

N° 180160

Do apt instal ao seu próprio repositório: criando uma infraestrutura de pacotes Debian com reprepro

Douglas Francisco Pereira Souza
Hércules Rodrigues Freire

*Palestra apresentada no
MINIDEBCONF, 7., 2026,
Campinas. 20 slides*

A série “Comunicação Técnica” compreende trabalhos elaborados por técnicos do IPT, apresentados em eventos, publicados em revistas especializadas ou quando seu conteúdo apresentar relevância pública.

PROIBIDO A REPRODUÇÃO, APENAS PARA CONSULTA.

**“Do apt install ao seu
próprio repositório:
criando uma infraestrutura
de pacotes Debian com
reprepro”**

- Douglas Francisco Pereira de Souza
- Hércules Rodrigues Freire

Assuntos abordados

- O que é um repositório de pacotes?
- Estrutura de um repositório de pacotes.
- O que são índices do repositório.
- O que é o APT e como ele instala pacotes.
- Como utilizar a ferramenta reprepro para criar um repositório de pacotes simples.
- Como consumir o próprio repositório de pacotes.



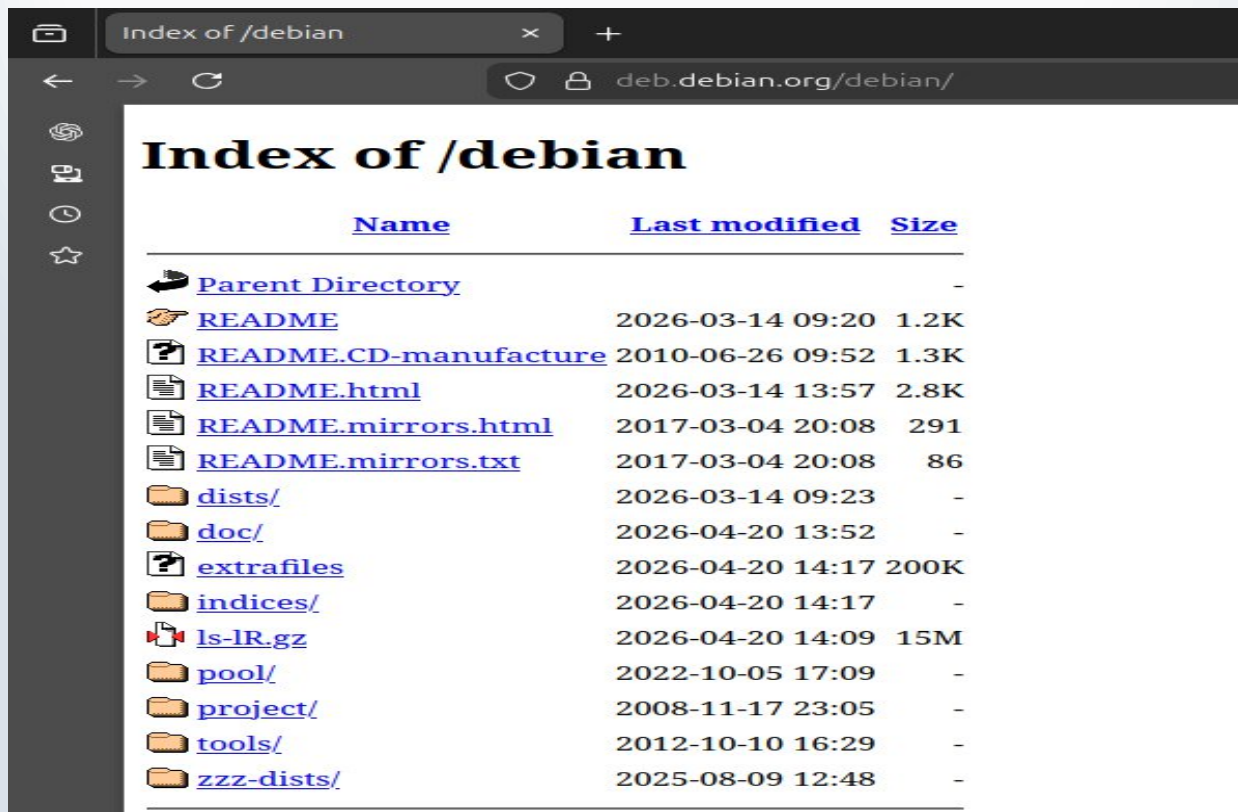
Repositório de Pacotes

- Infraestrutura que armazena, organiza e distribui softwares empacotados
- permite sua instalação e atualização automatizada por meio de gerenciadores de pacotes.
- Softwares empacotados em .deb, .rpm, .pkg.tar.xz, etc.
- Metadados com versões, dependências, descrição, assinatura digital, integridade e origem.

