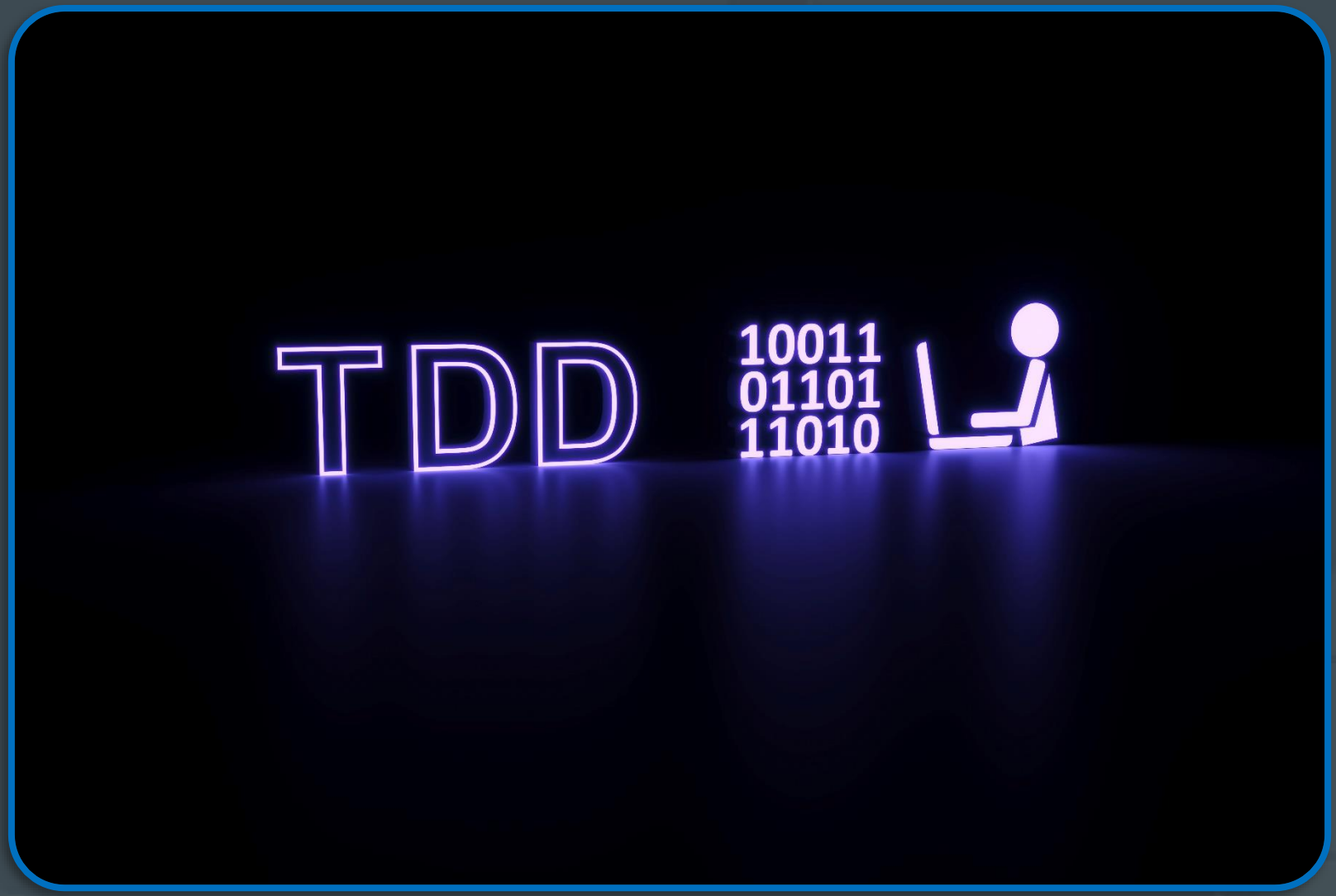
Aula prática 9

Escola Politécnica

Teste de software

Profª Maristela Weinfurter



Profit_Image/Shutterstock

TDD - Testes unitários

User story: transferência de dinheiro

User story	Como usuário do aplicativo BancoLegal, quero transferir dinheiro da minha conta bancária para outra conta usando o aplicativo móvel BancoLegal, para que eu possa gerenciar facilmente minhas finanças e pagar minhas contas no prazo.
Critérios de aceitação	▪ O usuário pode fazer login no aplicativo móvel ▪ O usuário pode entrar com o valor desejado para transferir ▪ O valor desejado está dentro do saldo atual do usuário ▪ As informações da conta do favorecido são autenticadas antes de transferir o valor ▪ Ambas as contas são atualizadas simultaneamente após a validação ▪ O e-mail e a confirmação no aplicativo são enviados ao usuário e ao favorecido

TDD - Testes unitários

Test case: transferência de dinheiro

Test case	TC 023: Autenticação do favorecido
1. Pré-condições 2. Procedimento	1. Início na UI de login do aplicativo móvel 2. Ações: a) O usuário informa login inválido e/ou senha inválida b) O usuário informa um valor desejado para transferir fora do limite configurado no aplicativo. c) O app verifica se foi informado um valor válido (deve ser número) d) O app valida se o valor desejado está dentro do saldo do usuário e) O usuário informa a chave do favorecido para transferência f) O app verifica se a chave é válida g) ...

