

# Guia Foca GNU/Linux

## Capítulo 20 - Sistema de gerenciamento de pacotes

---

Este capítulo ensina a operação básica do programa de manipulação de pacotes Debian, a instalação, remoção, consulta e checagem de arquivos `.deb`.

---

### 20.1 dpkg

O `dpkg` (Debian Package) é o programa responsável pelo gerenciamento de pacotes em sistemas Debian. Sua operação é feita em modo texto e funciona através de comandos, assim caso deseje uma ferramenta mais amigável para a seleção e instalação de pacotes, prefira o `dselect` (que é um front-end para o `dpkg`) ou o `apt` (veja [apt, Seção 20.2](#)).

`dpkg` é muito usado por utilizadores avançados da Debian e desenvolvedores para fins de instalação, manutenção e construção de pacotes.

---

#### 20.1.1 Pacotes

Pacotes Debian são programas colocados dentro de um arquivo identificados pela extensão `.deb` incluindo arquivos necessários para a instalação do programa, um sistemas de listagem/checagem de dependências, scripts de automatização para remoção parcial/total do pacote, listagem de arquivos, etc.

Um nome de pacote tem a forma `nome-versão_revisão.deb`

---

#### 20.1.2 Instalar pacotes

Use o comando: `dpkg -i [NomeDoPacote]` (ou `--install`) para instalar um pacote em seu sistema. Talvez ele peça que seja instalado algum pacote que depende para seu funcionamento. Para detalhes sobre dependências veja [Dependências, Seção 20.1.3](#). É preciso especificar o nome completo do pacote (com a versão e revisão).

---

#### 20.1.3 Dependências

Dependências são pacotes requeridos para a instalação de outro pacote. Na Debian cada pacote contém um programa com uma certa função. Por exemplo, se você tentar instalar o pacote de edição de textos `supertext` que usa o programa `sed`, você precisará verificar se o pacote `sed` está instalado em seu sistema antes de tentar instalar o `supertext`, caso contrário, o pacote `supertext` pedirá o `sed` e não funcionará corretamente. Note que o pacote `supertext` é apenas um exemplo e não existe (pelo menos até agora :-). O programa `dselect` faz o trabalho de checagem de dependências automaticamente durante a instalação dos pacotes.

A colocação de cada programa em seu próprio pacote parece ser uma dificuldade a mais para a instalação manual de um certo programa. Mas para os desenvolvedores que mantêm os mais de **18730** pacotes existentes na distribuição Debian, é um ponto fundamental, porque não é preciso esperar uma nova versão do `supertext` ser lançada para instalar a versão mais nova do pacote `sed`. Por este motivo também é uma vantagem para o utilizador.

---

#### 20.1.4 Listar pacotes existentes no sistema

Use o comando: `dpkg -l [pacote] (--list)` para isto.

Na listagem de pacotes também será mostrado o "status" de cada um na coluna da esquerda, acompanhado do nome do pacote, versão e descrição básica. Caso o nome do `[pacote]` seja omitido, todos os pacotes serão listados.

É recomendado usar "`dpkg -l|less`" para ter um melhor controle da listagem (pode ser longa dependendo da quantidade de programas instalados).

---

#### 20.1.5 Removendo pacotes do sistema

Use o comando: `dpkg -r NomeDoPacote (--remove)` para remover um pacote do sistema completamente. Somente é necessário digitar o nome e versão do pacote que deseja remover, não sendo necessário a revisão do pacote.

O comando `dpkg -r` não remove os arquivos de configuração criados pelo programa. Para uma remoção completa do programa veja [Removendo completamente um pacote, Seção 20.1.6](#).

---

#### 20.1.6 Removendo completamente um pacote

Use o comando: `dpkg -P [NomeDoPacote] (-a) (--purge)` para remover um pacote e todos os diretórios e arquivos de configuração criados. Não é necessário especificar a revisão do pacote. O comando `dpkg --purge` pode ser usado após uma remoção normal do pacote (usando `dpkg -r`).

Caso você usar diretamente o comando `dpkg --purge`, `dpkg` primeiro removerá o pacote normalmente (como explicado em [Removendo pacotes do sistema, Seção 20.1.5](#)) e após removido apagará todos os arquivos de configuração.

