

Clenio Emidio  
Esp.Gov.TI

# Testes, métricas e Qualidade de Software

# Apresentação

- Carga horária disciplina: 60h
- Horários
- Celular
- Avaliações
- Manter comunicação
- Representantes:
- Metodologia: Atividades teóricas, trabalhos, pesquisas, fóruns, prática no notebook.

# Apresentação



- <https://www.linkedin.com/in/clenio-emidio-12a37048/>

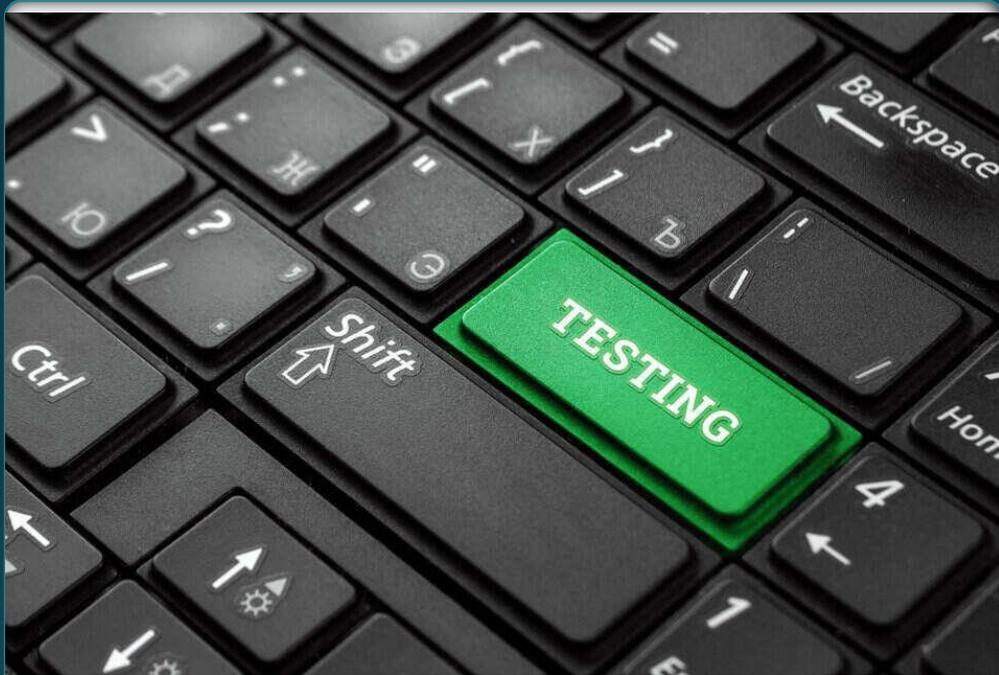


- @clenioemidio



- <http://lattes.cnpq.br/6532127787803030>

# 1-Teste de Software



- É um **conjunto** de **processos** com os quais se pretende **validar** um sistema ou aplicação, em momentos diferentes, para verificar seu correto funcionamento.
- O teste do software é a investigação do software a fim de fornecer informações sobre sua qualidade em relação ao contexto em que ele deve operar, se relaciona com o conceito de verificação e validação. Isso inclui o processo de utilizar o produto para encontrar seus defeitos
- Ex. Teste de carregamento, picos, segurança, sistema, ambiente, integração, desempenho, resistência, caixa preta, caixa branca, funcionais, não funcionais, testes de regressão...
- [SmartMeter](#), NeoLoad, Silk Performer, AppLoader, BlazeMeter, Pytest, Unittest, Junit, PHPunit, Selenium.

# 2-Métricas



- Métricas de software servem para apresentar **medidas**, preferencialmente **quantitativas**, que reflitam características específicas de processos e de produtos em construção, podendo ser utilizadas em diferentes dimensões, como esforço, tamanho, complexidade, dentre outras.
- **Medição de software** auxilia a tomada de decisão, pois através de dados quantitativos, é capaz de informar que aspectos do produto atendem ou não ao padrão de qualidade especificado, além de permitir a avaliação dos benefícios de novos métodos e ferramentas de engenharia de software, o entendimento e aperfeiçoamento do processo de produção, a avaliação do retorno do investimento e tornar o gerenciamento de projetos baseado em fatos e não “achismos”.

