

● ●

TRABALHO DE CONCLUSÃO DE CURSO I

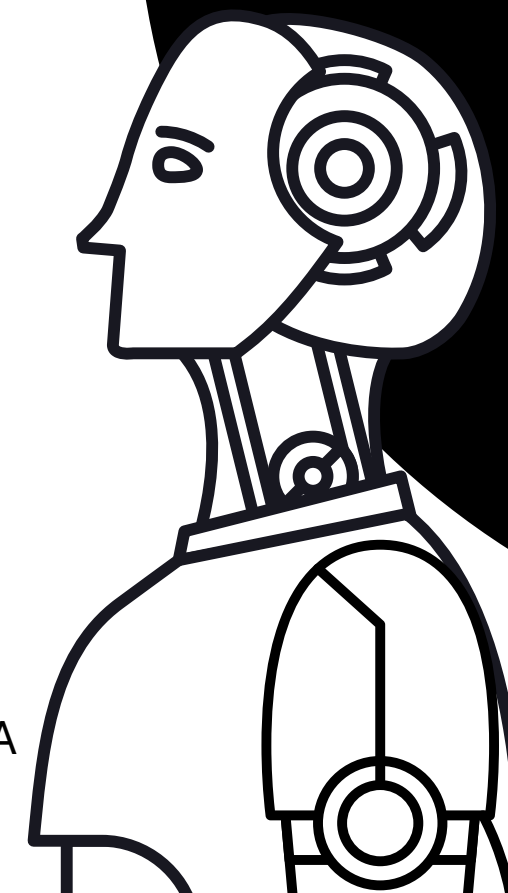
● ●

# HOOKCI

Um novo conceito de CI

GABRIEL DOLABELA MARQUES  
HENRIQUE CARVALHO ALMEIDA

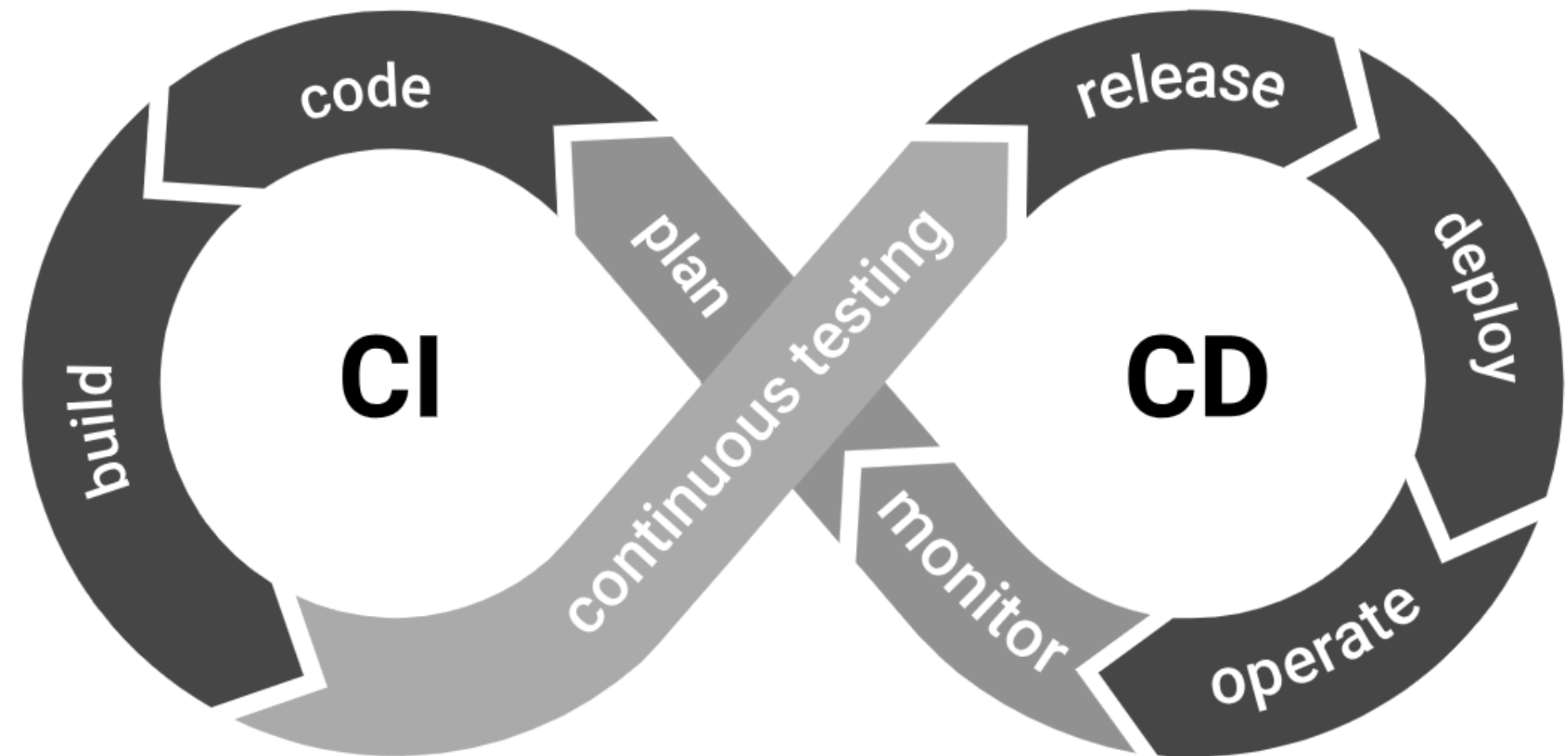
ORIENTADORES  
CLEITON SILVA TAVARES  
JOANA GABRIELA RIBEIRO DE SOUZA



## O PROBLEMA DA CI TRADICIONAL

Feedback lento e sobrecarga de servidores em CI remota geram gargalos e atrasam o desenvolvimento.

Ferramentas locais pecam pela inconsistência de ambientes, dificultando a padronização das validações de código entre a equipe.



## HOOKCI: A SOLUÇÃO

### Objetivo

- Realizar testes de CI antes de commits e pushes
- Reduzir falhas no repositório remoto
- Otimizar o ciclo de desenvolvimento

### Escopo

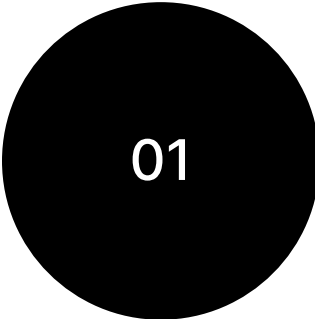
- Integração com Git (hooks)
- Execução local em Docker
- Configuração via YAML
- Uso por linha de comando (CLI)

HookCI redefine a CI ao trazê-la para o ambiente local do desenvolvedor, combinando a automação dos Git Hooks com o isolamento dos contêineres Docker.

Ao tentar um commit ou push, o HookCI intercepta a ação e executa testes configurados via YAML dentro de um contêiner Docker, permitindo a operação Git apenas em caso de sucesso da CI.

A configuração centralizada e a execução isolada proporcionam consistência entre desenvolvedores, otimizando o tempo e elevando a qualidade do código desde o início.

