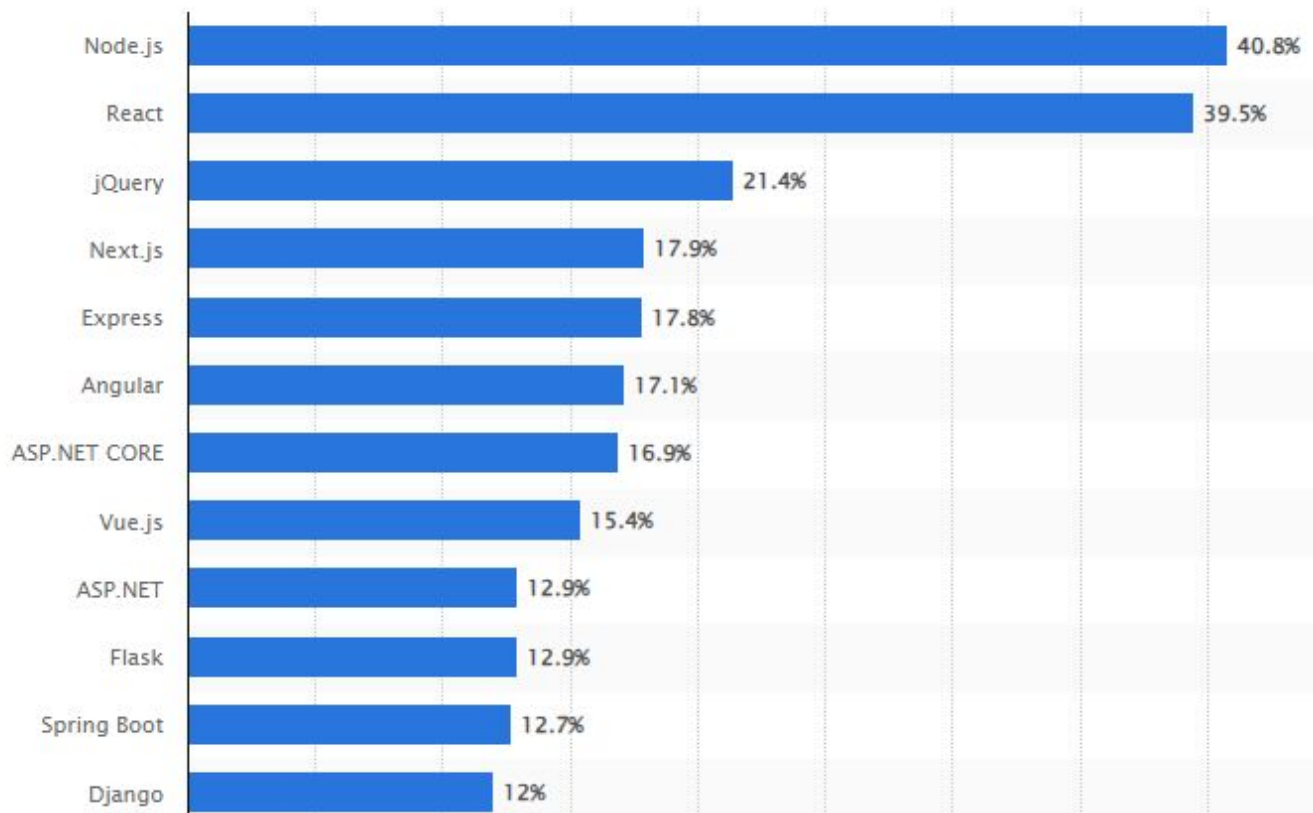
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PERNAMBUCO

Conteúdo

- Ideação e problema
- Requisitos
- Prototipagem
- Detalhes técnicos
- Atividade
- Referências