## Métricas de desempenho de software

- Escalabilidade. Estabilidade. Capacidade de resposta. Velocidade. Disponibilidade.

# 3- Qualidade de software

- A norma internacional ISO/IEC, define qualidade de software como A totalidade de características de um produto de software que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas.
- Segundo a NBR ISO 9000:2005, “qualidade é o grau no qual um conjunto de características inerentes satisfaz aos requisitos”.
- **Políticas de qualidade**, sobretudo quando se deseja buscar mercados externos ou expandir seus clientes internos aumentando a satisfação dos mesmos, além de alavancar a competitividade das empresas.

NORMAS	COMENTÁRIOS
ISO 9126	Características da qualidade de produtos de software.
<b>NBR 13596</b>	<b>Versão brasileira da ISO 9126.</b>
ISO 14598	Guias para a avaliação de produtos de software, baseados na utilização prática da norma ISO 9126.
ISO 12119	Características de qualidade de pacotes de software (software de prateleira, vendido como um produto embalado).
IEEE P1061	Standard for Software Quality Metrics Methodology. Norma que trata das metodologias para padronização da qualidade de software, incluindo algumas abordagens de medição.
ISO 12207	Software Life Cycle Process. Norma para a qualidade do processo de desenvolvimento de software.
NBR ISO 9001	Sistemas de qualidade – Modelo para garantia de qualidade em projeto, desenvolvimento, instalação e assistência técnica (processo).
NBR ISO 9000-3	Gestão de qualidade e garantia de qualidade. Aplicação da norma ISO 9000 para o processo de desenvolvimento de software.
NBR ISO 10011	Auditoria de Sistemas de Qualidade (processo).
CMMI	Capability Maturity Model Integration. Modelo da SEI (Instituto de Engenharia de Software do Departamento de Defesa dos USA) para avaliação da qualidade do processo de desenvolvimento de software. Não é uma norma ISO, mas é muito bem aceita no mercado.
SPICE ISO 15504	Projeto da ISO/IEC para avaliação do processo de desenvolvimento de software. Ainda não é uma norma oficial ISO, mas o processo está em andamento.

O que é software?

O que é a qualidade de software?

# Quanto ao seu objetivo, um software pode ser classificado de três maneiras:

- **Software aplicativo**

- São aplicações projetadas para resolver problemas específicos dos usuários.

- **Software de programação**

- São ferramentas pelas quais os desenvolvedores programam novas aplicações e softwares, utilizando uma linguagem de programação específica.

- **Software de sistema**

- São conjuntos de outros softwares que têm a finalidade de ofertar serviços e funções para outros programas.

# Tipos de código e formato de softwares

- **Código aberto**

- Um software de código aberto é aquele que é de fácil acesso e que um desenvolvedor pode fazer alteração em seu código-fonte, além de poder distribuir esta nova “versão” do software, desde de que o programador respeite as licenças daquela aplicação. Este tipo de software é conhecido também como *open source*.

- **Código fechado**

- Também conhecidos como *closed source*, são aqueles em que, ao contrário do código aberto, as alterações e distribuição do programa são proibidas.

- **Software livre**

- Este tipo de código refere-se aqueles que podem ser executados, copiados, distribuídos e modificados livremente. Isto não significa que a aplicação foi gratuita, é possível ter comprado sua cópia, ou tê-la adquirido gratuitamente, com os mesmos benefícios.

