

Embarcadero Conference 2024

Inovação faz parte do nosso DNA!

Inovando o Desenvolvimento de Software

Do Delphi ao pipeline contínuo

{Armando Corrêa Henrique Neto

Graduado em Ciência da Computação

Pós Graduação em Gerenciamento de Projetos

Pós Graduação em Arquitetura de Software

Delphi Developer certificado pela Embarcadero Academy (2021-2023)

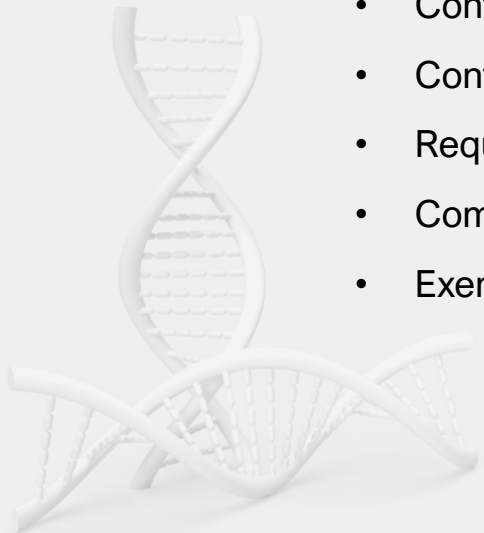
17 anos trabalhando como desenvolvedor de sistemas

Atualmente trabalhando como Analista de Sistemas na Senior Sistemas – Blumemau/SC



O que veremos hoje?

- Processos
- Controladores de Versão
- Repositórios de Código-Fonte e Branches
- Versionamento Semântico
- Análise Estática
- Continuous Integration – CI
- Continuous Delivery – CD
- Requisitos
- Como Mudar a Cultura?
- Exemplo na Prática



Quer me ver na
#ECON25?
Acesse o QRCode
e avalie minha palestra!



Processos

“Refere-se a uma sequência **estruturada** de atividades realizadas dentro de uma empresa para alcançar um objetivo. Esse processo envolve **pessoas, ferramentas, tecnologias e recursos.**”

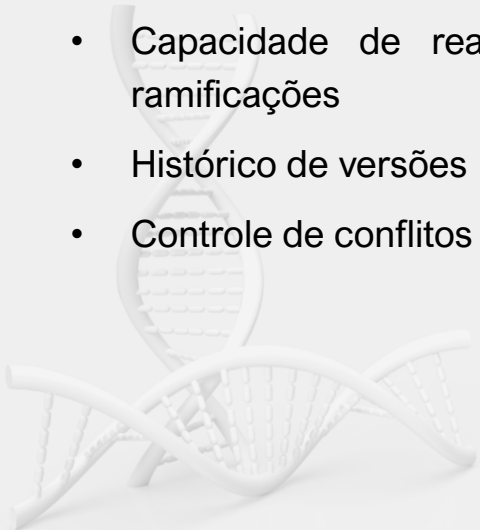


Controladores de Versão

“São ferramentas que nos auxiliam a gerenciar e controlar as mudanças no nosso código-fonte.”

Funcionalidades

- Trabalhar com ramificações para implementar novas funcionalidades
- Capacidade de realizar mesclagem de alterações entre as ramificações
- Histórico de versões
- Controle de conflitos

