



# Git: Economize DorFlex!

Gerenciamento de código com Git



# Quem Sou Eu ?

- Desenvolvedor de software há 4 anos sempre usando FOSS.
- Colaborador na Fundação Paulo Feitoza.
- Estudante de ciência da computação e fascinado pela área.
- Experiência com desenvolvimento WEB, Desktop e Mobile(Tablet).
- Atualmente estudando bastante compiladores.
- Eterno curioso... :-)





# Sumário

- O que é gerenciamento de código ?
- Introdução ao git
  - Conceitos básicos
- Fluxo de trabalho com git
- Aplicando conceitos básicos com GIT:
  - Trabalho local: Clone, Add, Commit, Checkout, Branch, Merge e Stash
  - Trabalho colaborativo: Push e Pull
- Revisão
- Exemplo prático usando com git



# Gerenciamento de código ?

- Controle de versão de código ou um documento qualquer durante o processo de desenvolvimento do mesmo.

# Gerenciamento de código ?

- Controle de versão de código ou um documento qualquer durante o processo de desenvolvimento do mesmo.
- Histórico do desenvolvimento de um software.



# Gerenciamento de código ?

- Controle de versão de código ou um documento qualquer durante o processo de desenvolvimento do mesmo.
- Histórico do desenvolvimento de um software.
- Controle de Ramificação.

# Gerenciamento de código ?

- Controle de versão de código ou um documento qualquer durante o processo de desenvolvimento do mesmo.
- Histórico do desenvolvimento de um software.
- Controle de Ramificação.
- Rastreamento de alterações feitas durante o desenvolvimento do software.




# Gerenciamento de código ?

- Controle de versão de código ou um documento qualquer durante o processo de desenvolvimento do mesmo.
- Histórico do desenvolvimento de um software.
- Controle de Ramificação.
- Rastreamento de alterações feitas durante o desenvolvimento do software.
- Trabalho em equipe.




# Introdução ao Git

- Git é um software open-source de gerenciamento de versão de código distribuído.
- 



# Introdução ao Git

- Git é um software open-source de gerenciamento de versão de código distribuído.
  - Desenvolvido por Linus Torvald.
- 



# Introdução ao Git

- Git é um software open-source de gerenciamento de versão de código distribuído.
- Desenvolvido por Linus Torvald.
- Arquitetado para suportar projetos de pequeno, médio e grande porte.



# Introdução ao Git

- Git é um software open-source de gerenciamento de versão de código distribuído.
- Desenvolvido por Linus Torvald.
- Arquitetado para suportar projetos de pequeno, médio e grande porte.
- Alternativa ao Mercurial, Bazaar, Subversion, CVS, VSS(Visual Source Safe) entre outros.




# Introdução ao Git

- Alguns projetos que usam GIT:
  - **Debian** :)
  - Kernel
  - Gnome
  - Qt
  - Ruby on Rails
  - Android
  - e etc.



# Conceitos básicos

- Clone: Faz cópia de um repositório (Área de trabalho) do endereço informado.
- 



# Conceitos básicos

- Clone: Faz cópia de um repositório (Área de trabalho) do endereço informado.
- Add: Adiciona o arquivo ou diretório informado para ser rastreado pelo Git.



# Conceitos básicos

- Clone: Faz cópia de um repositório(Área de trabalho) do endereço informado.
- Add: Adiciona o arquivo ou diretório informado para ser rastreado pelo Git.
- Commit: Armazena as modificações realizadas



# Conceitos básicos

- Clone: Faz cópia de um repositório(Área de trabalho) do endereço informado.
- Add: Adiciona o arquivo ou diretório informado para ser rastreado pelo Git.
- Commit: Armazena as modificações realizadas.
- Checkout/Branch: Manipulação de ramificações



# Conceitos básicos

- Clone: Faz cópia de um repositório(Área de trabalho) do endereço informado.
- Add: Adiciona o arquivo ou diretório informado para ser rastreado pelo Git.
- Commit: Armazena as modificações realizadas.
- Checkout/Branch: Manipulação de ramificações
- Push: Manda modificações commitadas para o repositório informado.



# Conceitos básicos

- Clone: Faz cópia de um repositório(Área de trabalho) do endereço informado.
- Add: Adiciona o arquivo ou diretório informado para ser rastreado pelo Git.
- Commit: Armazena as modificações realizadas.
- Checkout/Branch: Manipulação de ramificações
- Push: Manda modificações commitadas para o repositório informado.
- Pull: Traz as modificações commitadas do repositório informado.