O que é um repositório de pacotes?

Estrutura do repositório

- **dists/**: Diretório responsável por armazenar todos os metadados dos pacotes, das distribuições Debian e suas versões.
- **pool/**: Diretório responsável por armazenar todos os pacotes .deb de todas as distribuições Debian.



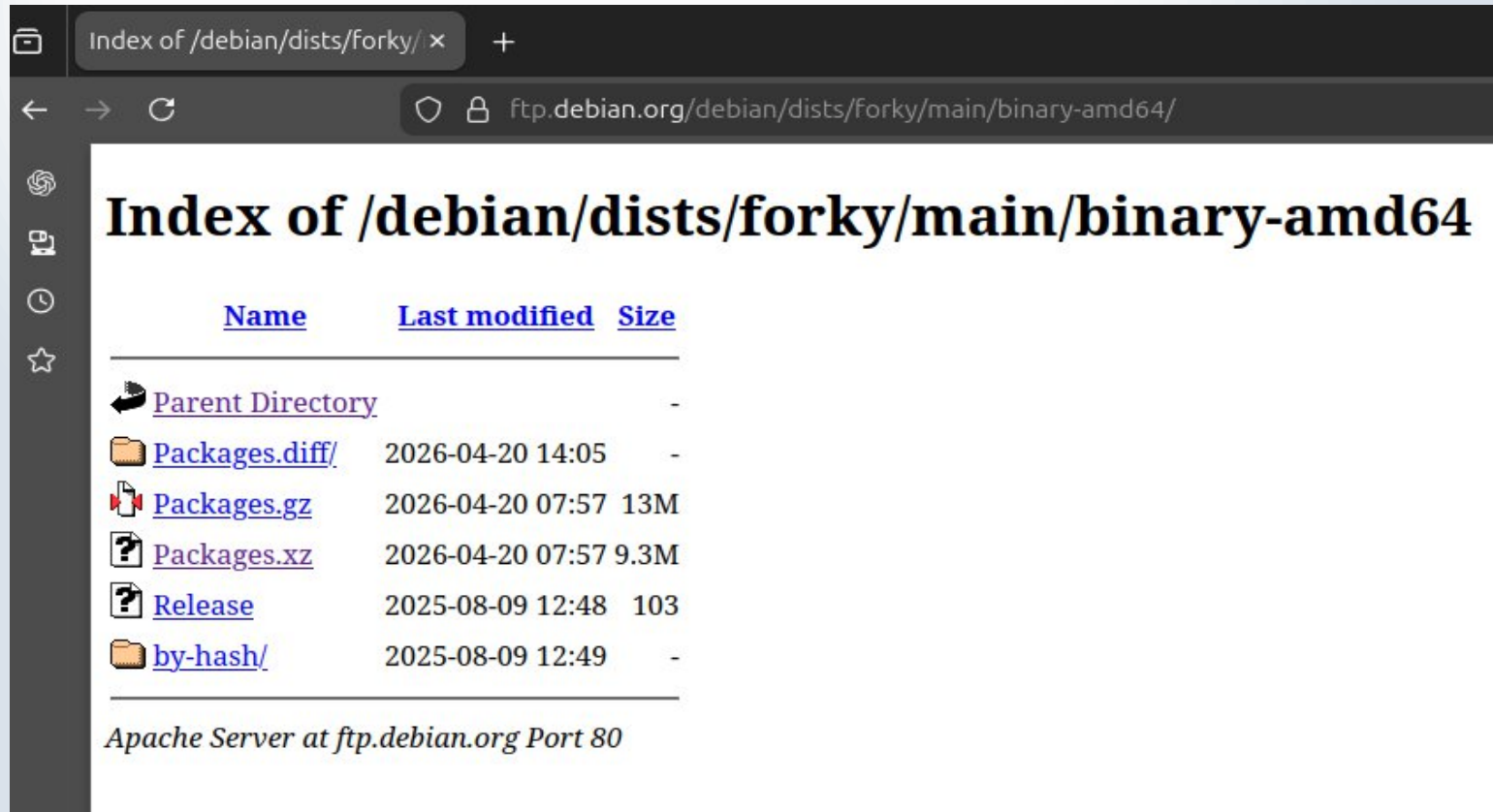
The screenshot shows a web browser window displaying the index of the Debian repository. The page title is "Index of /debian". The table below lists various files and directories with their last modified dates and sizes.

