

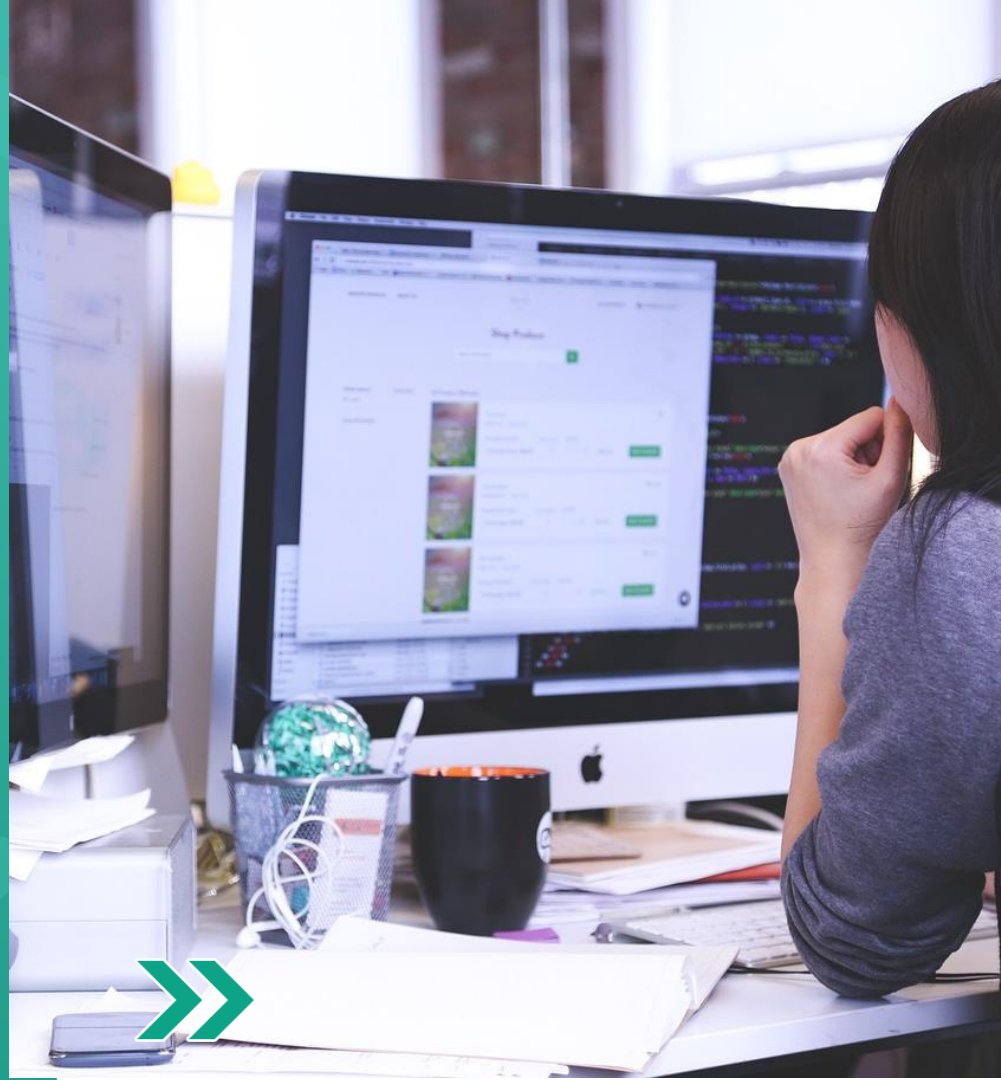


escola
britânica de
artes criativas
& tecnologia

Profissão: Engenheiro Front-End

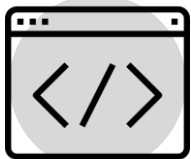


BOAS PRÁTICAS



Versionamento de código usando o Git

 **Controle a versão
(Github)**



Confira boas práticas da comunidade de Front-End por assunto relacionado às aulas.



