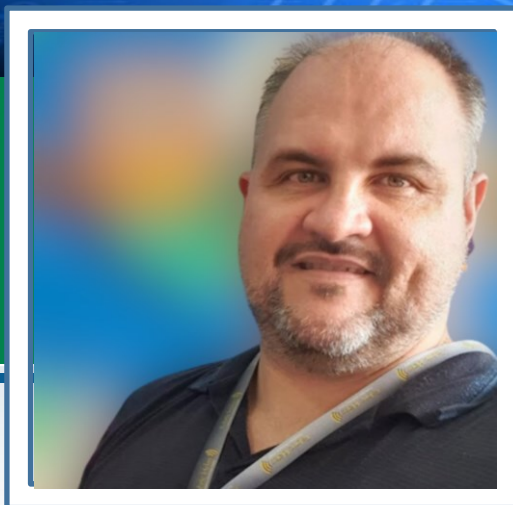




# Técnico em Desenvolvimento de Sistemas





# Programação Front-End

## Em instantes iniciaremos nossa aula

Me. Professor Ricardo B. deMello



**Se Liga que tá na hora da  
revisão!**

BBC

TRANS

USE...

#materiais de estudo



# Voltando em 3...2..1



# FRAMEWORK

Definição de framework como uma estrutura de software que fornece uma base para o desenvolvimento de aplicações

# CONCEITOS DE FRAMEWORK

Definição de frameworks como uma "mini" arquitetura reutilizável que fornece estrutura e comportamentos genéricos para abstrações de software em um contexto específico



# FRAMEWORK

Vantagens de utilizar frameworks:  
facilidades no desenvolvimento básico  
de código

Funcionalidades prontas para uso e  
maior produtividade.



# FRAMEWORK

Aplicações dos frameworks em diferentes setores: web, mobile, desktop e inteligência artificial.

# VANTAGENS DO USO DE FRAMEWORK

Alta qualidade e testes realizados pelos desenvolvedores

Aumento da produtividade ao evitar a implementação de recursos básicos em todos os projetos



# VANTAGENS DO USO DE FRAMEWORK

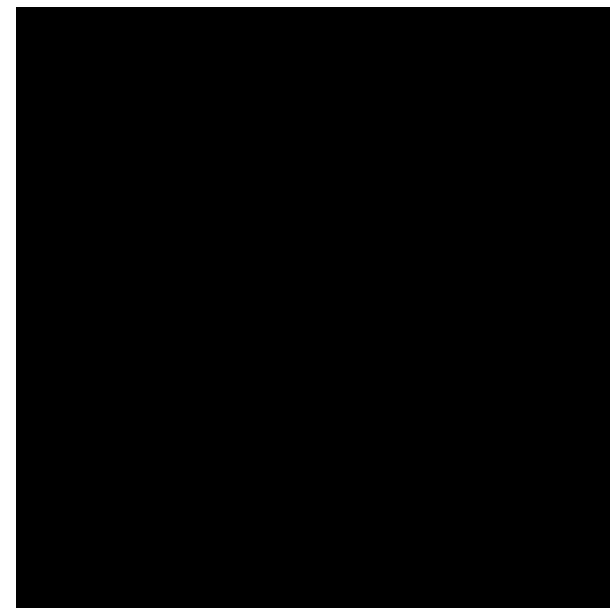
Segurança garantida pelos testes da comunidade

Otimização do tempo de desenvolvimento

Curva de aprendizado pequena, com documentação completa e exemplos de uso

# REACT

O React é uma biblioteca JavaScript de código aberto criada pelo Facebook em 2011, utilizada para construir interfaces de usuário (UI) em aplicações web e mobile.





