

TRABALHO PRÁTICO DE GESTÃO DE VERSÕES

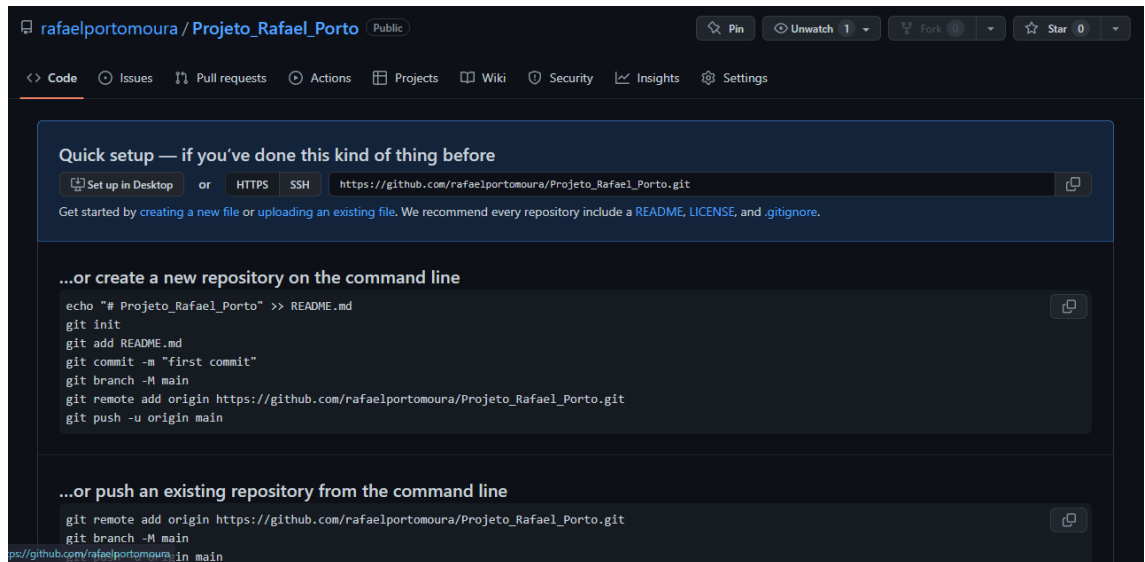
Nome: Rafael Porto Vieira de Moura

Matrícula: 201820274

Seção criação de projeto e clonagem:

- 1- Crie um repositório no GITHUB chamado “Projeto_<primeironome>_<segundoNome>”. Se fosse eu, professor, criando o repositório, ele se chamaria “Projeto_Antonio_Resende”

https://github.com/rafaelportomoura/Projeto_Rafael_Porto



- 2- Crie duas classes na linguagem que preferir, uma chamada A e a outra B. A classe A tem os atributos int A1 e float A2. A classe B tem os atributos int B1, float B2. O aluno pode alterar o tipo de variável para adequar a declaração a linguagem escolhida (int, integer, Integer,... ou float, real, decimal,...). Crie os gets e sets para esses atributos. A classe A deve ter os métodos MA1() e MA2(). A classe B deve ter os métodos MB1() e MB2(). Dentro de cada método coloque uma instrução de comando que manda imprimir o nome do método. Adicione (upload) dessas duas classes, A e B, no GitHub. **Atenção**, não se esqueça de apresentar o código das duas classes no relatório e a tela mostrando os dois arquivos inseridos no repositório do Github.

```
classA.cpp
```

```
#include <iostream>
```

```
using namespace std;
```

```
class A {
private:
    int A1;
    float A2;
public:
    A() {}
    int getA1() { return A1; }
    void setA1( int A1 ) { this->A1 = A1; }
    float getA2() { return A2; }
    void setA2( float A2 ) { this->A2 = A2; }
    void MA1() { cout << "MA1"; }
    void MA2() { cout << "MA2"; }
};

int main() {

    return 0;
}
```

classB.cpp

```
#include <iostream>

using namespace std;

class B {
private:
    int B1;
    float B2;
public:
    B() {}
    int getB1() { return B1; }
    void setB1( int B1 ) { this->B1 = B1; }
    float getB2() { return B2; }
```

