



# Arquitetura Ágil



**Certified  
Developer**  
The Ultimate Tech Degree

**DigitalHouse** >  
Coding School



Oi!

Como estão?





**O que vimos na aula  
passada?**





## O que vamos ver hoje?

1. **Qualidade e Modelagem de Software**
2. **Equipe de desenvolvimento**
3. **XP - Extreme Programming**



**1**

# Qualidade e modelagem de software



## Preciso de qualidade no meu código?

Quais os riscos de não  
ter qualidade?





**Quem é responsável  
pela qualidade?**

**O analista de testes?**



## Começando! Deixe seu código legível

Clean Code

Hein?



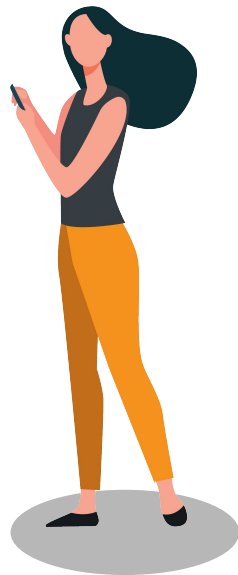




# Clean Code

## Fundamentos

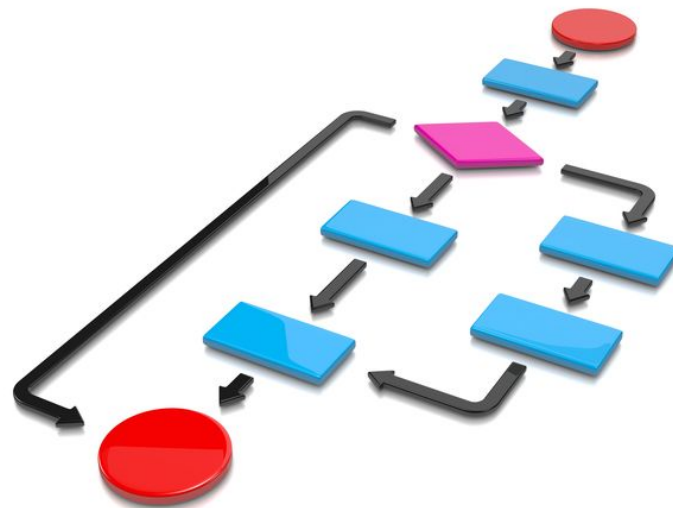
- **Simples:** O código não precisa fazer nada mais daquilo que ele precisa fazer, e deve ser direto ao ponto e resolver.
- **Eficiente:** Eficiente na performance, um código performático, ter a capacidade de ver se o código está sendo direto e performático.
- **Sem duplicidade:** Não existe motivo para duplicar um código. Além de ser crime, impacta na manutenção do código.





## TÉCNICAS DE MODELAGEM DE SOFTWARE

São metodologias padrão para a análise, implementação e documentação de sistemas orientados a objetos.

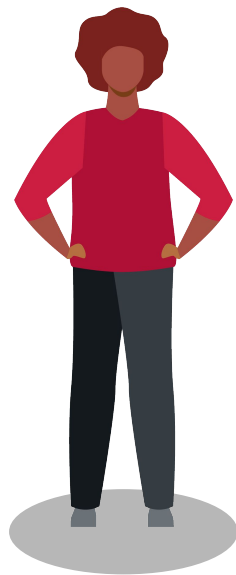




# Clean Code

## Fundamentos

- **Elegante:** Indentações, bem escrito, bonito de leitura nas nomenclaturas.
- **Feito com cuidado:** Pessoa teve preocupação em entregar o código. Não só na funcionalidade, mas também na escrita de código.
- **Fácil de ler:** Ir direto ao ponto de maneira simples e elegante, deixando-o fácil de ler.





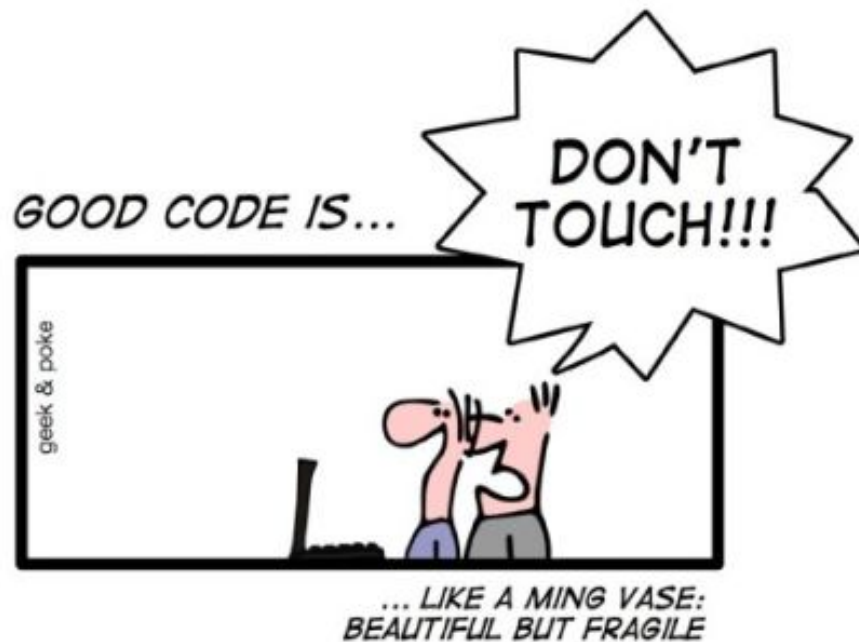
## Temos outras maneiras



- DoR, DoD e Critérios de aceite (Próxima aula)
- Solid - Princípios da orientação a objetos e design de código
- Sonar e Codacy
- Monitoramentos (Dynatrace, AppDynamics...)



