

# Refatoração

Um pouco além do código

# Vamos falar sobre

- O que é refatoração?
- Por que refatoramos?
- Como escolhemos as batalhas que queremos lutar?
- O que consideramos ao refatorar?

# Por que?

- Ouvi/usei o termo “Refatoração” para algo diferente do proposto pelos principais autores;

# Por que?

- Ouvi/usei o termo “Refatoração” para algo diferente do proposto pelos principais autores;
- Dificuldades em convencer o PO/PM/Gestor comprar a refatoração;

# Por que?

- Ouvi/usei o termo “Refatoração” para algo diferente do proposto pelos principais autores;
- Dificuldades em convencer o PO/PM/Gestor comprar a refatoração;
- Subestimei refatoração, gastando 4x o esperado;

# Por que?

- Ouvi/usei o termo "Refatoração" para algo diferente do proposto pelos principais autores;
- Dificuldades em convencer o PO/PM/Gestor comprar a refatoração;
- Subestimei refatoração, gastando 4x o esperado;
- Porque eu gostaria de saber destas coisas há alguns anos;

Várias coisas diferentes  
com o mesmo nome

Precisamos **refatorar** esta tela, para deixar o campo número ao lado do campo endereço.

A Web Page

← → ✕ 🖱️  🔍

## Participante

Nome

Endereço  CEP

Número  Cidade  UF

A Web Page

← → ✕ 🖱️  🔍

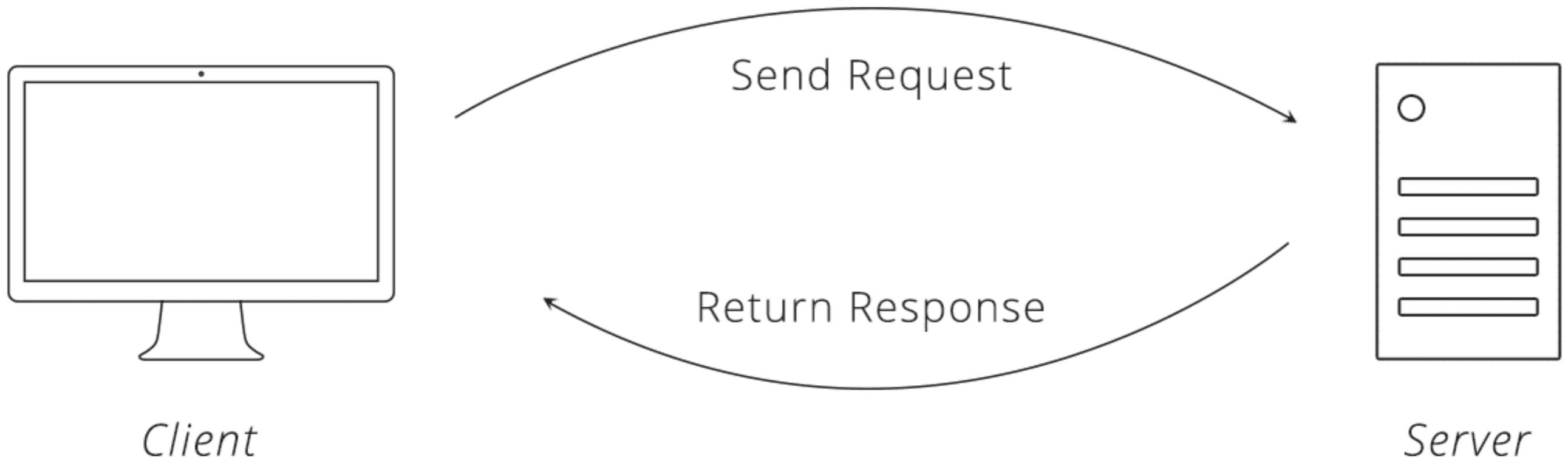
## Participante

Nome

Endereço  Número

CEP  Cidade  UF

Precisamos **refatorar** esta API para responder de 3000ms para 100ms



Vamos **refatorar** função X para corrigir aquele bug



# O que é refatoração?

Martin Fowler em seus livros sobre refatoração trabalha com 2 definições

- Refatoração (Refactoring) - substantivo
- Refatorar (Refactoring) - verbo

# Refatoração (refactoring) (substantivo)

- Uma alteração feita na estrutura interna do software para torná-lo mais fácil de ser entendido e menos custoso de ser modificado, sem alterar o seu comportamento observável.
- A change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior.



# Refatoração (refactoring) (substantivo)

- Uma alteração feita na estrutura interna do software para torná-lo mais fácil de ser entendido e menos custoso de ser modificado, **sem alterar o seu comportamento observável.**
- A change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behaviour.



# Refatoração (refactoring) (substantivo)

- Uma alteração feita na estrutura interna do software **para torná-lo mais fácil de ser entendido e menos custoso de ser modificado**, sem alterar o seu comportamento observável.
- A change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior.



# Refatoração

é para pessoas!!!



Precisamos **refatorar** esta tela, para deixar o campo número ao lado do campo endereço.

