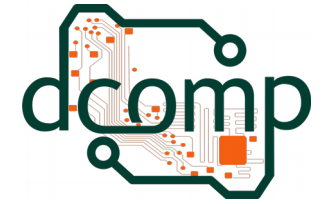




Universidade Federal do Espírito Santo  
Centro de Ciências Agrárias – CCA UFES  
Departamento de Computação



# Sistemas Operacionais Instalação e Configuração

**Sistemas de Software Livre**

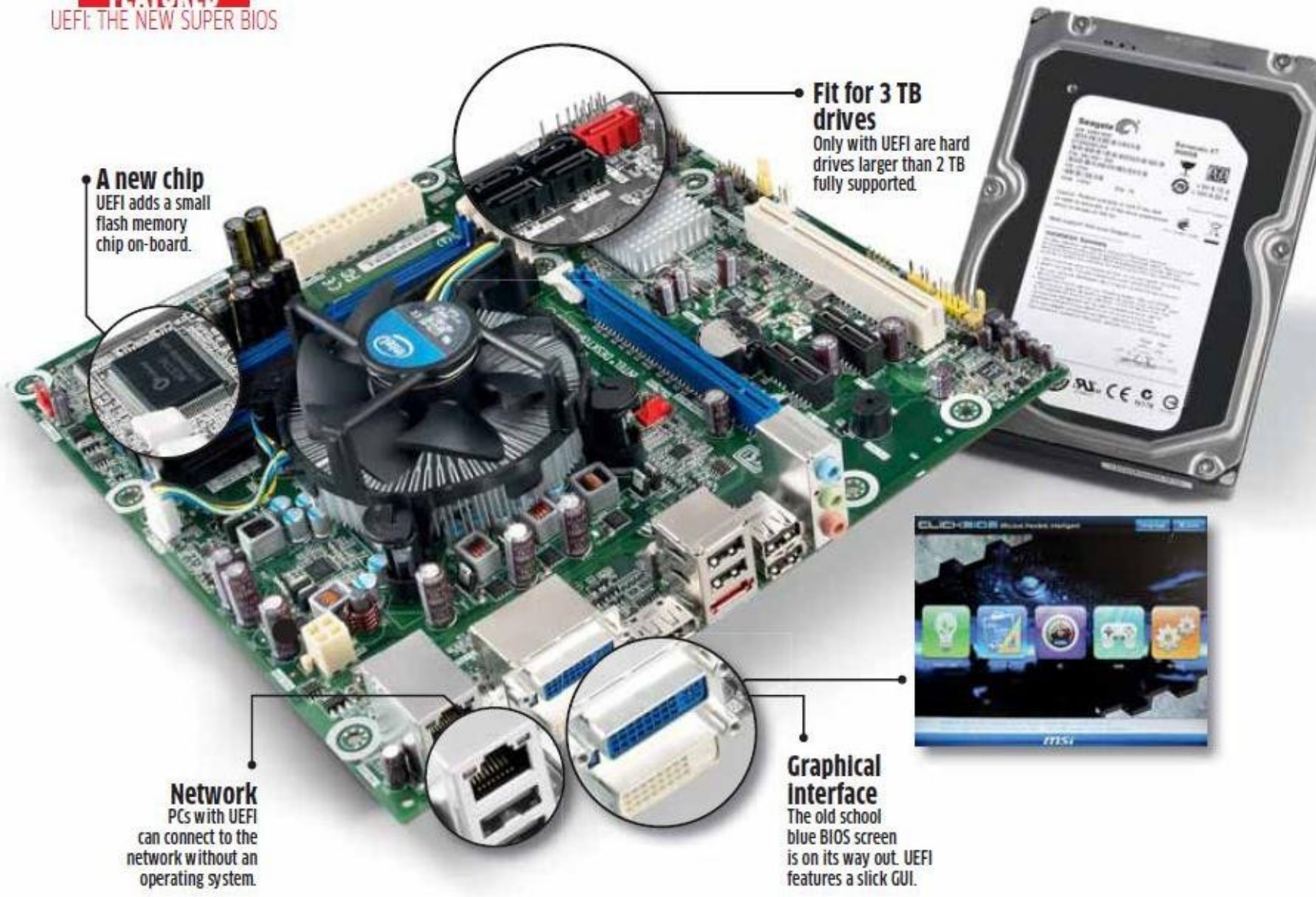
Site: <http://jeiks.net>

E-mail: [jacsonrcsilva@gmail.com](mailto:jacsonrcsilva@gmail.com)

