

Sistemas Operacionais – Clovis Jose Ramos Ferraro

Grupo 09

Data para entrega: 13/09/2025

MÓDULO VIRTUALBOX E VMWARE

São Paulo

2025

Sumário

1. INTRODUÇÃO	3
2. METODOLOGIA.....	4
2.1 LINUX	4
2.1.1 AMBIENTE DE TESTE	4
2.1.2. Acesso ao GRUB e Seleção da Mídia de Boot.....	4
2.1.3 Tela Inicial do Setup UEFI.....	5
2.1.4 Gerenciador de Dispositivos	6
2.1.5 Gerenciador de Boot	6
2.1.6 Gerenciador de Manutenção de Boot	7
2.1.7 Visualização da Tabela de Partições no Ubuntu.....	7
2.1.8 Verificação via Terminal com lsblk.....	8
2.2 WINDOWS	9
2.3 ANDROID	9
3. COMPARAÇÃO ENTRE SISTEMAS OPERACIONAIS.....	9
3.1 WINDOWS	9
3.1.1 Configurações de BIOS/UEFI.....	9
3.1.2 Acesso ao Boot em Máquinas Virtuais	11
3.1.3 Comando.....	11
3.2 Linux	11
3.3 ANDROID	12
3.3.1 Bios Android	13
3.3.2 Acessando a Bios no Android Studio.....	13
3.3.3 Partições de memória.....	13
3.3.4 FastBoot.....	15
3.4 Comparação Crítica.....	16
3.4.1 Tabela Comparativo do Processo de Boot, Inicialização e Instalação: Linux, Windows e Android:	17
4. ANÁLISE CRÍTICA	17
4.1 LINUX	17
4.2 WINDOWS	17
4.3 ANDROID	18
5. CONCLUSÃO	18
6. AUTOAVALIAÇÃO	19
7. REFERÊNCIAS	20

1. Introdução

Este trabalho tem como objetivo analisar como acontece a inicialização e a instalação em três sistemas operacionais bastante usados: Linux, Windows e Android. Para isso, vamos explorar conceitos fundamentais como o Bootstrapping e as configurações da BIOS/UEFI, que garantem o funcionamento adequado do sistema e a preparação correta dos dispositivos.

No caso do Linux, veremos como o processo de inicialização transforma o computador de um estado desligado para um ambiente pronto para rodar o sistema operacional. Já no Windows, o foco será a sequência de inicialização e o papel da BIOS/UEFI no gerenciamento do boot e dos dispositivos. Por fim, no Android, destacaremos o funcionamento do Fastboot, do boot.img e do Bootloader, elementos essenciais para controlar a forma como o aparelho é iniciado.

2. Metodologia

2.1 Linux

Ao conduzir os testes e experimentos, optou-se pelo uso de máquinas virtuais, uma escolha adequada para esse tipo de pesquisa. Essa ferramenta cria um "computador simulado", permitindo a instalação e personalização de sistemas Linux sem afetar o sistema operacional principal. Além disso, as máquinas virtuais aumentam a segurança durante os experimentos, pois falhas na divisão de discos não afetam o equipamento físico, o que possibilita a reexecução de testes de forma consistente e garante resultados confiáveis.

Os experimentos envolveram a análise de aspectos como a divisão de discos, examinando as estruturas MBR e GPT e avaliando suas composições e diferenças; os sistemas de arquivos, investigando como o Linux organiza e gerencia arquivos em diversos formatos de partições; e as notificações do kernel.

2.1.1 Ambiente de Teste

Para os experimentos a seguinte máquina virtual foi montada:

Distribuição Ubuntu Linux 22.04 LTS

Memória RAM: 4 GB

Processador: 2 vCPUs núcleos virtuais

Ferramenta virtualização: VirtualBox 6

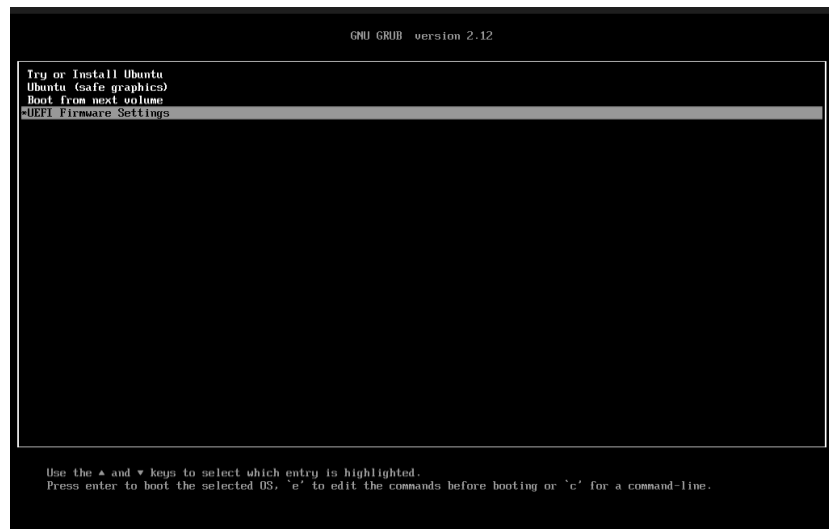
Terminal utilizado: Linux

A configuração, permitiu testes criação e gerenciamento particionamento, e tais ajustes e personalizações GRUB2, também acompanhou o boot na prática, garantindo a repetição dos experimentos em um ambiente seguro.

2.1.2. Acesso ao GRUB e Seleção da Mídia de Boot

Na inicialização da máquina, é exibido o menu do GRUB, com opções como "Try or Install Ubuntu", "Ubuntu (safe graphics)" e outras relacionadas à UEFI.

Neste momento, optou-se por acessar as configurações de firmware UEFI.



2.1.3 Tela Inicial do Setup UEFI

Ao selecionar “UEFI Firmware Settings”, o sistema carregou o menu de configuração UEFI. Nesta tela é possível escolher o idioma, gerenciar dispositivos e acessar as opções de boot.



2.1.4 Gerenciador de Dispositivos

Ao acessar o Device Manager, são exibidas opções de configuração relacionadas ao firmware UEFI. Cada uma dessas opções permite ajustar detalhes específicos, como o Secure Boot, entre outros.



2.1.5 Gerenciador de Boot

No Boot Manager, é possível visualizar os dispositivos disponíveis para inicialização. Ao selecionar qualquer um deles, são mostradas configurações específicas de boot, como ordem de prioridade e tipo de dispositivo.



