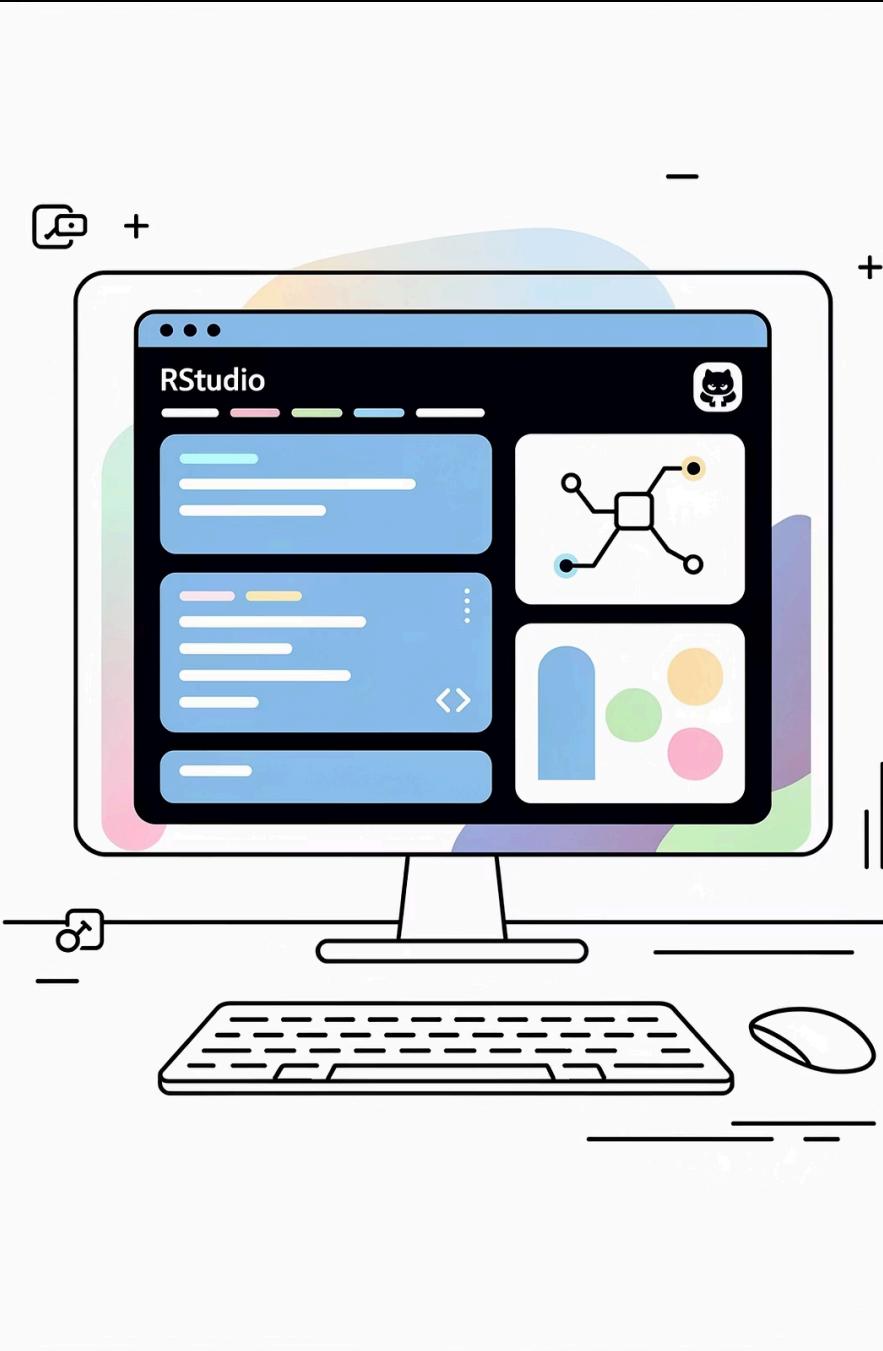


# Ambiente de Data Science do Data Analytics Lab

Guia de ferramentas e recursos disponíveis para análise de dados e desenvolvimento.