# Conteúdo

- Características de hardware
- BIOS/MBR e UEFI/GPT
- Particionamento
- Estrutura de diretórios
- Processo de instalação

**FEATURED**  
UEFI: THE NEW SUPER BIOS



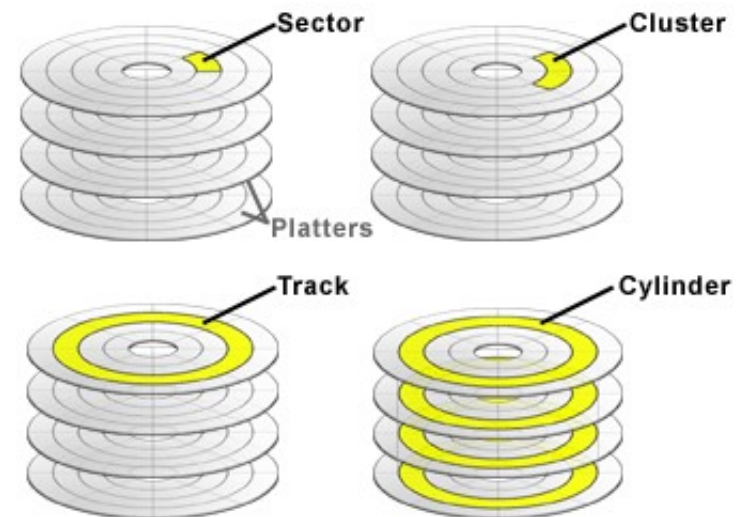
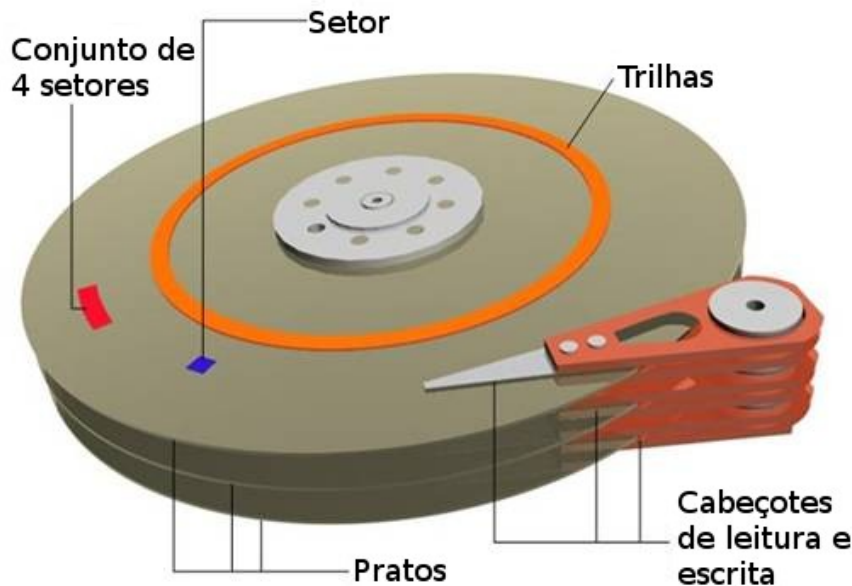
<https://lakkireddymadhu.files.wordpress.com/2011/07/uefi-12.jpg>