A Web Page

http://membro.guru-

Participan

Nome

Endereço CEP

Número Cidade

A Web Page

http://membro.guru-

Participante

Nome

Endereço Número

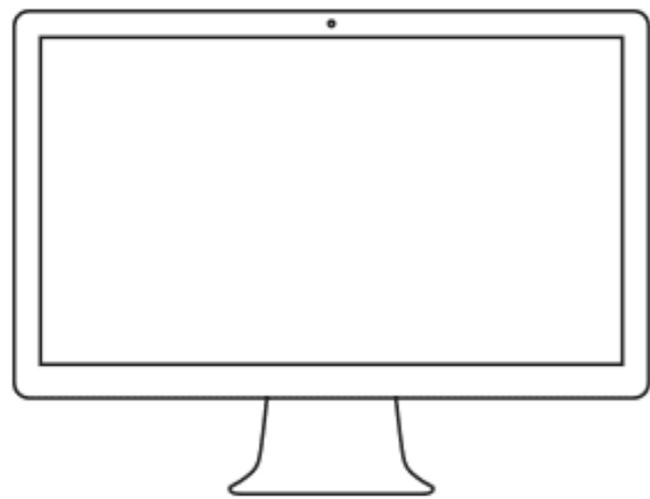
CEP Cidade UF

# Refatoração (refactoring) (substantivo)

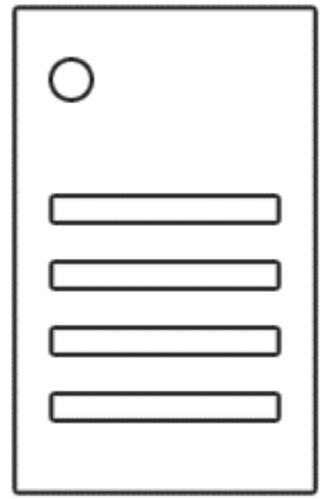
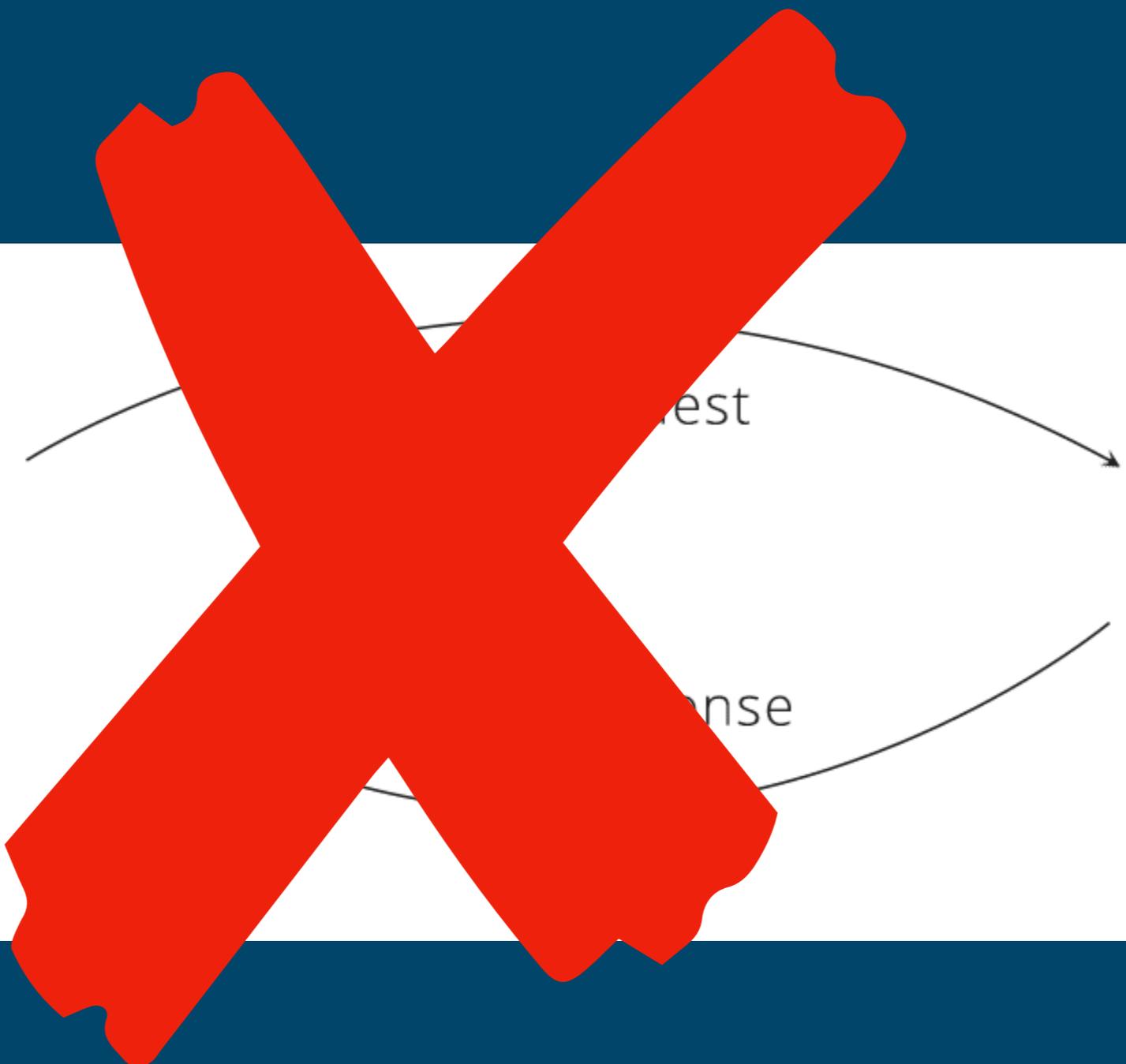
- Uma alteração feita na estrutura interna do software para torná-lo mais fácil de ser entendido e menos custoso de ser modificado, sem alterar o seu comportamento observável.



