

# CIREF: UMA FERRAMENTA PARA A REFATORAÇÃO DE *CODE SMELLS* NO PROCESSO DE INTEGRAÇÃO CONTÍNUA

Aluno: Marcos Gênesis da Silva

Orientadora: Profa. Dra. Carla Ilane Moreira Bezerra

Banca avaliadora: Profa. Dra. Mairieli Santos Wessel, Prof. Dr. Anderson Gonçalves Uchôa e Prof. Mr. Camilo Camilo Almendra



UNIVERSIDADE  
FEDERAL DO CEARÁ  
CAMPUS DE QUIXADÁ

TRABALHO DE CONCLUSÃO DE CURSO  
QUIXADÁ, 13 DE JULHO DE 2022

# Contextualização

---

- Refatorações de *Code Smells*
  - Ferramentas atuais são confusas e não estimulam a participação direta dos desenvolvedores das equipes (KIM et al., 2014)
- Integração Contínua
  - Executar os testes automatizados e garantir a qualidade estática do código (HUMBLE; FARLEY, 2010)
  - Qualidade contínua de código
- Bots para VCS's (Version Control System)
  - Não atrapalham o fluxo de desenvolvimento, algo considerável importante para os desenvolvedores considerarem um dia produtivo ou não (MEYER et al., 2014)

# Objetivos

---

- **Objetivo geral:**
  - Desenvolver o **CiRef**: um BOT automatizado para detecção e listagem de refatorações no processo de IC
- **Objetivos específicos:**
  - Identificar se o ambiente de IC é propício para as refatorações de *code smells*
  - Criar um ambiente de interação direta com os desenvolvedores para o gerenciamento de refatorações de *code smells*
  - Validar o índice de refatorações sugeridas pelo CiRef que foram aceitas pelos desenvolvedores.

# Fundamentação teórica

---

- ***Code smells***

- Introduzido por Fowler et al (1999) com 22 tipos diferentes (e.g, Data Class, God Class, entre outros)
- Não necessariamente são ruins
- “Muitas vezes (smells) são um indicador de problema, e não o problema” (Fowler, 2006a)
- Ferramentas de detecção
  - Jdeodorant
  - Sonar Qube

# Fundamentação teórica

---

- **Refatoração**

- Um terço do tempo de um projeto de desenvolvimento de software é alocado para a manutenibilidade do mesmo (Coleman et al., 1994; Guimaraes, 1983)
- Apenas 7% dos 42% totais de operações envolvendo *code smells*, de fato, removeram o *code smell* envolvido (Bavota et al., 2015)
- Kim et al. (2015) mostra que os desenvolvedores necessitam de ferramentas fáceis de integrar e de revisar alterações
- Técnicas de refatoração
  - *Move Method*
  - *Extract Class*