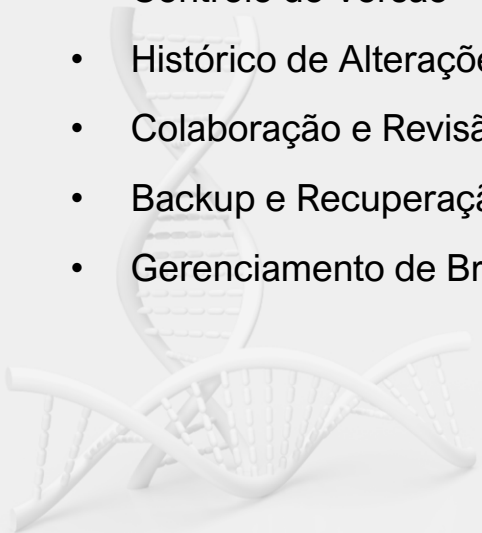


Repositórios de Código-Fonte e Branches

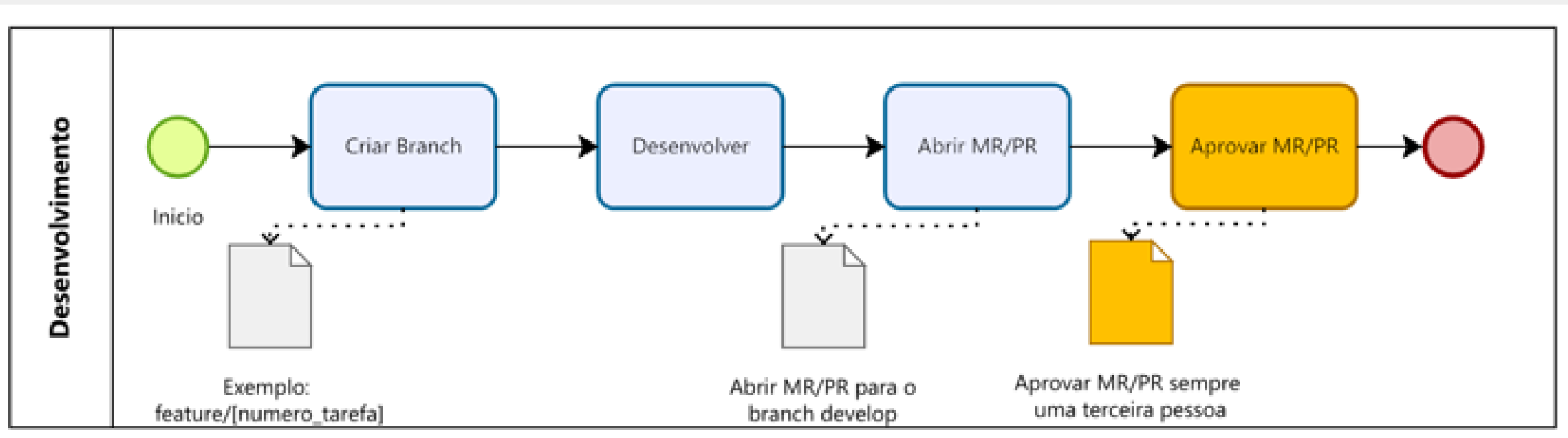
“São locais de armazenamento, gerenciamento e versionamento do código-fonte do nosso projeto.”

Funcionalidades

- Armazenamento Centralizado ou Distribuído
- Controle de Versão
- Histórico de Alterações
- Colaboração e Revisão de Código
- Backup e Recuperação
- Gerenciamento de Branches



Repositórios de Código-Fonte e Branches



Versionamento Semântico

“É uma convenção para atribuir números de versão ao sistema de maneira **padronizada**. Ajuda a informar de forma clara as mudanças realizadas no sistema.”

Funcionamento

- MAJOR
- MINOR
- PATCH

Importância

“Ajuda desenvolvedores e usuários a entenderem o impacto das mudanças entre as versões do sistema.”



Versionamento Semântico - Changelog

“É um arquivo de controle das atividades e suas versões.”



```
# {version}
[{date}]

### Major
* N/A.

### Minor
* [Task-002] - Título da Tarefa

### Patch
* [Task-002] - Título da Tarefa

# 1.1.2
[30/12/2023]

* [Task-número] - [Título da Tarefa]

# 1.1.1
[15/12/2023]

* [Task-número] - [Título da Tarefa]

# 1.1.0
[01/10/2023]

* [Task-número] - [Título da Tarefa]
* [Task-número] - [Título da Tarefa]

# 1.0.0
[01/06/2023]

* [Task-número] - [Título da Tarefa]
* [Task-número] - [Título da Tarefa]
```

Análise Estática

“São ferramentas capazes de analisar nosso código-fonte sem executá-lo, com a finalidade de identificar possíveis problemas, erros ou melhorias.”