Precisamos **refatorar** esta API para responder de 3000ms para 100ms



*Client*



*Server*

# Refatoração X Otimização de Performance

- Similares mas com propósitos diferentes.
- Refatoração: foco é legibilidade do código.
- Performance: foco é tornar o programa mais rápido.

Vamos **refatorar** função X para corrigir aquele bug



Correção de bug altera o comportamento observável.

# O que é refatoração?

Quando dividimos uma classe de mais de 1000 linhas em novas classes



# O que é refatoração?

Quando dividimos uma classe de mais de 1000 linhas em novas classes



Quando um método muito grande é quebrado em outros métodos ou classes



# O que é refatoração?

Quando dividimos uma classe de mais de 1000 linhas em novas classes

Quando um método muito grande é quebrado em outros métodos ou classes

Quando removemos código morto



# O que é refatoração?

Quando dividimos uma classe de mais de 1000 linhas em novas classes

Quando um método muito grande é quebrado em outros métodos ou classes

Quando removemos código morto

Quando substituímos um "número mágico"



# O que é refatoração?

Quando dividimos uma classe de mais de 1000 linhas em novas classes

Quando um método muito grande é quebrado em outros métodos ou classes

Quando removemos código morto

Quando substituímos um "número mágico"

E muito mais em <https://refactoring.com/catalog/>



# Refatorar (verbo)

- Reestruturar o software aplicando uma série de refatorações sem alterar seu comportamento observável.
- To restructure software by applying a series of refactorings without changing its observable behavior.



## Sessão de Refatoração

Extrair  
para uma  
classe

## Sessão de Refatoração

**Extrair  
para uma  
classe**

**Mover  
métodos**

**Remover  
código  
morto**

**Renomear  
variáveis**

Por que refatoramos?

# Por que refatoramos?

Qualquer tolo pode escrever código que o computador entenda. Bons programadores(as) escrevem código que humanos podem entender. Martin Fowler, 1999

Any fool can write code that computer can understand. Good programmers write code that humans can understand.

**Melhora o design do  
software**

# Melhora o design do software

- Sistema cresce, o custo e o tempo gasto acompanham.
- Um codebase mais enxuto é mais fácil de ser mantido.

**Projete seu código para  
ser mais fácil de entender,  
não mais fácil de escrever**

# Projete seu código para ser mais fácil de entender, não mais fácil de escrever

- 1º ciclo: computador entenda as instruções.
- 2º ciclo: próxima pessoa a ler este código.

**Respondemos mais  
rápido à mudanças**

# Respondemos mais rápido à mudanças

- Mais código para ler e entender = maior dificuldade em acertar da primeira vez.
- Código legível + fácil localização = menor carga cognitiva

Ajuda na redução do  
turnover

# Ajuda na redução do turnover

- Código complexo aumenta o ciclo de entrega

# Ajuda na redução do turnover

- Código complexo aumenta o ciclo de entrega
- Que aumenta a insatisfação dos participantes do projeto

# Ajuda na redução do turnover

- Código complexo aumenta o ciclo de entrega
- Que aumenta a insatisfação dos participantes do projeto
- Que aumenta a pressão no time

# Ajuda na redução do turnover

- Código complexo aumenta o ciclo de entrega
- Que aumenta a insatisfação dos participantes do projeto
- Que aumenta a pressão no time
- Devs / Devas começam a trabalhar em condições de forte tensão

# Ajuda na redução do turnover

- Código complexo aumenta o ciclo de entrega
- Que aumenta a insatisfação dos participantes do projeto
- Que aumenta a pressão no time
- Devs / Devas começam a trabalhar em condições de forte tensão
- Código mal estruturado + pressão intensa = perder pessoas

**Como escolhemos as  
batalhas que queremos  
lutar?**

# Diante de

- Codebase relativamente grande
- Projeto de + 8 anos
- +/- 100 engenheiros
- Produto em constante crescimento

# E mais

- **5 bilhões** de emails enviados
- **50 bilhões** de eventos (abertura, clique, etc) exibidos em tempo real
- **260 milhões** de conversões
- **280 mil** Landing Pages
- **150 mil** Fluxos de Automação rodando simultaneamente

- Um dos desafios da refatoração é sucessão - como dividir o trabalho de refatorar em etapas seguras e como ordenar estas etapas. Kent Beck
- One of the challenges of refactoring is succession - how to slice the work of a refactoring into safe steps and how to order those steps.

# Dando visibilidade aos problemas



# Dívida Técnica

- Metáfora usando a área financeira
- Citado por Ward Cunningham (@WardCunningham)
- Com prudência, dá pra adquirir e sobreviver