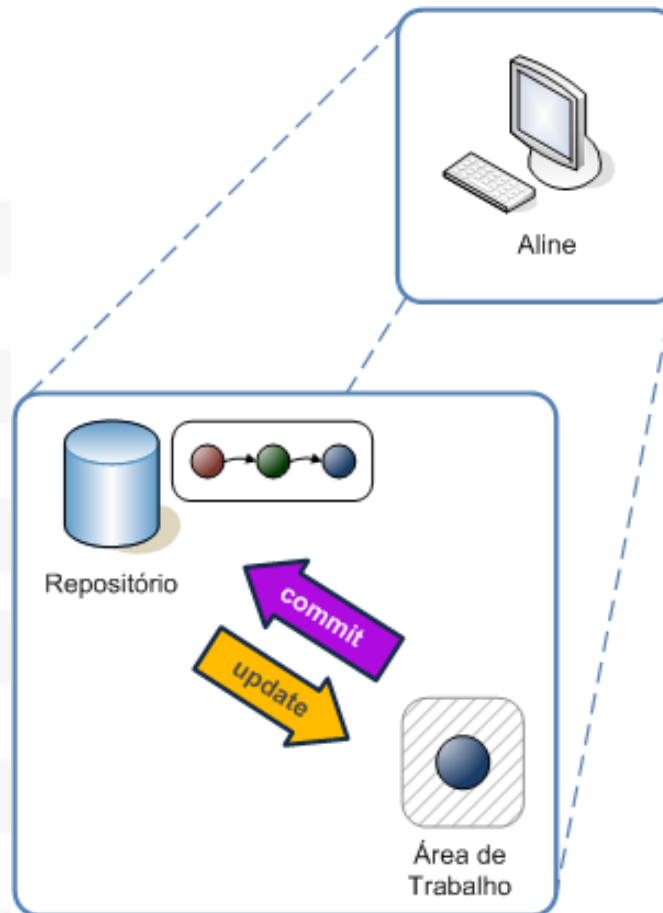


# Conceitos básicos

- Merge: Mesclar(Unir) duas ramificações.
  - Pode gerar conflitos

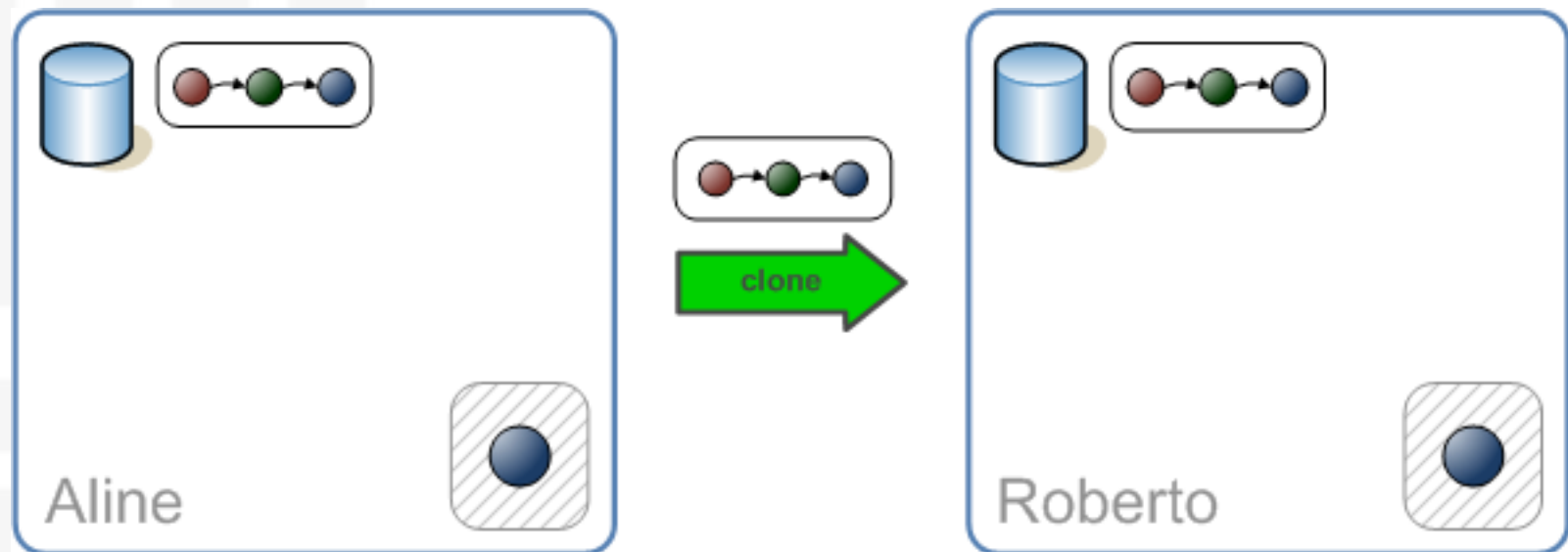
# Fluxo de trabalho

- Área de trabalho(working copy, repositório)