**Eduardo Ogasawara**

[eduardo.ogasawara@cefet-rj.br](mailto:eduardo.ogasawara@cefet-rj.br)  
<https://eic.cefet-rj.br/~eogasawara>

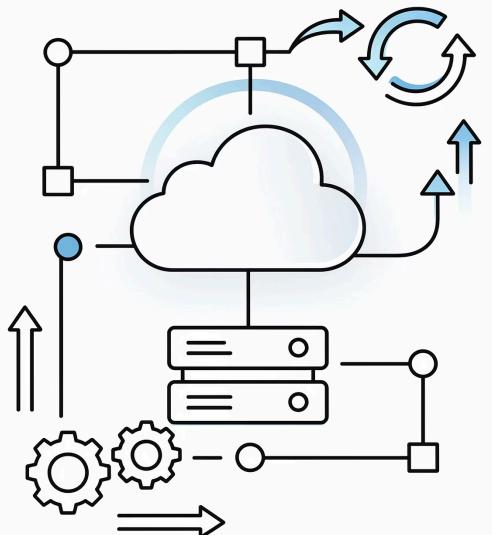


# Git + RStudio + GitHub via SSH

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



# Por que Versionar no Servidor?

## Controle de Código

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

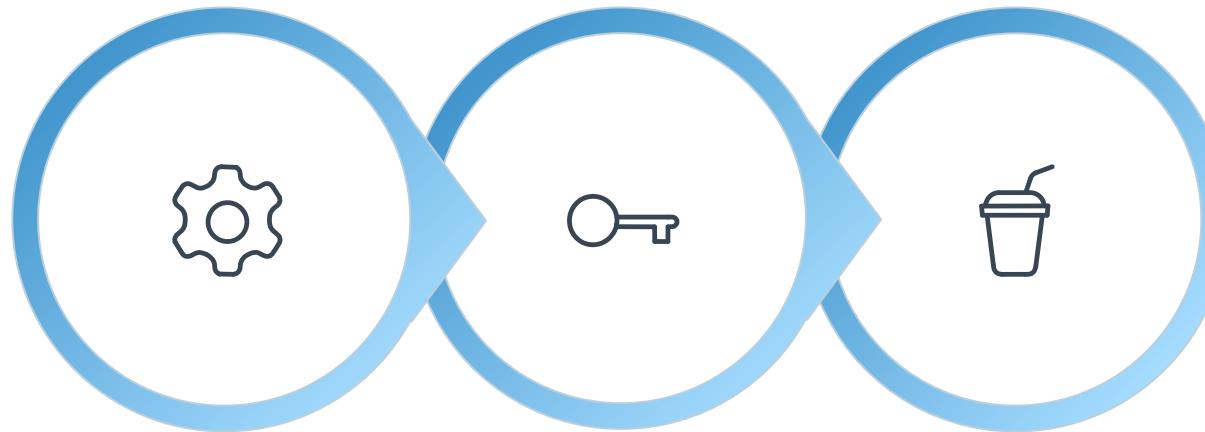
## Backup Seguro

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

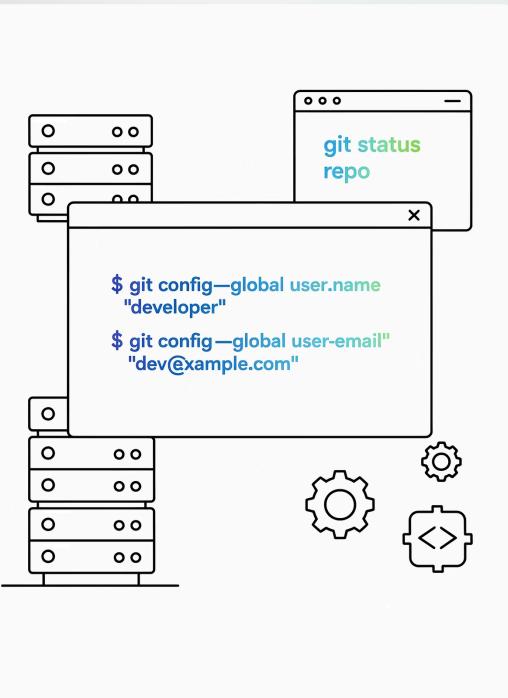
## No Terminal do Servidor

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:

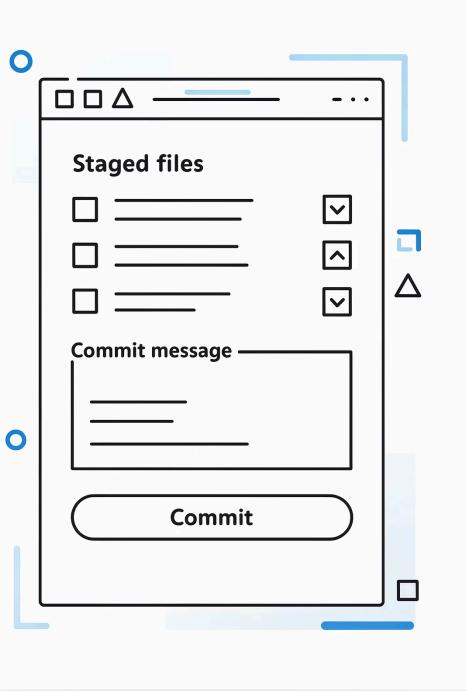
```
git@github.com:usuario/teste.git
```

## 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.

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

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.

# Pauta da Apresentação

01

## Servidor e Conta de Acesso

Informações de acesso e credenciais do servidor

02

## Acesso SSH ao Servidor

Conexão remota segura ao servidor

03

## RStudio Server

Ambiente de desenvolvimento integrado para R e Python

04

## Jupyter Notebook

Interface interativa para Python e R

05

## Code-Server

VS Code no navegador para desenvolvimento completo

# Servidor e Conta de Acesso

## Informações de Conexão

O Data Analytics Lab disponibiliza servidores dedicados para trabalhos de análise de dados e desenvolvimento de projetos em Data Science.



### Endereço do Servidor

`<servidor_dal>`



### Nome de Usuário

`<usuário>`



### Senha de Acesso

Enviada por e-mail para cada usuário

- Importante:** Guarde suas credenciais de acesso em local seguro. Você precisará delas para acessar todas as ferramentas do laboratório. Caso não tenha recebido a senha por e-mail, entre em contato com a coordenação do laboratório.



SSH

# Acesso SSH ao Servidor

Para usuários avançados que precisam de acesso direto ao terminal do servidor, disponibilizamos conexão via SSH (Secure Shell). Este método oferece controle total sobre o ambiente de linha de comando, permitindo executar scripts, gerenciar processos e realizar tarefas administrativas.

