Implementando sua stack de testes unitários e integrados em um projeto .NET de Crowdfunding

Eliézer Zarpelão e Marcos Freire

Apresentadores

Eliézer Zarpelão

Arquiteto de Software

Marcos Freire

.NET Ninja

Você acha realmente importante testar seu código???


Você viajaria num avião que não foi testado?

Testar faz parte do desenvolvimento de software!

(ou pelo menos deveria)

+ bugs

- qualidade

+ atrasos

- confiança

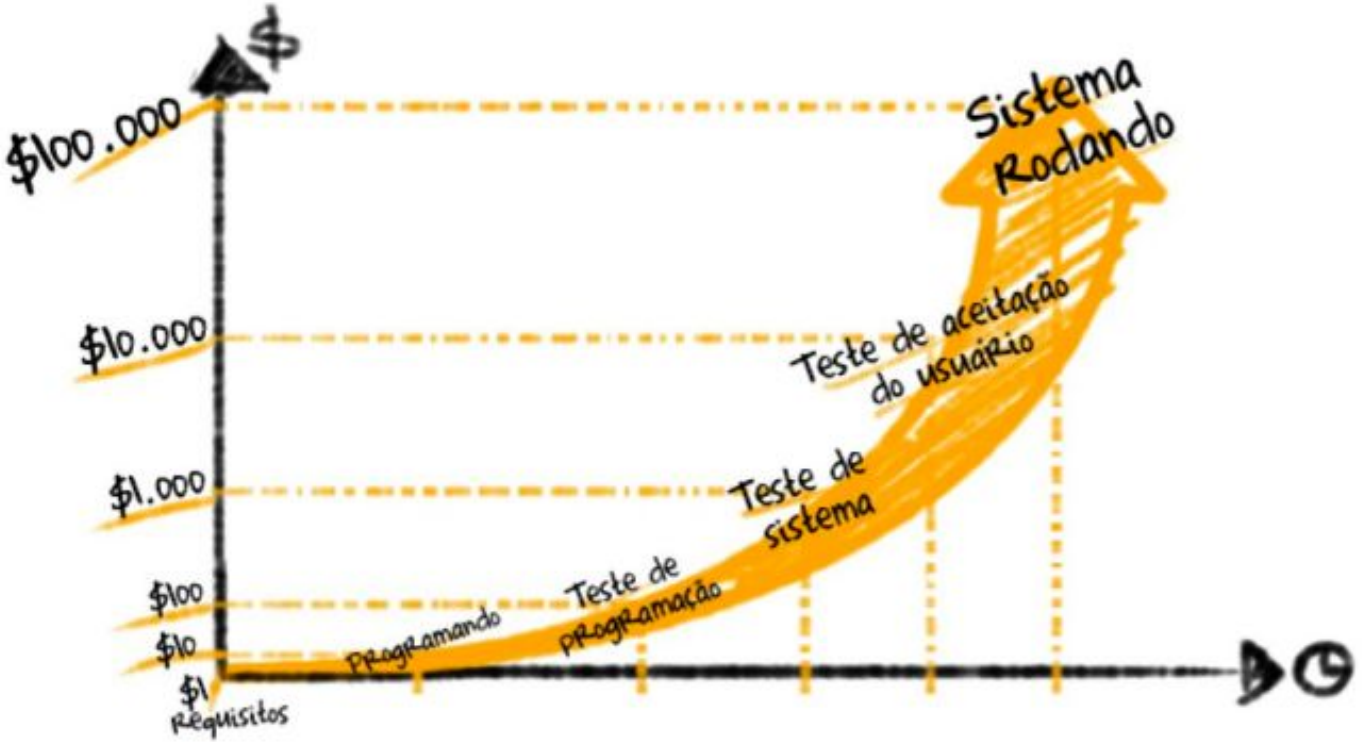
+ desmotivação

+ turnover

+ prejuízos

+ implicações legais

Regra 10 de myers



Fonte: https://www.pjf.mg.gov.br/secretarias/sarh/edital/interno/selecao2015/arquivos/apostilas/modulo4_p.pdf

Tipos mais comuns de testes

Testes de Unidade

Testes de Integração

Testes Automatizados

Existem muitos outros...