# REACT

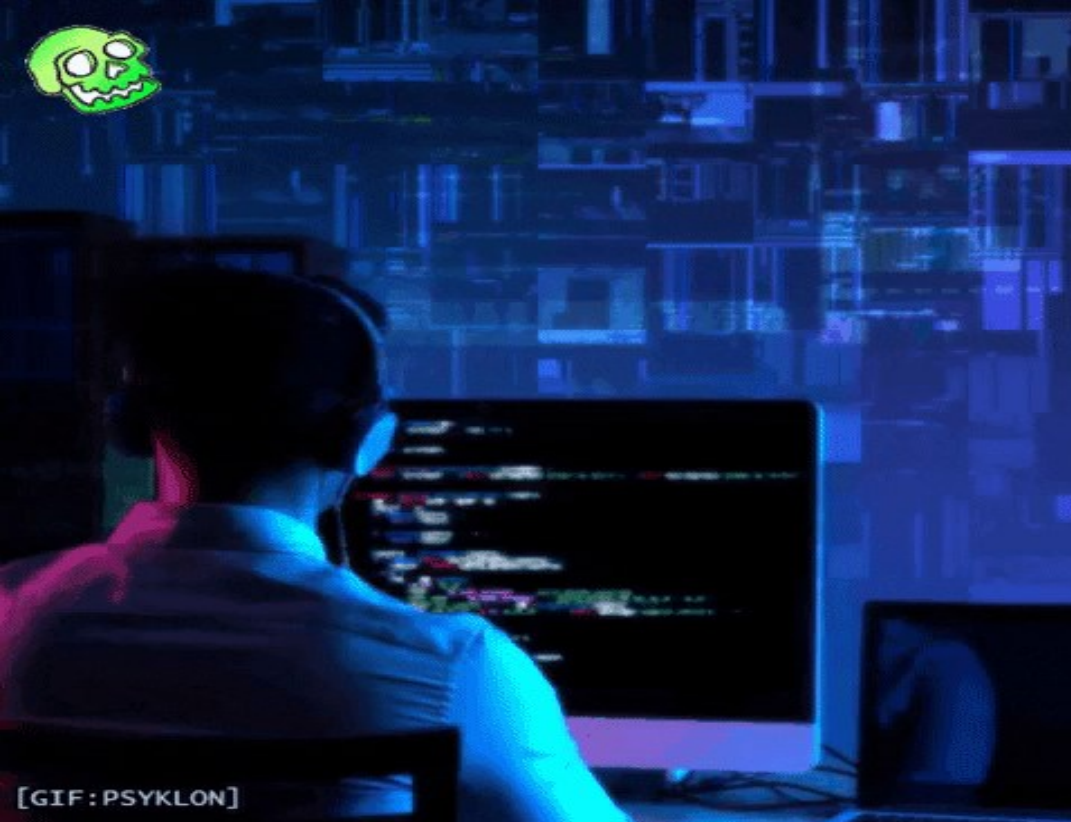
O React atualiza automaticamente a interface do usuário sempre que há mudanças nos dados da aplicação, sem a necessidade de atualizar a página inteira

# SÍNCRONO

Em requisições síncronas o programa aguarda até que essa operação seja concluída antes de continuar a executar outras tarefas

# ASSÍNCRONO

As requisições assíncronas permitem que o programa continue executando outras tarefas enquanto aguardam a conclusão da operação de E/S



```
var scrollHeight = ... + 0.02 * ...  
element.clientHeight + ...  
window.scroll(0, scrollHeight);  
}
```