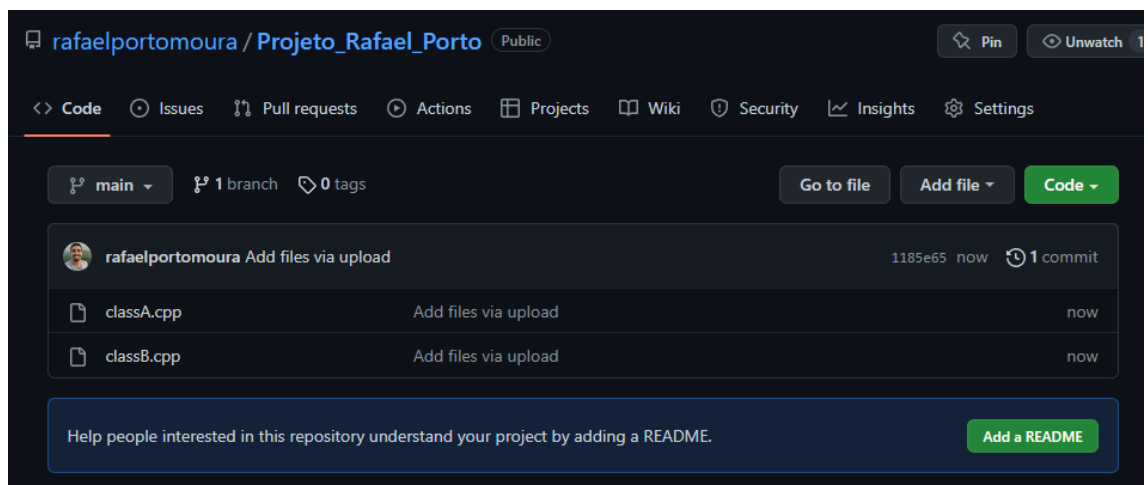
```

void setB2( float B2 ) { this->B2 = B2; }
void MB1() { cout << "MB1"; }
void MB2() { cout << "MB2"; }
};

int main() {

    return 0;
}

```



- 3- Faça o clone do repositório Criado no GitHub para sua máquina, isto é, o projeto criado no servidor do GitHub será copiado para sua máquina.

```

/mnt/c/Users/rafae/Documents/Projects/github
λ git clone git@github.com:rafaelportomoura/Projeto_Rafael_Porto.git
Cloning into 'Projeto_Rafael_Porto'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

```

Seção alteração e atualização do projeto:

- 4- Edite a Classe A clonada no seu computador, inserindo o método MA3() com uma linha de comando imprimindo na tela a string "". Crie uma classe C com os atributos String C1 e Int C2, os métodos MC1() e MC2(), cada um deles com um comando que imprime na tela o nome do método.

classA.cpp

```
#include <iostream>

using namespace std;

class A {
    private:
        int A1;
        float A2;
    public:
        A() {}
        int getA1() { return A1; }
        void setA1( int A1 ) { this->A1 = A1; }
        float getA2() { return A2; }
        void setA2( float A2 ) { this->A2 = A2; }
        void MA1() { cout << "MA1"; }
        void MA2() { cout << "MA2"; }
        void MA3() { cout << "Alteração a classe A partir
do clone"; }
};

int main() {

    return 0;
}
```

classC.cpp

```
#include <iostream>

using namespace std;

class C {
    private:
        int C1;
        string C2;
```

```

public:
C() {}
int getC1() { return C1; }
void setC1( int C1 ) { this->C1 = C1; }
string getC2() { return C2; }
void setC2( string C2 ) { this->C2 = C2; }
void MC1() { cout << "MC1"; }
void MC2() { cout << "MC2"; }
};

int main() {

    return 0;
}

```

- 5- Verifique o Status do repositório explicando a mensagem enviada pelo comando de quais arquivos estão desatualizados, atualizados, sob controle, sem controle etc.

```

Projeto_Rafael_Porto on main [!?]
λ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   classA.cpp

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        classC.cpp

no changes added to commit (use "git add" and/or "git commit -a")

```

A mensagem esta dizendo que a modificação no arquivo classA.cpp e um novo arquivo foi criado classC.cpp em relação a versão que está no servidor.