Frameworks web mais usados entre desenvolvedores em todo o mundo, em 2024

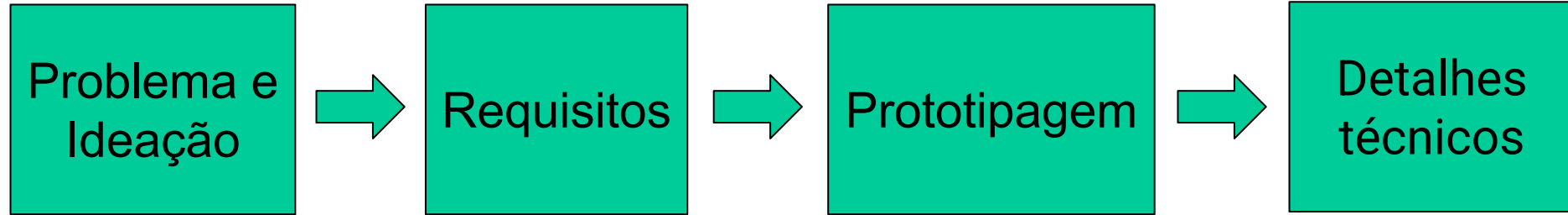


<https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PERNAMBUCO

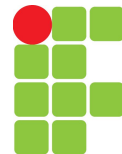
Construção de sistema web



Problema

Qual é o problema você vai resolver?

O que precisa ser resolvido ou criado?



Problema

- Os 5 porquês é uma ferramenta que busca descobrir a causa raiz de um problema a partir de seus sintomas. Na prática, consiste em perguntar cinco ou mais vezes o porquê de um problema até chegar a uma causa satisfatória que explique o fato originante do problema.



Problema

- Os 5 porquês

PROBLEMA: _____

POR QUE?

POR QUE?

POR QUE?

POR QUE?

POR QUE?

CAUSA RAIZ: _____



Problema

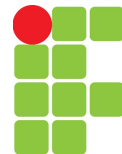
- Matriz de Definição de Problema: Esta ferramenta ajuda a equipe a re(pensar) o problema através de diferentes questões e pontos de vista.

1 Qual é a questão principal e por que ela é importante?	3 Quais as consequências desse problema mais afetam as pessoas?
2 Para quem isso é um problema?	4 Você consegue pensar esse problema de forma diferente?
5 Que fatores sociais e culturais têm influência neste problema?	6 Em uma frase é possível definir o problema?



ideação

Como ocorre a ideiação da construção de um sistema web?



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PERNAMBUCO

Ideação

Dicionário Oxford language:

1. ato ou efeito de idear.
2. formação e encadeamento das ideias; concepção.

Ideação vem da palavra ideias

Ideação é a geração de várias ideias.



Ideação

Ideação é a fase em que as ideias são apresentadas **sem nenhum julgamento (nenhuma ideia será descartada)** .

É o momento de pensar fora da caixa (**o que é pensar fora da caixa?**) e propor soluções para o problema.

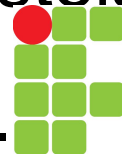


Ideação

A etapa idealizar é o momento de gerar o maior número de ideias possíveis para **a solução dos problemas.**

Mas utilizando o que?

a criatividade para estimular a criação de soluções que possam resolvê-los e esteiam
de acordo com o contexto do seu projeto.

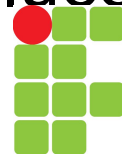


Ideação

Na prática:

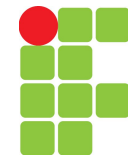
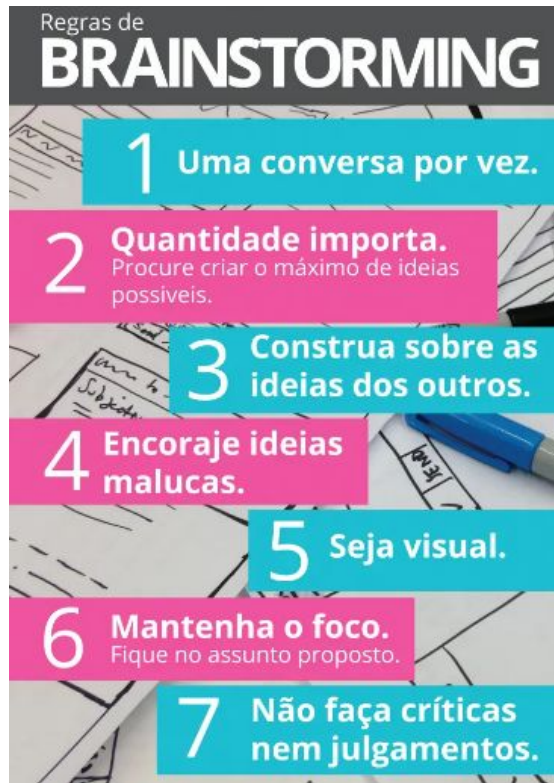
Formação de brainstorming

É uma técnica para estimular o surgimento de soluções criativas. A tempestade de ideias, em português, é feita em uma reunião e permite o compartilhamento de ideias e soluções valiosas.