Caso especifique a opção `-a` (ou sua equivalente `--pending`) no lugar do nome do pacote, todos os pacotes marcados para remoção serão removidos completamente do sistema.

Note que o `dpkg --purge` somente remove arquivos de configuração conhecidos pelo pacote. Em especial, os arquivos de configuração criados para cada utilizador do sistema devem ser removidos manualmente. Seria pedir demais que o `dpkg` também conhecesse os utilizadores de nosso sistema ;-).

---

#### 20.1.7 Mostrar descrição do pacote

Use o comando: `dpkg -I NomeDoPacote (--info)` para mostrar a descrição do pacote. Entre a descrição são mostradas as dependências do pacote, pacotes sugeridos, recomendados, descrição do que o pacote faz, tamanho e número de arquivos que contém.

---

### 20.1.8 Procura de pacotes através do nome de um arquivo

Use o comando: `dpkg -S arquivo (--search)` para saber de qual pacote existente no sistema o arquivo pertence.

---

### 20.1.9 Status do pacote

Use o comando: `dpkg -s pacote (--status)` para verificar o status de um pacote em seu sistema, se esta ou não instalado, configurado, tamanho, dependências, maintainer, etc.

Se o pacote estiver instalado no sistema, o resultado será parecido com o do comando `dpkg -c [pacote] (--contents)`.

---

### 20.1.10 Procurando pacotes com problemas de instalação

A checagem de pacotes com este tipo de problema pode ser feita através do comando:

```
dpkg -C (--audit)
```

Será listado todos os pacotes com algum tipo de problema, verifique os detalhes do pacote com "`dpkg -s`" para decidir como corrigir o problema.

---

### 20.1.11 Mostrando a lista de pacotes do sistema

Use o comando:

```
dpkg --get-selections
```

para obter uma lista de seleção dos pacotes em seu sistema. A listagem é mostrada na saída padrão, que pode ser facilmente redirecionada para um arquivo usando `dpkg --get-selections >dpkg.lista`.

A listagem obtida com este comando é muito útil para repetir os pacotes usados no sistema usando o `dpkg --set-selections`.

---

### 20.1.12 Obtendo uma lista de pacotes para instalar no sistema

Use o comando:

```
dpkg --set-selections <arquivo
```

para obter a lista de pacotes que serão instalados no sistema. O uso do `dpkg --get-selections` e `dpkg --set-selections` é muito útil durante uma necessidade de reinstalação do sistema GNU/Linux ou repetir a instalação em várias máquinas sem precisar selecionar algumas dezenas entre os milhares de pacotes no `dselect`.

Após obter a lista com `dpkg --get-selections`, use `dpkg --set-selections <arquivo` e então entre no `dselect` e escolha a opção `INSTALL`, todos os pacotes obtidos via `dpkg --set-selections` serão automaticamente instalados.

---

### 20.1.13 Configurando pacotes desconfigurados

Pacotes estão desconfigurados quando, por algum motivo, a instalação do mesmo não foi concluída com sucesso. Pode ter faltado alguma dependência, acontecido algum erro de leitura do arquivo de pacote, etc. Quando um erro deste tipo acontece, os arquivos necessários pelo pacote podem ter sido instalados, mas os scripts de configuração pós-instalação não são executados.

Use o comando:

```
dpkg --configure [NomeDoPacote]
```

Para configurar um pacote. O *NomeDoPacote* não precisa conter a revisão do pacote e extensão.

---

### 20.1.14 Listando arquivos de um pacote

Use o comando: `dpkg -c arquivo` (`--contents`) para obter a listagem dos arquivos contidos no pacote. É necessário digitar o nome completo do pacote. O comando `dpkg -c` é útil para listarmos arquivos de pacotes que não estão instalados no sistema.

Para obter a listagem de arquivos de pacotes já instalados no sistema, use o comando: `dpkg -L arquivo`. É necessário digitar somente o nome do pacote (sem a revisão e extensão).

---

## 20.2 apt

O `apt` é sistema de gerenciamento de pacotes de programas que possui resolução automática de dependências entre pacotes, método fácil de instalação de pacotes, facilidade de operação, permite atualizar facilmente sua distribuição, etc. Ele funciona através de linha de comando sendo bastante fácil de usar. Mesmo assim, existem interfaces gráficas para o `apt` como o `synaptic` (modo gráfico) e o `aptitude` (modo texto) que permitem poderosas manipulações de pacotes sugeridos, etc.