## Comando de Conexão SSH

```
ssh -p <porta> <usuário>@<servidor_dal>
```

## Estabelecendo um tunel via SSH

```
ssh -p <porta> -L <p_origem>:127.0.0.1:<p_destino> <usuário>@<servidor_dal>
```

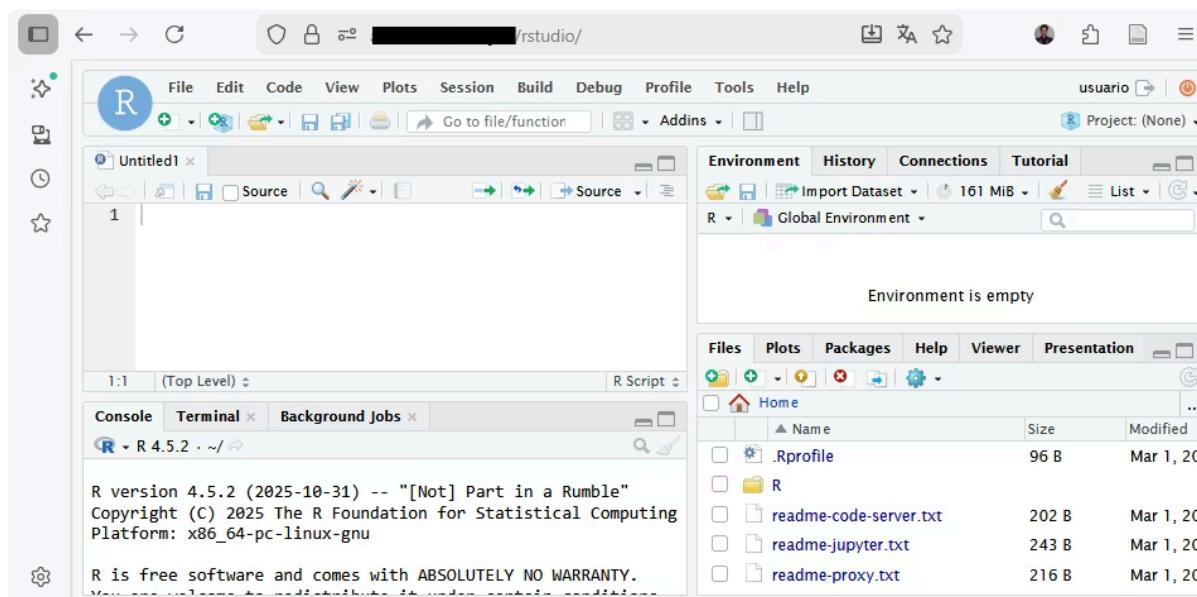
Este comando estabelece uma conexão SSH na porta <porta> e cria um túnel local da porta <p\_origem> para a porta <p\_destino> no servidor, permitindo acesso seguro aos serviços do servidor. Certifique-se de ter um cliente SSH instalado em seu sistema operacional.

# Acessando o RStudio Server

O RStudio Server é a interface web do RStudio, permitindo acesso remoto ao ambiente de desenvolvimento R diretamente pelo navegador. Esta ferramenta oferece todas as funcionalidades do RStudio Desktop, incluindo edição de código, visualização de dados, criação de gráficos e gerenciamento de pacotes.

**URL de Acesso:** [https://<servidor\\_dal>/rstudio/](https://<servidor_dal>/rstudio/)

Basta abrir esta URL em qualquer navegador moderno (Chrome, Firefox, Safari ou Edge) e fazer login com suas credenciais do laboratório.



# Interface Principal do RStudio

O ambiente RStudio oferece uma interface completa para desenvolvimento e depuração de código em R. A tela é dividida em quatro painéis principais, cada um com funcionalidades específicas para otimizar seu fluxo de trabalho.

## Painéis Principais

- **Editor de Scripts:** Escreva e edite seus códigos R
- **Console:** Execute comandos interativamente
- **Environment/History:** Visualize variáveis e histórico
- **Files/Plots/Packages:** Gerencie arquivos e visualizações

## Recursos Disponíveis

- Syntax highlighting e autocompletar
- Depurador integrado
- Gerenciador de pacotes
- Git integration

# Configuração para Integração com Python

O RStudio permite integração completa com Python através do pacote **reticulate**. Esta funcionalidade é especialmente útil para cientistas de dados que trabalham com ambas as linguagens, permitindo combinar o poder estatístico do R com as bibliotecas de machine learning do Python.