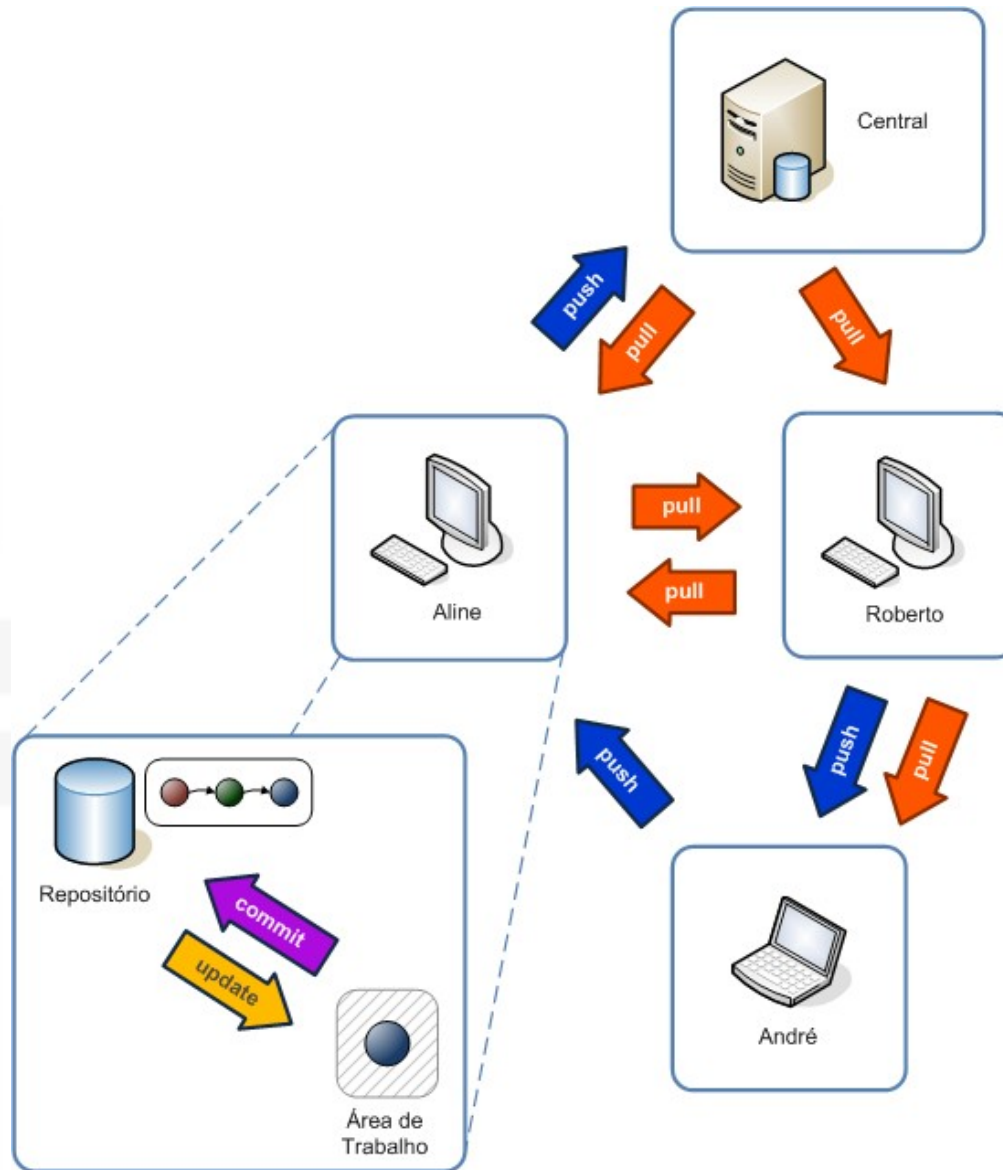
# Fluxo de trabalho

- Trabalho colaborativo



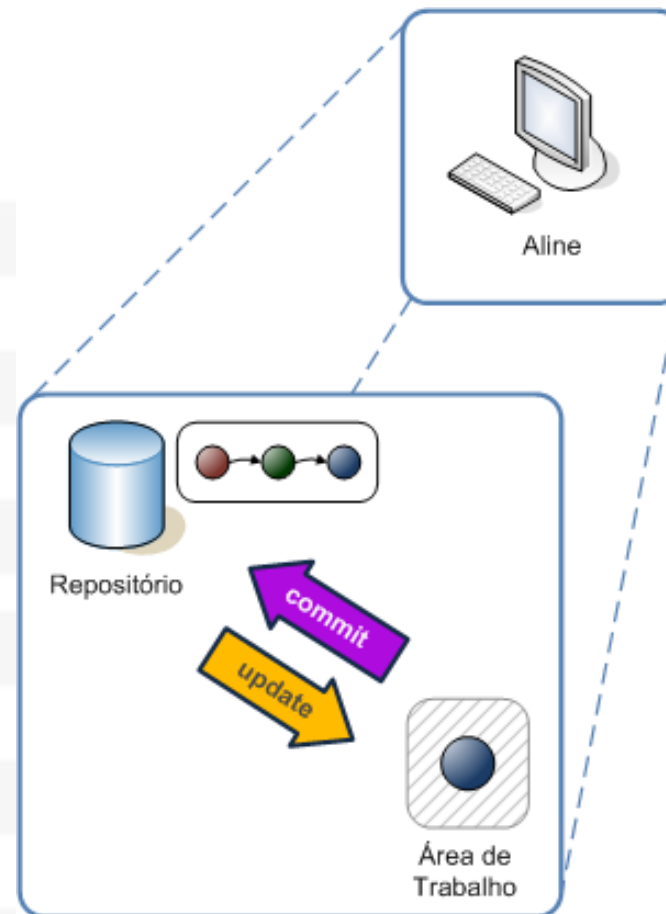
