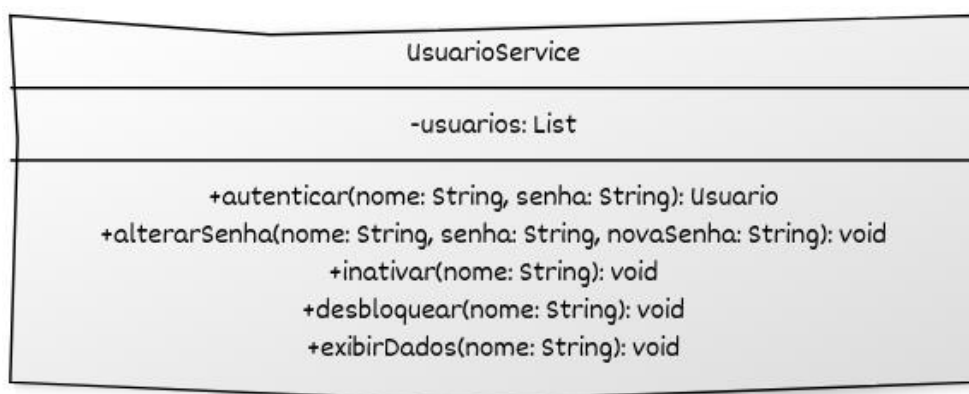




04 - ATIVIDADE AVALIATIVA - 2BIM

1. Modelagem da classe

- 1.1. Abra o IntelliJ, acesse o menu **File** → **New** → **Project** e coloque o nome de **pw2-seu_nome-av2bim**.
- 1.2. Crie o pacote **br.com.etechoracio.avaliacao.enums**.
- 1.3. Copie a classe **StatusUsuarioEnum** da rede para o pacote **enums**.
- 1.4. Crie o pacote **br.com.etechoracio.avaliacao.model**.
- 1.5. Copie a classe **Usuario** da rede para o pacote **model**.
- 1.6. Crie o pacote **br.com.etechoracio.avaliacao.exception**.
- 1.7. Crie a classe de exceção chamada **CredenciaisInvalidasException** dentro do pacote **model**. A exceção receberá uma mensagem de erro via construtor.
- 1.8. Altere o método **autenticar()** da classe **Usuario** para que lance a exceção **CredenciaisInvalidasException** nos casos de acesso negado ou usuário e senha inválidos colocando as respectivas mensagens de erro.
- 1.9. Crie a classe de exceção chamada **PrimeiroAcessoException** dentro do pacote **model**.
- 1.10. Altere o método **autenticar()** da classe **Usuario** para que lance a exceção **PrimeiroAcessoException** no caso de ser o primeiro acesso.
- 1.11. Crie o pacote **br.com.etechoracio.avaliacao.service**.
- 1.12. Crie a classe **UsuarioService** dentro do pacote **service**.



- 1.13. Inclua um atributo com uma lista de usuários e, no construtor da classe, adicione 3 usuários à lista: admin, TI e RH.
- 1.14. O método **autenticar()** verificará se o nome passado como parâmetro existe na lista, executará o método **autenticar()** do usuário encontrado na lista e retorná-lo.
- 1.15. O método **alterarSenha()** verificará se o nome passado como parâmetro existe na lista, chamará o método **autenticar()** do usuário encontrado e também o método **alterarSenha()** com a nova senha.



- 1.16. O método **inativar()** verificará se o nome passado como parâmetro existe na lista e executará o método **inativar()** do usuário encontrado.
- 1.17. O método **bloquear()** verificará se o nome passado como parâmetro existe na lista e executará o método **bloquear()** do usuário encontrado.
- 1.18. O método **exibir()** verificará se o nome passado como parâmetro existe na lista e executará o método **exibir()** do usuário encontrado.
- 1.19. Crie o pacote **br.com.etechoracio.avaliacao.test**.
- 1.20. Crie a classe **TesteAutenticacaoValida**, dentro do pacote **test**, com o método **main()**. Instancie um objeto do tipo **UsuarioService** e escreva o código para que ocorra uma autenticação válida.
- 1.21. Crie a classe **TesteBloqueioUsuario**, dentro do pacote **test**, com o método **main()**. Instancie um objeto do tipo **UsuarioService** e escreva o código para que ocorra um bloqueio de usuário após 3 tentativas de autenticação com senha inválida. Utilize o método **exibir()** do service para “printar” os dados do usuário.
- 1.22. Crie a classe **TestePrimeiroAcesso**, dentro do pacote **test**, com o método **main()**. Instancie um objeto do tipo **UsuarioService** e escreva o código para que ocorra a exceção **PrimeiroAcessoException** via **autenticar**, faça o tratamento dessa exceção efetuando a troca de senha e chamando novamente a autenticação. Utilize o método **exibir()** do service para “printar” os dados do usuário.