O `apt` pode utilizar tanto com arquivos locais como remotos na instalação ou atualização, desta maneira é possível atualizar toda a sua distribuição Debian via `ftp` ou `http` com apenas 2 simples comandos!

É recomendável o uso do método `apt` no programa `dselect` pois ele permite a ordem correta de instalação de pacotes e checagem e resolução de dependências, etc. Devido a sua facilidade de operação, o `apt` é o método preferido para os utilizadores manipularem pacotes da Debian.

O `apt` é exclusivo da distribuição `Debian` e distribuições baseadas nela e tem por objetivo tornar a manipulação de pacotes poderosa por qualquer pessoa e tem dezenas de opções que podem ser usadas em sua execução ou configuradas no arquivo `/etc/apt/apt.conf`. Explicarei aqui como fazer as ações básicas com o `apt`, portanto se desejar maiores detalhes sobre suas opções, veja a página de manual `apt-get`.

---

### 20.2.1 O arquivo `/etc/apt/sources.list`

Este arquivo contém os locais onde o `apt` encontrará os pacotes, a distribuição que será verificada (`stable`, `testing`, `unstable`, `Woody`, `Sarge`) e a seção que será copiada (`main`, `non-free`, `contrib`, `non-US`).

`Woody`(`Debian 3.0`) e `Sarge`(`Debian 3.1`) são os nomes das versões enquanto `stable` e `unstable` são links para as versões *estável* e *testing* respectivamente. Se desejar usar sempre uma distribuição estável (como a `Woody`), modifique o arquivo `sources.list` e coloque `Woody` como distribuição. Caso você desejar estar sempre atualizado mas é uma pessoa cuidadosa e deseja ter sempre a última distribuição estável da `Debian`, coloque `stable` como versão. Assim que a nova versão for lançada, os links que apontam de `stable` para `Woody` serão alterados apontando para `Sarge` e você terá seu sistema atualizado. Abaixo um exemplo simples de arquivo `/etc/apt/sources.list` com explicação das seções:

```
deb http://www.debian.org/debian stable main contrib non-free
```

```
deb http://nonus.debian.org/debian-non-US stable non-US
```

Você pode interpretar cada parte da seguinte maneira:

- `deb` - Identifica um pacote da `Debian`. A palavra `deb-src` identifica o código fonte.
- `http://www.debian.org/debian` - Método de acesso aos arquivos da `Debian`, site e diretório principal. O caminho pode ser `http://`, `ftp://`, `file://`.
- `stable` - Local onde serão procurados arquivos para atualização. Você pode tanto usar o nome de sua distribuição (`Woody`, `Sarge`) ou sua classificação (`stable`, `testing` ou `unstable`). Note que `unstable` é recomendada somente para desenvolvedores, máquinas de testes e se você tem conhecimentos para corrigir problemas. Nunca utilize `unstable` em ambientes de produção ou servidores críticos, use a `stable`.
- `main contrib non-us` - Seções que serão verificadas no site remoto.

Note que tudo especificado após o nome da distribuição será interpretado como sendo as seções dos arquivos (`main`, `non-free`, `contrib`, `non-US`). As linhas são processadas na ordem que estão no arquivo, então é recomendável colocar as linhas que fazem referência a pacotes locais primeiro e mirrors mais perto de você para ter um melhor aproveitamento de banda. O caminho percorrido pelo `apt` para chegar aos arquivos será o seguinte:

```
http://www.debian.org/debian/dists/stable/main/binary-i386
```

```
http://www.debian.org/debian/dists/stable/non-free/binary-i386
```

```
http://www.debian.org/debian/dists/stable/contrib/binary-i386
```

Você notou que o diretório `dists` foi adicionado entre `http://www.debian.org/debian` e `stable`, enquanto as seções `main`, `non-free` e `contrib` são processadas separadamente e finalizando com o

caminho `binary-[arquitetura]`, onde *[arquitetura]* pode ser *i386*, *alpha*, *sparc*, *powerpc*, *arm*, etc. dependendo do seu sistema. Entendendo isto, você poderá manipular o arquivo `sources.list` facilmente.