- 6- Garanta que todas as classes estejam sob controle de versão, adicionando as que não estiverem.
- 7- Aplique um commit na classe C enviando a mensagem "Criação da Classe C"

```
Projeto_Rafael_Porto on ✎ main [!?]
λ git add classC.cpp && git commit -m "Criação da Classe C"
[main 88f0f31] Criação da Classe C
1 file changed, 22 insertions(+)
create mode 100644 classC.cpp
```

- 8- Aplique um commit na classe A enviando a mensagem “Criado o Método MA3()”

```
Projeto_Rafael_Porto on ✎ main [!?]
λ git add classA.cpp && git commit -m "Criando o Método MA3()"
[main d041eff] Criando o Método MA3()
1 file changed, 2 insertions(+), 1 deletion(-)
```

- 9- Se houver algum arquivo desatualizado faça atualização com commit e descarregue todos os arquivos no servidor.

```
Projeto_Rafael_Porto on ✎ main [!?]
λ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 1.17 KiB | 30.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:rafaelportomoura/Projeto_Rafael_Porto.git
1185e65..5c82225 main -> main
```

Seção de rotulação (tag) de versões liberadas (releases) do projeto e uso de tags:

- 10- Crie uma Tag chamada “Versão 1.0”. Esta tag deve ser usada para rotular a última versão das classes A, B e C.

```
Projeto_Rafael_Porto on ✎ main took 2s
λ git tag -a "Versão 1.0" -m "Rotulando a versão 1.0 das classes A,B e C"
fatal: 'Versão 1.0' is not a valid tag name.

Projeto_Rafael_Porto on ✎ main
λ git tag -a Versão1.0 -m "Rotulando a versão 1.0 das classes A,B e C"
```

```
Projeto_Rafael_Porto on ✎ tags/Versão1.0
λ git checkout Versão1.0
HEAD is now at 5c82225 Conserta o Método setB2()
```

```
Projeto_Rafael_Porto on ✎ tags/Versão1.0
λ git push --set-upstream origin tags/Versão1.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 194 bytes | 3.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To github.com:rafaelportomoura/Projeto_Rafael_Porto.git
* [new tag] Versão1.0 -> Versão1.0
```

Tags

Versão1.0

8 minutes ago 5c82225 zip tar.gz

11- Crie uma nova Release da Tag “Versão 1.0”

Releases / Versão1.0

Release Versão 1.0

Compare ✎

rafaelportomoura released this now · 3 commits to main since this release Versão1.0 5c82225

Versão1.0

Rotulando a versão 1.0 das classes A,B e C

Assets 2

Source code (zip)

23 minutes ago

Source code (tar.gz)

23 minutes ago

12- Altere localmente a Classe B e C, criando, respectivamente, os métodos MB3() e MC3() para aquelas classes. Os métodos deverão ter o comando para imprimir na tela o nome do método.

classC.cpp

```
#include <iostream>

using namespace std;

class C {
private:
    int C1;
```

```

    string C2;
public:
    C() {}
    int getC1() { return C1; }
    void setC1( int C1 ) { this->C1 = C1; }
    string getC2() { return C2; }
    void setC2( string C2 ) { this->C2 = C2; }
    void MC1() { cout << "MC1"; }
    void MC2() { cout << "MC2"; }
    void MC3() { cout << "MC3"; }
};

int main() {

    return 0;
}