## Instalar reticulate

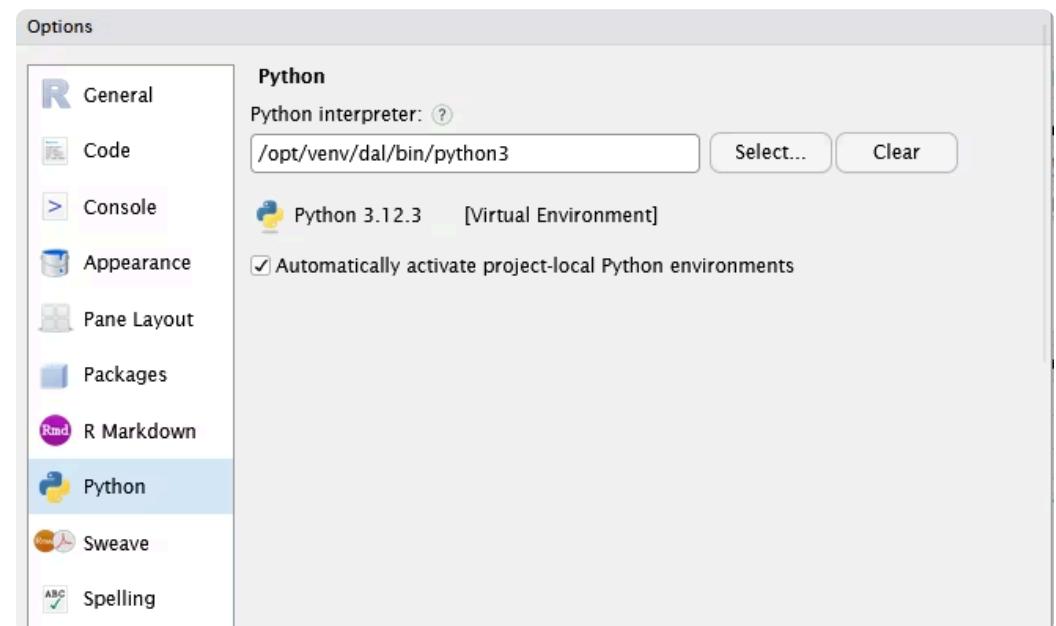
Execute: `install.packages("reticulate")`

## Configurar Python

Defina o interpretador Python desejado

## Executar Código

Use `py_run_string()` ou chunks Python



# Execução de Código Python no RStudio

Com o reticulate configurado, você pode executar código Python diretamente dentro do RStudio. O ambiente suporta tanto chunks Python em R Markdown quanto execução interativa no console.

## Vantagens da Integração

- Compartilhamento de objetos entre R e Python
- Uso de bibliotecas Python em projetos R
- Visualizações com matplotlib e seaborn
- Machine learning com scikit-learn e TensorFlow

## Casos de Uso

- Pré-processamento em Python, análise em R
- Modelos de deep learning com visualização em ggplot2
- Web scraping Python com análise estatística R

# Persistência de Sessões

Uma das grandes vantagens do RStudio Server é a persistência de sessões. Quando você inicia a execução de um script ou análise, pode fechar a janela do navegador sem interromper o processamento no servidor.

## Execute seu Código

Inicie processamentos longos ou análises complexas diretamente no RStudio

## Feche o Navegador

Seu trabalho continua sendo executado no servidor em segundo plano

## Retorne Quando Quiser

Acesse novamente e encontre sua sessão exatamente como deixou

- Dica Importante:** Esta funcionalidade é ideal para análises que levam horas ou até dias para serem concluídas. Você não precisa manter seu computador ligado - o servidor cuida de tudo!

# Reconectando à Sessão Ativa

Ao retornar ao RStudio Server após fechar o navegador, você pode facilmente reconectar-se à sua sessão em andamento. Basta fazer login novamente e acompanhar o progresso da execução do seu código.

O sistema mantém todas as variáveis carregadas na memória, o histórico de comandos e o estado completo do seu workspace. Isso significa que você pode continuar exatamente de onde parou, sem perder nenhum progresso ou ter que reexecutar análises anteriores.