**OBS:** Caso tenha mais de uma linha em seu arquivo `sources.list` de onde um pacote pode ser instalado, ele será baixado da primeira encontrada no arquivo. É recomendável colocar primeiro repositórios locais ou mais perto de você, como recomendado nesta seção.

---

#### **20.2.1.1 Endereços de servidores e mirrors nacionais da Debian**

Segue abaixo uma relação de servidores que podem ser colocados em seu arquivo `sources.list`:

Endereço Diretório Principal

-----

`ftp://ftp.debian.org.br /debian`

`ftp://ftp.br.debian.org /debian`

`ftp://ftp.debian.org /debian`

`ftp://download.sourceforge.net /debian`

`ftp://ftp.quimica.ufpr.br /debian`

`ftp://download.unesp.br /linux/debian`

---

#### **20.2.1.2 Um modelo de arquivo `sources.list`**

Você pode copiar o modelo do `sources.list` abaixo para ser usado em sua distribuição `Stable` ou personalizá-lo modificando a distribuição utilizada e servidores:

`# Arquivos principais da stable`

`deb ftp://ftp.debian.org.br/debian stable main non-free contrib`

`# Non-US da Stable`

`deb ftp://ftp.debian.org.br/debian-non-US stable/non-US main non-free contrib`

`# Atualizações propostas para Stable main e non-US`

`deb ftp://ftp.debian.org.br/debian dists/proposed-updates/`

`deb ftp://ftp.debian.org.br/debian-non-US dists/proposed-updates/`

`# Atualizações de segurança da Stable`

`deb ftp://nonus.debian.org/debian-security stable/updates main`

```
# Ximian é um conjunto de pacotes atualizados freqüentemente e compatíveis  
  
# com a distribuição Debian. Entre estes programas estão o Gimp 1.2 e outros  
  
# mais atuais e compatíveis com a Debian. Para usa-los inclua a seguinte linha no  
  
# seu sources.list  
  
# deb ftp://spidermonkey.ximian.com/pub/red-carpet/binary/debian-22-i386/ ./  
  
# Kde 1 e 2  
  
# deb ftp://kde.tdyc.com/pub/kde/debian woody main crypto optional qtlapps
```

---

### 20.2.2 O arquivo `/etc/apt/apt.conf`

Você pode especificar opções neste arquivo que modificarão o comportamento do programa `apt` durante a manipulação de pacotes (ao invés de especificar na linha de comando). Se estiver satisfeito com o funcionamento do programa `apt`, não é necessário modifica-lo. Para detalhes sobre o formato do arquivo, veja a página de manual do `apt.conf`. Na página de manual do `apt-get` são feitas referências a parâmetros que podem ser especificados neste arquivo ao invés da linha de comando.

---

### 20.2.3 Copiando a lista de pacotes disponíveis

O `apt` utiliza uma lista de pacotes para verificar se os pacotes existentes no sistema precisam ou não ser atualizados. A lista mais nova de pacotes é copiada através do comando `apt-get update`. Este comando pode ser usado com alguma freqüência se estiver usando a distribuição `stable` e sempre se estiver usando a `unstable` (os pacotes são modificados com muita freqüência). Sempre utilize o `apt-get update` antes de atualizar toda a distribuição.

---

### 20.2.4 Utilizando CDs oficiais/não-oficiais/terceiros com o `apt`

Para usar CDs da Debian ou de programas de terceiros, use o seguinte comando com cada um dos CDs que possui:

```
apt-cdrom add
```

Este comando adicionará automaticamente uma linha para cada CD no arquivo `/etc/apt/sources.list` e atualizará a lista de pacotes em `/var/state/apt/lists`. Por padrão, a unidade acessada através de `/cdrom` é usada. Use a opção `-d /dev/scd?` para especificar um outra unidade de CDs (veja [Identificação de discos e partições em sistemas Linux, Seção 5.12](#) para detalhes sobre essa identificação).

Durante a instalação de um novo programa, o `apt` pede que o CD correspondente seja inserido na unidade e pressionado <Enter> para continuar. O método acesso do `apt` através de CDs é inteligente o bastante para

instalar todos os pacotes necessários daquele CD, instalar os pacotes do próximo CD e iniciar a configuração após instalar todos os pacotes necessários.

Observação: - CDs de terceiros ou contendo programas adicionais também podem ser usados com o comando "apt-cdrom add".