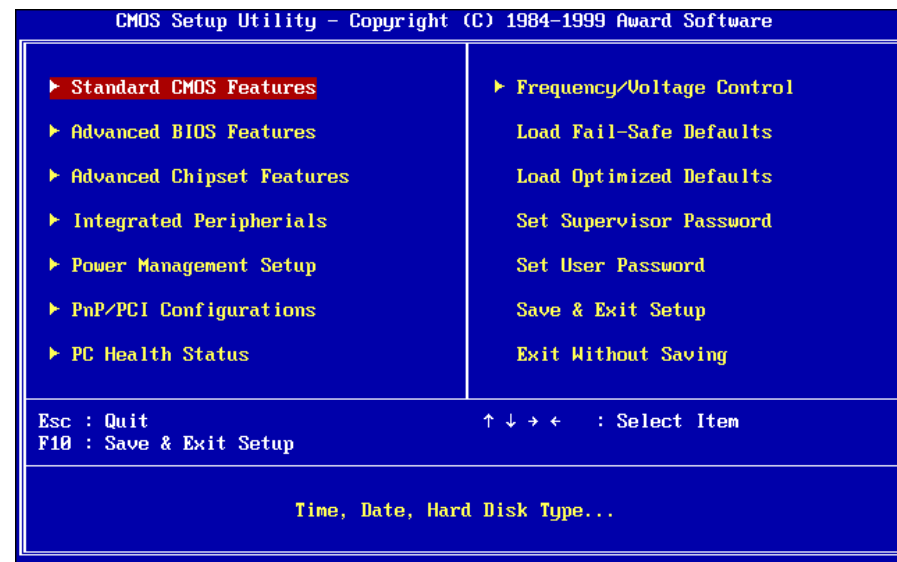
<https://www.itworld.com/article/2826910/hardware/goodbye-bios--hello-uefi.html>

# Geometria dos discos

- Discos são formados por pratos.
- Pratos são divididos em trilhas e setores.
- Setores possuem 512 bytes e populam os cilindros.



# BIOS e MBR



- BIOS: *Basic Input Output System*
  - Criado para o IBM PC,  
lançado em 12 de Agosto de 1981.
  - Código de ROM em Assembly, projetado por Dave Bradley.
  - Trabalha em conjunto com o MBR.
  
- MBR: *Master Boot Record*
  - Idealizado para ser utilizado com MSDOS.
  - Ocupa um setor de disco (512 bytes apenas).
  - Possui uma área para gestores de boot (ex: GRUB).
  - Utiliza uma tabela que pode controlar até quatro partições.
  - Limita o disco a  $2^{32}$  setores, ou seja 2.2 TB

# MBR

- Tamanho da MBR:  
 $446 + 16 \times 4 + 2 = 512$  bytes

Área para o  
gestor de boot  
(446 bytes)

Controle da 1ª Partição (16 bytes)

Controle da 2ª Partição (16 bytes)

Controle da 3ª Partição (16 bytes)

Controle da 4ª Partição (16 bytes)

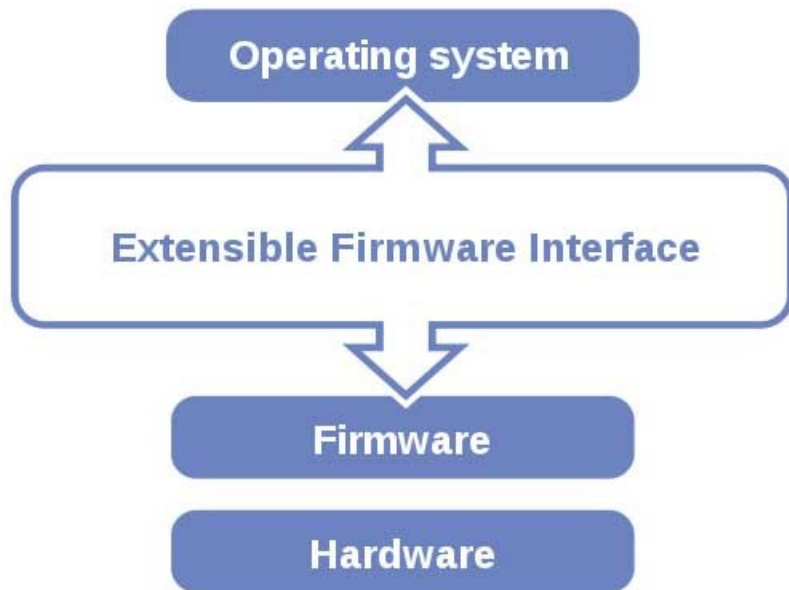
Ass. (2 bytes)

