

Git + RStudio + GitHub via SSH

Como configurar commits e controle de versão a partir do RStudio conectado a um servidor Linux, integrado diretamente a uma conta no GitHub.

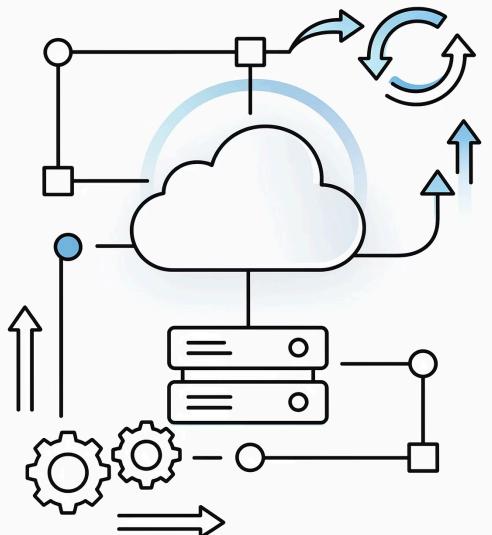
TUTORIAL

CONTROLE DE VERSÃO

Eduardo Ogasawara

eduardo.ogasawara@cefet-rj.br

<https://eic.cefet-rj.br/~eogasawara>



Por que Versionar no Servidor?

Controle de Código

Versione e acompanhe todo o código rodando remotamente no servidor Linux.

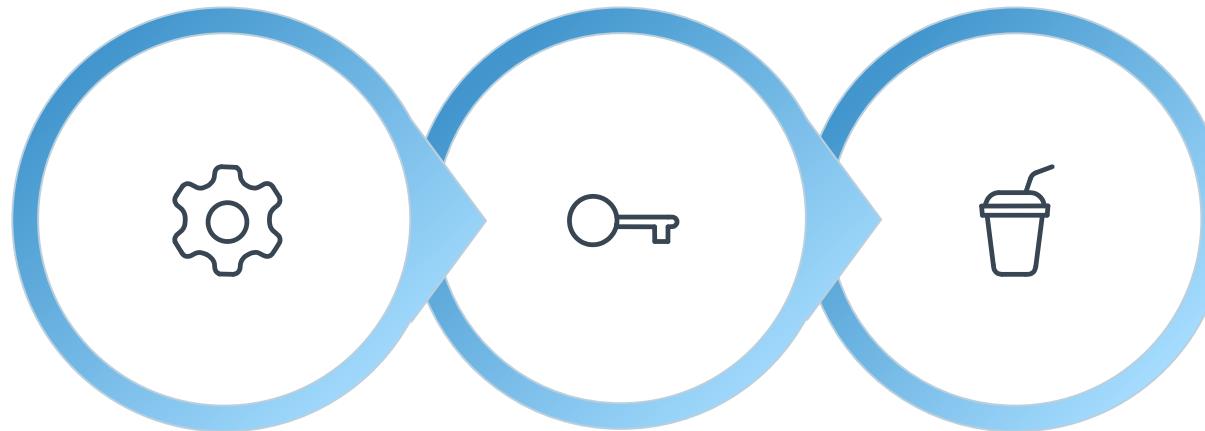
Backup Seguro

Mantenha backup de todas as operações realizadas no ambiente remoto.

Colaboração

Trabalhe em equipe de forma eficiente, com histórico completo de alterações.

Passo 1: Configurar a Chave SSH no RStudio



Acessar Git

Criar chave
RSA

Reiniciar
sessão

No RStudio, acesse as **configurações globais** e vá até a seção **Git**. O Git deve estar instalado como pré-requisito. Crie uma chave RSA — opcionalmente com passphrase para maior segurança. O sistema gera um arquivo privado e um arquivo público. Após aplicar, reinicie a sessão para confirmar a configuração.