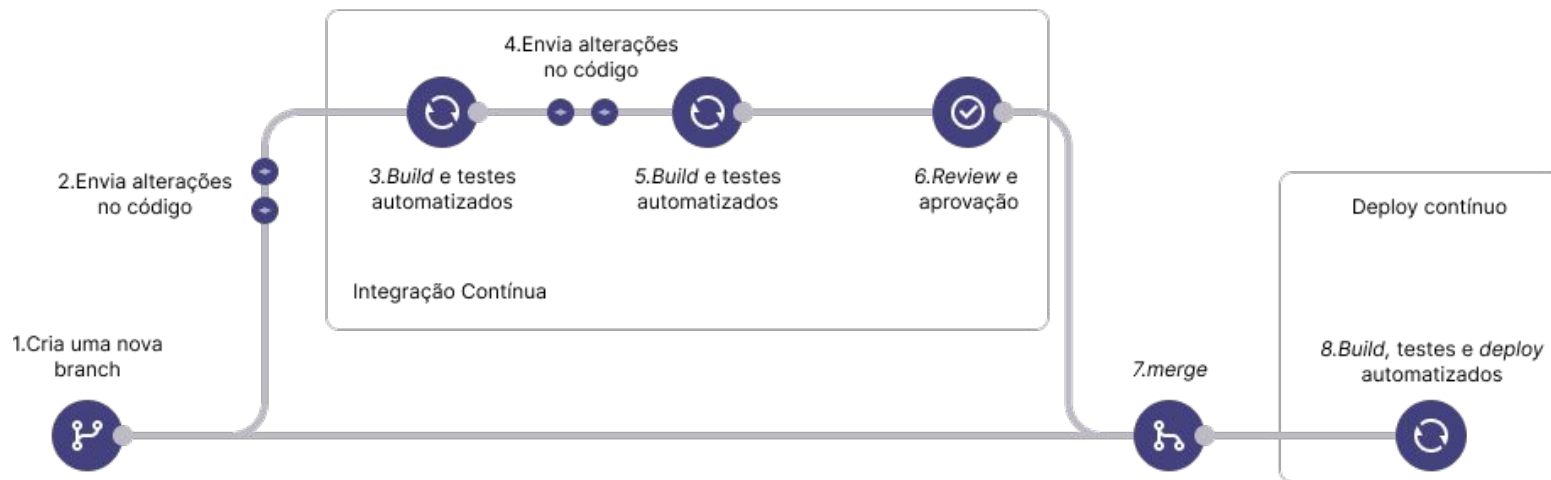
# Fundamentação teórica

---

- **Integração contínua (IC)**
  - Aumento da qualidade do código (ROSSI et al., 2016) e da produtividade dos desenvolvedores (STOL, 2017)
  - IC vem se tornando um protocolo base para o desenvolvimento de software (SCHAEFER et al., 2013)
  - A IC não fará os bugs sumirem, mas é fundamental na sua detecção e correção (MOHAMMAD, 2016)

# Fundamentação teórica

## Típico processo de integração contínua



Fonte: Adaptado documentação GitLab (disponível em <https://docs.gitlab.com/ee/ci/introduction/>)

# Fundamentação teórica

---

- **Bots para engenharia de software**
  - Um agente para a automação de rotinas e tarefas tediosas ou uma UI de estilo conversacional (LABEUF *et al.*, 2018)
  - Sumarização de resultados e centralização de informações, como alguns dos *tópicos* mais citados pelos desenvolvedores, identificados por Wessel *et al.* (2022)
  - Ferramentas atuais
    - A aceitação dos desenvolvedores é baixa devido à dificuldade de integração com o já existente fluxo de desenvolvimento e por hábitos de trabalho desses desenvolvedores (WYRICH; BOGNER, 2019)



