

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
GRADUAÇÃO – CIÊNCIA DA COMPUTAÇÃO

JULIA LLORENTE
NICOLE CAROLINA MENDES

PROJETO FASE 1: SOFTWARE PARA A DETECÇÃO DE FRAUDE DE
CARTÕES DE CRÉDITO

JOINVILLE

2022

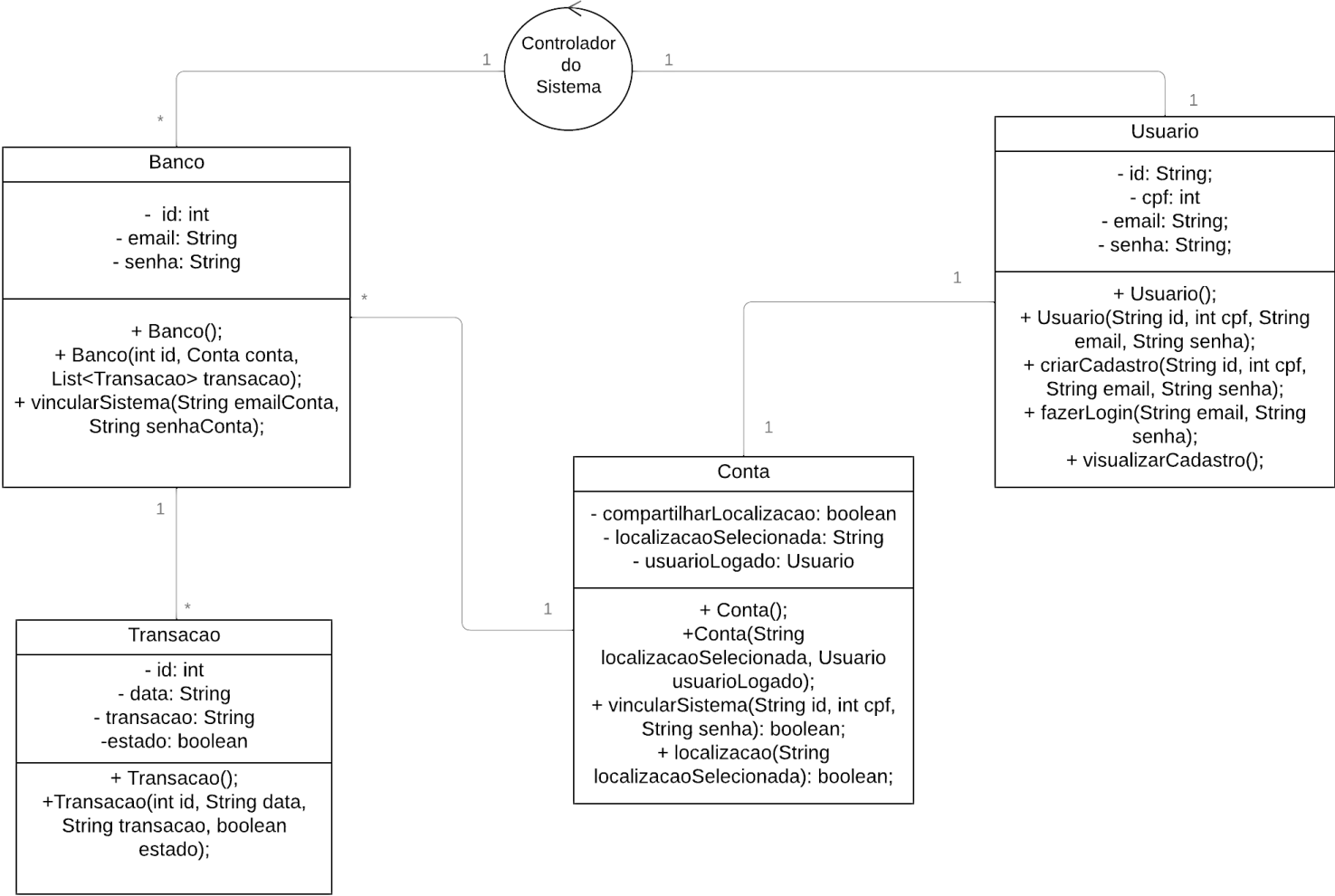
JULIA LLORENTE
NICOLE CAROLINA MENDES

**PROJETO FASE 1: SOFTWARE PARA A DETECÇÃO DE FRAUDE DE
CARTÕES DE CRÉDITO**

JOINVILLE

2022

DIAGRAMA DE CLASSES



TESTES UNITÁRIOS

O diagrama de classes sofreu algumas alterações devido a implementação do sistema, logo alguns atributos e métodos foram alterados. Além disso, algumas classes que não estavam sendo utilizadas foram retiradas e duas classes foram adicionadas para a implementação dos testes.

Os testes abaixo foram escolhidos por conta da dificuldade em relação aos testes e ao tema escolhido. Durante a implementação do código, foi visto que infelizmente o tema escolhido se tornou difícil durante a verificação dos testes unitários, pois o modo como foi criado os atributos e métodos da localização envolviam conceitos acima do nosso conhecimento.

ContaTest:

```
void localizacaoFalsa() {  
    Usuario usuario = new Usuario();  
    Conta conta = new Conta("Brasil", usuario);  
    assertFalse(conta.getCompartilharLocalizacao());  
}
```

Nessa função, verificamos que, após inicializar a conta, a localização deve ser falsa pois ainda não foi feito o set da conta para compartilhar a localização.

```
void compartilharLocalizacao() {  
    Usuario usuario = new Usuario();  
    Conta conta = new Conta("Brasil", usuario);  
    assertFalse(conta.localizacao(""));  
}
```

Nessa função, após inicializar a conta, verificamos que a opção de localização tem que ser falsa, pois ainda não foi solicitado ao usuário de ativar a localização.

```
void VincularSistema() {  
    Usuario usuario = new Usuario("1", 124, "banana@gmail.com", "123");  
    Conta conta = new Conta("Brasil", usuario);  
    assertTrue(conta.vincularSistema("1", 124, "123"));  
}
```

Nessa função, é feito a verificação de vinculação de conta, onde os dados inseridos devem estar corretos para a verificação ser concluída.

UsuarioTest:

```
void emailIncorreto() {  
    Usuario usuario = new Usuario("1", 124, "banana@gmail.com", "123");  
    Exception thrown = assertThrows(Exception.class, () ->  
    {usuario.fazerLogin("banana1@gmail.com", "senha123");});  
    assertTrue(thrown.getMessage().contentEquals("Email incorreto."));  
}
```

Nessa função é feita a verificação em relação ao e-mail inserido pelo usuário, onde, caso seja diferente, é lançado uma exceção.

```
void senhaIncorreta() {  
    Usuario usuario = new Usuario("1", 124, "banana@gmail.com", "123");  
    Exception thrown = assertThrows(Exception.class, () ->  
    {usuario.fazerLogin("banana@gmail.com", "senha12");});  
    assertTrue(thrown.getMessage().contentEquals("Senha incorreta."));  
}
```

Nessa função é feita a verificação em relação a senha inserida pelo usuário, onde, caso seja diferente, é lançado uma exceção.