# BIOS e MBR

- O MBR só controla quatro partições primárias.
- A saída para expansão foi criar a chamada partição estendida, que permite o uso de partições lógicas.
- A partição estendida só controla a primeira lógica.
- Cada partição lógica possui o seu próprio controle e um encadeamento com a próxima lógica (EBR).
- Se uma partição lógica for perdida, todas as outras também o serão.
- Evite o uso de partições lógicas.
- O padrão BIOS começou a desaparecer em 2009.

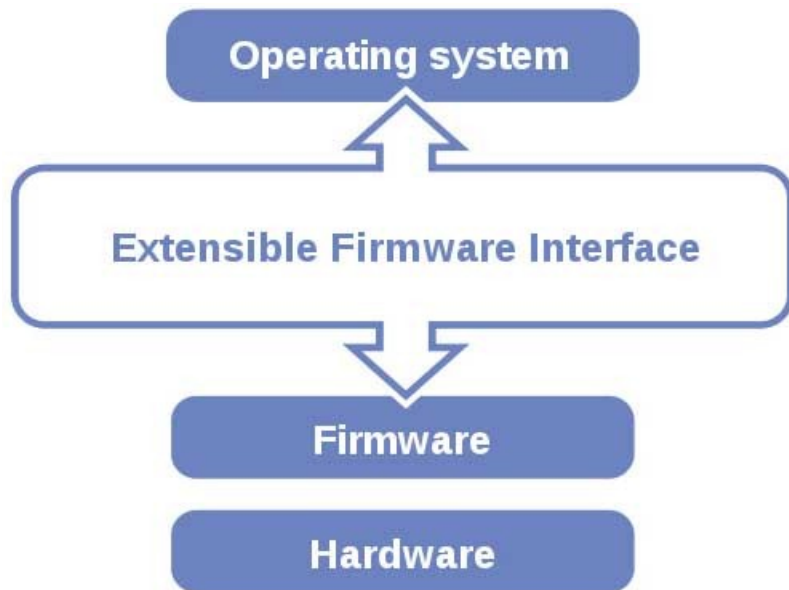


# UEFI



- *Unified Extensible Firmware Interface.*
- Criada pela Intel na década de 1990 para substituir a BIOS.
- Em 2005 foi criada a aliança UEFI.
- Participantes: Intel, AMD, Apple, Dell, HP, IBM, Lenovo e outros.
- Suporte a x86, x86\_64, ARM e IA64.
- Utilização de tabela de particionamento GPT (no lugar do MBR).
- O UEFI é um firmware feito em C, o que o torna extremamente flexível.
- O site do UEFI é <<http://www.uefi.org>>

# UEFI



- A UEFI é uma especificação de *firmware* e não uma implementação.
  - Cada um pode implementar o que necessitar.
- O boot com UEFI é muito mais rápido do que com BIOS,
  - pois é um padrão moderno com técnicas avançadas.
  - Pode ser até 6x mais rápido.
- A UEFI pode prover mouse, conexão de rede, design de tela personalizado e moderno, etc.