Ideação

- As sete regras do brainstorming:



Ideação

- As sete regras do brainstorming:
- Uma conversa de cada vez. Todas as ideias precisam ser ouvidas, para que se possa construir em cima delas.
- Quantidade é melhor que qualidade. Defina um objetivo exorbitante – e o ultrapasse. A melhor forma de encontrar uma boa ideia é ter várias ideias.
- Construa em cima das ideias dos outros. Acrescente às ideias utilizando “e”. Evite o uso da expressão “mas”.



Ideação

- As sete regras do brainstorming:
- Encoraje as ideias ousadas. Mesmo que algo não pareça realista, pode estimular uma ideia em outra pessoa.
- Seja Visual. Aceite erros e falhas – pense no exagero como parte do processo. Não desestimule seu colega de grupo que apontou uma ideia equivocada.



Ideação

- As sete regras do brainstorming:
- Foque o tópico. Para aproveitar melhor a sessão, mantenha em mente a questão de brainstorm.
- Evite o julgamento. Não há más ideias nesta altura. Haverá bastante tempo para selecioná-las depois.



Requisitos

Como ocorre os requisitos
da construção de um
sistema web?

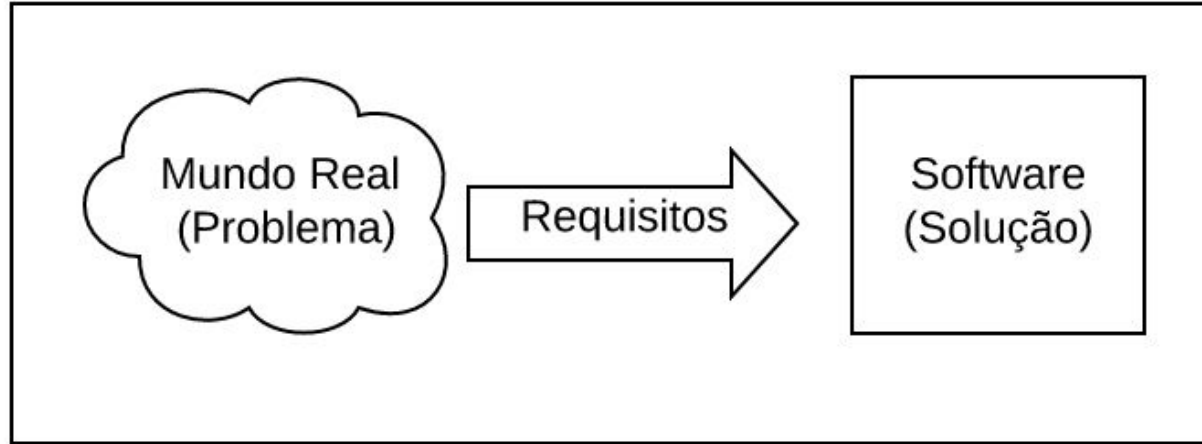


Requisitos

- Descrições do que o sistema deve fazer
- Inclui: os serviços fornecidos pelo sistema, suas qualidades específicas e suas restrições operacionais.
- Esses requisitos refletem as necessidades dos clientes de um sistema



Requisitos



Requisitos de sistemas

- requisitos de sistema são técnicos, precisos e escritos pelos próprios desenvolvedores.
- Normalmente, um requisito de usuário é expandido em um conjunto de requisitos de sistema.
- Suponha, por exemplo, um sistema bancário.



Requisitos de sistemas

- Um requisito de usuário — especificado pelos funcionários do banco — pode ser o seguinte: o sistema deve permitir transferências de valores para uma conta corrente de outro banco, por meio de TEDs.
- Esse requisito dá origem a um conjunto de requisitos de sistema, os quais vão detalhar e especificar o protocolo a ser usado para realização de tais transferências bancos.



Requisitos de sistemas

- Exemplos de requisito do sistema
- Documentação técnica sendo gerada dentro de uma metodologia - PMBOK;
- O custo do projeto não ultrapasse 250 mil reais;
- Que a entrega do projeto seja realizada até 31 de julho de 2024;
- Que atenda o mínimo de 90% da população que utiliza de consultas em bancos privados
- Que possua o protocolo HTTPS (Hypertext Transfer Protocol Secure na transferência bancária.