TDD - Testes unitários

Test case: transferência de dinheiro

Test case	TC 023: Autenticação do favorecido
3. Resultado esperado	3. Mensagens de erro no aplicativo
4. Dados de entrada	4. Login, senha, valor e favorecido válidos
5. Prioridade	5. Alta
6. Ambiente	6. IOS e Android
7. Técnica	7. Manual
8. Iteração	8. 1a. iteração

TDD - Testes unitários

- Observando nossa card de Caso de Teste, vamos caminhar no desenvolvimento dos testes unitários
- Vamos utilizar a linguagem Python para compreendermos como os testes caminham dentro de um conceito de desenvolvimento de software orientado a testes

bsd studio/Shutterstock

TDD - Testes unitários

```
import unittest


class TestSaldo(unittest.TestCase):
    def test_saldo_cliente(self):
        resultado = saldo(1000, 1000)
        self.assertEqual(resultado, 2)


if __name__ == '__main__':
    unittest.main()
```

Bakhtiar Zein/Shutterstock

TDD - Testes unitários

```
E
======================================================================
ERROR: test_saldo_cliente (__main__.TestSaldo)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "tdd_saldo.py", line 9, in test_saldo_cliente
    resultado = saldo(1, 1)
NameError: global name 'saldo' is not defined
----------------------------------------------------------------------
Ran 1 test in 0.000s
FAILED (errors=1)
```

TDD - Testes unitários

```
def saldo(a, b):
    return a - b
```

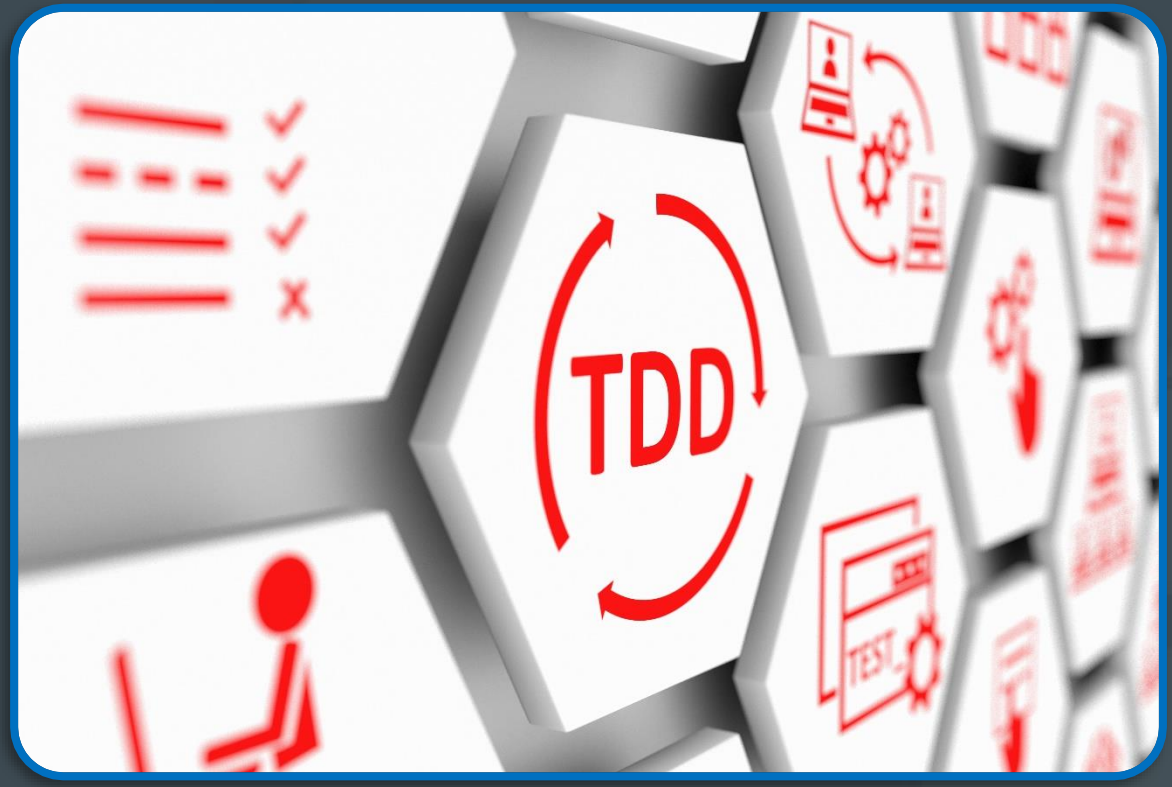
```
class TestSaldo(unittest.TestCase):
  # [...]
```

TDD - Testes unitários

```
def saldo(a, b):
    return a - b


class TestSaldo(unittest.TestCase):
  def test_saldo_cliente(self):
    # [...]
  def test_saldo_consolidado(self):
      with self.assertRaises(ValueError) as error:
          saldo('a', 'b')
          self.assertEqual(error.msg, u'Somente números são
permitidos')
```

- E assim vamos incrementando, um novo teste e um novo código (componente), até que o código esteja completo



Profit_Image/Shutterstock

Fechar