# Fluxo de trabalho

- Trabalho colaborativo centralizado



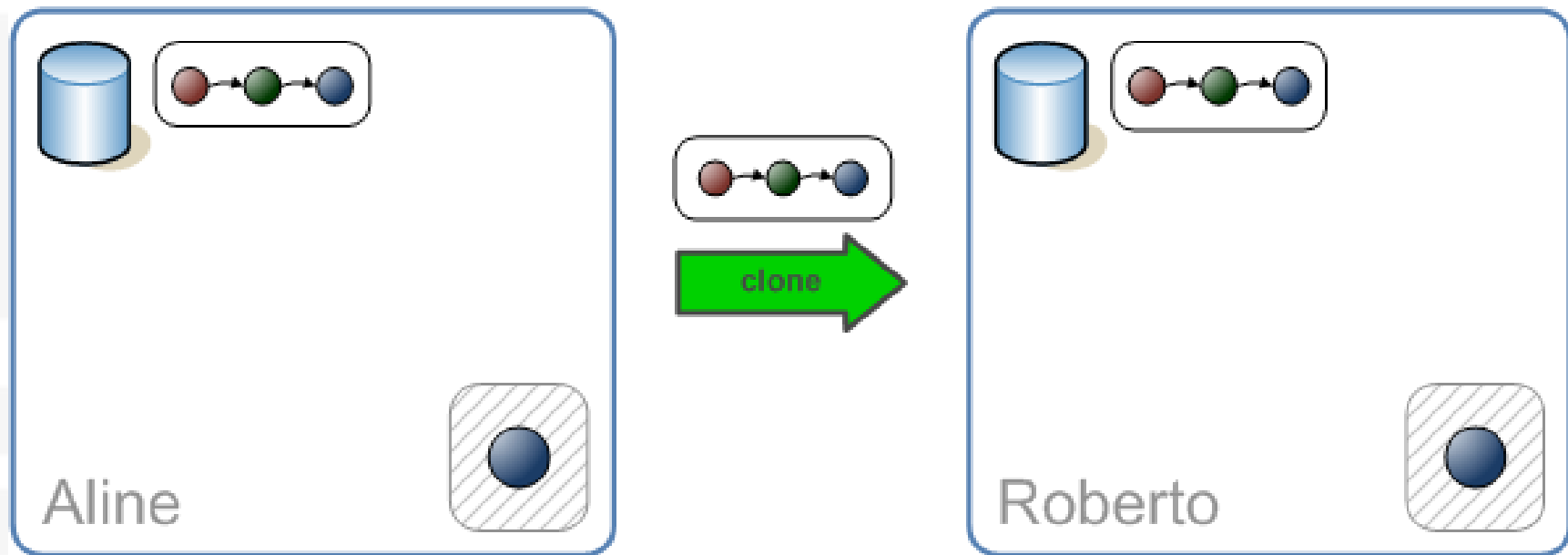
# Exemplo de um fluxo real de desenvolvimento