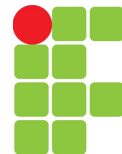


Requisitos funcionais

- Descreve funcionalidade e serviços do sistema

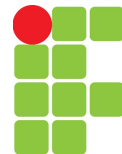
Depende do:

- Tipo do software
 - Usuários esperados
 - Onde o software é usado
- Cuidado com a redação dos requisitos



Requisitos funcionais

- Exemplos de RF
- [RF001] O sistema deve cadastrar o cliente (entrada)
- [RF002] O sistema deve emitir um recibo para o cliente (saída)
- [RF003] O sistema deve transformar uma fita disponível em fita emprestada, quando a fita for alugada pelo cliente (mudança de estado)
- [RF004] Usuário pode pesquisar todo ou um sub conjunto dos clientes da loja



Requisitos não funcionais

- Definem propriedades e restrições do sistema (tempo, espaço, etc)
- Requisitos de processo também podem especificar o uso de determinadas linguagens de programação, método de desenvolvimento
- Os requisitos não funcionais são tão importantes quanto os requisitos funcionais



Requisitos não funcionais

- Medidas de Requisitos

Propriedade	Métrica
Velocidade	Transações processadas por segundo Tempo de resposta ao usuário/evento Tempo de <i>refresh</i> da tela
Tamanho	Kbytes Número de chips de RAM
Facilidade de uso	Tempo de treinamento Número de telas de ajuda
Confiabilidade	Tempo médio para falhar Probabilidade de indisponibilidade Taxa de ocorrência de falhas Disponibilidade
Robustez	Tempo de reinício após falha Porcentagem de eventos que causam falhas Probabilidade de que os dados sejam corrompidos por falhas
Portabilidade	Porcentagem de declarações dependentes de sistemas-alvo Número de sistemas-alvo



Requisitos não funcionais

- Requisitos do Produto Final
 - Produto deve comportar-se de forma particular (velocidade de execução, confiabilidade, etc.)
- Requisitos Organizacionais
 - Conseqüência de políticas e procedimentos organizacionais (padrões de processo usados, requisitos de implementação, etc.)
- Requisitos Externos
 - Conseqüência de fatores externos ao sistema e ao processo de desenvolvimento (legislação, etc.)



Comparação de requisitos

- Requisitos funcionais: o que um sistema deve fazer
 - suas funcionalidades
- Requisitos não-funcionais: sob que restrições
 - Desempenho, segurança, disponibilidade, etc



prototipagem

Como ocorre a
prototipagem da
construção de um sistema
web?



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PERNAMBUCO

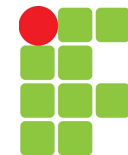
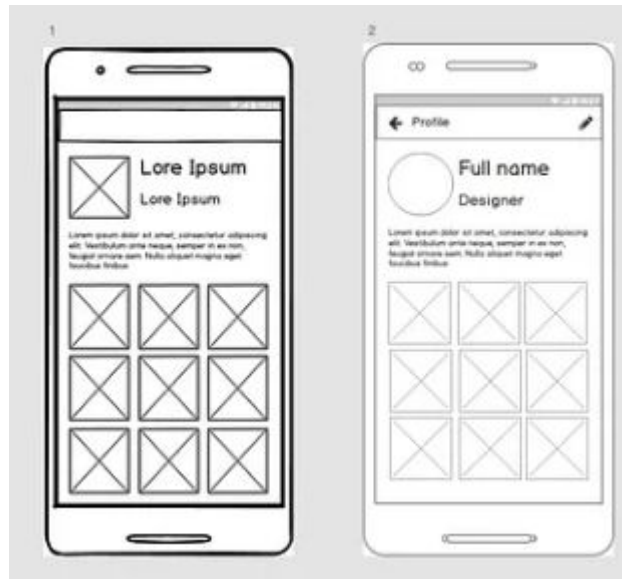
Prototipagem

- Nessa etapa já temos uma ideia sobre o que a nossa aplicação deverá fazer, mas é neste momento que desenhamos a interface do usuário da maneira mais prática para o usuário usar a aplicação.
- Quais os tipos de prototipagem?



Prototipagem de baixa fidelidade

- Os protótipos de baixa fidelidade funcionam como visualizações iniciais práticas do seu produto ou serviço. Exemplo: papel



Prototipagem de alta fidelidade

- é um recurso para testar o produto ou serviço com design e interatividade semelhantes à entrega final. Exemplo: figma



Prototipagem

- Ferramentas de prototipagem



Quais os detalhes técnicos
na construção de um
sistema web?