```

classB.cpp

```

#include <iostream>

using namespace std;

class B {
private:
    int B1;
    float B2;
public:
    B() {}
    int getB1() { return B1; }
    void setB1( int B1 ) { this->B1 = B1; }
    float getB2() { return B2; }
    void setB2( float B2 ) { this->B2 = B2; }
    void MB1() { cout << "MB1"; }
    void MB2() { cout << "MB2"; }
    void MB3() { cout << "MB3"; }
};

```



```
int main() {  
  
    return 0;  
}
```

- 13- Crie uma classe D com os atributos D1 e D2, e os 2 métodos MD1() e MD2(). As assinaturas dos métodos e tipos de dado são de sua escolha.

```
classD.cpp  
  
#include <iostream>  
  
using namespace std;  
  
class D {  
    private:  
        int D1;  
        string D2;  
    public:  
        int getD1() { return D1; }  
        void setD1( int D1 ) { this->D1 = D1; }  
        string getD2() { return D2; }  
        void setD2( string D2 ) { this->D2 = D2; }  
        void MD1() { cout << "MD1"; }  
        void MD2() { cout << "MD2"; }  
};  
  
int main() {  
    return 0;  
}
```

- 14- Atualize o servidor com as alterações feitas, fazendo commit com mensagem adequada para cada classe.

```
Projeto_Rafael_Porto on ✎ main [!?]
λ git add classB.cpp && git commit -m "Adiciona método MB3"
[main 609b0fa] Adiciona método MB3
1 file changed, 1 insertion(+)
```

```
Projeto_Rafael_Porto on ✎ main [!?]
λ git add classC.cpp && git commit -m "Adiciona método MC3"
[main 92d3ab7] Adiciona método MC3
1 file changed, 1 insertion(+)
```

```
Projeto_Rafael_Porto on ✎ main [!?]
λ git add classD.cpp && git commit -m "Criação da classe D"
[main 7314655] Criação da classe D
1 file changed, 20 insertions(+)
create mode 100644 classD.cpp
```

```
Projeto_Rafael_Porto on ✎ main [!:]
λ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 1.01 KiB | 25.00 KiB/s, done.
Total 9 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To github.com:rafaelportomoura/Projeto_Rafael_Porto.git
5c82225..7314655 main -> main
```

- 15- Crie uma nova Tag chamada “Versão 2.0”. Esta tag deve ser usada para rotular o contendo a última versão das classes A, B, C e D, solicitadas anteriormente.

```
Projeto_Rafael_Porto on ✎ main took 5s
λ git tag -a Versão2.0 -m "Rotulando versão 2.0"
```

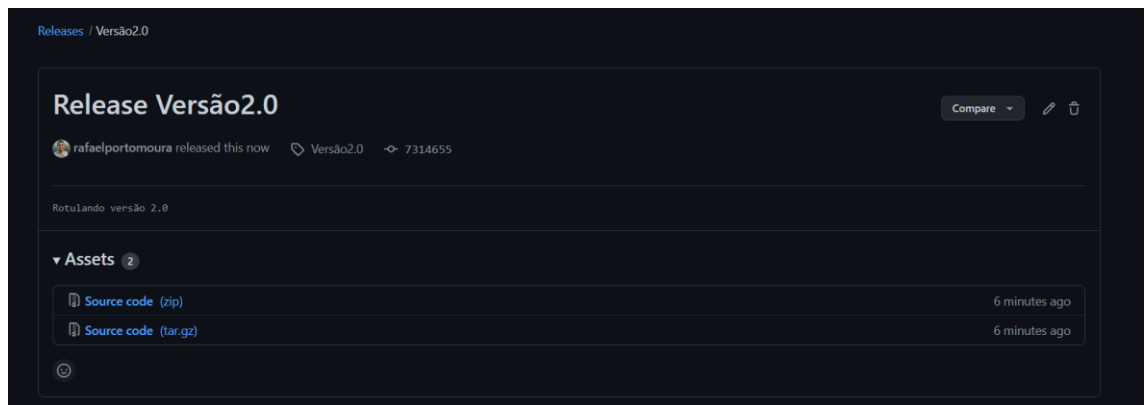
```
Projeto_Rafael_Porto on ✎ main
λ git push --set-upstream origin tags/Versão2.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 176 bytes | 2.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To github.com:rafaelportomoura/Projeto_Rafael_Porto.git
* [new tag] Versão2.0 -> Versão2.0
```

