



EVENTO  
ONLINE  
GRATUITO

# 20º CONGRESSO CIENTÍFICO

19º CONGRESSO DE INICIAÇÃO CIENTÍFICA PIBIC – CNPq | 3º WORKSHOP DE PROJETOS DE EXTENSÃO

FUNDAÇÃO HERMÍNIO OMETTO

*"Desafios para Construção de um Mundo Sustentável"*

## De 2 a 3 | JUNHO



**PARTICIPANTES  
DE TODO O BRASIL**

REALIZAÇÃO:

Coordenadoria de  
Comunidade e Extensão

**FHO**

FUNDAÇÃO  
HERMÍNIO OMETTO



**TEST DRIVEN DEVELOPMENT (TDD) NA PRÁTICA.  
UMA DEMONSTRAÇÃO  
COM USO DO FRAMEWORK DJANGO**

**ORLANDO SARAIVA JR**



**Orlando Saraiva Jr**

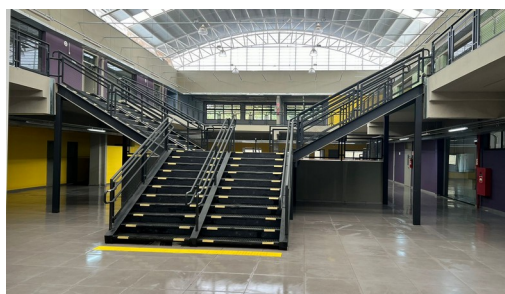
**Assistente de Suporte Acadêmico**  
Unesp Rio Claro



**Docente Fatec Araras**  
Fatec Araras



**Docente Fatec Rio Claro**  
Fatec Rio Claro



## Precisamos conversar sobre qualidade...

A missão de uma empresa de software é desenvolver produtos inovadores de alta qualidade e serviços a um preço competitivo para seus clientes e fazê-lo à frente de seus concorrentes.

Isso requer uma visão clara do negócio, uma cultura de inovação, uma ênfase na qualidade, conhecimento detalhado do domínio do negócio e uma sólida estratégia de desenvolvimento de produtos.

Os clientes têm expectativas muito altas em relação à qualidade e esperam produtos de software de alta qualidade sejam consistentemente entregues no prazo e dentro do orçamento.

## O que é qualidade de software ?

Existem várias definições de qualidade, como a definição proposta por Philip Crosby como **“conformidade com os requisitos”**.

Juran define qualidade como **“adequação ao uso”**, e esta é uma definição melhor, embora não fornece um mecanismo para julgar a melhor qualidade quando dois produtos são igualmente aptos para serem usados.

O **padrão ISO 9126** para tecnologia da informação (ISO/IEC 1991) é uma estrutura para a **avaliação da qualidade do produto de software**.

## Critérios de Qualidade ISO 9126

Característica	Descrição
Funcionalidade	Isso indica até que ponto a funcionalidade necessária está disponível em o software
Confiabilidade	Isso indica até que ponto o software é confiável
Usabilidade	Isso indica até que ponto os usuários do software o julgam fácil de usar
Eficiência	Esta característica indica a eficiência do software
Manutenibilidade	Isso indica até que ponto o produto de software é fácil de modificar e manter
Portabilidade	Isso indica a facilidade de transferir o software para um ambiente diferente

## Cultura Organizacional

Cada organização tem uma cultura distinta, e isso reflete a maneira como as coisas são feitas na empresa.

A cultura organizacional inclui o espírito da organização, seus valores fundamentais, sua história, suas histórias de sucesso, seu pessoal, incidentes divertidos, e assim por diante.

A cultura da organização pode ser favorável ou desfavorável para desenvolvimento de software de alta qualidade.

Ocasionalmente, é necessária uma mudança na cultura da organização, e isso pode ser difícil, pois envolve mudança de suas formas fundamentais de trabalho.

## A jornada ágil

A transição para o desenvolvimento ágil na prática de projetos de TI está em andamento por muito tempo, desde a disseminação da programação orientada a objetos no final dos anos 1980 e início dos anos 1990.

A abordagem orientada a objetos mudou a maneira como o software é desenvolvido.



## A jornada ágil

Os principais objetivos da orientação a objetos eram

- Aumento da produtividade através da reutilização.
- Redução na quantidade de código por meio de herança e associação.
- Facilitação de alterações de código com blocos de código menores e intercambiáveis.
- Limitação do efeito de erros por encapsulamento dos módulos de código

## Jornada Ágil

No entanto, essa melhoria tecnológica também teve um preço:  
**o aumento da complexidade.**

Ao decompor o código em pequenos blocos reutilizáveis, o número de relacionamentos, ou seja, dependências entre blocos de código, aumentaram.

Em procedimentais software, a complexidade residia nos blocos aninhados individuais, cuja lógica de fluxo foi aumentando.