---

## 20.2.5 Instalando novos pacotes

Use o comando `apt-get install [pacotes]` para instalar novos pacotes em sua distribuição. Podem ser instalados mais de um pacotes ao mesmo tempo separando os nomes por espaços. Somente é preciso especificar o nome do pacote (sem a versão e revisão).

Se preciso, o `apt` instalará automaticamente as dependências necessárias para o funcionamento correto do pacote. Quando pacotes além do solicitado pelo utilizador são requeridos para a instalação, o `apt` mostrará o espaço total que será usado no disco e perguntará ao utilizador se ele deseja continuar. Após a instalação, o pacote será automaticamente configurado pelo `dpkg` para ser executado corretamente em seu sistema.

---

## 20.2.6 Removendo pacotes instalado

Use o comando `apt-get remove [pacotes]` para remover completamente um pacote do sistema. Podem ser removidos mais de um pacote ao mesmo tempo separando os nomes dos pacotes com espaços. O `apt-get remove` remove completamente o pacote mas mantém os arquivos de configuração, exceto se for adicionada a opção `--purge`.

É preciso especificar somente o nome do pacote (sem a versão e revisão).

---

## 20.2.7 Atualizando sua distribuição

O `apt` tem uma grande característica: Atualizar toda a sua distribuição de uma forma inteligente e segura.

O `apt` lê a listagem de pacotes disponíveis no servidor remoto, verifica quais estão instalados e suas versões, caso a versão do pacote seja mais nova que a já instalada em seu sistema, o pacote será imediatamente atualizado.

A cópia dos arquivos pelo `apt` pode ser feita via FTP, HTTP ou através de uma cópia local dos arquivos no disco rígido (um *mirror* local). Em nenhuma circunstância os pacotes existentes em seu sistema serão removidos ou sua configuração apagada durante um `upgrade` na distribuição.

Os arquivos de configuração em `/etc` que foram modificados são identificados e podem ser mantidos ou substituídos por versões existentes nos pacotes que estão sendo instalado, esta escolha é feita por você. Se estiver atualizando a Debian Potato (2.2) para Woody (3.0) (ou versão superior), execute os seguintes comandos antes de iniciar a atualização:

```
export LANG=C
```

```
export LC_ALL=C
```

```
export LC_MESSAGES=C
```



para retornar as variáveis de localização ao valor padrão (inglês). Isto é necessário por causa de modificações no sistema de locales, e o excesso de mensagens de erro do perl causaram alguns problemas em meus testes.

Após isto, a atualização da distribuição Debian pode ser feita através de dois simples comandos:

```
apt-get update #Para atualizar a lista de pacotes (obrigatório)
```

```
apt-get -f dist-upgrade #Para atualizar a distribuição
```

A opção `-f` faz com que o `apt` verifique e corrija automaticamente problemas de dependências entre pacotes.

Recomendo executar o comando `apt-get -f --dry-run dist-upgrade | less` para ver o que vai acontecer sem atualizar a distribuição, se tudo ocorrer bem, retire o `--dry-run` e vá em frente.

A distribuição usada na atualização pode ser:

- Para a mesma versão que utiliza - Para quem deseja manter os pacotes sempre atualizados entre revisões, copiar pacotes que contém correções para falhas de segurança (veja a página web em <http://www.debian.org/> para acompanhar o boletim de segurança).
- Para uma distribuição *stable* - Mesmo que o acima, mas quando uma nova distribuição for lançada, o link simbólico de *stable* será apontado para próxima distribuição, atualizando instantaneamente seu sistema.
- Para a distribuição *testing* - Atualiza para a futura distribuição Debian que será lançada, é como a *unstable*, mas seus pacotes passam por um período de testes de 2 semanas na *unstable* antes de serem copiados para esta.
- *unstable* - Versão em desenvolvimento, recomendada somente para desenvolvedores ou utilizadores que conhecem a fundo o sistema GNU/Linux e saibam resolver eventuais problemas que apareçam.

A *unstable* é uma distribuição em constante desenvolvimento e podem haver pacotes problemáticos ou com falhas de segurança. Após o período de desenvolvimento, a distribuição *unstable* se tornará *frozen*.