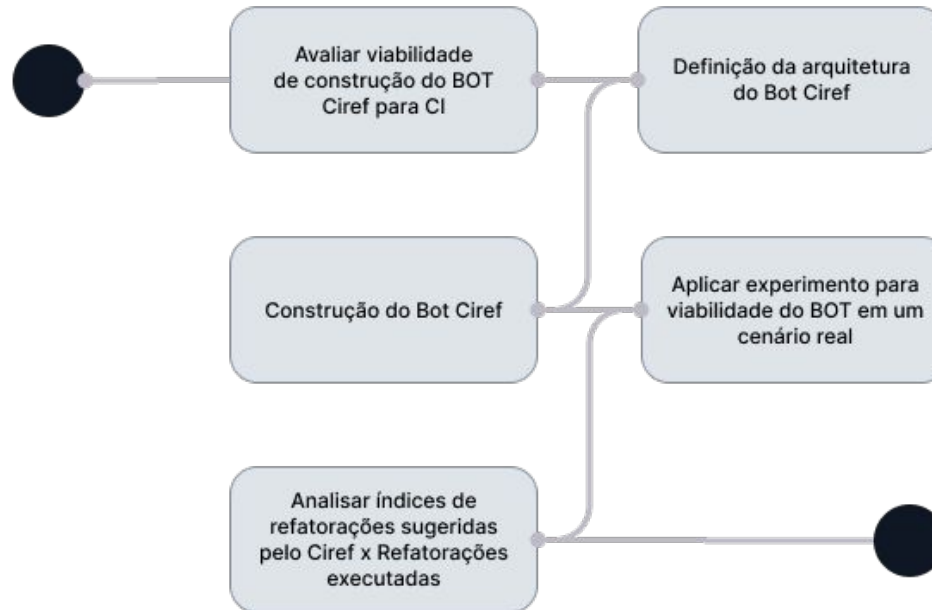
# Trabalhos relacionados

---

<b>Trabalhos</b>	<b>Identifica e apresenta code smells encontrados</b>	<b>Leva em consideração a opinião dos desenvolvedores</b>	<b>Avalia a qualidade de código</b>	<b>Possui integração com GitHub</b>
<b>Presente Trabalho</b>	Sim	Sim	Sim	Sim
<b>Alizadeh <i>et al.</i> (2019)</b>	Não	Não	Sim	Sim
<b>Vassallo <i>et al.</i> (2018b)</b>	Não	Sim	Não	Não
<b>Wyrich e Bogner (2019)</b>	Não	Não	Não	Sim
<b>Saidani <i>et al.</i> (2021)</b>	Não	Sim	Não	Não

# Procedimentos metodológicos

---



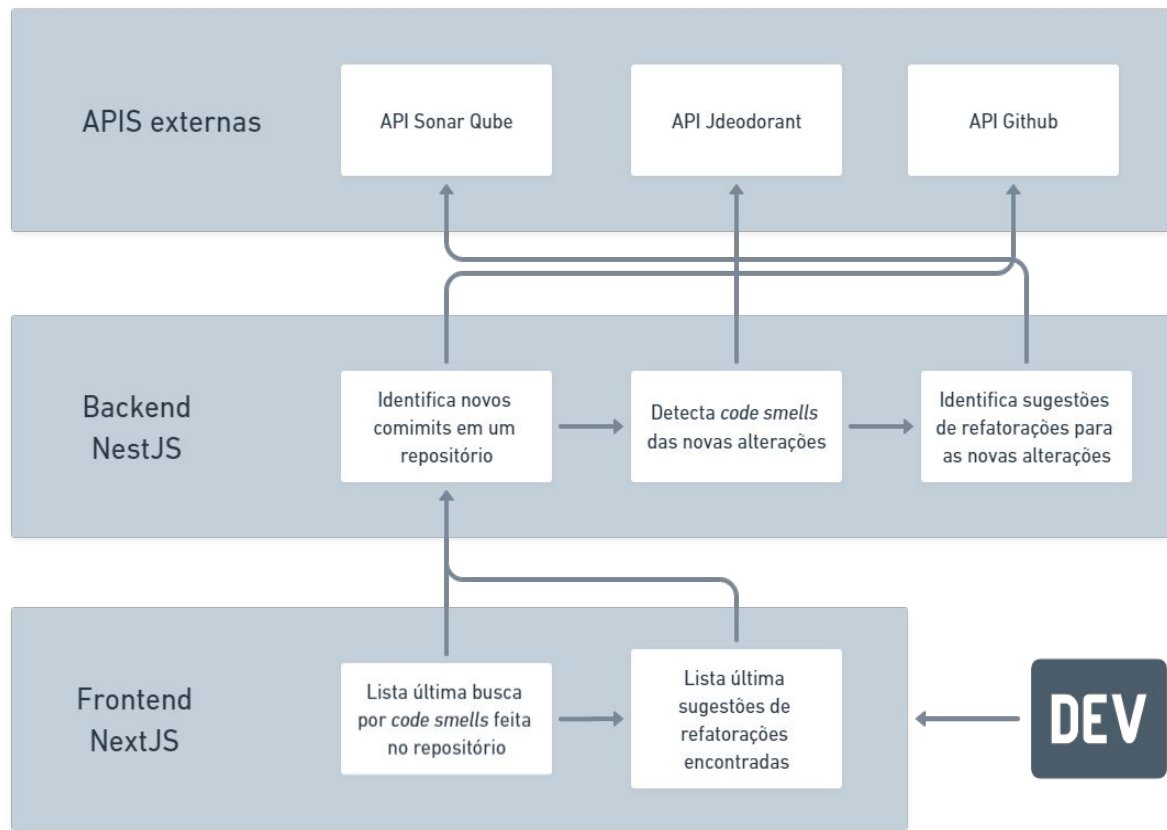
# Procedimentos metodológicos

---

- Avaliar viabilidade de construção do *bot* Ciref para CI
  - Github APP
    - Principalmente voltado para Pull Requests
  - Api do GitHub
    - Versátil
    - Independe de gatilhos disparados pelos desenvolvedores