## Benefícios

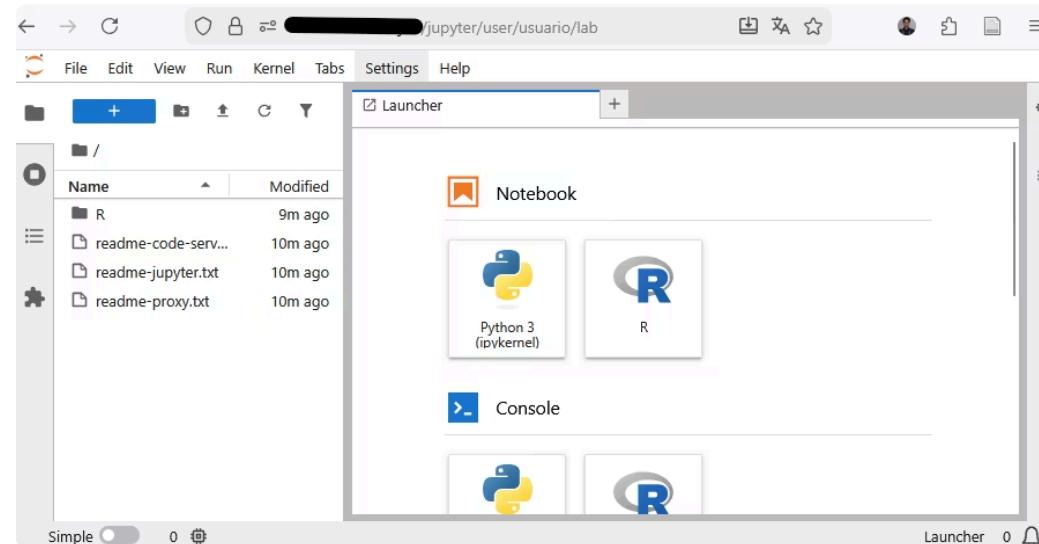
- Economia de tempo
- Maior flexibilidade
- Trabalho remoto eficiente
- Processamento contínuo

# Acessando o Jupyter Notebook

O Jupyter Notebook é uma aplicação web interativa que permite criar e compartilhar documentos contendo código executável, equações, visualizações e texto narrativo. É amplamente utilizado em ciência de dados, machine learning e computação científica.

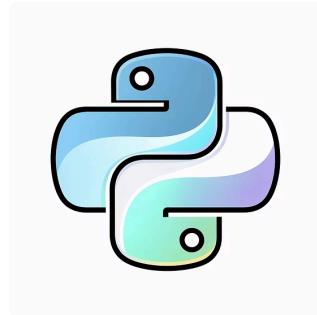
**URL de Acesso:** [https://<servidor\\_dal>/jupyter/](https://<servidor_dal>/jupyter/)

O Jupyter oferece uma interface baseada em células, onde você pode misturar código, markdown e resultados de execução em um único documento interativo. Esta abordagem é ideal para análise exploratória de dados, prototipagem rápida e criação de relatórios reproduzíveis.



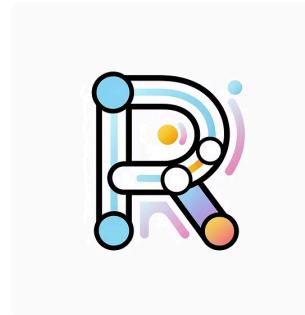
# Supporte para R e Python no Jupyter

Nosso servidor Jupyter está configurado com kernels tanto para R quanto para Python, oferecendo máxima flexibilidade na escolha da linguagem de programação. Você pode criar notebooks em qualquer uma das linguagens ou até combinar ambas em projetos diferentes.



## Python Kernel

Acesse bibliotecas como NumPy, Pandas, Matplotlib, Scikit-learn e TensorFlow para análise de dados e machine learning.



## R Kernel

Utilize pacotes R como tidyverse, ggplot2, caret e shiny para análise estatística e visualização avançada.

# Interface do Jupyter Notebook

A interface do Jupyter é intuitiva e organizada, permitindo gerenciar facilmente seus notebooks, arquivos e diretórios. Você pode criar novos notebooks, fazer upload de arquivos e organizar seus projetos de forma eficiente.