- *frozen* - Versão congelada, nenhum pacote novo é aceito e somente são feitas correções de falhas. Após todas as falhas estarem corrigidas, a distribuição *frozen* se tornará *stable*

A distribuição que será usada na atualização pode ser especificada no arquivo `/etc/apt/sources.list` (veja a seção correspondente acima). Caso o método de atualização usado seja via HTTP ou FTP, será necessário usar o comando `apt-get clean` para remover os pacotes copiados para seu sistema (para detalhes veja a seção seguinte).

---

## 20.2.8 Removendo pacotes baixados pelo apt

Use o comando `apt-get clean` para apagar qualquer arquivo baixado durante uma atualização ou instalação de arquivos com o `apt`. Os arquivos baixados residem em `/var/cache/apt/archives` (download completo) e `/var/cache/apt/archives/partial` (arquivos sendo baixados - parciais).

Este local de armazenamento é especialmente usado com o método `http` e `ftp` para armazenamento de arquivos durante o download para instalação (todos os arquivos são primeiro copiados para serem instalados e configurados).

O `apt-get clean` é automaticamente executado caso seja usado o método de acesso `apt` do `dselect`.

---

## 20.2.9 Procurando por pacotes através da descrição

O utilitário `apt-cache` pode ser usado para esta função. Ele também possui outras utilidades interessante para a procura e manipulação da lista de pacotes.

Por exemplo, o comando `apt-cache search clock` mostrará todos os pacotes que possuem a palavra *clock* na descrição do pacote.

---

## 20.2.10 Procurando um pacote que contém determinado arquivo

Suponha que algum programa esteja lhe pedindo o arquivo `perlcc` e você não tem a mínima idéia de que pacote instalar no seu sistema. O utilitário `auto-apt` pode resolver esta situação. Primeiro instale o pacote `auto-apt` e execute o comando `auto-apt update` para que ele copie o arquivo `Contents-i386.gz` que será usado na busca desses dados.

Agora, basta executar o comando:

```
auto-apt search perlcc
```

para que ele retorne o resultado:

```
usr/bin/perlcc interpreters/perl
```

O pacote que contém este arquivo é o `perl` e se encontra na seção `interpreters` dos arquivos da Debian. Para uma pesquisa que mostra mais resultados (como `auto-apt search a2ps`), é interessante usar o `grep` para filtrar a saída:

```
auto-apt search a2ps|grep bin/
```

```
usr/bin/psmandup text/a2ps
```

```
usr/bin/pdiff text/a2ps
```

```
usr/bin/psset text/a2ps
```

```
usr/bin/composeglyphs text/a2ps
```

```
usr/bin/a2psj text/a2ps-perl-ja
```

```
usr/bin/a2ps text/a2ps
```

```
usr/bin/fixps text/a2ps
```

```
usr/bin/ogonkify text/a2ps
```

```
usr/bin/fixnt text/a2ps
```

```
usr/bin/card text/a2ps
```

```
usr/bin/texi2dvi4a2ps text/a2ps
```

Serão mostrados somente os binários, diretórios de documentação, manpages, etc. não serão mostradas.

---

### 20.2.11 Modos eficazes de compilação do código fonte para a Debian

O Debian como qualquer distribuição de Linux, possui o diretório `/usr/local` que segundo a FHS é o local apropriado para colocação de programas que não fazem parte da distribuição, que seria no caso o de fontes compilados manualmente. Um dos grandes trabalhos de quem pega o código fonte para compilação é a instalação de bibliotecas de desenvolvimento para a compilação ocorrer com sucesso.

O `auto-apt` facilita magicamente o processo de compilação da seguinte forma: durante o passo `./configure` no momento que é pedida uma bibliotecas, dependência, etc. o `auto-apt` para o processo, busca por pacotes no repositório da Debian, pergunta qual pacote será instalado (caso tenha mais de uma opção), instala e retorna o `./configure` do ponto onde havia parado.

Para fazer isso, execute o comando:

```
auto-apt run ./configure
```

E ele se encarregará do resto :-)

---

### 20.2.12 Verificando pacotes corrompidos

Use o comando `apt-get check` para verificar arquivos corrompidos. A correção é feita automaticamente. A lista de pacotes também é atualizada quando utiliza este comando.

---

### 20.2.13 Corrigindo problemas de dependências e outros erros

Use o comando `apt-get -f install` (sem o nome do pacote) para que o `apt-get` verifique e corrija problemas com dependências de pacotes e outros problemas conhecidos.