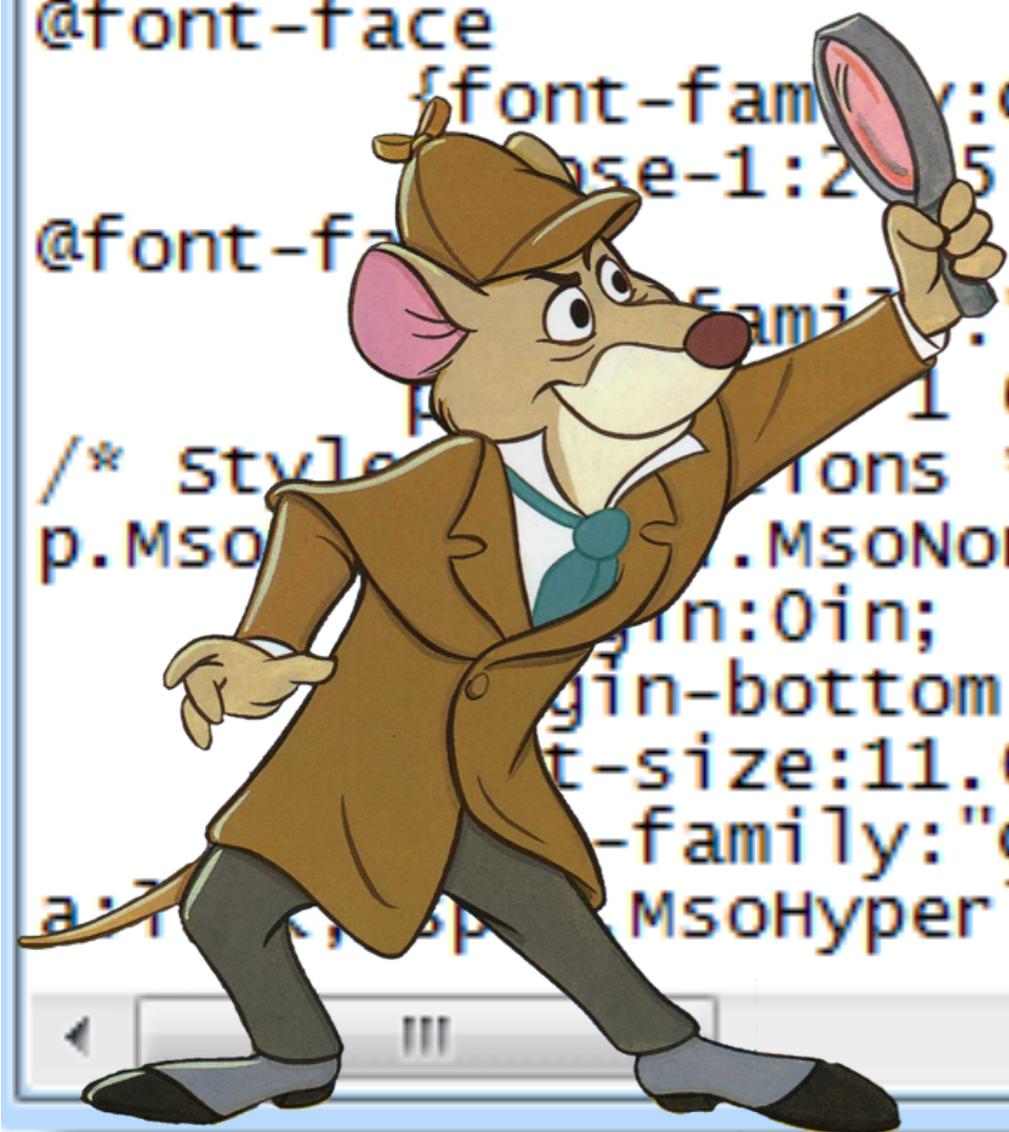


File Edit Format View Help

```

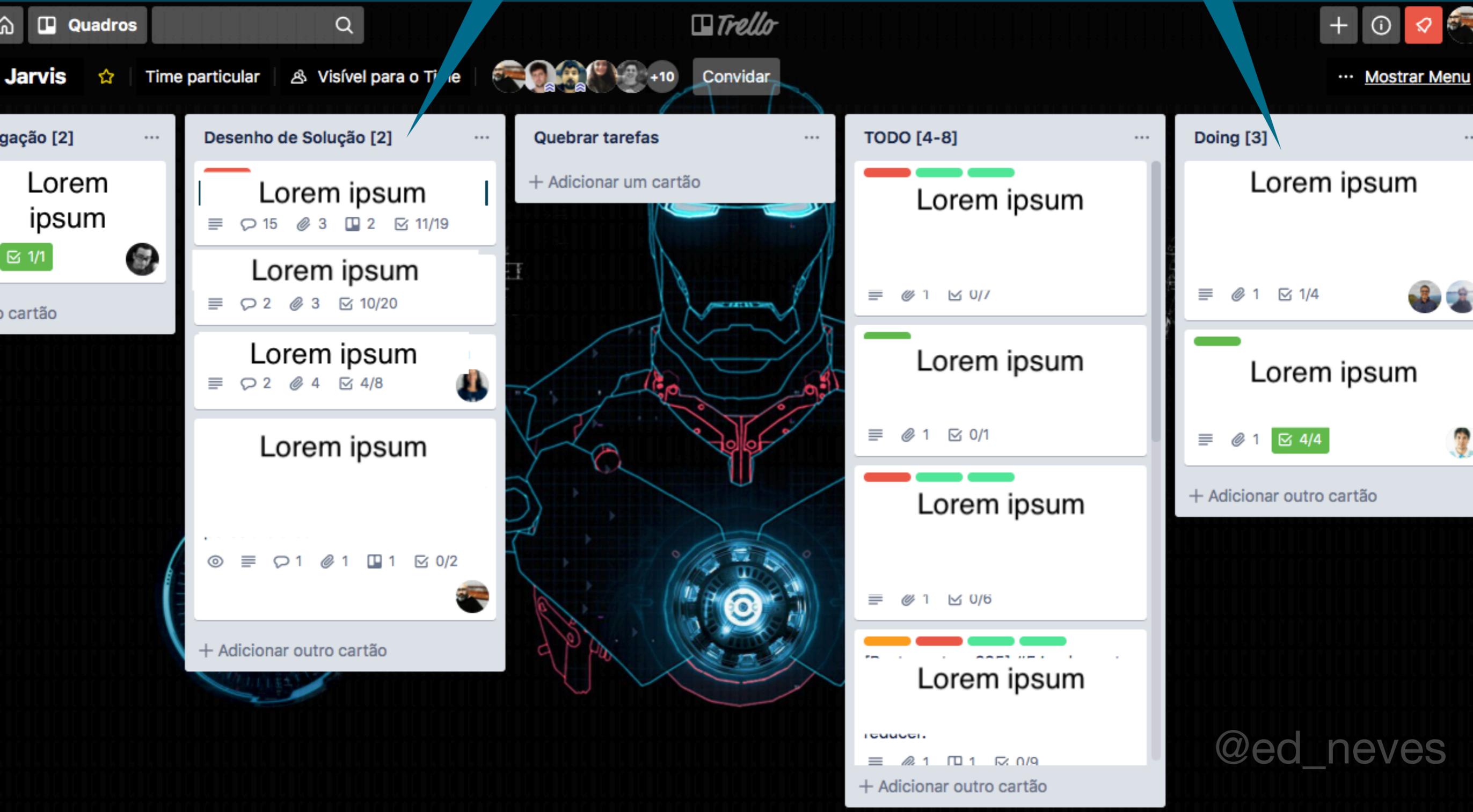
<html xmlns:v="urn:schemas-microsoft-com:vm1" xmlns:o="urn:
/* Font Definitions */
@font-face
    {font-family:\5B8B\4F53;
    panose-1:2 1 6 0 3 1 1 1 1 1;}
@font-face
    {font-family:"Cambria Math";
    panose-1:2 4 5 3 5 4 6 3 2 4;}
@font-face
    {font-family:Calibri;
    panose-1:2 5 5 2 2 2 4 3 2 4;}
@font-face
    {font-family:"@5B8B\4F53";
    panose-1:2 1 6 0 3 1 1 1 1 1;}
/* style definitions */
p.MsoNormal, div.MsoNormal
    {margin:0in;
    margin-bottom:.0001pt;
    font-size:11.0pt;
    font-family:"Calibri","sans-serif";}
a:link, span.MsoHyperlink