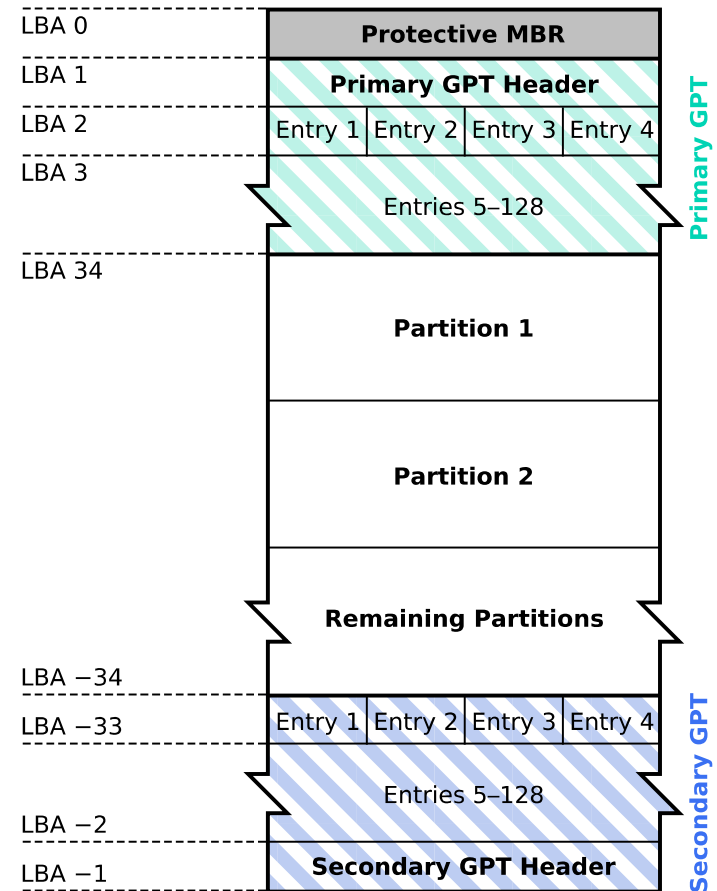
# Partições GPT

- *GUID Partition Table* (GPT).
  - GUID:
    - Globally Unique Identifier.
    - São utilizados 128 bits (16 bytes) para identificar um dispositivo unicamente no mundo.
    - Apesar de raro, há casos de colisão de GUID.
  - Trabalha com LBA, em vez de CHS (que possui limite de 8 GB).
  - Endereça até  $2^{64}$  setores.
- Para conhecimento:
  - Byte:  $2^0$ ;
  - KiloByte:  $2^{10}$ ;
  - MegaByte:  $2^{20}$ ;
  - GigaByte:  $2^{30}$ ;
  - TeraByte:  $2^{40}$ ;
  - PetaByte:  $2^{50}$ ;
  - ExaByte:  $2^{60}$ ;
  - ZettaByte:  $2^{70}$ ;

# Partições GPT

- Ocupa um total de 34 setores para a estrutura e mais 34 como backup.
- O backup é estabelecido no fim do disco.
- Dos 34 setores, 32 realizam o controle de partições.
- Permite até 128 partições de disco.
- Utiliza o conceito de *Protective MBR*, contra erros de sistemas.

## GUID Partition Table Scheme



[https://commons.wikimedia.org/wiki/File:GUID\\_Partition\\_Table\\_Scheme.svg](https://commons.wikimedia.org/wiki/File:GUID_Partition_Table_Scheme.svg)

# Comportamento com SOs

- O MacOS suporta GPT.
  - No entanto, o boot só ocorrerá se o hardware utilizar o firmware UEFI.
- MS Windows:
  - até o Windows 7, somente as versões de 64 bits suportam GPT.
  - A partir do Windows 8, todos suportam.
  - Similar ao MacOS, deverá haver o UEFI.
  - O Windows usa o MBR híbrido como alternativa geral, tanto para 64 como para 32 bits.
- O Windows também pode utilizar loaders externos, como Paragon, para implementar suporte ao GPT.
- O GNU/Linux (GRUB2) já está totalmente preparado para o GPT.