FUNCIONALIDADES DO PRODUTO



VALIDAÇÃO PREVENTIVA DE CÓDIGO



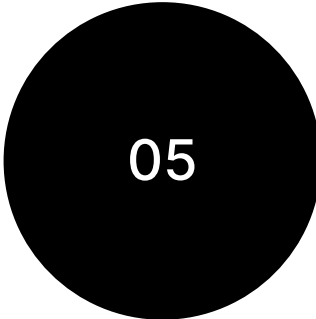
RÁPIDA INICIALIZAÇÃO EM PROJETOS



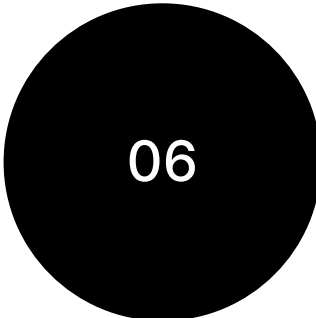
RETORNO IMEDIATO AOS DESENVOLVEDORES



ISOLAMENTO DO AMBIENTE DE CI



DEPURAÇÃO EM CASO DE ERRO



FLEXIBILIDADE NA CONFIGURAÇÃO E EXECUÇÃO

## ATORES DO SISTEMA

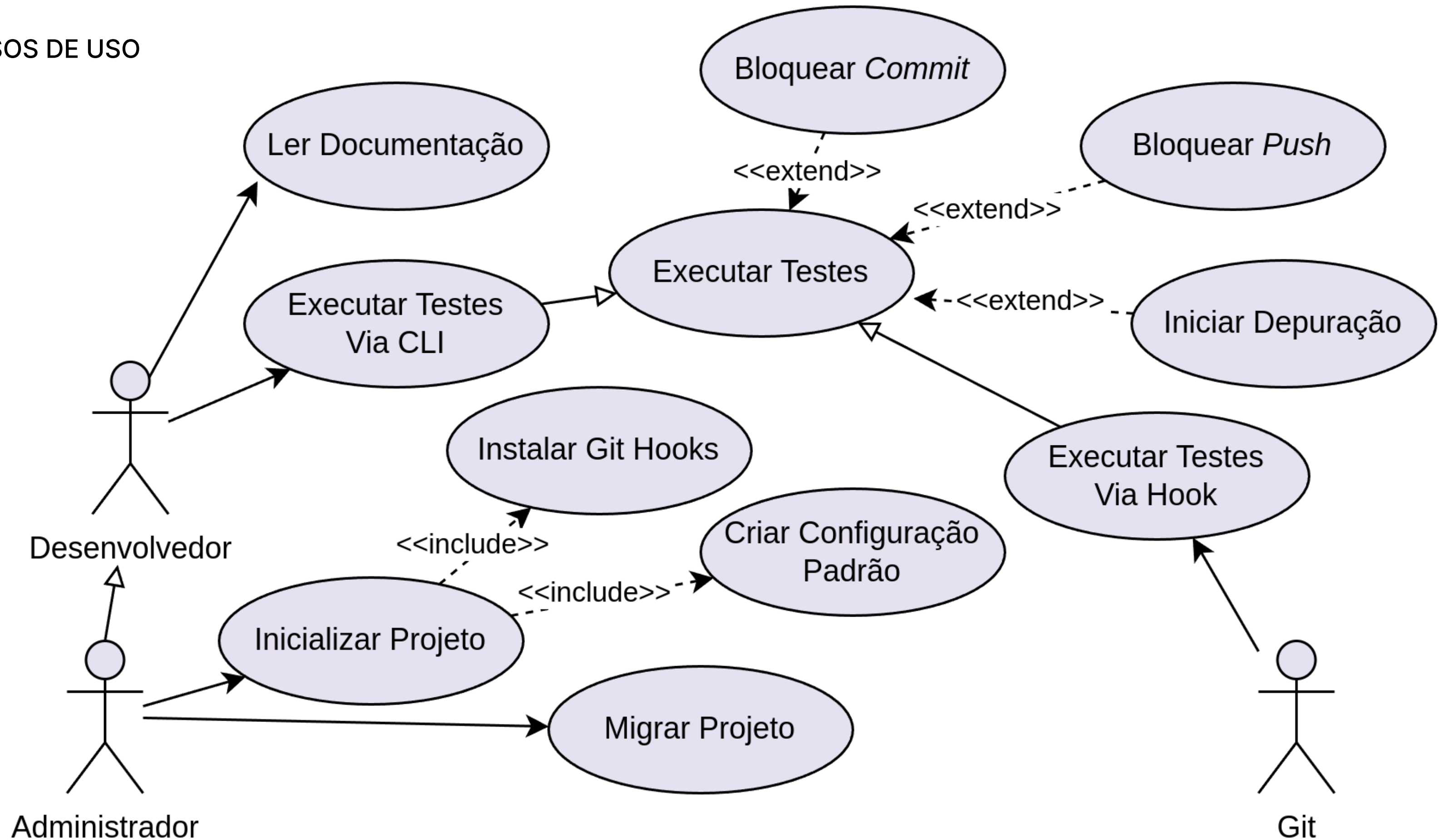


- Administrador
  - Configura o HookCI
  - Define padrões de teste para a equipe

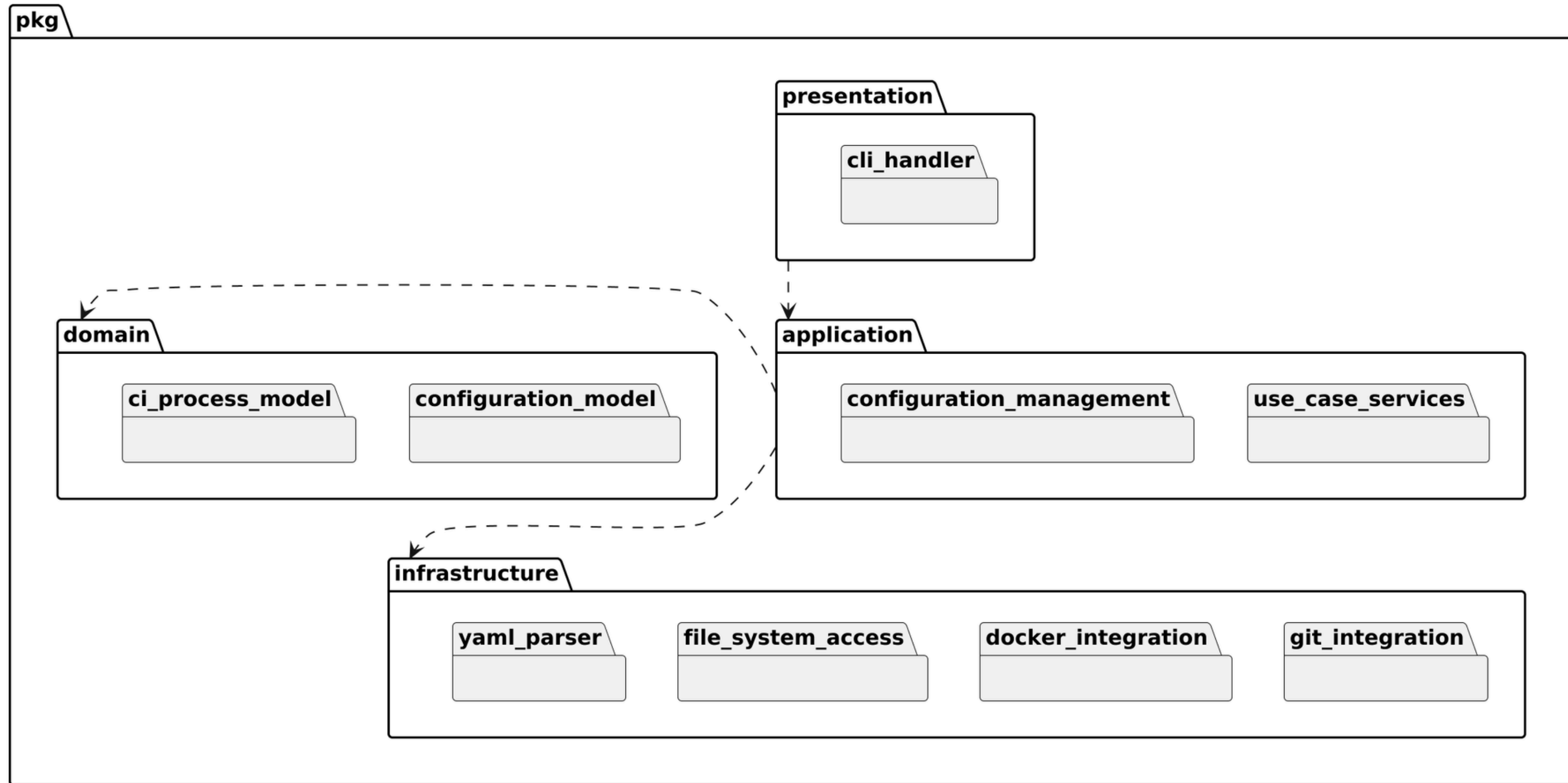


- Desenvolvedor
  - Valida código antes de integrar
  - Recebe feedback local

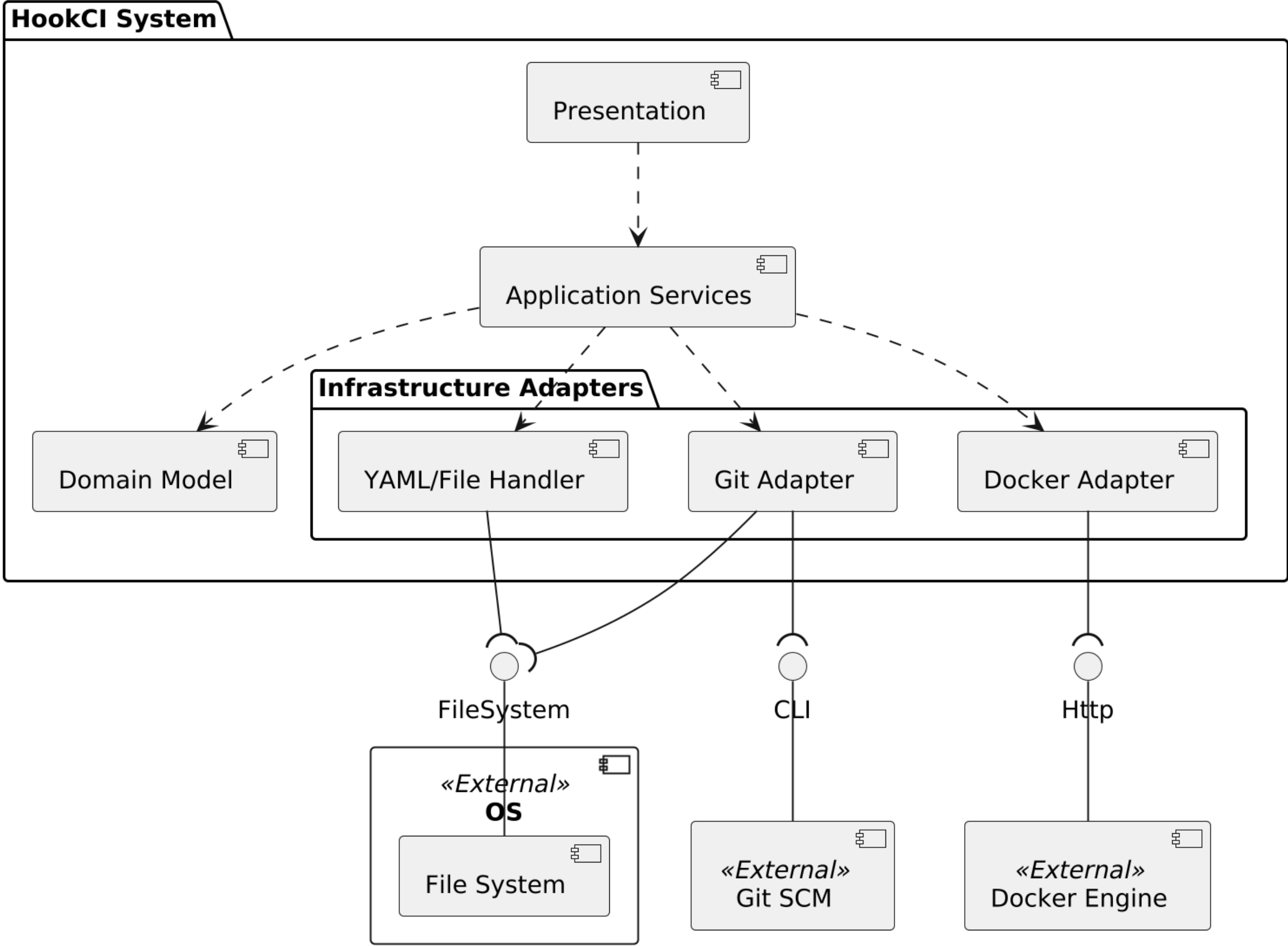
## CASOS DE USO



# ARQUITETURA



COMPONENTES





## INTERFACE

```
$ hookci
```

### HookCI - Help

#### Available Commands:

- init** - Initializes HookCI in the current repository.
- help** - Displays this help message.