Detalhes técnicos

- Definir arquitetura do software
- Definir tecnologias (serviços externos, libs, frameworks, hospedagem etc.)
- Configuração inicial do projeto e workflow



Arquitetura de software

- Arquitetura de software é a estrutura fundamental ou o esqueleto de um sistema de software, que define seus componentes, suas relações e seus princípios de projeto e evolução.

ISO/IEC/IEEE 42010:2022



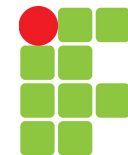
Os tipos de Arquitetura de software

- Arquitetura client-server (cliente-servidor)
- Segundo Martin Fowler, no livro Padrões de arquitetura de aplicações corporativas, os sistemas eram separados duas camadas:
- Cliente: responsável por manter a interface com o usuário e um ou outro código da aplicação. Por exemplo, Delphi e VB disponibilizavam componentes visuais para trabalhar com banco de dados.
- Servidor: normalmente um banco de dados relacional, como o SQL.



Os tipos de Arquitetura de software

- Arquitetura client-server (cliente-servidor)



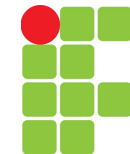
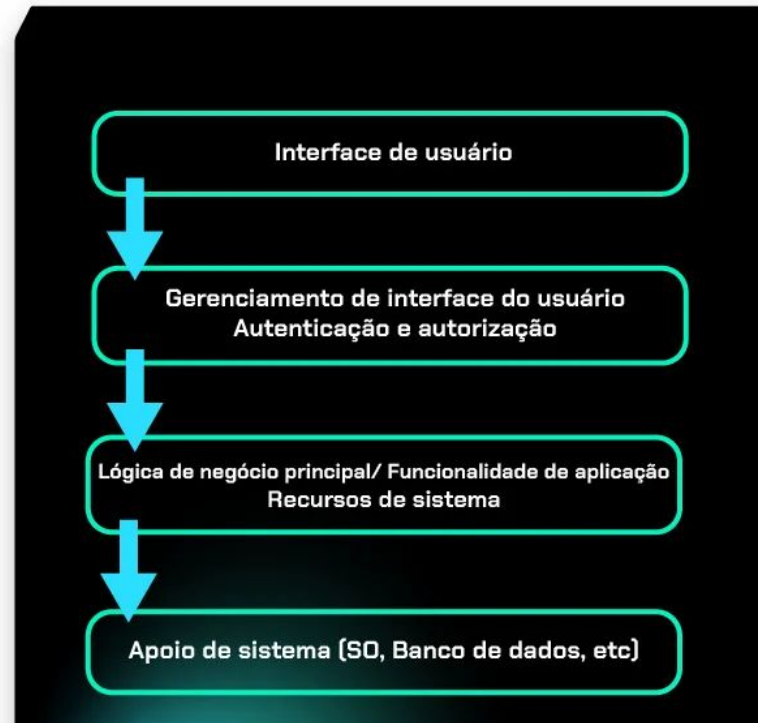
Os tipos de Arquitetura de software

- Arquitetura em camadas (Layers)
- Em uma organização do sistema em camadas, cada camada possui uma responsabilidade e funcionalidade específicas.
- no padrão de arquitetura em camadas, as funcionalidades de uma camada dependem dos recursos e serviços disponibilizados pela camada abaixo dela.



Os tipos de Arquitetura de software

- Arquitetura em camadas (Layers)