# Procedimentos metodológicos

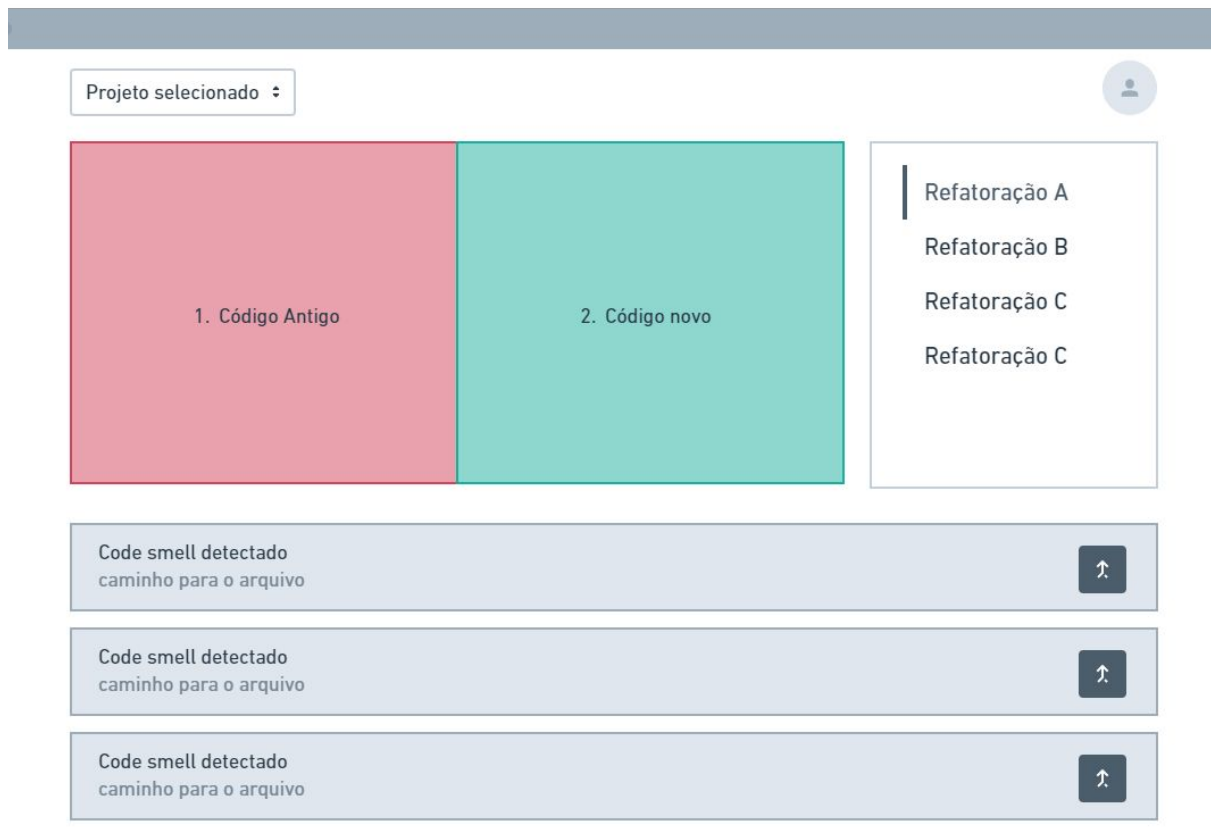
- Definição da arquitetura do *bot* Ciref



Fonte: Elaborado pelo autor

# Procedimentos metodológicos

- Construção do *bot* CiRef



Fonte: Elaborado pelo autor

# Procedimentos metodológicos

---

- Aplicar experimento para viabilidade do *bot* em um cenário real
  - Ciref será exposto a um ambiente real
  - Será coletado também, o índice de satisfação dos desenvolvedores com o Ciref
- Analisar índices de refatorações sugeridas pelo Ciref x Refatorações executadas
  - Recursos gráficos (dispersões, barras ou outros)
  - Condicionados aos dados coletados no passo anterior