- Aline cria um repositório e começa a trabalhar



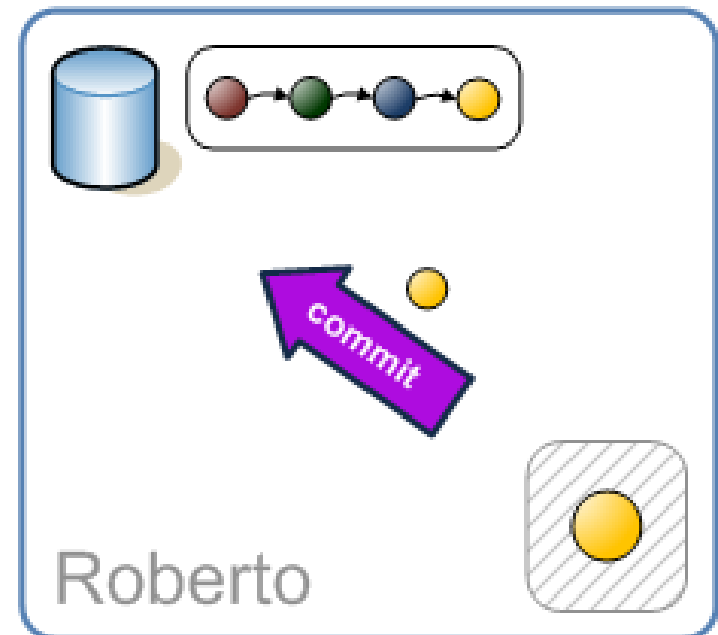
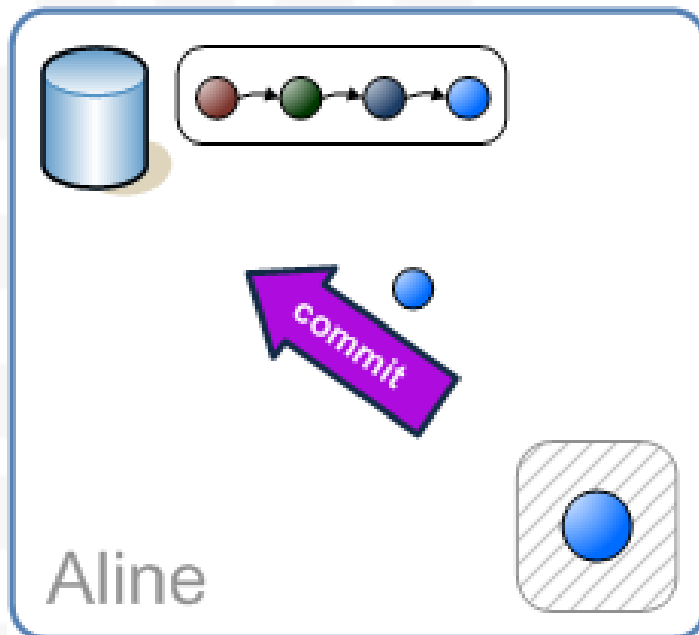
# Exemplo de um fluxo real de desenvolvimento

- Roberto também irá participar do projeto



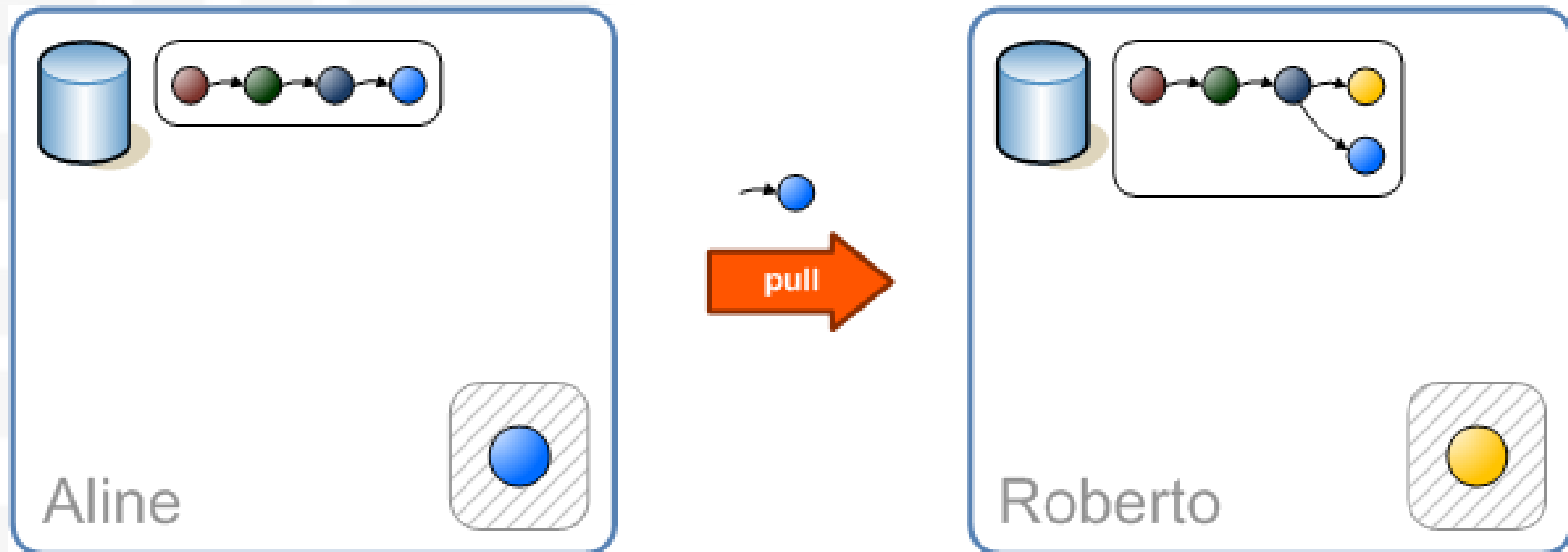
# Exemplo de um fluxo real de desenvolvimento

- Os dois começam a trabalhar (Cada um no seu repositório)