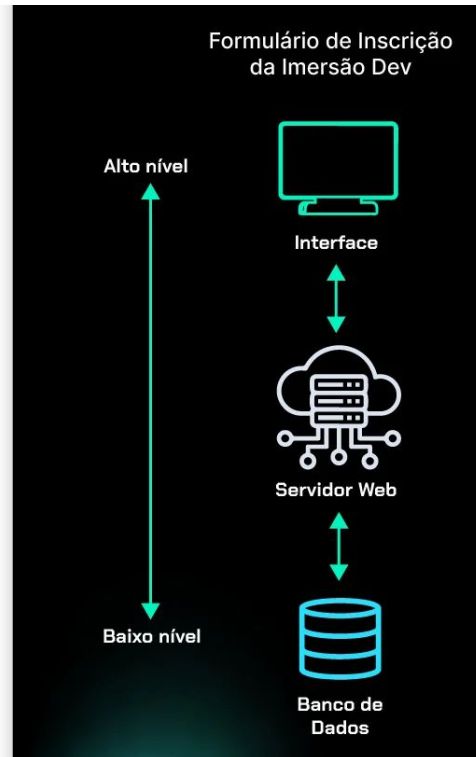
Os tipos de Arquitetura de software

- Arquitetura em camadas (Layers)
- O diagrama apresenta as camadas do mais alto para o mais baixo. Temos três camadas, que são:
- Interface com front-end para o formulário de inscrição.
- Um servidor com uma API para tratar os dados das pessoas inscritas.
- Um banco de dados para armazenamento das informações.



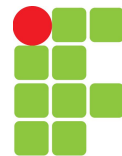
Os tipos de Arquitetura de software

- Arquitetura em camadas (Layers)



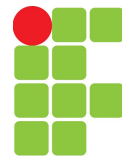
Os tipos de Arquitetura de software

- Arquitetura Model-View-Controller (MVC)
- A arquitetura MVC faz a separação da apresentação e a interação dos dados do sistema, que é estruturado em três componentes lógicos:
- model (modelo), view (visualização, visão ou vista) e controller (controlador).



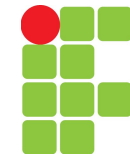
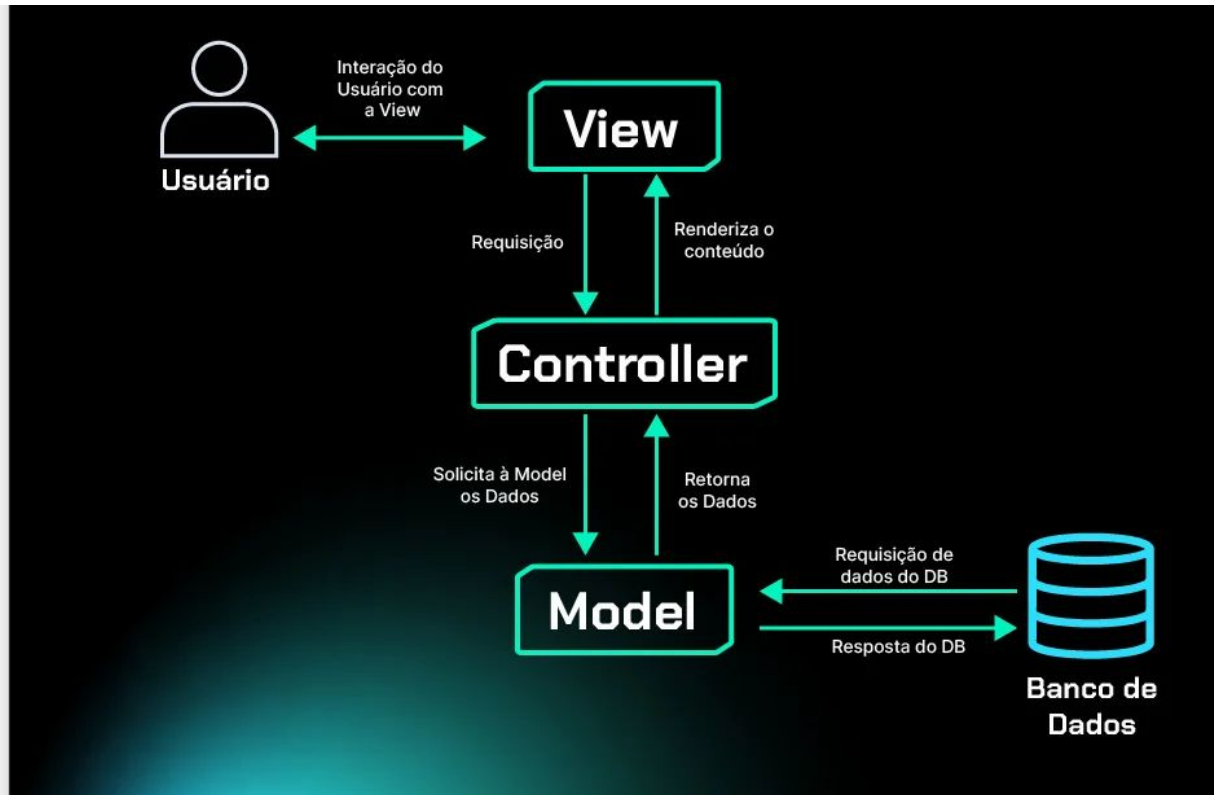
Os tipos de Arquitetura de software

- Arquitetura Model-View-Controller (MVC)
- Model é responsável por estabelecer as regras de negócio, interagir com o sistema de dados e fazer as operações associadas a esses dados.
- View define e gerencia como os dados são apresentados ao usuário.
- Controller é a camada intermediária entre model e view, interage com o usuário (por meio de teclas, cliques do mouse, requisições etc.) e é responsável por responder de acordo.