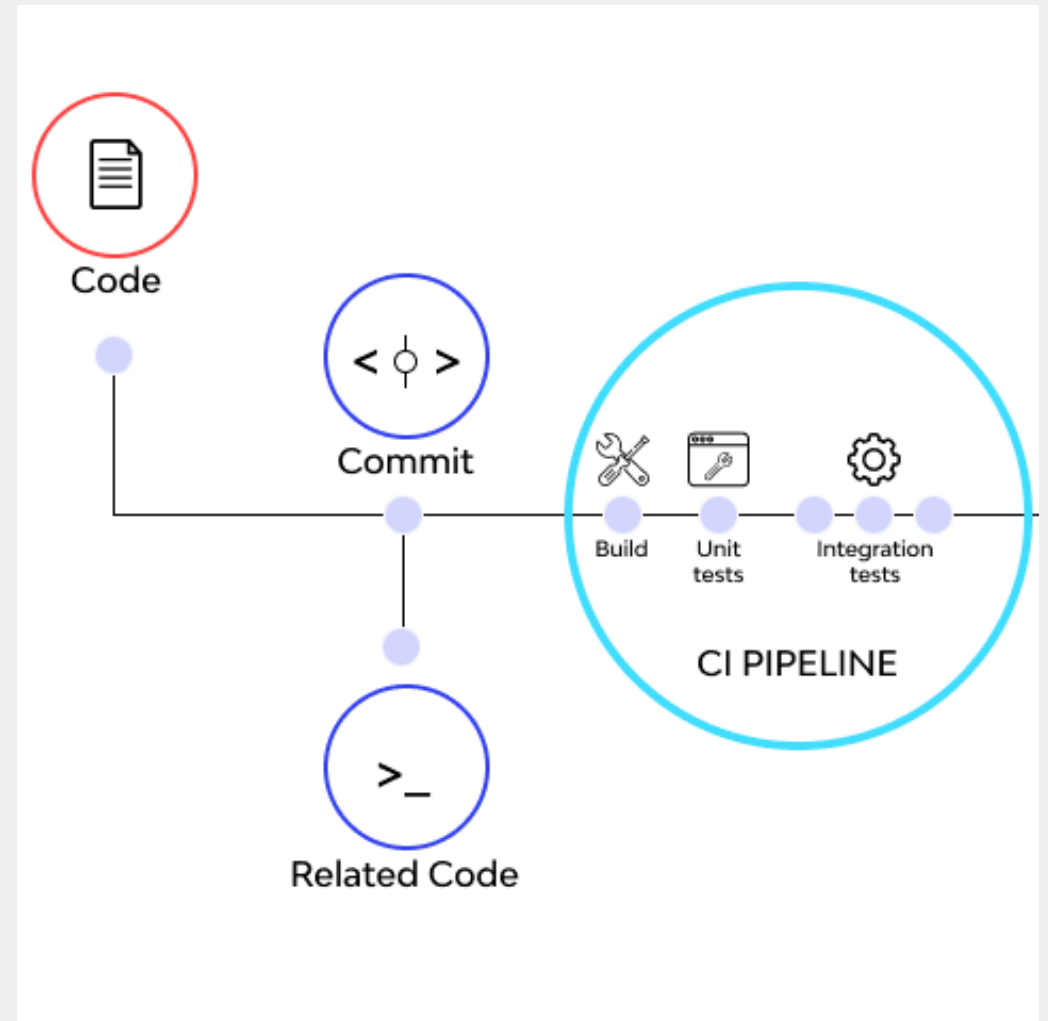
Funcionalidades

- Detecção de Erros e Vulnerabilidades
- Verificação de Conformidade com Padrões
- Análise de Complexidade
- Identificação de Código Morto
- Sugestão de Melhoria

