



Community Webcast

CÓDIGO LIMPO:

UMA ABORDAGEM
INTRODUTÓRIA E
PRÁTICA DE COMO
LIMPAR CÓDIGOS

Um pouco de mim...

QUEM SOU

YAN DE LIMA JUSTINO. interessado por adoção de boas práticas de Arquitetura e engenharia de desenvolvimento.

O QUE FAÇO

Desenvolvedor Sênior na Tec-Soft Informática; responsável por pesquisas e implementação de melhores práticas de desenvolvimento de software; palestrante em eventos técnicos.

AGENDA

O QUE É CÓDIGO LIMPO?
NOMES SIGNIFICATIVOS
FUNÇÕES
COMENTÁRIOS
CLASSES

HÁ DUAS RAZÕES PELAS QUAIS VOCÊ ESTÁ ASSISTINDO ESTE WEBCAST:

**“Você é um programador e deseja se tornar
um ainda melhor. Ótimo. Precisamos de
programadores melhores”** Rober C. Martin



“Qualquer um pode escrever código
que um computador possa entender.
Bons programadores escrevem código
que humanos podem entender”

Martin Fowler

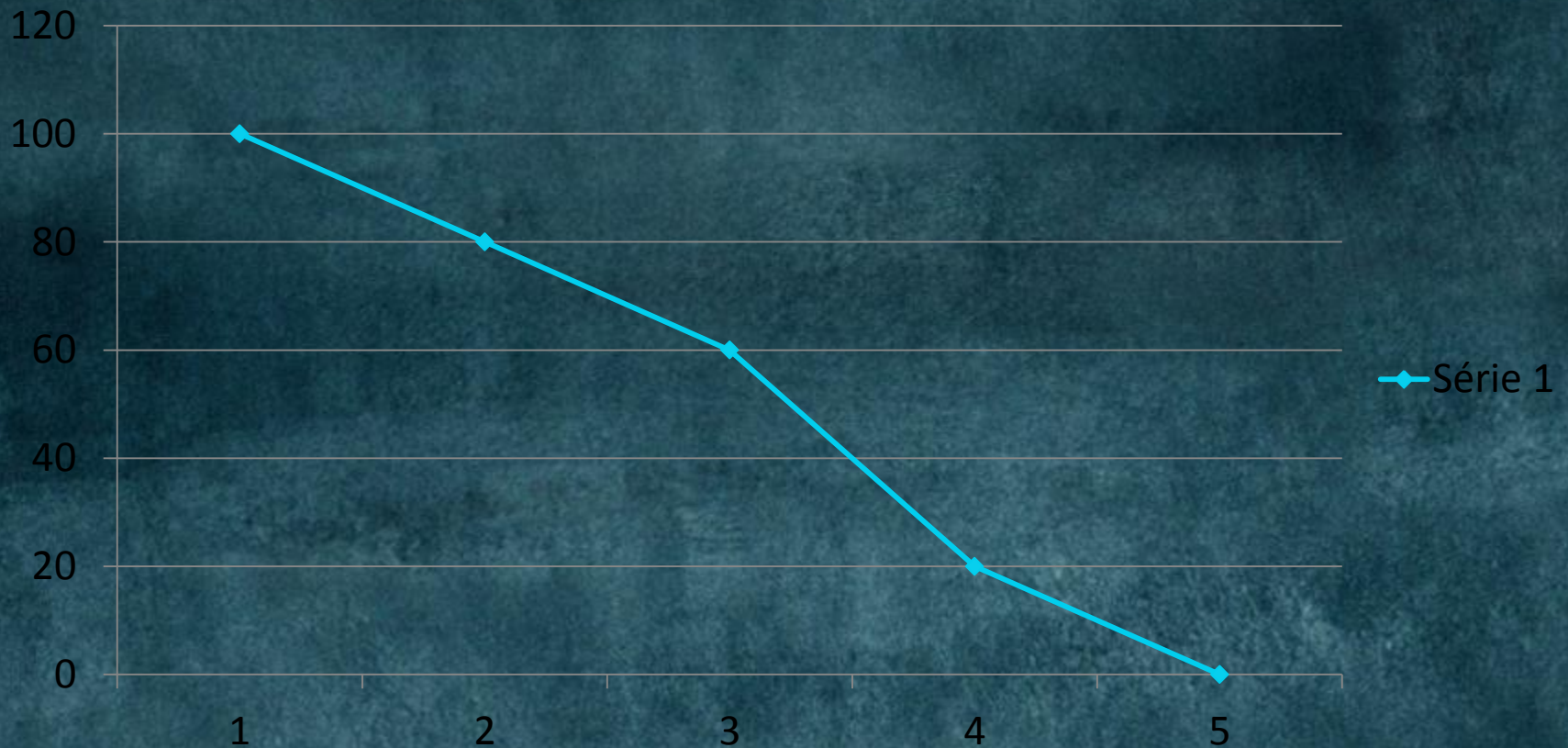
O QUE É CÓDIGO LIMPO?



Escrito por Robert C. Martin, O livro apresenta técnicas de como escrever códigos limpos, advindas de anos de experiências. Representa uma escola de pensamento sobre código

Custo de ter um código confuso

Produtividade X Tempo



“Não basta escrever um código bom. Ele Precisa ser mantido sempre limpo”.

Regra dos escoteiros:

Deixe a área do acampamento mais
Limpa do que como você a encontrou”



BLAH

blah
blah
blah

BLAH

4 BLAH

BLAH

Blah BLAH

BLAH

blah

BLAH

BLAH

BLAH

blah

BLAH

BLAH

BLAH

B

blah

Blah

Blah

BLAH

BLAH

blah

BLAH

Nomes Significativos



Nomes Significativos

- Use Nomes que revelem seu propósito
- Evite informações Erradas
- Use Nomes Pronunciáveis
- Passíveis de Busca
- Evite a notação Húngara
- Nomes de Classes
- Nomes de Métodos



Mãos à obra



Funções

- Faça Funções Pequenas
- Quer Faça Apenas uma Coisa
- Use Nomes Descritivos
- Use no máximo três parâmetros em funções
- Prefira exceções a retorno de código de erro



Mãos à obra



Comentários

- Explique-se No Código
- Use comentários para coisas relevantes: como alertas sobre consequências, Destaque, TODO
- Evite comentários redundantes



Mãos à obra



Classes



- Organização
 - Funções publicas abaixo após lista de variáveis e de propriedades
 - Tarefas privadas chamadas por uma função publica logo depois desta.
- Encapsulamento
 - As vezes é necessário tornar uma variável ou função como *protected* de modo que possa ser testada
- Classes pequenas
- Princípio da Responsabilidade única

Mãos à obra



POR TANTO,

“Um programador sem sensibilidade ao código pode visualizar um módulo confuso e reconhecer a bagunça, mas não saberá o que fazer a respeito dela. Já um com essa sensibilidade verá alternativas”.



OBRIGADO