No software orientado a objetos, a complexidade foi terceirizada para o arquitetura.

## Lidando com a complexidade

Havia duas respostas para esse desafio.  
Uma resposta era a **modelagem**.

Durante a década de 1990, diferentes linguagens de modelagem foram propostas: OMT, SOMA, OOD, etc. deles prevaleceu: **UML**.

Ao representar a arquitetura de software em um modelo, deve ser possível encontrar a estrutura ideal e obter e manter a visão geral.

O desenvolvedores iriam - como esperado - criar o modelo "adequado" e então implementá-lo em o código concreto.

## Lidando com a complexidade

No entanto, a modelagem provou ser muito tediosa, mesmo com o melhor suporte de ferramentas. O desenvolvedor precisou de muito tempo para descobrir o modelo em todos os detalhes.

Outra resposta ao desafio de aumentar a complexidade foi “Extreme Programming” (Beck, 1999).

Como o modelo apropriado para o software não foi previsível, os desenvolvedores começaram a traduzir os requisitos diretamente no código em estreita comunicação com o usuário e em iterações curtas (Beck, 2000).

O “**Desenvolvimento Orientado a Testes**” acabou sendo uma consequência útil da Programação Extrema (Beck, 2003).



## Manifesto Ágil

No manifesto, os autores enfatizam os seguintes quatro “revolucionários” valores:

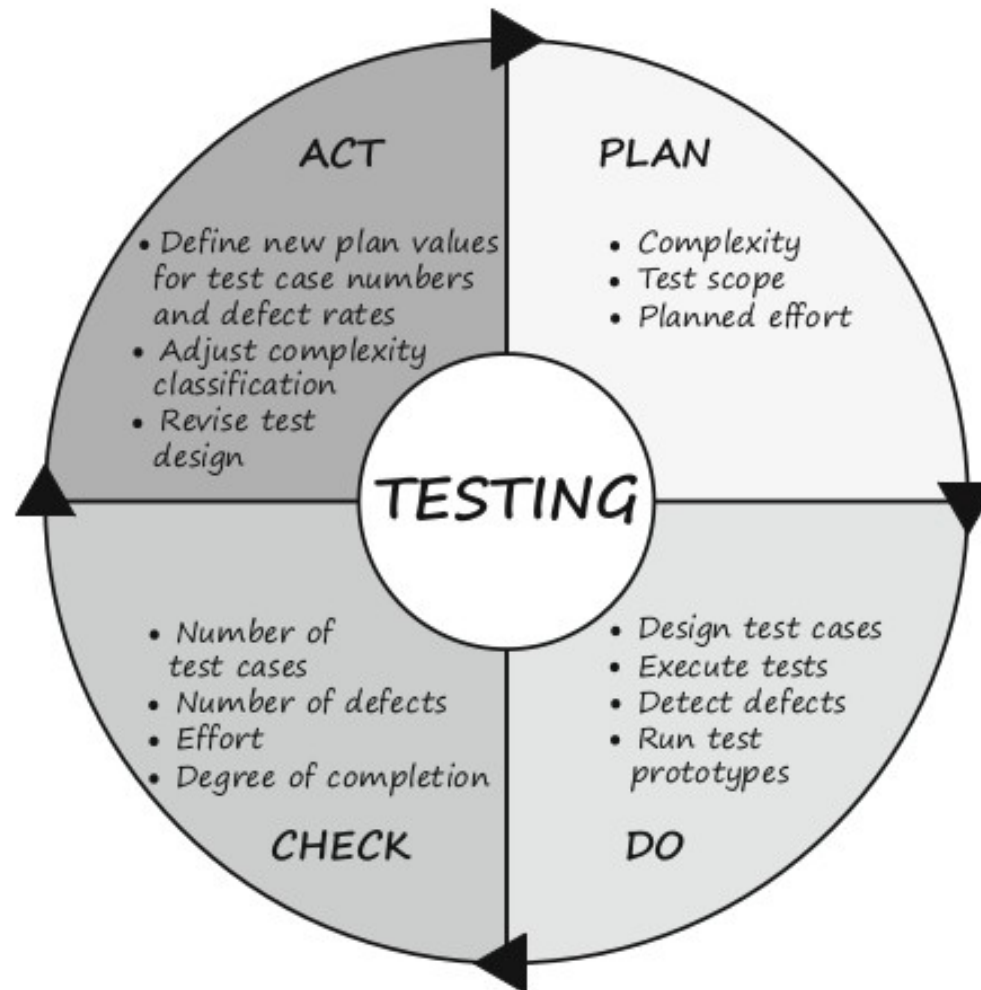
**Indivíduos e interações** sobre processos e ferramentas.