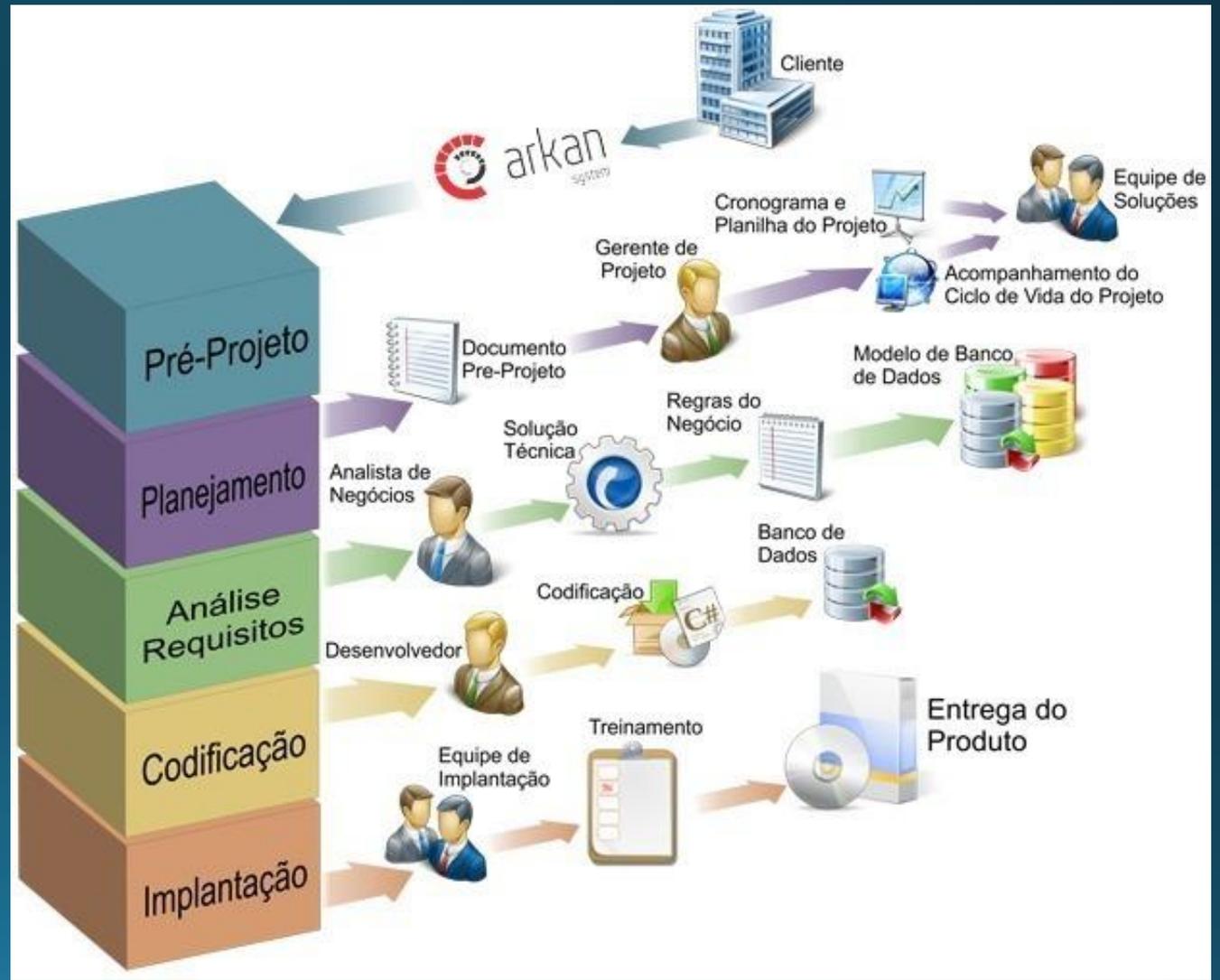
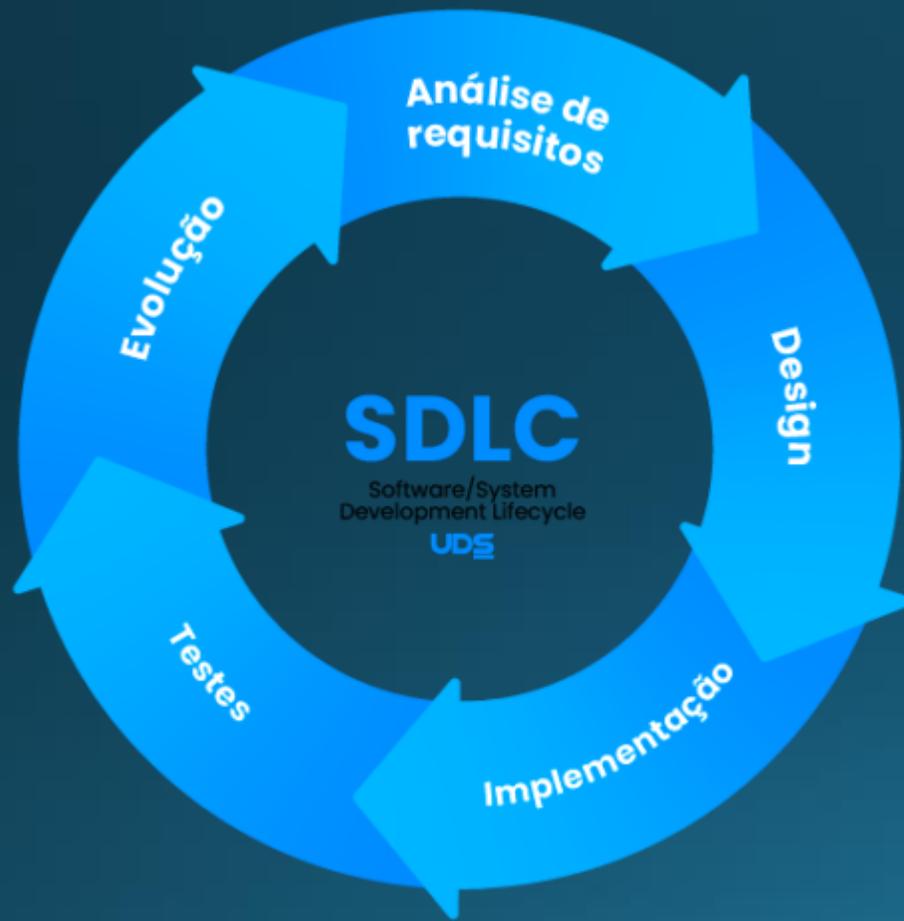
- **Software proprietário**