<u>Name</u>	<u>Last modified</u>	<u>Size</u>
Parent Directory		-
README	2026-03-14 09:20	1.2K
README.CD-manufacture	2010-06-26 09:52	1.3K
README.html	2026-03-14 13:57	2.8K
README.mirrors.html	2017-03-04 20:08	291
README.mirrors.txt	2017-03-04 20:08	86
dists/	2026-03-14 09:23	-
doc/	2026-04-20 13:52	-
extrafiles	2026-04-20 14:17	200K
indices/	2026-04-20 14:17	-
ls-lR.gz	2026-04-20 14:09	15M
pool/	2022-10-05 17:09	-
project/	2008-11-17 23:05	-
tools/	2012-10-10 16:29	-
zzz-dists/	2025-08-09 12:48	-

O que é um repositório de pacotes?

Repositório de Pacotes

- Packages dentro do diretório **dists/**.
- Arquivo responsável pelos metadados de todos os pacotes **.deb**



The screenshot shows a web browser window with the address bar displaying `ftp.debian.org/debian/dists/forky/main/binary-amd64/`. The page title is "Index of /debian/dists/forky/main/binary-amd64". Below the title is a table with columns for "Name", "Last modified", and "Size". The table lists several items: "Parent Directory", "Packages.diff/", "Packages.gz", "Packages.xz", "Release", and "by-hash/". The "Release" item has a size of 103. At the bottom of the page, it says "Apache Server at ftp.debian.org Port 80".

<u>Name</u>	<u>Last modified</u>	<u>Size</u>
Parent Directory		-
Packages.diff/	2026-04-20 14:05	-
Packages.gz	2026-04-20 07:57	13M
Packages.xz	2026-04-20 07:57	9.3M
Release	2025-08-09 12:48	103
by-hash/	2025-08-09 12:49	-

Apache Server at ftp.debian.org Port 80

O que é um repositório de pacotes?

```
Package: gnome-tour
Version: 49.0-2
Installed-Size: 2860
Maintainer: Debian GNOME Maintainers <pkg-gnome-maintainers@lists.aliases.debian.org>
Architecture: amd64
Depends: libadwaita-1-0 (>= 1.0.1), libc6 (>= 2.34), libcairo2 (>= 1.2.4), libgcc-s1 (>= 4.2), libgdk-pixbuf-2.0-0 (>= 2.22.0), libglib2.0-0t64 (>= 2.68.0), libgtk-4-1 (>= 4.0.0), libpango-1.0-0 (>= 1.14.0)
Description: guided tour and greeter for GNOME
Homepage: https://apps.gnome.org/Tour/
Description-md5: 9a3164d8a0d6b87e4c4654103351c3fe
Section: gnome
Priority: optional
Filename: pool/main/g/gnome-tour/gnome-tour_49.0-2_amd64.deb
Size: 736744
MD5sum: 394c17c4877fafb02369730903140751
SHA256: c501be19f4fdf8a7cb1152f22c507342945d1846c28e566db0934bb356522619
```

```
Package: gnome-tweaks
Version: 49.0-1
Installed-Size: 1145
Maintainer: Debian GNOME Maintainers <pkg-gnome-maintainers@lists.aliases.debian.org>
Architecture: all
Provides: gnome-tweak-tool
Depends: gir1.2-adw-1 (>= 1.4.0), gir1.2-gdesktopenums-3.0, gir1.2-glib-2.0, gir1.2-gnomedesktop-4.0, gir1.2-gtk-4.0 (>= 4.10.0), gir1.2-gudev-1.0 (>= 238), gir1.2-notify-0.7, gir1.2-pango-1.0, gnome-settings-daemon, gnome-shell-common (>= 3.24), gsetting-s-desktop-schemas (>= 46.0), mutter-common, python3-gi (>= 3.48.0), dconf-gsettings-backend | gsettings-backend, user-session-migration, python3:any
Suggests: gnome-shell-extension-prefs | gnome-shell-extension-manager
Description: tool to adjust advanced configuration settings for GNOME
Homepage: https://wiki.gnome.org/Apps/Tweaks
Description-md5: 941073ce1ecfd5c85f09b1ba2d0728f3
Section: gnome
Priority: optional
Filename: pool/main/g/gnome-tweaks/gnome-tweaks_49.0-1_all.deb
Size: 192788
MD5sum: 36e6718cea46267e104de12a8807798e
SHA256: f234cffe028a37328c46da06ba6cb6bab1af719251ad1f80e38f714a263814c5
```

```
Package: gnome-usage
Version: 48.0-1
Installed-Size: 835
Maintainer: Debian GNOME Maintainers <pkg-gnome-maintainers@lists.aliases.debian.org>
Architecture: amd64
Depends: vnstat, dconf-gsettings-backend | gsettings-backend, libadwaita-1-0 (>= 1.6), libc6 (>= 2.34), libcairo2 (>= 1.2.4), libgee-0.8-2 (>= 0.8.3), libglib2.0-0t64 (>= 2.80.0), libgraphene-1.0-0 (>= 1.5.4), libgtk-4-1 (>= 4.12.0), libgtop-2.0-11 (>= 2.34.0), libjson-glib-1.0-0 (>= 1.5.2), libnm0 (>= 1.44.0), libpango-1.0-0 (>= 1.14.0), libtinysparql-3.0-0 (>= 3.8.1)
Description: simple system monitor app for GNOME
Homepage: https://wiki.gnome.org/Apps/Usage
Description-md5: 26c89b55692cfb38eff700ad618b1016
Tag: uitoolkit::gtk
Section: gnome
Priority: optional
Filename: pool/main/g/gnome-usage/gnome-usage_48.0-1_amd64.deb
Size: 169884
MD5sum: 93a9e9b1d86c0d9fab90396d8d408c06
SHA256: ae8335ec4db00c58eed96de53524da156882fc30fec1fc787d854d875ff4547c
```

```
Package: gnome-user-docs
Version: 50.0-1
Installed-Size: 75792
Maintainer: Debian GNOME Maintainers <pkg-gnome-maintainers@lists.aliases.debian.org>
Architecture: all
Depends: yelp
```