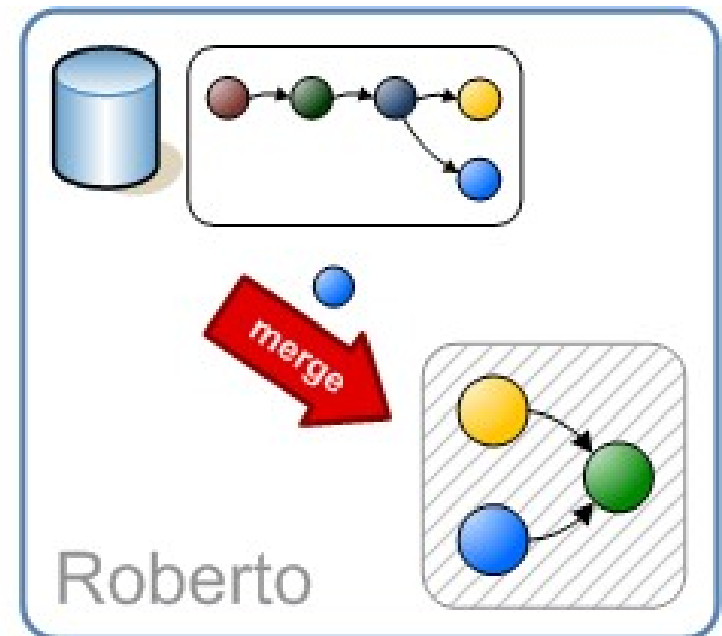
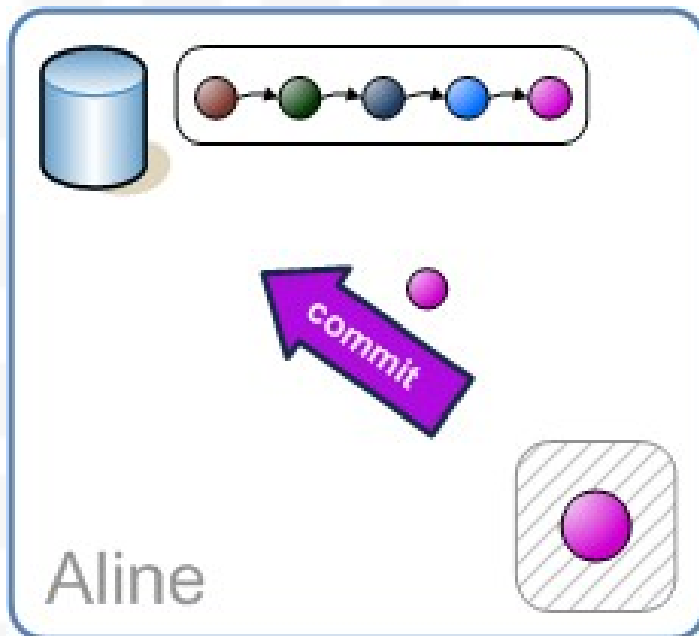
# Exemplo de um fluxo real de desenvolvimento

- Aline falou que terminou uma funcionalidade e o Roberto quer ou precisa usar essas modificações



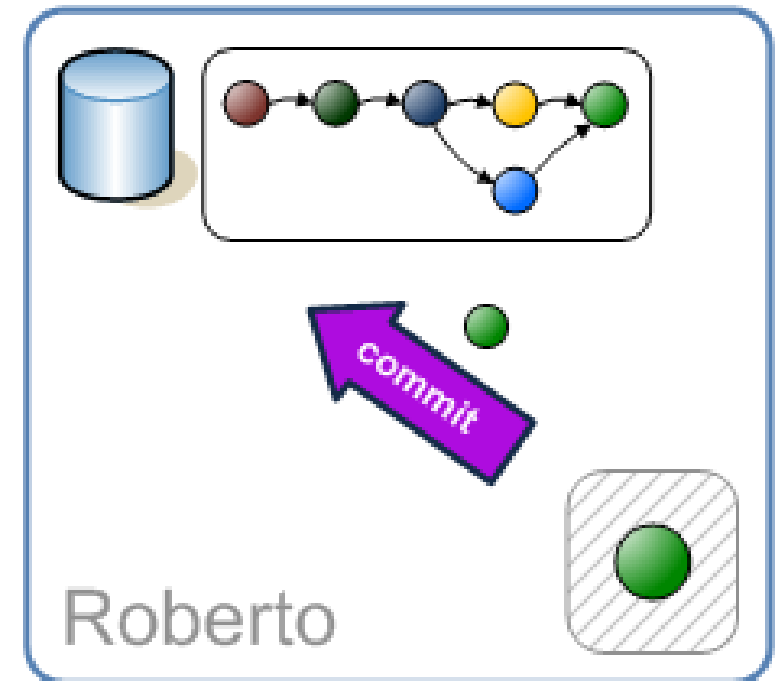
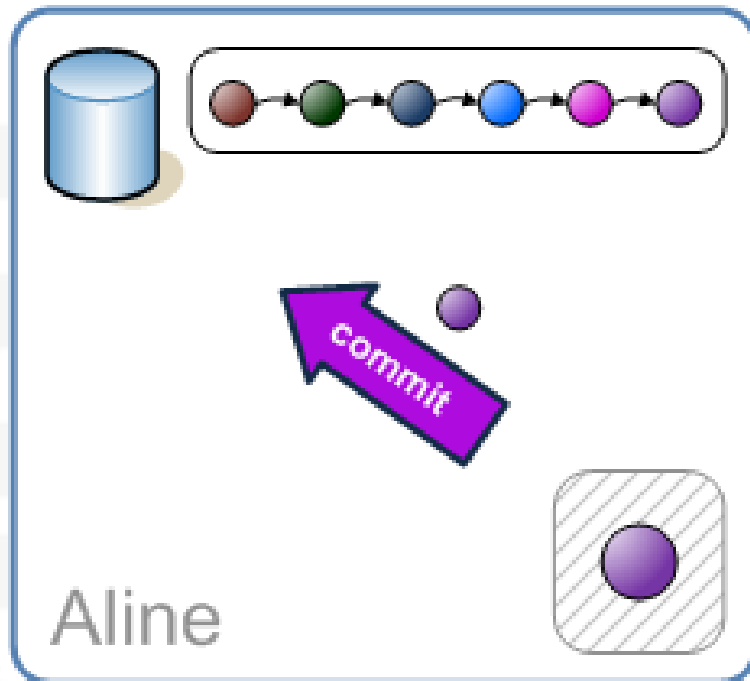
# Exemplo de um fluxo real de desenvolvimento