Tags

Versão2.0

5 minutes ago 7314655 zip tar.gz

- 16- Crie uma nova Release da Tag “Versão 2.0”



17- Faça um clone para sua máquina do software cuja Tag é “Versão 1.0”. Isto é possível? Caso não seja possível, como um desenvolvedor deve proceder para clonar uma versão de software que corresponda a uma Tag desejada?

Não é possível clonar somente uma branch ou tag. Porém para que o desenvolvedor consiga ver a versão desejada no código ele pode usar o comando `git checkout <tag ou branch>` sendo <tag ou branch> uma variável para a tag ou branch desejável.

Seção de resolução de conflitos (merge): Aqui, devemos simular dois programadores alterando ao mesmo tempo um arquivo. O aluno pode usar duas contas de email ou ele pode fazer 1 (uma) alteração via interface web do GITHUB e uma segunda alteração na pasta local do projeto que está na sua máquina. Ao tentar dar um commit e push, o sistema acusará conflito. Assim o segundo usuário a retornar o arquivo para o servidor terá um conflito e deverá unir suas alterações com as alterações feitas pelo outro programador.

18- Clone a última versão do projeto na sua máquina local.

```

rafael@DESKTOP-J82L40H MINGW64 ~/Documents/Projects/github/Projeto_Rafael_Porto (main)
$ git show main
commit 73146557a70c95f5da45763a23fcbcd5122a441a (HEAD -> main, tag: Versão2.0)
Author: rafaelportomoura <rafaelportomoura@gmail.com>
Date: Sun Jul 3 14:04:58 2022 -0300

```

Criação da classe D

```

diff --git a/classD.cpp b/classD.cpp
new file mode 100644
index 0000000..7bff9f9
--- /dev/null
+++ b/classD.cpp
@@ -0,0 +1,20 @@
+#include <iostream>
+
+using namespace std;
+
+class D {
+ private:
+ int D1;
+ string D2;
+ public:
+ int getD1() { return D1; }
+ void setD1( int D1 ) { this->D1 = D1; }
+ string getD2() { return D2; }
+ void setD2( string D2 ) { this->D2 = D2; }
+ void MD1() { cout << "MD1"; }
+ void MD2() { cout << "MD2"; }
+};
+
+int main() {
+ return 0;
+}
\ No newline at end of file

```

19- Abra a classe D e inclua o novo método MD3() nela, imprimindo o nome do método.

classD.cpp
<pre> #include <iostream> using namespace std; class D { private: int D1; string D2; public: int getD1() { return D1; } void setD1(int D1) { this->D1 = D1; } string getD2() { return D2; } void setD2(string D2) { this->D2 = D2; } void MD1() { cout << "MD1"; } </pre>

```

void MD2() { cout << "MD2"; }
void MD3() { cout << "MD3"; }
};

int main() {
    return 0;
}

```

20- Salve localmente, SEM atualizar o servidor. (isto simula um usuário que está alterando um arquivo numa máquina)

```

Projeto_Rafael_Porto on main [!] took 1m 7s
λ git add classD.cpp && git commit -m "Adiciona método MD3"
[main b73873a] Adiciona método MD3
1 file changed, 1 insertion(+)

```

21- Acesse a classe D da última versão pelo github (browser). Crie o método MD4(), imprimindo o nome do método. Salve essa alteração no github. (isto é para simular outro usuário alterando o mesmo arquivo em outra máquina)

rafaelportomoura Atualiza classe D pela interface do GitHub Latest commit dbf841b now History

