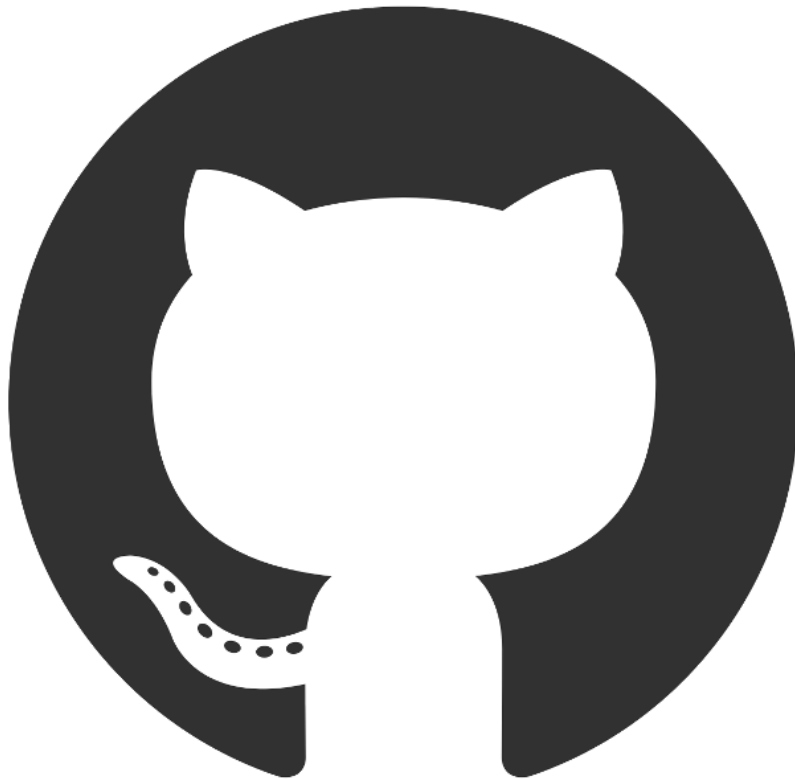


Controle de Configuração

Para realizar o controle de configuração do projeto, manter um histórico de todas suas versões e assim poder analisar as mudanças realizadas, será utilizado o Github.



Padronização

No Github, é de extrema importância padronizar as ações dos desenvolvedores de modo a manter o projeto organizado e facilitar futuras manutenções. Para isso, as seguintes regras devem ser seguidas para que o controle de configuração do Sistema seja respeitado:

- Cada desenvolvedor terá uma branch no repositório do projeto

```
git checkout -b $nome_desenvolvedor
```

- Para modificar um ou mais arquivos do projeto, o desenvolvedor deve seguir os seguintes passos:

- Atualizar a própria branch

```
git checkout $nome_desenvolvedor  
git pull origin master
```

- Realizar as mudanças nos arquivos e fazer o commit para sua branch

```
git add/rm $arquivos_modificados  
git commit -m "v $numero_versao"  
git push origin $nome_desenvolvedor
```

- Certificar que a branch do desenvolvedor não ficou desatualizada durante seu trabalho, o que pode acontecer se, durante esse período, houver outras atualizações na master

```
git checkout $nome_desenvolvedor  
git rebase master
```

- Aplicar as mudanças realizadas para a branch master

```
git checkout master  
git merge $nome_desenvolvedor
```

- Para determinar a variável `$numero-versao`, será seguido o seguinte padrão:
 - Durante o desenvolvimento inicial do projeto, o primeiro commit será realizado com o número de versão sendo 0.01
 - Ao realizar um novo commit, o desenvolvedor deve verificar qual foi a versão do último commit realizado e adicionar 0.01 para obter a nova versão
 - No fim da fase de desenvolvimento, o projeto será postado na versão 1.00
 - Para realizar manutenções ou atualizações da versão postada, será mantido o mesmo padrão de somar 0.01 à versão atual para obter a nova versão
 - Ou seja, commits com versão 0.xx representarão, em ordem cronológica, a versão do projeto em fase de desenvolvimento; enquanto os commits na versão 1.xx serão versões do projeto postado
- A partir do comando `git reset`, é possível restaurar versões anteriores do projeto
- Assim, será possível manter um rastreamento e armazenamento de todas as versões do projeto

Equipe

Eduardo Menezes

Fernando Rodrigues

Igor Dias

Lucas do Vale

Rodrigo Almeida

Rodrigo Teixeira

Professores

Karla Donato Fook

Johnny Cardoso Marques