Testes de Unidade / UnitTest

~~Testes Unitários~~

Unidade: menor parte testável de um software

Orientação a Objetos: classe

Testes de Integração

Encontrar falhas de integração entre as unidades, e não mais em testar as funcionalidades da mesma

Testes Automatizados

“Simula” ações do usuário

Aceitação: caixa preta

Regressão: garante integridade de versões passadas

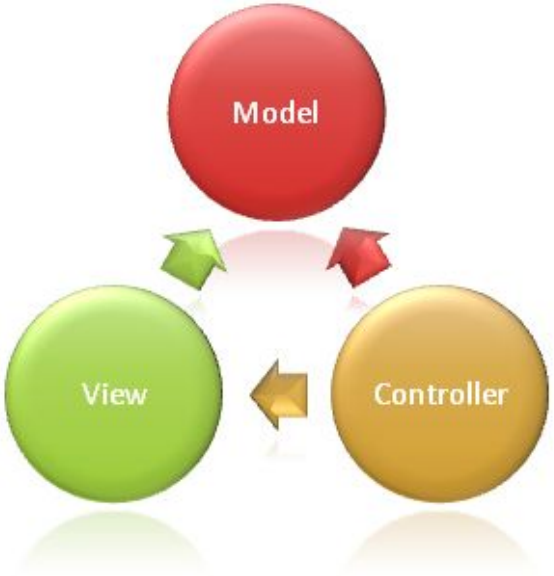
Projeto Vaquinha

Projeto demonstração de testes

.Net Core 3.1

ASP.NET Core MVC

MVC: Model View Controller



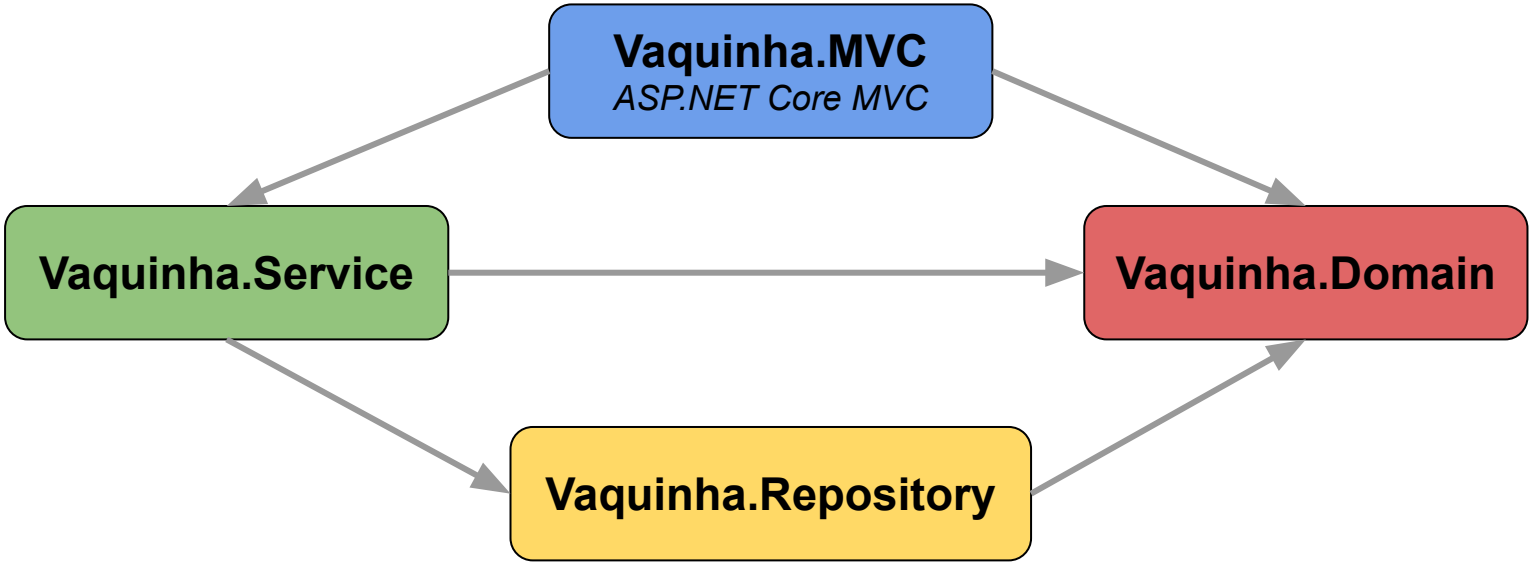
Model: Armazena, manipula e gera os dados.

View: Apresenta o conteúdo por meio da interface do usuário.

Controller: Cuida da interação do usuário, trabalham com o model e, em última análise, selecionam uma view a ser renderizada.

Fonte: https://docs.microsoft.com/pt-br/aspnet/core/mvc/overview

Estrutura do Projeto



Vaquinha - Estrutura Testes

Automated.UI.Tests

Integration.Tests

Unit.Tests

KEEP
CALM
AND
TEST
ON