1 contributor

21 lines (18 sloc) 393 Bytes Raw Blame

```

1 #include <iostream>
2
3 using namespace std;
4
5 class D {
6     private:
7         int D1;
8         string D2;
9     public:
10        int getD1() { return D1; }
11        void setD1( int D1 ) { this->D1 = D1; }
12        string getD2() { return D2; }
13        void setD2( string D2 ) { this->D2 = D2; }
14        void MD1() { cout << "MD1"; }
15        void MD2() { cout << "MD2"; }
16        void MD4() { cout << "MD4"; }
17    };
18
19    int main() {
20        return 0;
21    }

```

22- Volte para o terminal ou interface gráfica que você está utilizando para trabalhar no repositório local. Atualize o servidor com as alterações feitas localmente na classe D (foi criado o método MD3()). Mostre a mensagem retornada ao tentar atualizar o servidor.

```
Projeto_Rafael_Porto on main [0]
λ git push
To github.com:rafaelportomoura/Projeto_Rafael_Porto.git
! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'git@github.com:rafaelportomoura/Projeto_Rafael_Porto.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

- 23- Faça o merge e verifique se os métodos MD3 e MD4 estão presentes na classe D ou se houve alguma perda de informação.

```
Projeto_Rafael_Porto on main [0] took 2s
λ git pull
Auto-merging classD.cpp
CONFLICT (content): Merge conflict in classD.cpp
Automatic merge failed; fix conflicts and then commit the result.
```

Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes | Start Live Share Session

```
<<<<<< HEAD (Current Change)
    void MD3() { cout << "MD3"; }
=====
    void MD4() { cout << "MD4"; }
>>>>>> dbf841b6ef5368d5050902f1079ca487e53b914c (Incoming Change)
};
```

classD.cpp

```
#include <iostream>

using namespace std;

class D {
private:
    int D1;
    string D2;
public:
    int getD1() { return D1; }
    void setD1( int D1 ) { this->D1 = D1; }
    string getD2() { return D2; }
    void setD2( string D2 ) { this->D2 = D2; }
    void MD1() { cout << "MD1"; }
    void MD2() { cout << "MD2"; }
    void MD3() { cout << "MD3"; }
    void MD4() { cout << "MD4"; }
```

```
};
```

```
int main() {  
    return 0;  
}
```

```
Projeto_Rafael_Porto on  (git)-[main|merge]- [8=] took 3s  
λ git add classD.cpp && git commit -m "Merge do conflito"  
[main ce041d1] Merge do conflito  
  
Projeto_Rafael_Porto on  main [⇄]  
λ git push  
Enumerating objects: 10, done.  
Counting objects: 100% (10/10), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (6/6), done.  
Writing objects: 100% (6/6), 576 bytes | 14.00 KiB/s, done.  
Total 6 (delta 4), reused 0 (delta 0)  
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.  
To github.com:rafaelportomoura/Projeto_Rafael_Porto.git  
dbf841b..ce041d1  main -> main
```

Seção de verificar diferença do mesmo arquivo em versões diferentes e também a diferença entre duas versões utilizando as tags de diferentes versões:

24- Consulta Artefato: Mostre a diferença entre as duas últimas versões das classes B e D

classD.cpp

Adiciona método MD3

rafaelportomoura on 7/3/2022, 2:50:32 PM

b73873a Soft Hard + Tag + Branch ⇄ More

Atualiza classe D pela interface do GitHub

Rafael Moura on 7/3/2022, 2:27:32 PM

dbf841b Soft Hard + Tag + Branch ⇄ More

```
1 #include <iostream>  
2  
3 using namespace std;  
4  
5 class D {  
6     private:  
7         int D1;  
8         string D2;  
9     public:  
10    int getD1() { return D1; }  
11    void setD1( int D1 ) { this->D1 = D1; }  
12    string getD2() { return D2; }  
13    void setD2( string D2 ) { this->D2 = D2; }  
14    void MD1() { cout << "MD1"; }  
15    void MD2() { cout << "MD2"; }  
16  
17 };  
18  
19 int main() {  
20     return 0;  
21 }
```