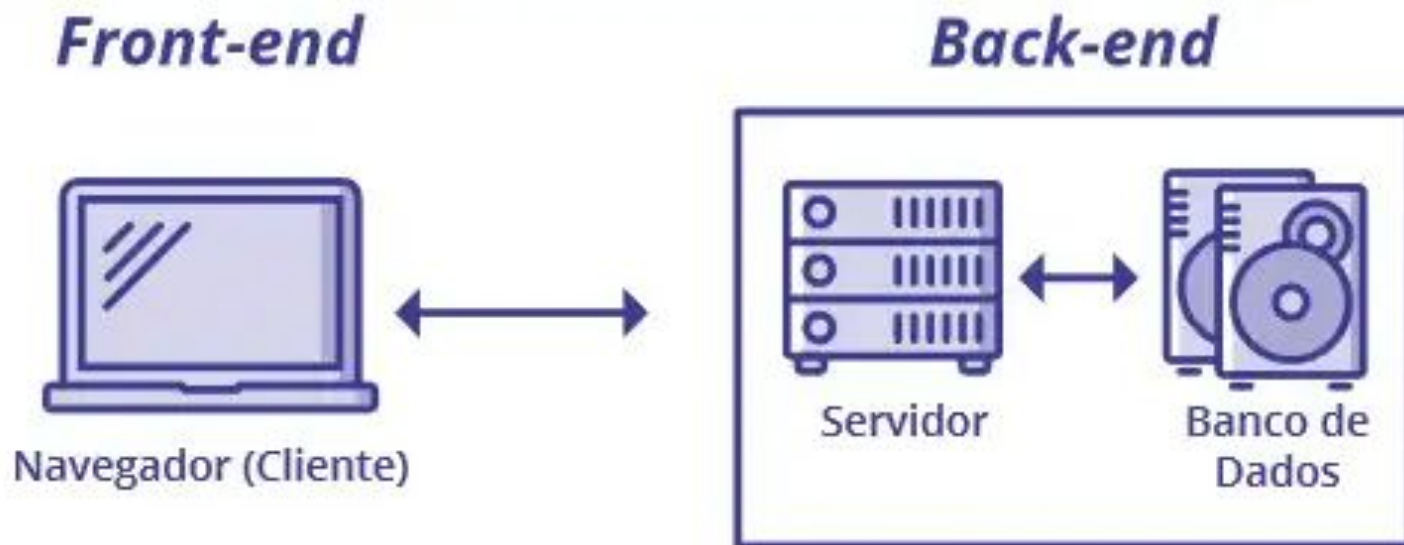


Os tipos de Arquitetura de software

- Arquitetura Model-View-Controller (MVC)



Definir tecnologia



Definir tecnologia

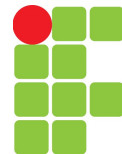


Configurações iniciais



Atividade

- Proponha um sistema web com a indicação do problema e ideação (solução)
- Definir os requisitos do sistema web
- Esboço de baixa fidelidade do sistema web
- Propor os detalhes técnicos do sistema web



Próxima aula

- Arquiteturas de sistema web



Referências

BIBLIOGRAFIA BÁSICA:

WILLIAMSON, Ken. Introdução ao AngularJS: Um guia para o desenvolvimento com o AngularJS. Novatec Editora, 2015.

MACHADO, Kheronn Khennedy. Angular 11 e Firebase: Construindo uma aplicação integrada com a plataforma do Google. Casa do Código, 2021.

MORAES, William Bruno. Construindo aplicações com NodeJS. Novatec Editora, 2015.

VALENTE, Marco Tulio De Oliveira et al. Engenharia de software moderna: princípios e práticas para desenvolvimento de software com produtividade. 2020.





**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PERNAMBUCO**