- test** - Manually runs the tests.
- version** - Shows the current version of the tool.
- migrate** - Migrates old config to new format.

## INTERFACE

```
$ hookci inot
```

### HookCI - Error

```
Unknown command: 'inot'
```

```
Did you mean: init?
```

```
Use 'help' to see the available commands.
```

## INTERFACE

```
$ hookci init
```

### HookCI - Init

Initializing HookCI...

- Creating YAML config file...
- Installing Git Hooks...
- Checking dependencies...

## INTERFACE

```
$ hookci run
```

### HookCI - Running Tests

- ✓ Starting test run...
- ✓ Creating Docker container...
- ⌚ Running test suite...
- ⌚ Finalizing results...

```
$ hookci run
```

### HookCI - Running Tests

- ✓ Starting test run...
- ✓ Creating Docker container...
- ✓ Running test suite...
- ✓ Finalizing results...

**Tests completed successfully! ✓**

CRONOGRAMA

AGO/1

Gabriel - Implementação da inicialização pela criação do YAML padrão e estrutura inicial.  
Henrique - Implementação da estrutura base da CLI com parsing, do comando help e configuração do ambiente de desenvolvimento Docker.

US02, US03  
US01, NF:  
Usabilidade da CLI

AGO/2

Gabriel - Desenvolvimento do parser YAML para carregar a configuração e validação básica da estrutura.  
Henrique - Implementação da instalação automática dos Git hooks durante o init.

US05, US14  
US04

SET/1

Gabriel - Implementar execução de testes ao chamar “hookci run” com suporte a YAML.  
Henrique - Integração inicial com Docker através da execução de comandos de steps em contêineres e montagem do repositório no contêiner.

US09  
US10, US18

SET/2

Gabriel - Implementar execução de testes via Git hooks, integrando com a lógica de run.  
Henrique - Implementação do sistema de logging, verbosidade e suporte para imagem Docker/Dockerfile.

US08  
US11, US12, US16,  
US17

CRONOGRAMA

OUT/1	Gabriel - Implementação de testes não essenciais na CI.	US15
	Henrique - Implementação do modo de depuração em caso de falha nos testes.	US13
OUT/2	Gabriel - Implementação da funcionalidade de migração de configuração através do comando "hookci migrate".	US06, US07
	Henrique - Criação da documentação com GNU Roff	Documentação
NOV/1	Gabriel - Configuração de pipeline CI/CD para o projeto HookCI. Criação de testes unitários.	CI/CD NF: Testes
	Henrique - Pipeline de CI auto-testada pelo HookCI	CI/CD
NOV/2	Gabriel - Criação de Casos de Teste de Aceitação.	NF: Testes
	Henrique - Empacotamento do sistema via PyInstaller e .deb.	CI/CD

FIM

**OBRIGADO PELA ATENÇÃO**