Controle a versão (Github)

Acompanhe algumas dicas para fazer o versionamento de código usando o Git:

- **Faça commits frequentes e granulares:**
É uma boa prática fazer commits frequentes e com granularidade adequada. Commits pequenos e atômicos facilitam a revisão de código, a localização de bugs e o rastreamento de alterações. Evite fazer commits com um grande número de alterações não relacionadas.
- **Crie commits significativos:**
Sempre que fizer alterações no seu código que fazem sentido como uma unidade lógica, crie um commit. Certifique-se de que seus commits sejam significativos e autodescritivos. Evite fazer commits com muitas alterações não relacionadas em um único commit.
- **Use mensagens de commit claras:**
Ao criar um commit, forneça uma mensagem clara e concisa que explique as alterações realizadas. Uma boa mensagem de commit descreve o que foi feito e, opcionalmente, o porquê. Use um estilo consistente para suas mensagens de commit.



Controle a versão (Github)

Acompanhe algumas dicas para fazer o versionamento de código usando o Git:



- **Faça o merge de branches:**
Quando você concluir o trabalho em um branch e estiver satisfeito com as alterações, faça o merge dessas alterações de volta para o branch principal. Use o comando `git merge nome-do-branch` para mesclar as alterações. Certifique-se de resolver quaisquer conflitos de mesclagem que possam ocorrer durante o processo.
- **Utilize repositórios remotos:**
Além de manter um repositório local, você também pode criar um repositório remoto no GitHub, GitLab ou em outro serviço de hospedagem de código. Isso permite que você faça backup do seu código, colabore com outros desenvolvedores e implante seu projeto em ambientes de produção.
- **Explore recursos avançados do Git:**
O Git possui recursos mais avançados, como tags, rebasing, cherry-picking e muito mais. À medida que você se familiariza com o básico do Git, explore e aprenda sobre esses recursos para melhorar seu fluxo de trabalho e eficiência no versionamento de código.

Controle a versão (Github)

Acompanhe algumas dicas para fazer o versionamento de código usando o Git:



- Utilize branches de feature:**
Crie branches separados para cada nova funcionalidade que você estiver desenvolvendo. Isso permite que você trabalhe em paralelo com outras pessoas e isole as alterações relacionadas a uma determinada funcionalidade. Uma boa prática é nomear os branches de feature de forma descritiva, como feature/nome-da-funcionalidade.
- Aprenda a resolver conflitos de mesclagem:**
Conflitos de mesclagem podem ocorrer ao mesclar branches ou ao atualizar seu branch com as alterações do branch principal. É importante aprender a resolver esses conflitos, analisar as alterações conflitantes e decidir como combiná-las corretamente.
- Faça uso de tags:**
As *tags* do Git são marcadores usados para identificar pontos específicos na história do seu projeto. Elas são úteis para marcar versões estáveis, lançamentos ou pontos de referência importantes. Use tags para marcar versões importantes do seu código e facilitar a navegação no histórico do projeto.

Controle a versão (Github)

Acompanhe algumas dicas para fazer o versionamento de código usando o Git:



- Utilize ferramentas de revisão de código:**
À medida que você se envolve em projetos de desenvolvimento front-end, é comum fazer revisões de código. Utilize ferramentas como o GitHub, GitLab ou Bitbucket para facilitar o processo de revisão de código, deixar comentários e discutir alterações com seus colegas.
- Mantenha seu repositório organizado:**
Organize seu repositório Git de forma lógica, mantendo uma estrutura de diretórios coerente. Separe os arquivos em pastas de acordo com suas responsabilidades e mantenha um fluxo de trabalho consistente para adicionar, modificar e excluir arquivos.
- Utilize ramificações estratégicas:**
Além dos branches de feature, você também pode criar ramificações específicas para outras finalidades, como correção de bugs (bugfix/nome-do-bug), experimentações (experiment/nome-da-experimentacao) ou preparação para lançamento (release/nome-da-versao).

Controle a versão (Github)

Acompanhe algumas dicas para
fazer o versionamento de código
usando o Git:



Aprenda a reverter alterações:

Às vezes, é necessário desfazer alterações no código ou reverter commits anteriores. Aprenda a usar os comandos git revert e git reset para reverter alterações de forma segura e mantendo o histórico do projeto coerente.



Bons estudos!