- Mas roberto já tinha feito algumas alterações e commitado.



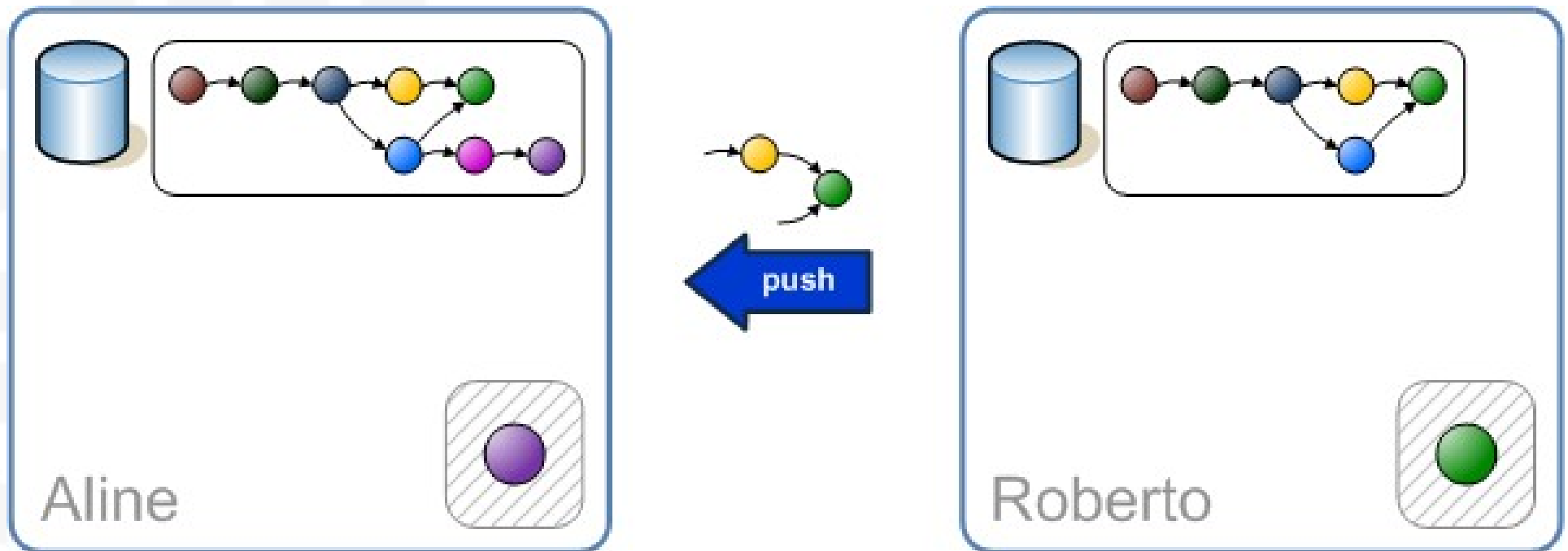
# Exemplo de um fluxo real de desenvolvimento

- Após o merge ele commita as alterações feitas



# Exemplo de um fluxo real de desenvolvimento

- E então roberto deseja enviar de volta para aline as suas modificações já integrado com as modificações dela.