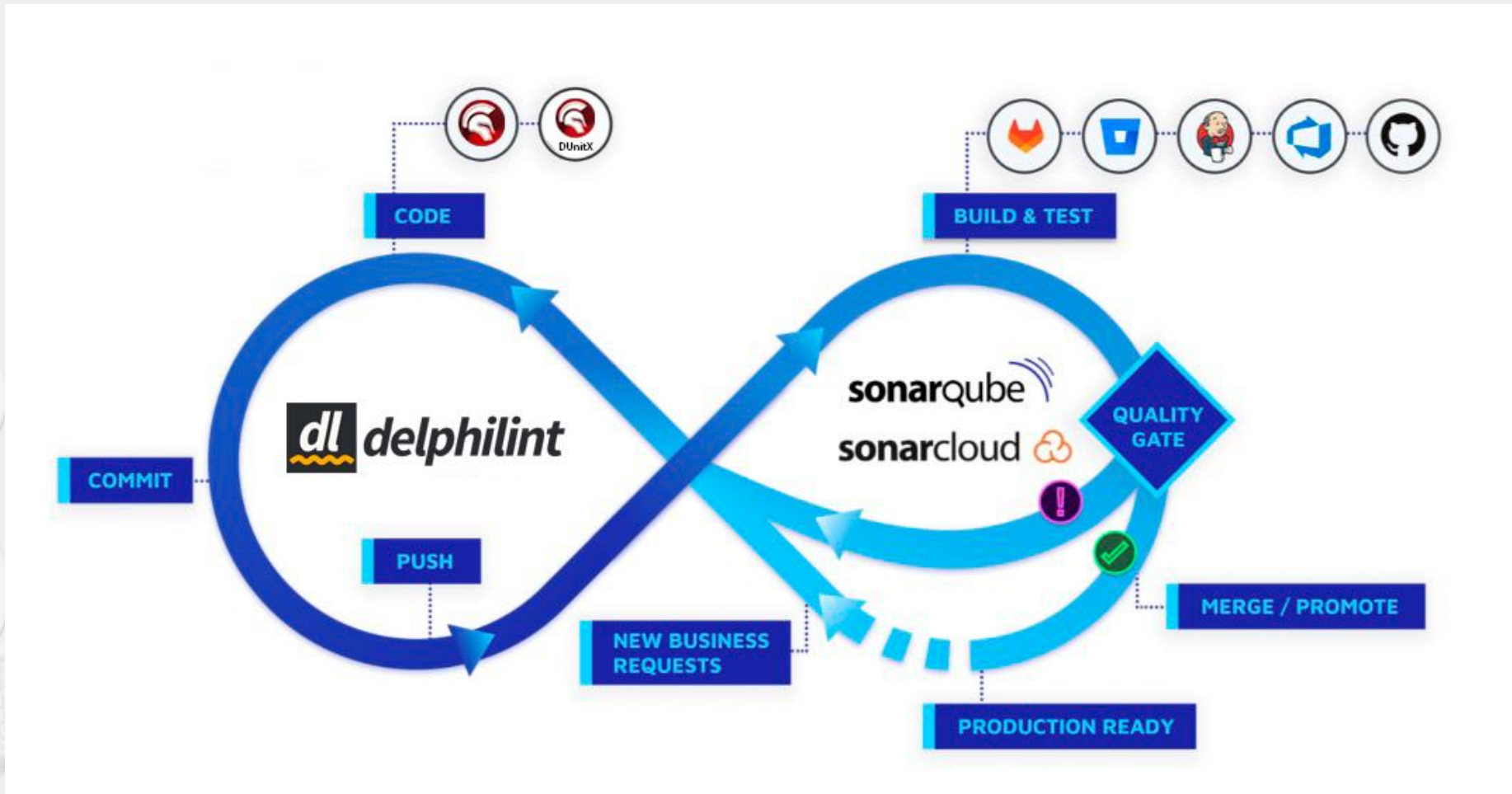
The SonarQube logo features the word "sonarqube" in a lowercase, sans-serif font. To the right of the text is a blue icon consisting of three curved lines, resembling a sonar wave or a stylized 'S'.The ESLint logo consists of a blue hexagonal icon with a white geometric pattern inside, followed by the word "ESLint" in a bold, sans-serif font.The Pylint logo features the Python logo (two interlocking snakes, one blue and one yellow) to the left of the word "Pylint" in a bold, sans-serif font. Below "Pylint" is a small tagline "Star your Python code!" preceded by five small diamond icons.The checkstyle logo features the word "checkstyle" in a bold, sans-serif font. A red wavy line runs underneath the text, and a yellow pencil icon is positioned at the end of the line.

Continuous Integration

“É uma prática onde desenvolvedores integram diversas vezes seu código aos repositórios de código-fonte. Cada integração é verificada por uma build automatizada.”