- Este é um exemplo de software no qual é possível utilizar, modificar e distribuir o código, desde que o usuário tenha comprado uma licença e solicitado permissão para distribuir e modificar o código-fonte.

- **Software freeware**

- É um tipo de programa que é oferecido gratuitamente, sendo permitida a instalação e execução do software sem restrição. Em alguns casos, o mesmo software possui a versão gratuita (freeware) e a versão paga (premium).

# Ciclo de Desenvolvimento de Software



# Ferramentas IDE

- Notepad
- Notepad++
- Sublime
- Eclipse
- Netbeans
- Visual Studio Code
- Visual Studio
- Flutter
- IntelliJ
- PyCharm
- Jupyter

## FRAMEWORK

Docker

Kubernetes

ReactJS

Angular

Vue.js

Laravel

# Exemplos de teste

```
piloto.py x
piloto.py > Piloto
1 class Piloto:
2     aumento = 1.10
3     def __init__(self, nome, sobrenome, salario) :
4         self.nome= nome
5         self.sobrenome = sobrenome
6         self.salario = salario
7
8     @property
9     def email(self):
10        return '{}.{}@email.com.br'.format(self.nome, self.sobrenome)
11
12    @property
13    def nome_completo(self):
14        return '{} {}'.format(self.nome, self.sobrenome)
15
16    def aumentar_salario(self):
17        self.salario = float(self.salario * self.aumento)
18
```

```
test_ex8.py x
test_ex8.py > ...
1
2 import unittest
3 senha = input("Digite uma senha: ")
4 def test_senha_8_digitos(self):
5     senha = "12345678"
6     if len(senha) != 8:
7         raise ValueError("A senha deve ter 8 dígitos")
8
9
10 class TestSenha(unittest.TestCase):
11     def test_senha_8_digitos(self):
12         self.test_senha_8_digitos()
13
14
15 if __name__ == "__main__":
16     unittest.main()
```

# Bibliografia

- <https://www.devmedia.com.br/qualidade-de-software-engenharia-de-software-29/18209>
- <https://www.computerweekly.com/br/tip/23-metricas-de-desenvolvimento-de-software-que-devem-ser-monitoradas>