**Software que trabalha** sobre uma documentação completa.

**Colaboração do cliente** em vez de negociação de contratos.

**Responder à mudança** ao invés de seguir um plano.

# Ciclo PDCA





# Django

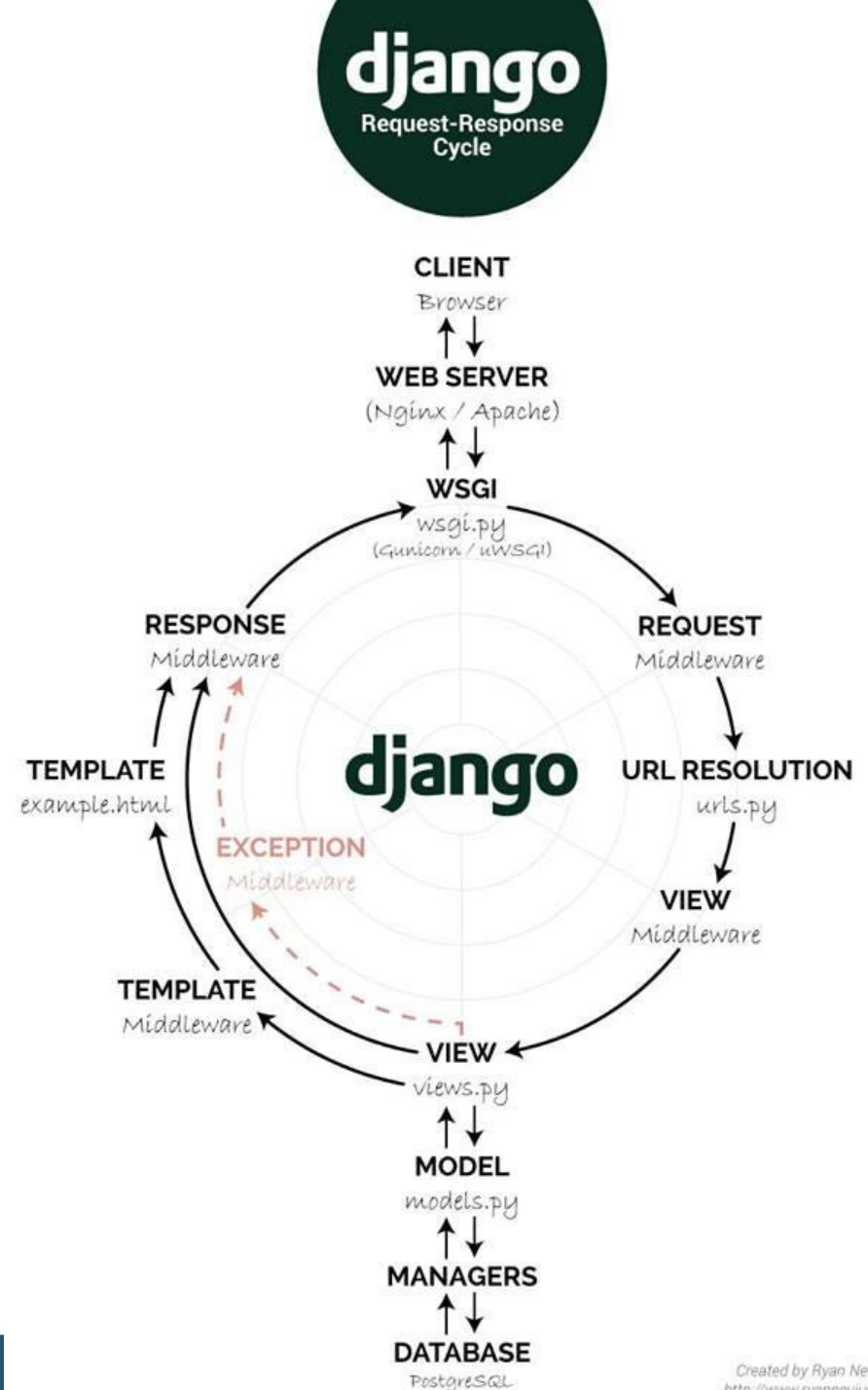
Django é um **framework web** de alto nível escrito em Python, criado para facilitar o desenvolvimento rápido e seguro de aplicações web. Ele segue o padrão de arquitetura MTV (Model-Template-View), que separa a lógica de dados, a apresentação e a interação do usuário.

O Django oferece uma série de recursos prontos — como um sistema de rotas, um ORM (mapeamento objeto-relacional), autenticação de usuários e um painel administrativo automático — que ajudam os desenvolvedores a focarem na lógica do negócio sem se preocupar com detalhes repetitivos da infraestrutura.