## Vamos Debater!



## **2** | **Equipe de desenvolvimento**



**Certified Tech  
Developer**  
The Ultimate Degree

# Quais funções vocês conhecem?

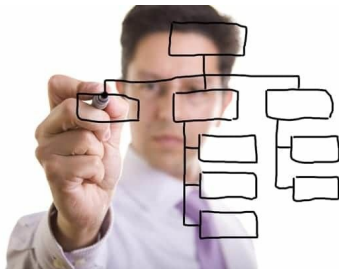






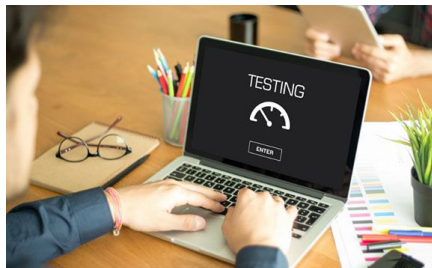
# Equipe de desenvolvimento

**Analista  
Arquiteto**



**Desenvolvedor**

**Tester / QA**



**Infra estrutura**





## Equipe de desenvolvimento

- DevOps
  - Automation Engineer
  - Full-Stack Developer
- 
- UI (User interface)
  - UX (User experience)
  - PM (Product Manager)



## Especialidades de um desenvolvedor

### Backend

É aquele que trabalha na parte que se encarrega do servidor, gerenciamento de dados e tudo que represente uma base de suporte à aplicação, seja em banco de dados ou infraestrutura. Eles geralmente conhecem mecanismos de banco de dados (MySQL, PostgreSQL, SQL Server, etc). Eles têm experiência com linguagens como Ruby, PHP, Python, Java

### Frontend

Funciona na parte que mostra os dados de um aplicativo para os usuários. Sua principal tarefa é coletar e enviar dados de e para o back-end do aplicativo. Os programadores de front-end entendem APIs, JSON, XML, eles são muito bons em JavaScript, HTML e CSS.

### Full Stack

Eles têm experiência de back-end e front-end. Eles geralmente não são necessários em um estágio embrionário de desenvolvimento.



## Tipos de desenvolvedor

### Corporativo

#### DEV Desktop

Fazem aplicações corporativas usando linguagens como Java, Visual Basic.Net, C#, C++, Python.

### Web

#### DEV Web

Utilizam linguagens de servidor como PHP, ASP.Net, JSP, Ruby, Python, NodeJS, entre outros. Também sabem de HTML, CSS, JavaScript

### Mobile

#### DEV Mobile

Fazem Aplicações para Android, iOS o Windows Phone usando linguagens nativas (Java, C++, Objective-C)



# Vamos montar nosso time?

Vamos ao Miro



**3**

## **XP - Extreme Programming**



## O que é XP?





## O que é XP

### Extreme Programming

É uma metodologia ágil para desenvolvimento de software;

Com foco na agilidade das equipes e na qualidade dos projetos;

Indicado para desenvolvimento de software com requisitos vagos e em constante mudança;

A principal tarefa é a codificação.





# XP - Extreme Programming

## Princípios básicos

- Já que **testar é bom**, que todos testem o tempo todo;
- Já que **revisão é bom**, que se revise o tempo todo;
- Se **projetar é bom**, então refatorar o tempo todo;
- Se **teste de integração é bom**, então que se integre o tempo todo;
- Se **simplicidade é bom**, desenvolva uma solução não apenas que funcione, mas que seja a mais simples possível;
- Se **iterações curtas é bom**, então mantenha-as realmente curtas.





# XP - Extreme Programming

## Programação em pares

- Enquanto uma pessoa escreve o código, a outra pessoa observa, comenta e fornece feedback. Esse conceito ajuda a disseminar e compartilhar o conhecimento. Além do sistema sempre ser revisto por duas pessoas, evitando e diminuindo assim a possibilidade de defeitos.





# XP - Extreme Programming

**Desafio**

The Kahoot! logo is centered on a dark blue rectangular background. The word "Kahoot!" is written in a large, white, bold, sans-serif font. A lighter blue, stylized arrow points from the right towards the text.

**Kahoot!**

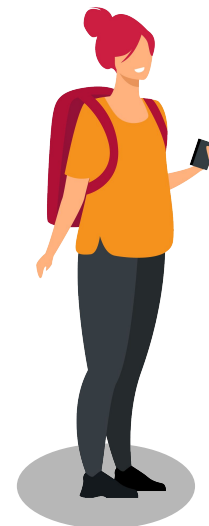


Como estamos até aqui?



## Leitura para próxima aula:

**Material Online - Aula 9 - Agile  
Testing**



DigitalHouse>  
Coding School