# Revisão

- Para centralizar o processo de trabalho com repositórios distribuídos é preciso um trabalho de centralização e revisão.
- Exemplo de fluxo com revisão:
  - [gitorious.org](https://gitorious.org)

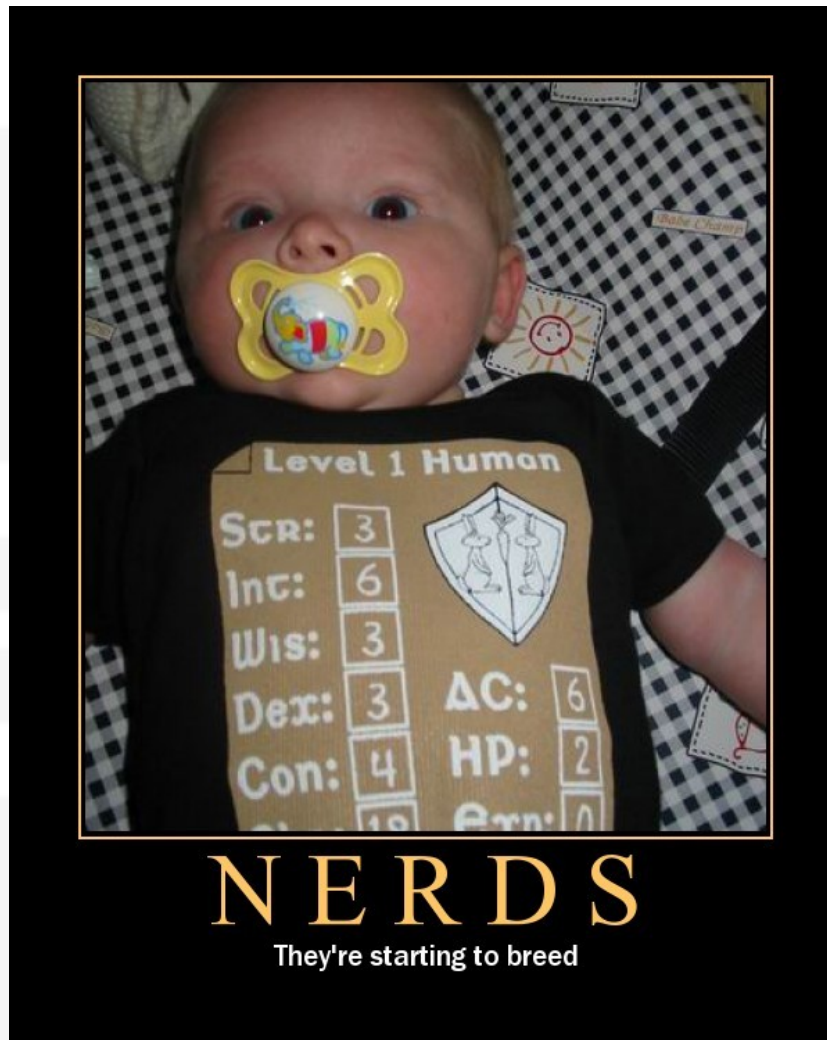


# Exemplo prático

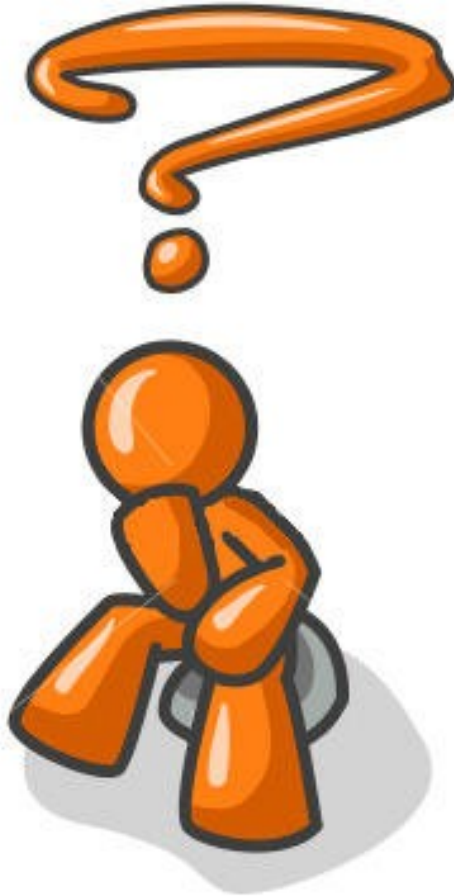
- Exemplo prático
  - Criação de repositório
  - Ramificação
  - Merge
  - Criação de Patch
  - Clone
  - Push
  - Pull

# Resumindo...

- Economize DorFlex usando NEOSALDINA! :-P



# Dúvidas



# Agradecimentos

- André Felipe Dias, Msc (Pronus Engenharia de Software) por permitir usar imagens de um artigo escrito em seu blog( [pronus.eng.br](http://pronus.eng.br) ), muito bom por sinal.
- Nome do Artigo: **Conceitos Básicos de Controle de Versão de Software — Centralizado e Distribuído**



# Contato

- Blog: <http://danielmartins.eti.br/blog>
- Email: [root@danielmartins.eti.br](mailto:root@danielmartins.eti.br)