## Navegador de Arquivos

Explore e gerencie seus notebooks e datasets

## Criação de Notebooks

Inicie novos projetos com apenas um clique

## Upload de Dados

Importe arquivos CSV, JSON e outros formatos

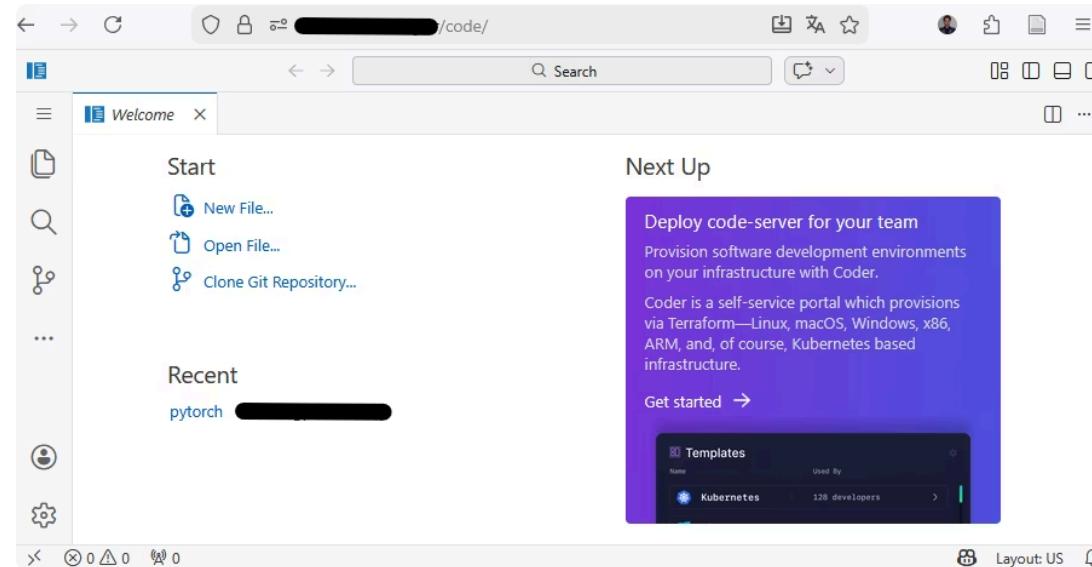
## Organização de Projetos

Crie pastas e estruture seu trabalho

# Acessando o Code-Server

O Code-Server é uma versão do Visual Studio Code que roda diretamente no seu navegador, oferecendo uma experiência de desenvolvimento profissional completa.

O acesso é feito de forma direta através da URL: [https://<servidor\\_dal>/code](https://<servidor_dal>/code)



# Interface do Code-Server

O Code-Server oferece uma interface idêntica ao VS Code desktop, com suporte completo a extensões, temas, atalhos de teclado e todas as funcionalidades que desenvolvedores esperam de um IDE moderno.



## Explorador de Arquivos

Navegue pela estrutura de diretórios do seu projeto



## Busca Global

Encontre texto em todos os arquivos rapidamente



## Controle de Versão

Integração nativa com Git e GitHub



## Depuração

Debug avançado para múltiplas linguagens



## Extensões

Marketplace completo com milhares de extensões



## Terminal Integrado

Execute comandos sem sair do editor

# Editando Arquivos no Terminal Linux

O editor vi/vim é uma ferramenta poderosa para edição de arquivos no terminal. Embora possa parecer intimidador no início, dominar comandos básicos do vi é essencial para administração de servidores Linux.

## Comandos Essenciais do Vi

- **i** - Entrar no modo de inserção
- **Esc** - Sair do modo de inserção
- **:w** - Salvar arquivo
- **:q** - Sair do editor
- **:wq** - Salvar e sair
- **:q!** - Sair sem salvar

# Recursos Completos do Code-Server

Com o Code-Server configurado e em execução, você tem acesso a um ambiente de desenvolvimento profissional completo, acessível de qualquer lugar através do navegador. Esta ferramenta é ideal para projetos complexos que exigem recursos avançados de edição e depuração.

## Desenvolvimento Multiplataforma

Supporte para Python, R, JavaScript, C++, Java e dezenas de outras linguagens

## IntelliSense Avançado

Autocompletar inteligente, sugestões de código e documentação inline

## Refatoração de Código

Ferramentas poderosas para reorganizar e otimizar seu código

# Comece Agora Mesmo!

Você tem acesso a uma infraestrutura completa para seus projetos de Data Science. Escolha a ferramenta que melhor se adapta ao seu fluxo de trabalho.



## RStudio

[https://<servidor\\_dal>/rstudio](https://<servidor_dal>/rstudio)



## Jupyter Notebook

[https://<servidor\\_dal>/jupyter](https://<servidor_dal>/jupyter)



## Code-Server

[https://<servidor\\_dal>/code](https://<servidor_dal>/code)