

Como utilizar o SSH (Secure Shell)

SSH (Secure Shell) é um protocolo de rede criptográfico usado para operar serviços de rede de forma segura sobre uma rede insegura. Foi projetado como uma substituição segura para protocolos como Telnet e outros protocolos de shell remoto não seguros.

1 Conectar-se a uma sessão SSH

Para conectar a uma sessão SSH, abra o prompt de comando no seu computador e insira

```
ssh usuario@xx.x.xx.xxx
```

Em seguida, será solicitado a sua senha cadastrada.

2 Transferir arquivos via WinSCP

O WinSCP é um cliente gráfico gratuito para Windows, utilizado para transferir arquivos entre um computador local e um servidor remoto. Ele suporta vários protocolos de rede, como SFTP (Secure File Transfer Protocol), SCP (Secure Copy Protocol), FTP (File Transfer Protocol) e WebDAV. WinSCP é amplamente utilizado para transferir arquivos de forma segura e também para gerenciar arquivos no servidor, como copiar, mover e editar diretamente. Para baixar no computador acesse o link <https://winscp.net/eng/download.php>.

Uma vez baixado no computador, ao executar o WinSCP, irá aparecer a seguinte tela:

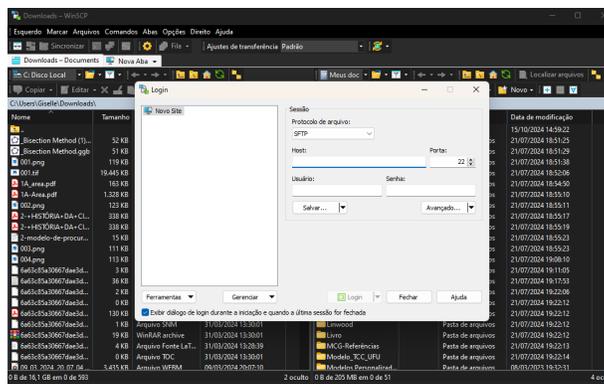


Figura 1: Tela inicial do WinSCP.

No campo “Host”DS, deve ser inserido o IP da máquina que irá acessar. Em seguida, deve ser inserido o nome de usuário e a senha para efetuar o login. Após feito o login, do lado esquerdo ficarão as pastas do usuário e do lado direito as pastas da máquina que está sendo acessada. Com isso, pode-se transferir os arquivos normalmente. Uma vez transferidos os arquivos, deve-se retornar ao terminal onde foi realizada a conexão remota via SSH.

3 O que é o screen e como utilizar

O comando *screen* no Linux é essencial para que tarefas de longa duração não sejam encerradas quando uma sessão no SSH é interrompida. Uma das grandes funcionalidades é utilizá-lo para executar códigos que irão demandar tempo para que seja finalizado. Antes de executar um código, rode no terminal:

```
screen
```

Após isso, irá aparecer a seguinte tela:



```
GNU Screen version 4.09.00 (GNU) 30-Jan-22
Copyright (c) 2018-2020 Alexander Naumov, Amadeusz Slawinski
Copyright (c) 2015-2017 Juergen Weigert, Alexander Naumov, Amadeusz Slawinski
Copyright (c) 2010-2010 Juergen Weigert, Sadrul Habib Chowdhury
Copyright (c) 2008-2009 Juergen Weigert, Michael Schroeder, Micah Cowan, Sadrul Habib Chowdhury
Copyright (c) 1993-2007 Juergen Weigert, Michael Schroeder
Copyright (c) 1987 Oliver Laumann

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General
Public License as published by the Free Software Foundation; either version 3, or (at your option) any later
version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License along with this program (see the file
COPYING); if not, see https://www.gnu.org/licenses/ or contact Free Software Foundation, Inc., 51 Franklin
Street, Fifth Floor, Boston, MA 02111-1301 USA.

Send bugreports, fixes, enhancements, t-shirts, money, beer & pizza to screen-devel@gnu.org

[Press Space for next page; Return to end.]
```

Figura 2: Terminal com o comando *screen*.

Pressione *espaço* e uma nova tela *screen* será criada. Rode o comando abaixo para executar o código que deseja

```
julia nome_do_arquivo.jl
```

Para sair da tela *screen* e deixar o programa rodando o código, digite o seguinte comando

```
ctrl+A, depois D
```

Para consultar as telas *screen* existentes utilize o seguinte comando:

```
screen -ls
```

Assim, irá aparecer a seguinte tela:

```
giselle_27@DESKTOP-T4VJ0JB:~$ screen -ls
There is a screen on:
  1094.pts-0.DESKTOP-T4VJ0JB      (11/04/24 13:18:44)      (Detached)
1 Socket in /run/screen/S-giselle_27.
giselle_27@DESKTOP-T4VJ0JB:~$
```

Figura 3: Terminal com a seção *screen* que foi iniciada.

Para entrar na sessão novamente digite o seguinte comando

```
screen -r ID_da_sessao

%Exemplo da Figura 3

screen -r 1094
```

4 Desconectar da conexão via SSH

O comando **screen** utilizado no passo anterior permite que um comando ou script continue em execução em segundo plano, mesmo depois que o usuário se desconecta da sessão SSH ou encerra o terminal. Para se desconectar, utilize o comando

```
exit
```

5 Consultar execução

Para verificar o código que foi disparado, primeiro se conecte à máquina via SSH e depois insira o comando

```
top
```

O comando “**top**” no SSH (ou em qualquer terminal Unix/Linux) é uma ferramenta interativa que exibe em tempo real informações sobre os processos em execução no sistema. Ele serve para monitorar o uso de recursos do sistema, como CPU, memória, e para visualizar quais processos estão consumindo esses recursos. Alguns comandos associados ao “**top**” são:

- **q**: Sair do top.
- **h**: Exibir ajuda.

- **k**: Matar um processo. Será solicitado o PID do processo.
- **P**: Ordenar os processos pelo uso de CPU (padrão).
- **M**: Ordenar os processos pelo uso de memória.

Caso você queira acompanhar a compilação com mais detalhes (por exemplo, exibindo as threads) use “**htop**”.