A ferramenta APT

- Consumido através de ferramentas chamadas Gerenciador de Pacotes.
- Ecossistema Debian APT.
- APT (Advanced Package Tool) foi desenvolvido pelo projeto Debian em 31 de março de 1998, escrito principalmente em C, C++ e Shell Script.
- Considerado o frontend do DPKG por muitos.
- Resolve dependências entre os pacotes, verifica informações, versões disponíveis, instala e remove pacotes com comandos simples pelo terminal.

- Ferramentas usadas para gerenciar pacotes no sistema.
- APT

Funcionamento do APT

- Verifica os índices do repositório apontado para buscar os pacotes.
- Packages, Release, InRelease
- Contém md5sum, sha256, descrição, nome e endereço dos pacotes, codenome da distribuição, etc.
- Arquivos em **/etc/apt/sources.list** ou **/etc/apt/sources.list.d/<NOME>.sources**.
- apt update baixa os índices do repositório em **/var/lib/apt/lists/**
- sudo apt install <PACOTE.deb>
- Verifica índices em **/var/lib/apt/lists** e checa informações necessárias para instalar o pacote
- Após resolver as dependências, o apt baixa todos os pacotes .deb em **/var/cache/apt/archives/** e instala um por um com **dpkg**.



Como o APT funciona?

• **A ferramenta Reprepro**

- Ferramenta para gerenciar repositórios Debian.
- Cria um repositório de pacotes e servir ele local ou remotamente.
- Criar seu próprio repositório com pacotes personalizados.
- Foi desenvolvido por Bernhard R. Link por volta de 2003-2004 para facilitar a criação e gerenciamento de novos repositórios.
- Utiliza um banco de dados Berkeley DB.
- Repositório oficial da Debian

◀ Usando o reprepro

Como utilizar o REPREPRO para construir seu próprio repositório de pacotes

- Criar um diretório de trabalho com o nome a sua escolha **mkdir /repo**
- Criar um arquivo chamado **/repo/conf/distributions**
- O arquivo **/repo/conf/distributions** deve conter pelo menos os seguintes campos:

Codename: Nome que será chamada a distribuição, por exemplo forky

Components: main, non-free, non-free-firmware e/ou contrib

Architectures: Arquitetura dos pacotes, pode ser amd64, arm64, i386, entre outros

SignWith: ID da chave pública gpg para assinar os índices e pacotes

- Outros campos podem ser adicionados para maior controle da configuração, como criar regras de update, pull, logs, etc.