```

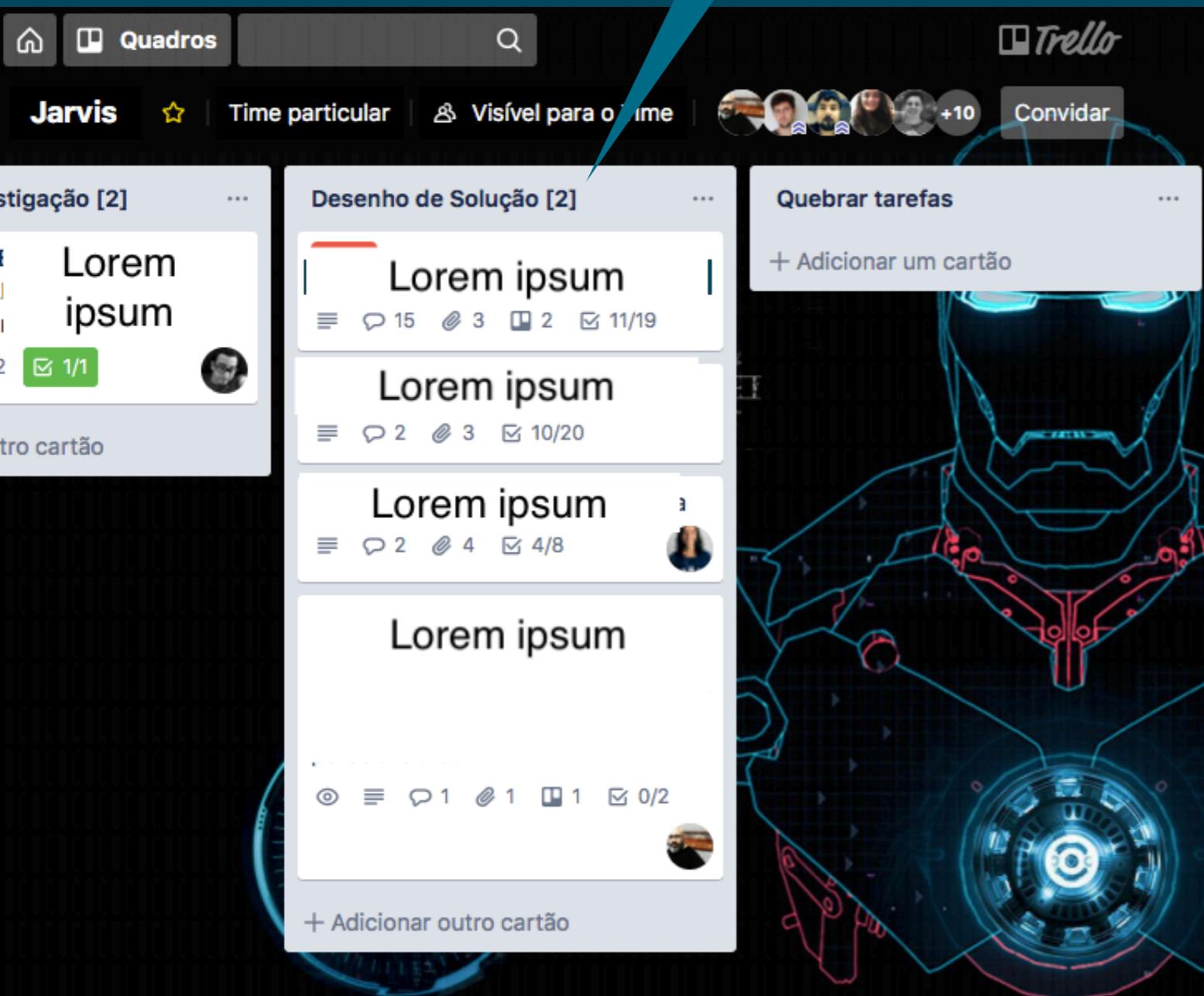


Preferencialmente

Mas com liberdade aqui também



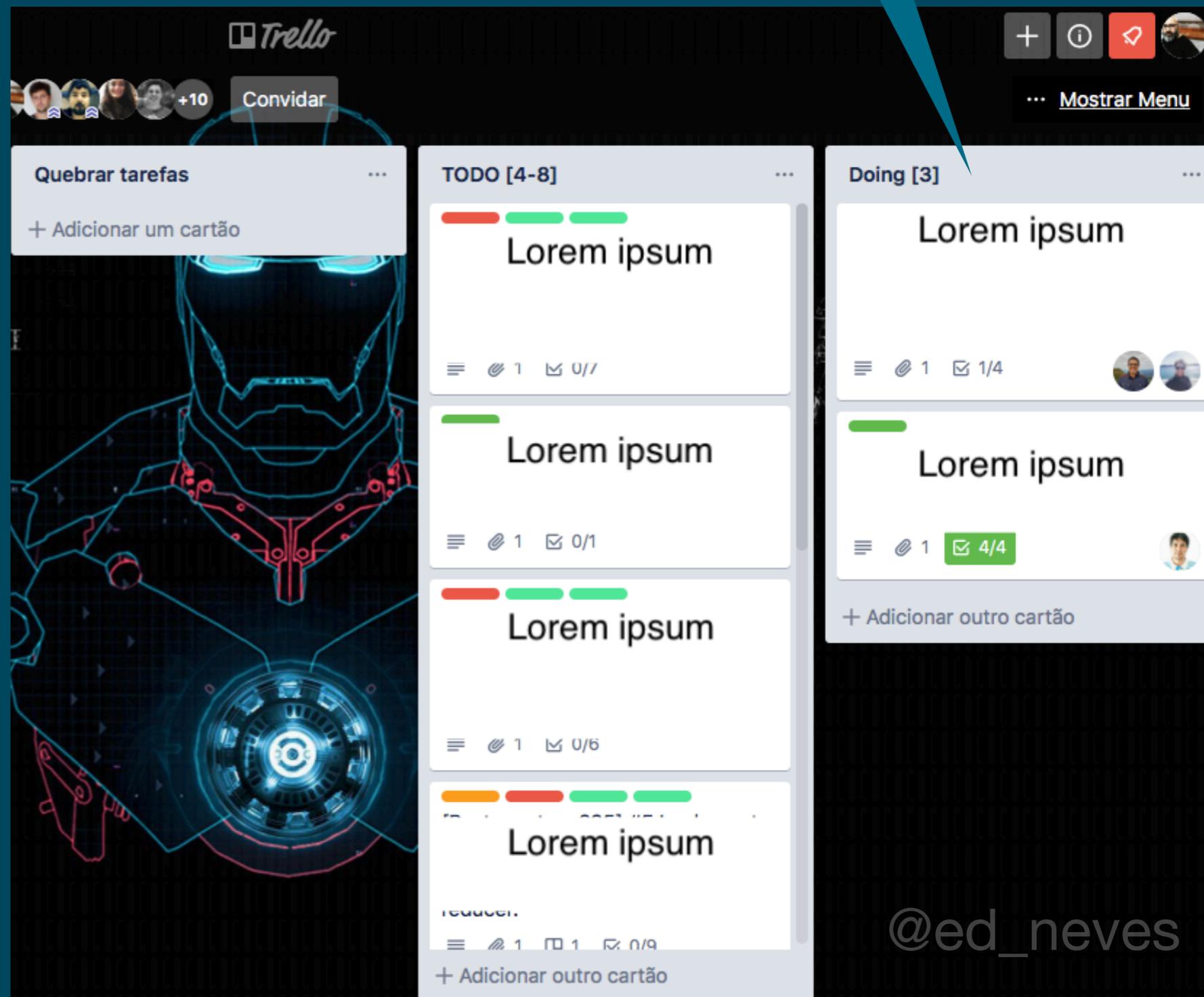
Preferencialmente



- Visão de todo o time
- Momento de análise e descobertas
- Maior liberdade para prova de conceito

Mas com liberdade aqui também

- Momento mais próximo do código
- Maior clareza da implementação



# Escolhendo as batalhas que queremos lutar

- Realmente temos um problema?
- Está atrapalhando o nosso dia a dia?
- É o melhor momento para atacarmos?

# Escolhendo as batalhas que queremos lutar

- Temos orçamento para isso?
- Por que esta refatoração agora e não outra?
- Temos conhecimento do contexto?

# Como refatoramos?

- Código novo: Assim que seu primeiro ciclo de escrita de código é concluído (Red, Green)

# Como refatoramos?

- Código já existente: seu comportamento já está implementado, mas precisa ser melhorado.

# Refatorar código novo?

2 chapéus

- Adicionando funcionalidade
  - Escrever os testes
  - Fazer os testes passarem
  - Não alterar código existente

# Refatorar código novo?

2 chapéus

- Refatorando
  - Reestruturar o código
  - Manter os testes passando
  - Não adicionar novas funcionalidades

# Cuidados ao refatorar

- Ferramentas de apoio a identificação de *mau cheiro* no código (*Bad Smells*).
- Mas o mapa nunca pode ser mais importante que o terreno

# Cuidados ao refatorar

- Desenvolvimento guiado por testes (TDD)
  - Perguntamos para o sistema escrevendo um teste.
  - Respondemos a pergunta escrevendo código que passa no teste.
  - Melhoramos a pergunta consolidando ideias e removendo coisas desnecessárias.
  - Repetição: Mantemos o diálogo com novas perguntas.