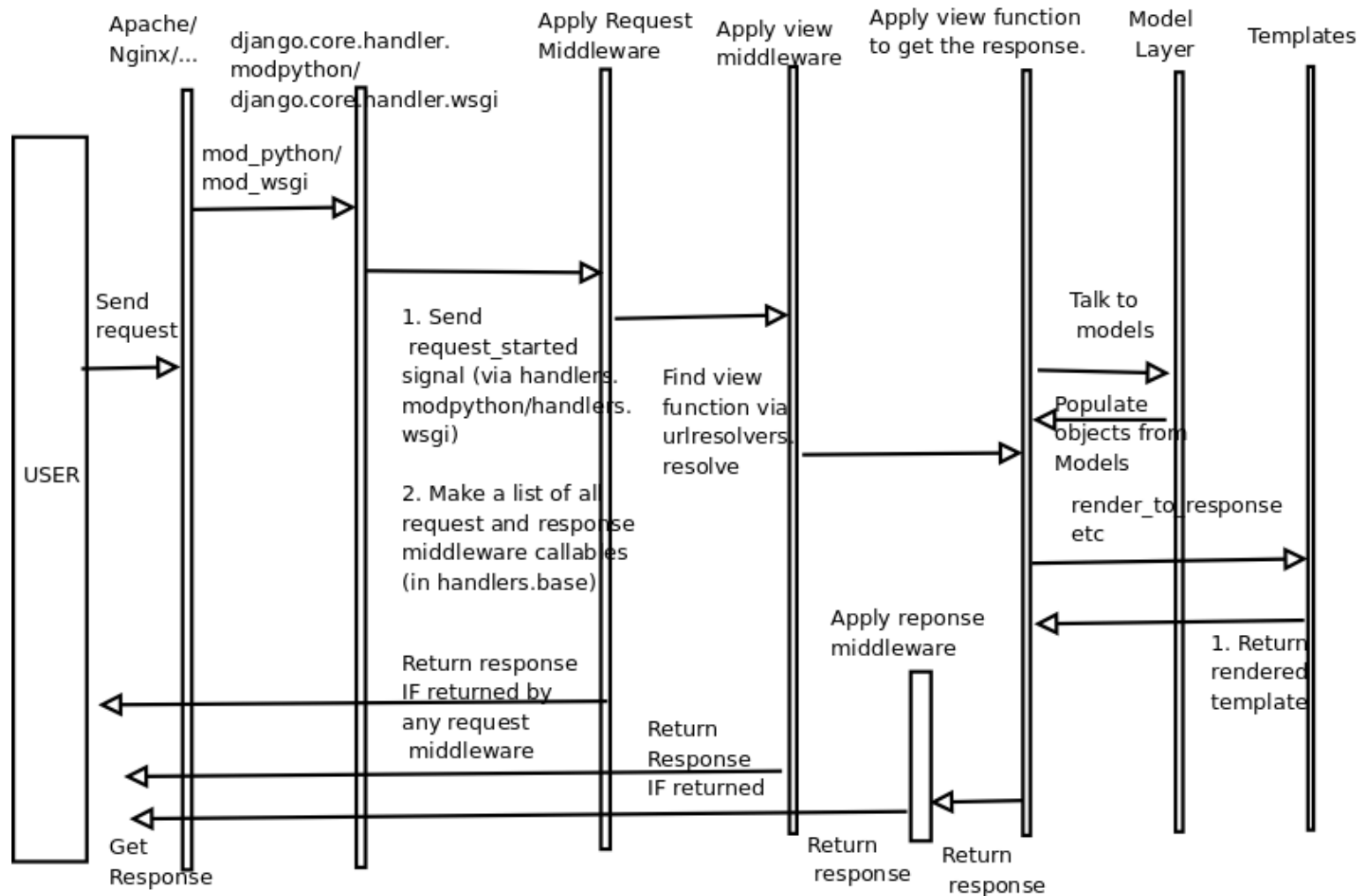
Além de promover a reutilização de código e a escalabilidade, Django é conhecido por suas boas práticas de segurança embutidas, como proteção contra ataques comuns (SQL injection, CSRF, XSS). É uma ferramenta poderosa para construir desde sites simples até sistemas complexos e robustos, com o suporte de uma comunidade ativa e uma vasta documentação.



# Django



# Django



**Orlando Saraiva Jr**

**[orlandosaraivajr.github.io](https://orlandosaraivajr.github.io)**



FUNDAÇÃO  
HERMÍNIO OMETTO

# FHO Qualifica



FUNDAÇÃO  
HERMÍNIO OMETTO

Cursos de Curta e Média Duração

*Desenvolver habilidades  
é investir no futuro*

Cursos oferecidos nos formatos:

- ✓ Híbrido
- ✓ Presencial
- ✓ Remoto

Aponte o seu  
celular para ler o  
QRCode



✉ [contatopos@fho.edu.br](mailto:contatopos@fho.edu.br) ☎ (19) 99604-7774

ou acesse o site:  
[qualifica.fho.edu.br](http://qualifica.fho.edu.br)