# Procedimentos metodológicos

- Cronograma de execução

ATIVIDADES	2022													
	Jun		Jul		Ago		Set		Out		Nov		Dez	
Criação do frontend em NextJS	X	X	X											
Desenvolver API com Jdeodorant			X	X										
Construção do backend em NestJS			X	X	X	X								
Integrar backend com API do GitHub			X	X	X	X								
Integrar backend com API do SonarQube			X	X	X	X								
Integrar backend com API Jdeodorant construída			X	X	X	X								
Integrar backend com frontend							X	X						
Disponibilizar Ciref para projeto real escolhido									X	X				
Analisar os dados obtidos pelo experimento e gerar gráficos e tabelas para comparação										X	X			
Revisão final da monografia												X		
Defesa da monografia													X	-

# Resultados Preliminares

Criação do frontend em NextJS

## Ciref BOT

marcosgenesis  
visualizar perfil

src/app/index.java	src/app/Car.java	src/app/Home.java	Ops...
1 const a = 10	1 const a = 10	1 const a = 10	
2 - const b = 10	2 + const b <b>oo</b> = 10	2 + const b oo = 10	
3 - const c = () => console.log('foo')			
4 - if(a > 10) {	3 + if(a === 10) {	3 + if(a === 10) {	
5 console.log('bar')	4 console.log('bar')	4 console.log('bar')	
6 }	5 }	5 }	
7 -			
8 - console.log('done')			

### Possíveis Refatorações

EXTRACT METHOD  
./src/app/Car/Door.java

11 Criar PR

EXTRACT METHOD  
./src/app/Car/Door.java

11 Criar PR

EXTRACT METHOD  
./src/app/Car/Door.java

11 Criar PR

EXTRACT METHOD  
./src/app/Car/Door.java

11 Criar PR

EXTRACT METHOD  
./src/app/Car/Door.java

11 Criar PR

EXTRACT METHOD  
./src/app/Car/Door.java

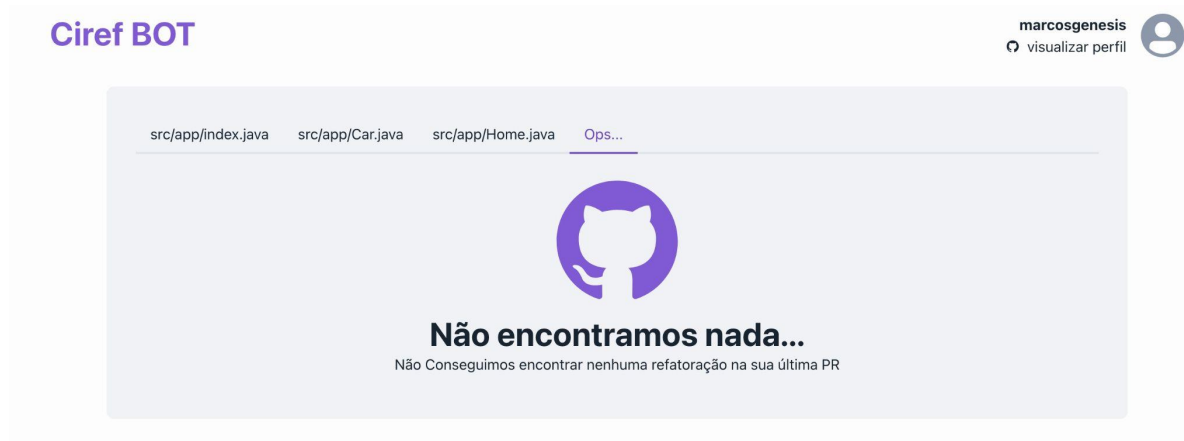
11 Criar PR

Fonte: Elaborado pelo autor



# Resultados Preliminares

Criação do frontend em NextJS



Fonte: Elaborado pelo autor

# Muito Obrigado!!!

---