# Considerações extras para GPT

- Conversão de GPT para MBR e vice-versa:
  - É possível, mas algumas vezes, a conversão não dará certo.
  - Quase sempre necessitará de intervenção humana para ajustes (fstab, GRUB etc).
  - Há o problema das partições lógicas na conversão (bilateral).
- Remoção do MBR: Limpeza (*zero fill*) do primeiro setor do disco.
- Remoção do GPT: Limpeza dos primeiros e dos últimos 34 setores.
- O UEFI e o GPT são uma realidade, pois HDs já possuem mais que 2 TB.
  - BIOS e MBR já acabaram.
- UEFI é um substituto da BIOS. Então, também interage com placas de vídeo, dispositivos USB, etc.
- Dispositivos pequenos ainda tenderão a usar MBR por um tempo.
  - Isso, inclusive, facilita a portabilidade.
- O GNU/Linux (GRUB2) é 100% compatível com o GPT.

# Mão na massa

```
dcfldd if=/dev/zero of=imagem.img \  
  bs=1024 count=$((1024*1024))
```

```
cfdisk imagem.img  
# escolha dos  
# tente criar mais que 4 partições  
hexdump -C -n 512 imagem.img
```

---

```
dcfldd if=/dev/zero of=imagem.img \  
  bs=1024 count=$((1024*1024))
```

```
cfdisk imagem.img  
# escolha gpt  
# tente criar mais que 4 partições  
hexdump -C -n 2048 imagem.img  
hexedit imagem.img  
# → ENTER → 0x40000000 → ENTER  
# F10 para sair
```

# Estrutura de Diretórios

- /usr/bin
  - Executáveis acessíveis por todos. Todos podem executá-los.
- /bin
  - Tipo o /usr/bin, mas aqui são somente os executáveis vitais do sistema. São executáveis simples e pequenos.
- /usr/sbin
  - Tipo o /usr/bin, mas são executáveis que devem ser acessados/executados somente pelo administradores.
- /sbin
  - Tipo o /bin, mas são executáveis que devem ser acessados/executados somente pelos administradores para gerenciar/manter o sistema.
- /usr/lib
  - Contém as bibliotecas dinâmicas de suporte para os arquivos executáveis de /usr/bin e /usr/sbin.
- /lib
  - Tipo o /usr/lib, mas contém as bibliotecas necessárias para o processo de inicialização.
- /etc
  - Contém arquivos de configuração.



# Estrutura de Diretórios

- /var
  - O nome se origina de "variável", pois tudo nesse diretório modifica frequentemente.
- /home
  - Contém os diretórios que pertencem aos usuários do sistema. Onde eles tem permissão geral de manipulação de seu conteúdo.
- /root
  - Tipo o /home, mas é o diretório pertencente ao administrador.
- /usr/share/doc, /usr/share/man
  - O "share" vem de "compartilhar", pois é um diretório independente de plataforma que pode ser compartilhado até mesmo pela rede.
- /usr/local, /opt
  - Diretórios obsoletos. Utilizados para trabalhar com programas que não são gerenciados pelo sistema e sim pelo usuário.
- /media
  - Diretório onde o acesso as mídias de armazenamento são fornecidas ao usuário.
- /mnt
  - Tipo o /media, mas obsoleto.

# Processo de Instalação

- Iniciar o instalador
- Definir usuários e senhas
- Particionar o disco rígido
- Instalar o sistema base
- Instalar o software adicional
- Instalar o sistema de inicialização (GRUB)
- Terminar a instalação

*Vamos ver isso na prática...*

# Referências

- Palestra do João Eriberto Mota Filho:  
<http://eriberto.pro.br/palestras/gpt.pdf>
- UEFI:  
<http://www.uefi.org>
- TLDP:  
<http://www.tldp.org/HOWTO/HighQuality-Apps-HOWTO/fhs.html>
- Manual de instalação Debian:  
<https://www.debian.org/releases/stable/i386/install.pdf.pt>