```
1 #include <iostream>  
2  
3 using namespace std;  
4  
5 class D {  
6     private:  
7         int D1;  
8         string D2;  
9     public:  
10    int getD1() { return D1; }  
11    void setD1( int D1 ) { this->D1 = D  
12    string getD2() { return D2; }  
13    void setD2( string D2 ) { this->D2  
14    void MD1() { cout << "MD1"; }  
15    void MD2() { cout << "MD2"; }  
16+   void MD3() { cout << "MD3"; }  
17  
18 };  
19  
20 int main() {  
21     return 0;  
22 }
```

classB.cpp

Adiciona método MB3

rafaelportomoura on 7/3/2022, 2:03:13 PM

609b0fa Soft Hard + Tag + Branch

Conserta o Método setB2()

rafaelportomoura on 7/3/2022, 3:02:46 AM

5c82225 Soft Hard + Tag + Branch

```
1 #include <iostream>
2
3 using namespace std;
4
5 class B {
6     private:
7         int B1;
8         float B2;
9     public:
10        B() {}
11        int getB1() { return B1; }
12        void setB1( int B1 ) { this->B1 = B1; }
13        float getB2() { return B2; }
14        void setB2( float B2 ) { this->B2 = B2; }
15        void MB1() { cout << "MB1"; }
16        void MB2() { cout << "MB2"; }
17    };
18
19 int main() {
20
21     return 0;
22 }
```

```
1 #include <iostream>
2
3 using namespace std;
4
5 class B {
6     private:
7         int B1;
8         float B2;
9     public:
10        B() {}
11        int getB1() { return B1; }
12        void setB1( int B1 ) { this->B1 = B1; }
13        float getB2() { return B2; }
14        void setB2( float B2 ) { this->B2 = B2; }
15        void MB1() { cout << "MB1"; }
16        void MB2() { cout << "MB2"; }
17        void MB3() { cout << "MB3"; }
18    };
19
20 int main() {
21
22     return 0;
23 }
```

25- Consulta Versões: Mostre a diferença entre as versões 1 e 2.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base: Versão1.0 ← compare: Versão2.0

3 commits

3 files changed

1 contributor

Commits on Jul 3, 2022

Adiciona método MB3

rafaelportomoura committed 1 hour ago

609b0fa <>

Adiciona método MC3

rafaelportomoura committed 1 hour ago

92d3ab7 <>

Criação da classe D

rafaelportomoura committed 1 hour ago

7314655 <>

Merge do conflito

origin/HEAD

origin/main

main

ce041d1

Soft

Hard

Tag

Branch

More

Adiciona método MD3

rafaelportomoura on 7/3/2022, 2:50:32 PM

b73873a

Soft

Hard

Tag

Branch

More

Atualiza classe D pela interface do GitHub

Rafael Moura on 7/3/2022, 2:27:32 PM

dbf841b

Soft

Hard

Tag

Branch

More

Criação da classe D

Versão 2.0

rafaelportomoura on 7/3/2022, 2:04:58 PM

7314655

Soft

Hard

Tag

Branch

More

Adiciona método MC3

rafaelportomoura on 7/3/2022, 2:03:58 PM

92d3ab7

Soft

Hard

Tag

Branch

More

Adiciona método MB3

rafaelportomoura on 7/3/2022, 2:03:13 PM

609b0fa

Soft

Hard

Tag

Branch

More

Conserta o Método setB2()

Versão 1.0

rafaelportomoura on 7/3/2022, 3:02:46 AM

5e82225

Soft

Hard

Tag

Branch

More

Criando o Método MA3()

rafaelportomoura on 7/3/2022, 3:01:44 AM

d041eff

Soft

Hard

Tag

Branch

More

Criação da Classe C

rafaelportomoura on 7/3/2022, 3:00:54 AM

88f0f31

Soft

Hard

Tag

Branch

More

Add files via upload

Rafael Moura on 7/3/2022, 2:24:07 AM

1185e65

Soft

Hard

Tag

Branch

More