# Versionamento



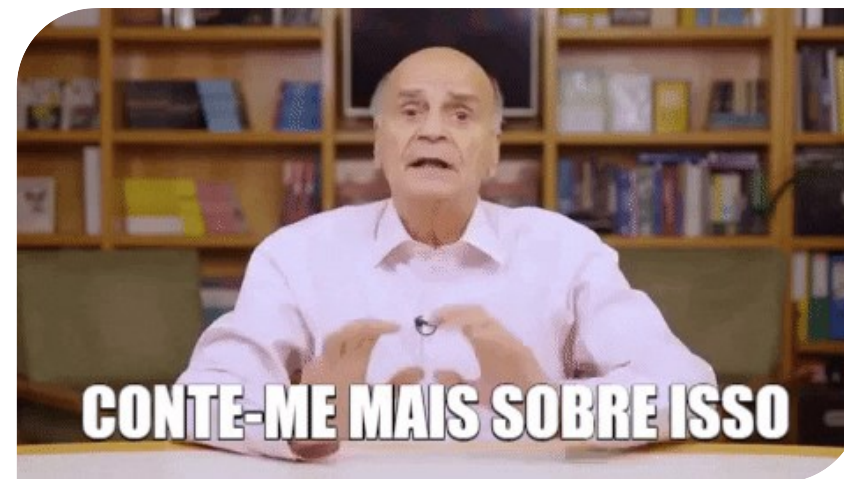
# VERSIONAMENTO

Acompanhar as alterações feitas em um arquivo, permitindo que os usuários mantenham um histórico das mudanças e revertam para versões anteriores, se necessário.

# VERSIONAMENTO

Colaboração: Permite que várias pessoas trabalhem em um mesmo arquivo sem sobrescrever as alterações umas das outras. Cada alteração é registrada como uma nova versão.

# VERSIONAMENTO



Histórico: Mantém um histórico das mudanças feitas em um arquivo, o que pode ser útil para rastrear quando e por quem uma alteração específica foi feita.

# VERSIONAMENTO

Recuperação: Permite a recuperação de versões anteriores de um arquivo em caso de erros, perda de dados ou necessidade de voltar a uma versão anterior.



# CONTROLE DE VERSÃO

Controle de versão é a **prática de gerenciar alterações** em arquivos de código-fonte ao longo do tempo, permitindo que você volte a versões anteriores, se necessário.

# GIT

**GIT** é um sistema de **controle de versão**, ou versionamento, que permite que você **rastreie alterações** em seus arquivos ao longo do tempo. Com isso, você pode criar **diferentes versões** de um arquivo e alterar entre elas facilmente.



# ESTADOS DE UM ARQUIVO NO GIT

Qualquer arquivo possui **3 estados no GIT**

- ***Committed (registrado)***: significa que os dados desses arquivos já estão armazenados, versionados de forma segura pelo GIT localmente.

# ESTADOS DE UM ARQUIVO NO GIT

Qualquer arquivo possui **3 estados no GIT**

• ***Modified (modificado)***: significa que você alterou o arquivo, mas ainda não fez o *commit* no seu banco de dados.



# ESTADOS DE UM ARQUIVO NO GIT

Qualquer arquivo possui **3 estados no GIT**

- ***Staged (organizar)***: significa que você marcou a versão atual de um arquivo modificado para fazer parte de seu próximo *commit*.

# COMANDOS GIT

**git init:** inicializa um novo repositório git;

**git add:** adiciona alterações aos arquivos ao stage;

**git commit:** cria um novo commit com as alterações adicionadas;

# COMANDOS GIT

**git status:** exibe o status atual do repositório, incluindo arquivos modificados e não adicionados ao stage;

**git log:** exibe um histórico detalhado de commits;

# COMANDOS GIT

**git clone:** clona um repositório remoto existente na internet ou em sua própria máquina;

**git push:** envia as alterações locais para um repositório remoto;

**git pull:** obtém as alterações mais recentes de um repositório remoto;

# COMANDOS GIT



**git branch:** exibe uma lista de branches (ramos) no repositório atual;

**git checkout:** muda para um branch (ramo) no repositório atual;



# COMANDOS GIT

**git merge:** combina as alterações de um ramo para outro;

**git remote:** gerencia as conexões com repositórios remotos;

**git diff:** exibe as diferenças entre duas versões de um arquivo.

# Kahoot!

<https://bit.ly/Kahoot17e18>



**Hora do Teste !**



**Me conta aí!**

Me conta aí!



<https://forms.gle/ZjUx2fZk55Y1tuMy6>



# PARANÁ

GOVERNO DO ESTADO  
SECRETARIA DA EDUCAÇÃO  
E DO ESPORTE



Unicesumar

EDUCAÇÃO  
PR.GOV.BR