# Cuidados ao refatorar

- Trecho a ser refatorado com cobertura de testes

# Cuidados ao refatorar

- Baby Steps
  - Você está no zero, onde tem que chegar primeiro? No um! Só depois disso é que você se preocupa com o dois. (Henrique Bastos)

# Cuidados ao refatorar

- Intervenções cirúrgicas
  - Teste após cada mudança

# Cuidados ao refatorar

- Mantenha-se a pouquíssimos passos de distância do código que passava no teste.

# Esteja alinhado ao negócio

"Negócios são bem servidos via refatoração contínua, ainda assim, as práticas de refatoração devem coexistir harmoniosamente com as prioridades do negócio."

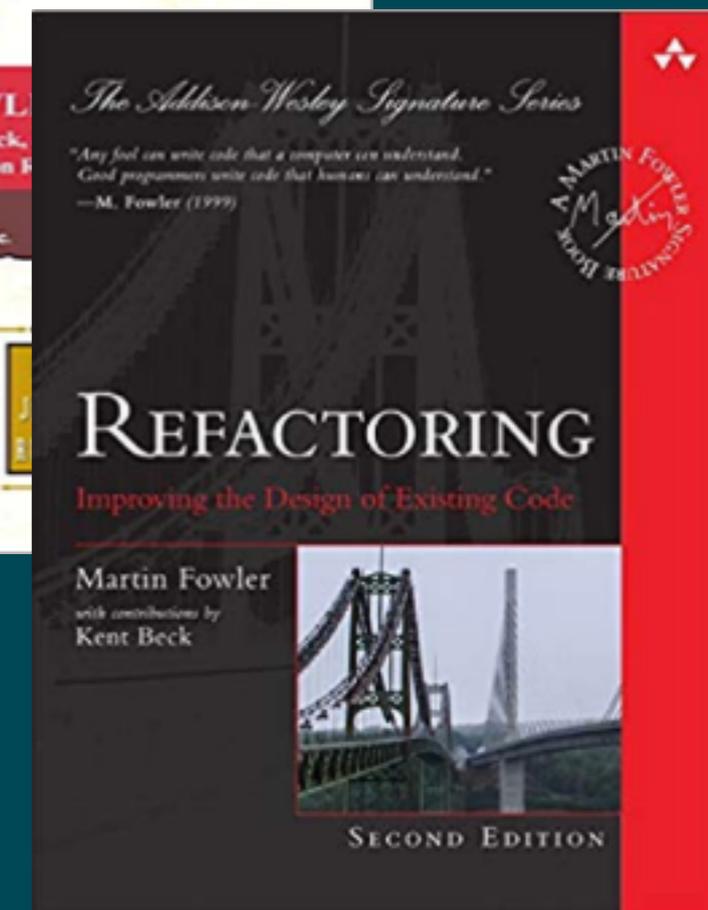
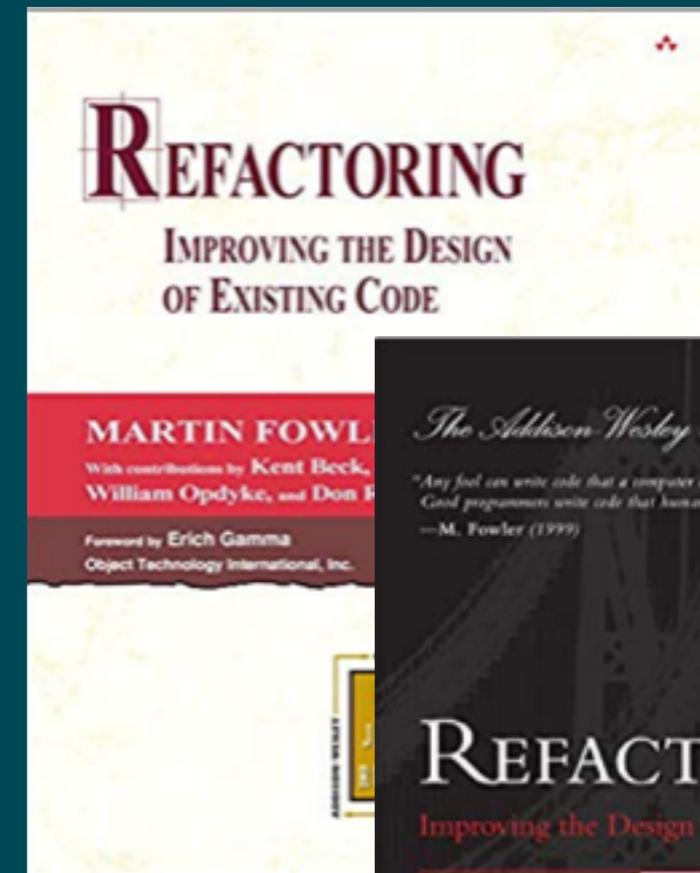
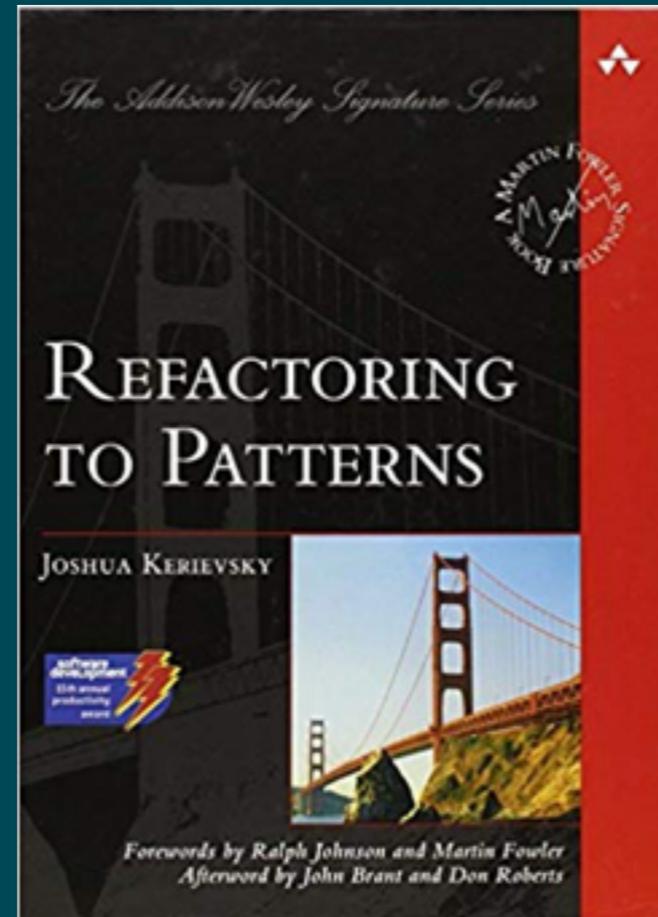
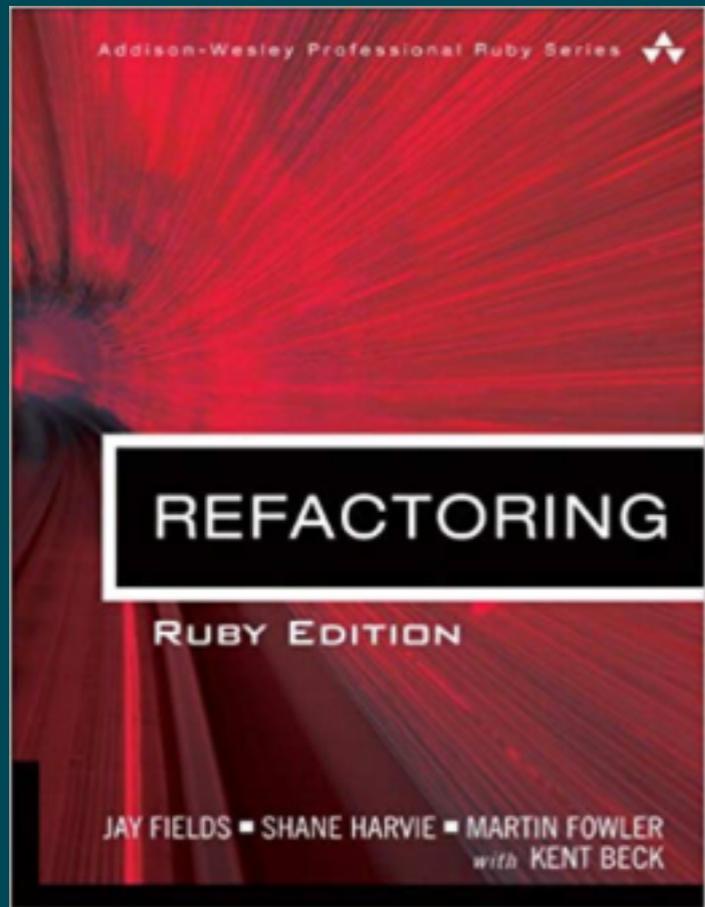
[@JoshuaKerievsky](#) (Refactoring to Patterns)