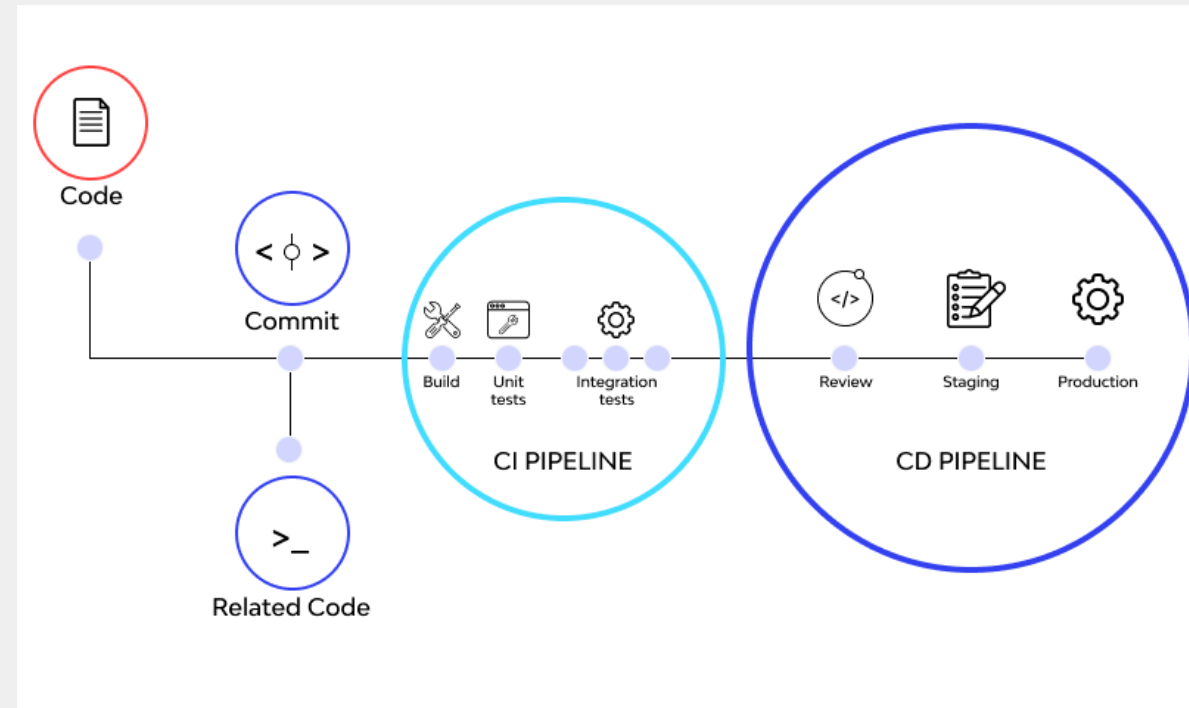


Continuous Integration

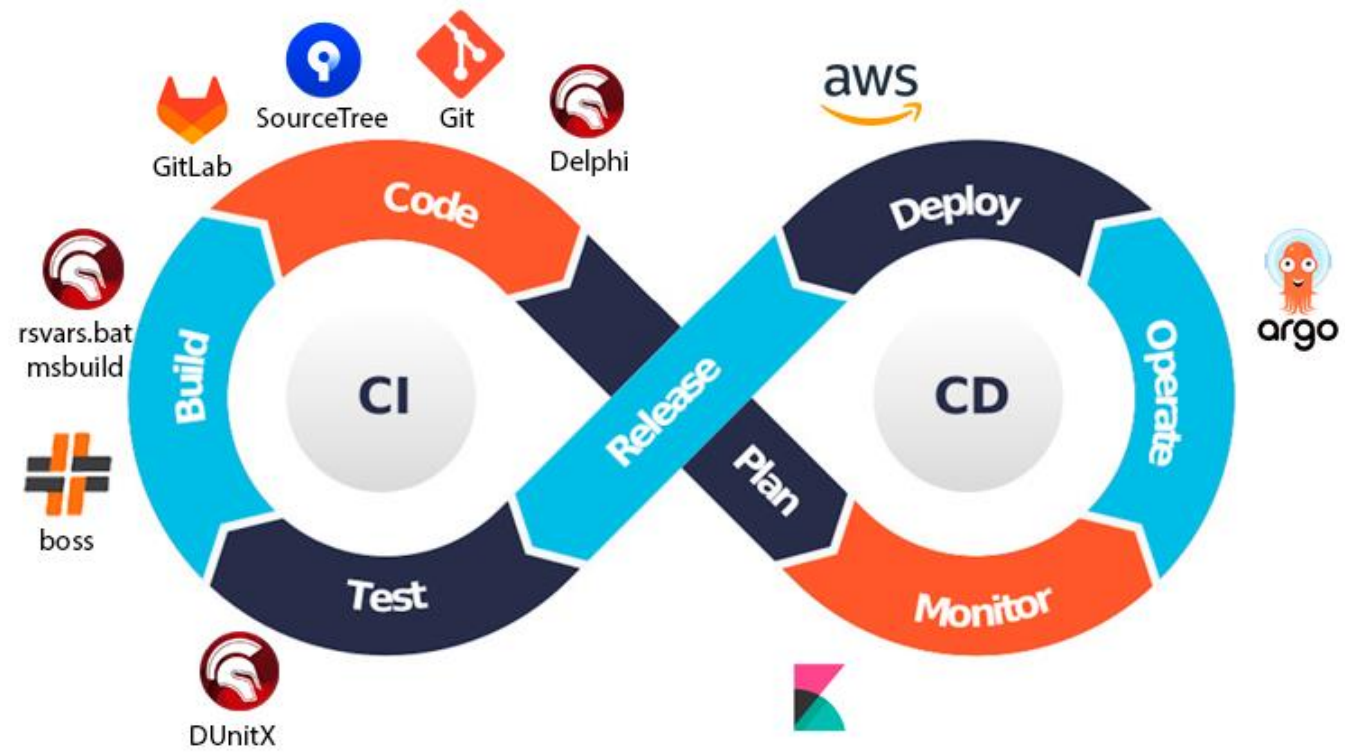


Continuous Delivery

“É uma prática onde há a garantia em que o software esteja sempre pronto para ser entregue em produção. É um processo que trabalha em conjunto com o CI.”

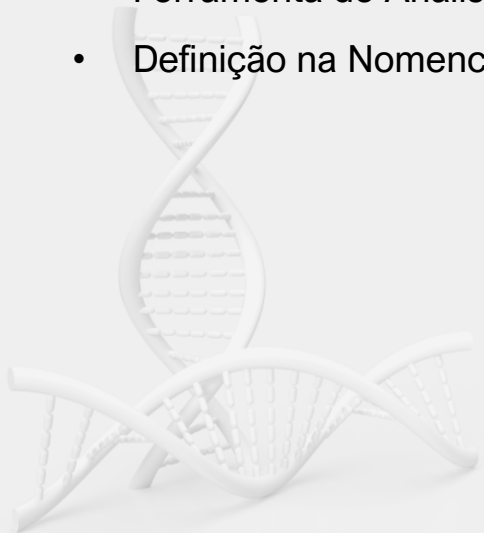


Continuous Delivery



Requisitos

- Delphi Professional+ (Compilação via linha de comando)
- Servidor ou VM (Instalação do Delphi)
- Ferramenta de Controle de Versão (Git)
- Repositório de Código-Fonte (GitLab ou GitHub) – Com suporte a Pipelines
- Ferramenta de Análise Estática (SonarQube)
- Definição na Nomenclatura das Branches e Changelog



Quer me ver na
#ECON25?
Acesse o QRCode
e avalie minha palestra!



Como Mudar a Cultura?

- Step 1: Implantação do Git e GitLab ou GitHub.
- Step 2: Criar uma esteira (pipeline) somente para realizar o build.
- Step 3: Adicionar na esteira a execução dos testes unitários.
- Step 4: Adicionar na esteira integração com o SonarQube.
- Step 5: Adicionar na esteira o ciclo de CD.



Exemplo na Prática

- Exemplo 1: Pipeline com Falha.
- Exemplo 2: Pipeline com Sucesso.
- Exemplo 3: Deploy versão Major, Minor ou Patch.

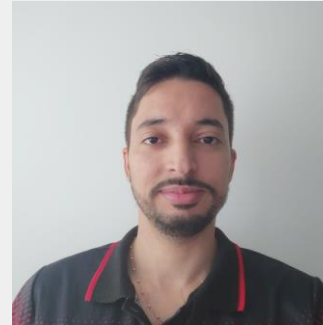


Embarcadero Conference 2024


Inovação faz parte do nosso DNA!





Quer me ver na
#ECON25?
Acesse o QRCode
e avalie minha palestra!



Armando Neto

 [@armandonetodev](https://twitter.com/armandonetodev)

 [armandochneto](https://www.linkedin.com/in/armandochneto)

 kikobattery@hotmail.com

