**Grupo**:

* Pedro Paulo da Silva - 1421422
* Stephanie Fay - 1721631
* Thiago Lages de Alencar - 1721629

Modelo da classe DireitoDeUso

class DireitoDeUso {

private:

string nome;

string apelido;

string email;

char direitos\_de\_uso[NUMERO\_DE\_DIREITOS];

public:

DireitoDeUso(char direitos[]);

DireitoDeUso(string nome, string apelido, string email, char direitos[]);

~DireitoDeUso();

bool TemDireito(char idDireito);

// Getters

string GetNome();

string GetApelido();

string GetEmail();

char\* GetDireitosDeUso();

/\*

EU NÃO ENTENDO DE HACKERS, ENTÃO POSSO ESTAR FALANDO BESTEIRA AQUI.

Setters (SetNome/SetApelido/SetEmail/SetDireitos) são desnecessários já que tiram a necessidade desse objeto.

Uma vez que você quer evitar que hackers alterem os direitos/informações, você deve fazer esse objeto ser criado apenas quando o usuário loga no sistema. Caso algum direito tenha sido alterado, o sistema deve criar um objeto novo

\*/

}

Só três errinhos

deve existir somente o construtor com 4 parâmetros

não deve existir GetDireitosDeUso – o TemDireito já realiza a função relevante

O componente sempre retorna o objeto construído, logo deve conter a condição de retorno – autorizado, não autorizado, erro, troca senha, cancela.

Deve ter um método que retorna a condição.

E DireitoDeUso deve verificar se está autorizado.

DireitoDeUso(char direitos[])

**AE**:

login\_bem\_sucedido == true

**AS**:

this->nome == 𝝓 **∧** this->apelido == 𝝓 **∧** this->email == 𝝓 **∧**

(**∀** d **∊** D **|** (**∃** c **∊** C **|** this->direitos\_de\_uso[ d ] == direitos[ c ]))

DireitoDeUso(string nome, string apelido, string email, char direitos[])

**AE**:

login\_bem\_sucedido == true **∧**

nome != 𝝓 **∧** apelido != 𝝓 **∧** email != 𝝓

**AS**:

this->nome != 𝝓 **∧** this->apelido != 𝝓 **∧** this->email != 𝝓 **∧**

(**∀** d **∊** D **|** (**∃** c **∊** C **|** this->direitos\_de\_uso[ d ] == direitos[ c ]))

~DireitoDeUso()

**AE**:

**AS**:

this->nome == 𝝓 **∧** this->apelido == 𝝓 **∧** this->email == 𝝓 **∧**

(**∀** d **∊** D **|** direitos\_de\_uso[ d ] == 𝝓 )

bool TemDireito(char idDireito)

**AE**:

**AS**:

(**∃** d **∊** D **|** this->direitos\_de\_uso[ d ] == idDireito) **∨**

**!**(**∃** d **∊** D **|** this->direitos\_de\_uso[ d ] == idDireito)

char\* GetNome( )

**AE**:

só vale se condição retorno == autoriza ou troca senha

**AS**:

nova\_string == this->nome

char\* GetApelido( )

**AE**:

só vale se condição retorno == autoriza ou troca senha

**AS**:

nova\_string == this->apelido

char\* GetEmail( )

**AE**:

só vale se condição retorno == autoriza **AS**:

nova\_string == this->email

char\* GetDireitosDeUso( )

**AE**:

não deve existir

**AS**:

(**∀** d **∊** D **|** (**∃** c **∊** C **|** this->direitos\_de\_uso[ d ] == novo\_array[ c ]))