



Publicar aplicações web

Prof. Renan Ponick

Roteiro das aulas / Calendário

1ª Segunda:

- Apresentação;
- Navegadores e buscadores;

2ª Segunda:

- Hospedagem, registro, tráfego;
- Pesquisa 1;

3ª Segunda:

- Aplicativos.
- Pesquisa 2;

4ª Segunda:

- Boas práticas usando site do Google SEO e W3C;

5ª Segunda:

- Git;

6ª Segunda:

- GitHub;

7ª Segunda:

- Build e Deploy;
- Backups;

8ª Segunda:

- Revisão;
- Kahoot;

9ª Segunda:

- Git Hub Pages;

10ª Segunda:

- Railway

11ª Segunda:

- Testes.

12ª Segunda:

- Avaliação Final;



Lema

"A riqueza de informações consome a nossa atenção."

Herbert Simon



GitHub

Agora que todos já tem conta no GitHub, vamos configurá-lo.

SSH - GitBash

Verificar se existe chave ssh.

```
ls -al ~/.ssh
```

Adicionar uma nova chave. (ID)

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Inicializar agente-ssh.

```
eval "$(ssh-agent -s)"
```

Adicionar chave ssh ao agente.

```
ssh-add ~/.ssh/id_ed25519
```

SSH - GitBash

Copiar chave ssh.

```
clip < ~/.ssh/id_ed25519.pub
```

Adicionar chave no github

Github -> Settings -> SSH and GPG keys -> New SSH key -> Colar

Coloque um título que identifique a chave

Testar conexão

```
ssh -T git@github.com  
yes
```



Referência

<https://docs.github.com/pt/authentication/connecting-to-github-with-ssh/about-ssh>



Mão na massa

Agora que já temos tudo configurado vamos FINALMENTE realizar os comando que você pesquisou aula passada.

Iniciando um projeto

Você foi contratado para desenvolver um programa WEB, utilizando o git e github para versionamento.

O sistema irá conter uma página inicial de bem vindo.
Conterá uma segunda página, para implementação futura, de um formulário.

Processo

Inicializar um projeto - index e form;
Criar a branch dev - proteger contra deleção;
Melhorar o form e subir em feat;
Merger feat em dev;
Mergear dev em main;
Realizar um revert;

Comandos

git status

git add <filename ou . >

git restore --staged <filename ou . >

git branch <branchname>

git checkout <branchname>

git checkout -b <branchname>

git commit -m "<description>"

git push

git branch -D <branchname>

git fetch

git pull

Continuando...

Crie uma branch de DEV e bloqueie contra exclusão.

Após ajuste o formulário para enviar o nome e a senha de um usuário.

Boas praticas

NUNCA COMMITAR NA MAIN/MASTER/PROD

Para isso abrimos branches:

feat: Algo relacionado com features que você adicionar;

fix: Algo para corrigir;

docs: Algo relacionado a documentações, README e afins;

style: Algo relacionado com estilização;

refactor: Algo relacionado com refatoração(refazer);

perf: Algo relacionado a performance;

test: Algo com testes;

chore: Algo para coisas relacionados a build, configs e afins. Exemplo: Use esse, seja atualizando a versão do pacote ou instalando novas dependências.