# Referências

**Catálogo de refatorações:** <https://refactoring.com/catalog/>

**Tipos de débitos técnicos:** <https://agilemichaeldougherty.wordpress.com/2015/07/24/types-of-technical-debt/>

# Referências



# Resultados Digitais

6&7&8 NOV 2019  
FLORIPA - BR

RD  
SUMMIT



8 anos 700 RDoers 13000 Clientes 20 Países  
1 ano de escritório em São Paulo

# Somos remote-friendly

+ 30% do time de engenharia é remoto

[bit.ly/matrix-rd](https://bit.ly/matrix-rd)

Criamos uma ferramenta open source para o trabalho remoto

☰ Matrix

🔍 Search...



EDMILTON NEVES

Lounge



ENTER ROOM ENTER MEETING

WAR ROOM CDP

ENTER ROOM ENTER MEETING

Data Services



ENTER ROOM ENTER MEETING

CDP

ENTER ROOM ENTER MEETING

SBT

ENTER ROOM ENTER MEETING

CORE

ENTER ROOM ENTER MEETING

Vikings

ENTER ROOM ENTER MEETING

Mushin

ENTER ROOM ENTER MEETING

MATRIX

ENTER ROOM ENTER MEETING

Inception

ENTER ROOM ENTER MEETING

MoTeam



ENTER ROOM ENTER MEETING

Adoption

ENTER ROOM ENTER MEETING

JARVIS



ENTER ROOM ENTER MEETING

COMMANDOS

ENTER ROOM ENTER MEETING

OLIMPO

ENTER ROOM ENTER MEETING

Wanderlust

ENTER ROOM ENTER MEETING

# Estamos contratando

<https://grnh.se/da017d1f2>

# Obrigado!

Twitter: @ed\_neves

Telegram: @edmilton\_neves