**VANTAGENS**

**VANTAGENS DO GIT**

1. **Sistema distribuído**

Isso significa que, diferentemente de outros sistemas de controle de versionamento populares na época em que foi lançado, o GIT possui repositórios, e não somente um único local com o histórico de seu trabalho.

1. **Fácil de trabalhar**

A maioria das operações do Git são rápidas, especialmente porque são efetuadas na cópia do seu repositório local.

1. **Os repositórios são menores**

Um repositório do Git é menor do que utilizar o SubVersion, por exemplo.

1. **Capacidade de mover ou adicionar arquivos**

Se você quiser mover um arquivo para dentro do seu repositório o Git acompanha a movimentação. Isso não era possível em ferramentas antigas de controle de versionamento como CVS. Mover um arquivo exigiria a criação de um novo arquivo e a remoção do antigo, portanto perdendo o histórico de modificações.

1. **Você pode ignorar certos arquivos**

Às vezes, você tem arquivos sendo gerados pelo seu projeto, como arquivos de log ou arquivos gerados pelo sua IDE, que você não quer armazenar no repositório do seu projeto porque eles não são parte do código-fonte.

Você pode mandar o Git ignorar certos arquivos nos diretórios do repositório local usando o arquivo .gitignore. Os arquivos listados no .gitignore são excluídos do processo de controle de versionamento como se não estivessem lá. Você pode compartilhar essas regras commitando o arquivo ou apenas mantê-las localmente.

1. **Branches**

Algumas vezes você precisa trabalhar numa nova feature experimental sem interferir no código principal do seu projeto.

Você pode fazer isso criando novos branches para experimentar o código dessas features experimentais.

Branches também permitem que diferentes desenvolvedores trabalhem em diferentes features sem interferir no trabalho de cada um. Quando as features estiverem prontas, eles podem fazer o merge das mudanças do branch para o branch principal.

1. **Verifique o status de suas modificações**

Verificar o status das modificações que você fez no seu repositório é bastante simples. O comando git status permite que você veja o que aconteceria se você commitasse suas modificações num determinado momento. Isso pode ajudar a evitar os erros do uso de diferentes branches externos do seu projeto.

1. **Stash Branches**

Se você está trabalhando em um branch do seu projeto mas não quer commitar as modificações, é possível salvar o status atual do branch para retornar a ele no futuro. Você pode trocar para outro branch e inserir as modificações ocultas nele.

Com o comando git stash você pode salvar o status atual. Com o comando git stash pop você aplicará as modificações ocultas.

1. **Cherry Pick Modificações dos Branches**

O Git permite selecionar um commit de algum branch e aplicar dentro do branch atual. Essa operação é útil para testes. Imagine que você quer testar uma modificação temporária ou escolher um commit feito em outros branches.

1. **Encontre a versão que introduziu um bug usando a Busca Binária**

Se você tem um problema no seu código e quer saber quando ele foi introduzido e que tipo de problema é, o comando git bisect permite que você volte à cada commit feito até encontrar o ponto-chave onde a irregularidade foi introduzida.

**VANTAGENS DO GITHUB**

1. **Edição de código**

Ao acessar o GitHub, é possível editar os seus códigos de maneira simples e rápida por já possuir edição de texto direto na plataforma. Isso facilita quando o desenvolvedor precisa fazer uma alteração rápida em um projeto.

1. **GitHub Wiki**

Com o GitHub Wiki é possível criar uma “documentação” do seu projeto, fornecendo informações diversas sobre seu repositório.

Para criar o Wiki do seu projeto, basta clicar sobre a aba “wiki”, conforme a imagem abaixo, e clicar sobre o botão “Create the first page”:

Este recurso é muito utilizado para descrição geral de um projeto, lista de versões e suas alterações, passos para instalação do projeto, lista de contribuidores, etc.

1. **Criar listas de tarefas**

Uma outra funcionalidade provida pelo GitHub é a criação de lista de tarefas para descrever as funcionalidades a serem desenvolvidas no projeto. Estas listas auxiliam o desenvolvedor a gerenciar as novas funcionalidades ou correções que devem ser implementadas.

1. **GitHub Pages**

O GitHub Pages é um recurso que permite criarmos uma página web estática para nossos repositórios totalmente gratuita.

Temos aqui no blog um artigo de como criar páginas para repositórios com o GitHub Pages, e caso esteja interessado, recomendo a leitura e o passo a passo.

1. **Criar project boards**

Com o GitHub podemos gerenciar de maneira mais fácil nossos projetos no próprio repositório, fazendo com que o mesmo funcione similar ao que fazemos quando utilizamos o Trello (um aplicativo de planejamento). Desta forma, torna-se algo extremamente útil na hora de desenvolver.

**REFERÊNCIAS**

<https://www.treinaweb.com.br/blog/5-utilidades-do-github> | Acessado em 03/04/2023, às 21:10

<http://ditech.com.br/git-beneficios-ferramenta-projetos/> | Acessado em 03/04/2023, às 21:05