Passo 2: Registrar a Chave Pública no GitHub

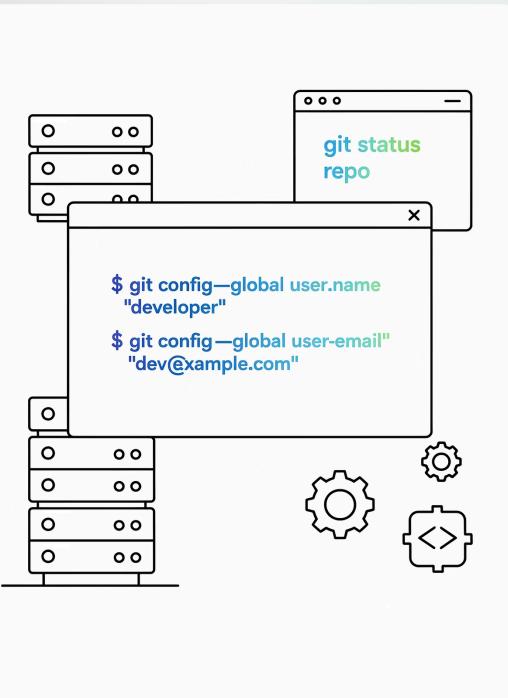
No Terminal do Servidor

Acesse a pasta `~/.ssh` e localize o arquivo `id_rsa.pub`. Copie todo o conteúdo desse arquivo público.

No GitHub

Vá em **Settings → SSH Keys**, clique em **New SSH Key**, dê um nome identificador e cole o conteúdo copiado do arquivo público. Salve a chave.

- ☐ Atenção: o RStudio pode ter usado um e-mail local ao gerar a chave. Certifique-se de associar o e-mail correto da conta GitHub.



Passo 3: Configurar Usuário Git Global

No terminal do servidor, navegue até a pasta raiz e execute os comandos de configuração global do Git para associar sua identidade aos commits:

```
git config --global user.email "seu@email.github.com"  
git config --global user.name "<usuario>"
```

Esses dados serão usados em todos os commits realizados a partir desse servidor, identificando corretamente o autor das alterações no histórico do GitHub.

Passo 4: Clonar o Repositório via SSH

No RStudio, crie um **Novo Projeto** usando controle de versão com Git. Ao informar a URL do repositório, use o formato SSH — diferente do HTTPS:

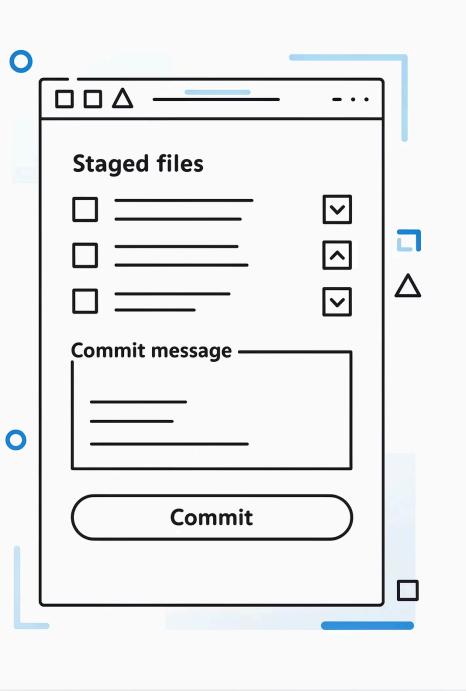
```
git@github.com:usuario/teste.git
```

Formato SSH

Substitua `https://github.com/` por `git@github.com:` e adicione `.git` ao final do endereço do repositório.

Escolha o Diretório

Selecione a pasta de destino no servidor onde o projeto será clonado. O padrão (default) funciona bem para a primeira configuração.



Passo 5: Primeiro Commit e Push

01

Criar arquivo de teste

Crie `HelloWorld.r` com `print("Hello World")`, execute e confirme o funcionamento.

02

Marcar arquivos (Stage)

Na aba Git do RStudio, marque o arquivo `HelloWorld.r` e o arquivo de projeto para staging.

03

Commit e Push

Clique em **Commit**, escreva a mensagem "*meu primeiro commit*", confirme e clique em **Push** para enviar ao GitHub.

Fluxo Contínuo de Trabalho



Abrir Projeto
File → Open Project → pasta do repositório clonado.

Editar Código
Faça alterações nos arquivos R normalmente no RStudio.

Push / Pull
Envie alterações ao GitHub ou traga atualizações com Pull.

Commit
O Git detecta mudanças; escreva a mensagem e confirme o commit.

Uma vez configurado, o projeto fica pronto para operações contínuas de versionamento. O GitHub registra cada alteração com histórico completo, visível a todos os colaboradores.



Resultado: Projeto Versionado no GitHub

Histórico Completo

Cada commit registra quem alterou, o quê e quando — rastreabilidade total do código.

Trabalho em Equipe

Colaboradores podem clonar, contribuir e sincronizar alterações com Pull e Push.

Sem Perda de Código

Backup automático no GitHub evita perdas por falhas no servidor remoto.