



## 04 - ATIVIDADE AVALIATIVA - 2BIM

### 1. Criação do projeto

- 1.1. Abra o *IntelliJ* e crie um novo projeto. Caso necessário, acesse o menu **File** → **New** → **Project....**
- 1.2. Primeiramente, defina o local de gravação em **D:/2AI/pw2/nomes-dupla/pw2-atividade-2bim**.
- 1.3. Coloque o nome de projeto **pw2-atividade-2bim** e clique em **Create**.
- 1.4. Acesse a aba **Terminal** na parte inferior do IntelliJ.
- 1.5. Configure o usuário e e-mail do *git* com os dados da dupla:  

```
git config --global user.name "nome1-nome2"
git config --global user.email nome1-nome2@test.com
```
- 1.6. Inicialize o repositório com `git init`.
- 1.7. Use `git add .` para adicionar os arquivos do projeto.
- 1.8. Confirme as alterações com `git commit -m "Commit inicial"`.
- 1.9. Acesse o **github.com** e entre em sua conta.
- 1.10. Crie o repositório **pw2-atividade-2bim** e, após a criação, copie o último bloco de linhas exibido.
- 1.11. Retorne ao IntelliJ, cole o conteúdo copiado e realize a autorização quando solicitado.
- 1.12. Após o procedimento, dê um refresh na tela do **github.com** para confirmar o upload do projeto.

### 2. Status de usuário

- 2.1. Crie a enumeração **StatusUsuarioEnum** com os valores ATIVO, INATIVO e BLOQUEADO.
- 2.2. Acesse a aba **Terminal** na parte inferior do IntelliJ.
- 2.3. Verifique a situação dos arquivos com `git status`.
- 2.4. Adicione os arquivos do projeto ao git com `git add ..`
- 2.5. Confirme as alterações com `git commit` e a mensagem "Criacao da enumeracao StatusUsuarioEnum".



- 2.6. Envie as alterações ao **github.com** com o comando `git push`.
- 2.7. Após o procedimento, dê um *refresh* na tela do **github.com** para confirmar o upload do projeto.

---

### 3. Classe Usuario

---

- 3.1. Crie a classe **Usuario** com os seguintes atributos privados: código, nome, senha, tentativas de acesso, primeiro acesso (verdadeiro/falso), data de inativação e status.
- 3.2. Crie os métodos *getters* e *setters* para os atributos privados.
- 3.3. Codifique um construtor para que, ao criar um objeto do tipo **Usuario**, sempre defina a senha padrão **etec#123**, que se trata de um primeiro acesso (`true`) e o status **ATIVO**.
- 3.4. Quando o usuário trocar a senha, via método `setSenha()`, colocar `false` na flag do primeiro acesso, status será **ATIVO** e zerar as tentativas de acesso.
- 3.5. Além disso, inclua um método para inativar o usuário que incluirá a data no momento da inativação (`LocalDate.now()`), o primeiro acesso para `false` e status **INATIVO**.
- 3.6. Codifique, também, outro método que exiba um log dos dados conforme abaixo:  
Código: 1  
Usuário: horacio.augusto  
Senha: etec#123  
Trocar senha: true  
Status: ATIVO
- 3.7. Crie a classe **UsuarioTeste** que instancie um novo usuário, preencha o código e o nome do usuário e exiba o *log* de registro criado. Em seguida, inative o usuário e exiba novamente o log de registro.
- 3.8. Acesse a aba **Terminal** na parte inferior do IntelliJ.
- 3.9. Verifique a situação dos arquivos com `git status`.
- 3.10. Adicione os arquivos do projeto ao git com `git add ..`
- 3.11. Confirme as alterações com `git commit` e a mensagem "Inclusao da classe Usuario".
- 3.12. Envie as alterações ao **github.com** com o comando `git push`.
- 3.13. Após o procedimento, dê um *refresh* na tela do **github.com** para confirmar o upload do projeto.



---

## 4. Autenticação de usuários

---

- 4.1.** Crie um novo método chamado `autenticar()` que receberá uma senha como parâmetro e retornará uma *String*. O método verificará:
- se o status do usuário é `INATIVO` ou `BLOQUEADO`, retornará "Acesso negado";
  - se as tentativas de acesso é maior ou igual a 3, o status será `BLOQUEADO` e retornará "Acesso negado";
  - se o atributo senha for igual a senha informada no parâmetro, zerará as tentativas de acesso e retornará "Acesso liberado". Se forem diferentes, incrementará mais um nas tentativas de acesso e retornará "Usuário/senha inválidos".
- 4.2.** Altere a classe **UsuarioTeste** para chamar o método `autenticar()` e exiba o log de registro.
- 4.3.** Faça os testes de autenticação, passando senhas diferentes até o usuário ser bloqueado.
- 4.4.** Acesse a aba **Terminal** na parte inferior do IntelliJ.
- 4.5.** Verifique a situação dos arquivos com `git status`.
- 4.6.** Adicione os arquivos do projeto ao git com `git add ..`
- 4.7.** Confirme as alterações com `git commit` e a mensagem "Inclusao de metodo de autenticacao".
- 4.8.** Envie as alterações ao [github.com](https://github.com) com o comando `git push`.
- 4.9.** Após o procedimento, dê um *refresh* na tela do [github.com](https://github.com) para confirmar o upload do projeto.