Usando o reprepro

Como utilizar o REPREPRO para construir seu próprio repositório de pacotes

- Gerar uma chave gpg para assinar os índices:

```
gpg --full-generate-key
```

e para verificar o ID é só rodar o comando:

```
gpg --list-keys
```



Usando o reprepro

Como utilizar o REPREPRO para construir seu próprio repositório de pacotes

- Exemplo de um arquivo `/repo/conf/distributions`:

Codename: meu-repo

Components: main non-free non-free-firmware contrib

Architectures: amd64

Update: debian-testing

SignWith: ****

Log: meu-repo-log



Usando o reprepro

Como utilizar o REPREPRO para construir seu próprio repositório de pacotes

- Necessário configurar um arquivo chamado `/repo/conf/updates`:

Name: debian-testing

- Suite: testing
 - Method: <http://deb.debian.org/debian/>
 - VerifyRelease: blindtrust
 - Architectures: amd64
-
- Regras de update (trazer pacotes de outro repositório para o nosso repositório). Com ele é possível criar um repositório baseado em Debian.



Usando o reprepro

Como utilizar o REPREPRO para construir seu próprio repositório de pacotes

- Utilizar a linha de comando do reprepro para gerenciar o repositório:
- **reprepro --basedir /repo update meu-repo**
- **reprepro --basedir /repo includedeb meu-repo pacote.deb**
- **reprepro list meu-repo**

- Exemplo:

Baixar um pacote com o apt, depois incluir esse pacote dentro da nossa distribuição

- **apt download nginx**

Usando o reprepro

1:hercules@hercules-MS-7E27: /var/lib/apt/lists ▾

root@49c5ea2907e6:/repo# ls

conf

root@49c5ea2907e6:/repo# cat conf/distributions

Codename: meu-repo

Components: main non-free non-free-firmware contrib

Architectures: amd64

Update: debian-testing

SignWith: 897E2DAFA6F1223C23B5A034F3E31530BB9A0AB6

Log: meu-repo-log

root@49c5ea2907e6:/repo# apt download nginx

Get:1 http://deb.debian.org/debian testing/main amd64 nginx amd64 1.28.3-2 [627 kB]

Fetched 627 kB in 0s (11.8 MB/s)

Warning: Download is performed unsandboxed as root as file '/repo/nginx_1.28.3-2_amd64.deb' couldn't be accessed by user '_apt'. - pkgAcquire::Run (13: Permission denied)

root@49c5ea2907e6:/repo# ls

conf nginx_1.28.3-2_amd64.deb

root@49c5ea2907e6:/repo#

1/1 ▾

+

📄

📄

1: hercules@hercules-MS-7E27: /var/lib/apt/lists ▾

```
root@49c5ea2907e6:/repo# ls
```

```
conf db nginx_1.28.3-2_amd64.deb
```

```
root@49c5ea2907e6:/repo# reprepro --basedir /repo includedeb meu-repo nginx_1.28.3-2_amd64.deb
```

```
Exporting indices...
```

```
root@49c5ea2907e6:/repo# reprepro list meu-repo
```

```
meu-repo|main|amd64: nginx 1.28.3-2
```

```
root@49c5ea2907e6:/repo#
```

Como utilizar o APT para consumir seu próprio repositório de pacotes

- Após executar o reprepro foram criados os diretórios **db**, **dists**, **logs** e **pool**

```
1/1 v + [ ] [ ]
1: hercules@hercules-MS-7E27: /var/lib/apt/lists v
root@49c5ea2907e6:/repo# ls
conf db dists logs nginx_1.28.3-2_amd64.deb pool
root@49c5ea2907e6:/repo#
```



Usando o reprepro

Servindo o repositório via http/https

- Agora temos uma distribuição chamada **meu-repo** que contém o pacote do **nginx** na **versão 1.28.3_2** para a arquitetura **amd64**.
- Disponibilizar o acesso à esse servidor por **http/https** usando apache ou o próprio nginx e adicionar o ip no `sources.list` da nossa distro que podemos instalar o pacote do nosso servidor com **apt install**.
- **ln -s /repo /var/www/html/repo**

Repositório criado, e agora?

Conclusão

- Saímos de um pacote .deb isolado e construímos um repositório que pode ser completo.
- Capaz de ser consumido por qualquer sistema baseado em Debian e customizado com nossos próprios pacotes.
- Controle total sobre distribuição de software, versionamento e atualização de sistemas.
- Podemos utilizar o gerenciador de pacotes APT para instalar e remover pacotes de nosso próprio repositório.
- Entendemos o básico do funcionamento do APT e sua comunicação com um repositório de pacotes.
- Aprendemos como criar nosso próprio repositório e como gerenciá-lo através do **reprepro**.

Conclusão

Muito obrigado!