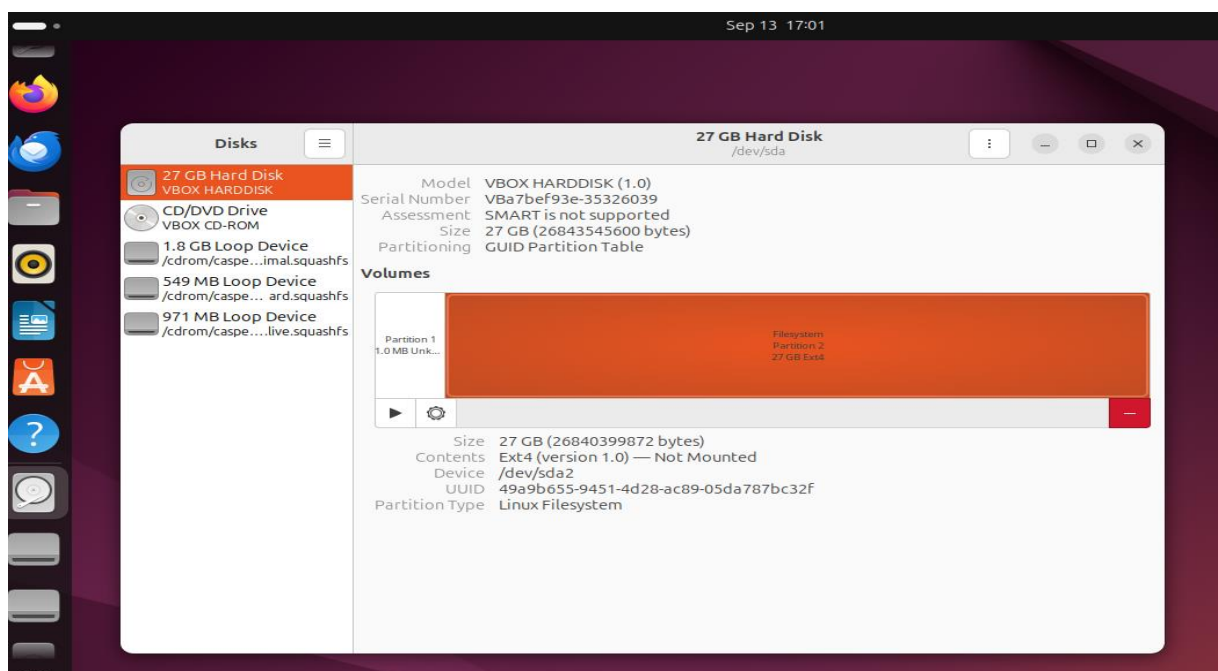
2.1.6 Gerenciador de Manutenção de Boot

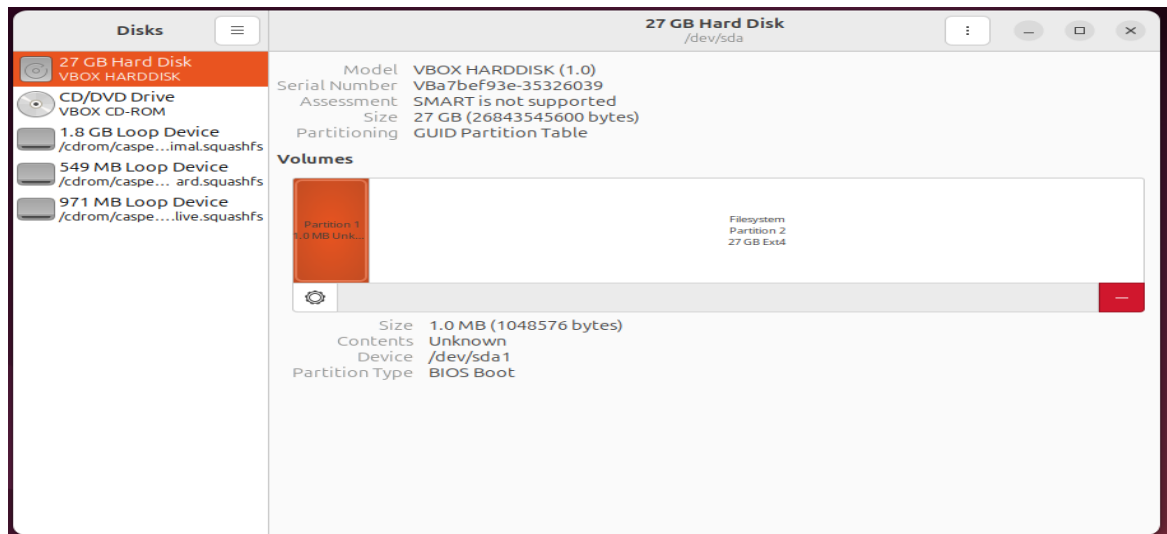
No Boot Maintenance Manager, o sistema apresenta configurações adicionais, como ajuste de prioridade, timeout de boot e carregamento de arquivos UEFI. Cada item acessado dentro dessa opção oferece parâmetros específicos de personalização.



2.1.7 Visualização da Tabela de Partições no Ubuntu

Após o boot do Ubuntu, foi utilizado o aplicativo Discos, onde foi possível visualizar o disco rígido de 27 GB, utilizando uma tabela de partição do tipo GUID (GPT). A partição principal está formatada em Ext4, típica do Ubuntu.





2.1.8 Verificação via Terminal com lsblk

Utilizando o comando `lsblk`, foi possível verificar a estrutura de dispositivos de blocos no sistema. A saída mostra o disco `sda`, com duas partições (`sda1` e `sda2`). A partição `sda2` contém o sistema de arquivos `Ext4`.

```

ubuntu@ubuntu: ~
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ubuntu:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0   1.7G  1 loop /rofs
loop1       7:1      0 523.3M  1 loop
loop2       7:2      0 925.9M  1 loop
loop3       7:3      0 245.1M  1 loop /snap/firefox/6565
loop4       7:4      0  11.1M  1 loop /snap/firmware-updater/167
loop5       7:5      0     4K  1 loop /snap/bare/5
loop6       7:6      0  73.9M  1 loop /snap/core22/2045
loop7       7:7      0  91.7M  1 loop /snap/gtk-common-themes/1535
loop8       7:8      0  10.8M  1 loop /snap/snap-store/1270
loop9       7:9      0   210M  1 loop /snap/thunderbird/769
loop10      7:10     0  49.3M  1 loop /snap/snapd/24792
loop11      7:11     0   516M  1 loop /snap/gnome-42-2204/202
loop12      7:12     0   576K  1 loop /snap/snapd-desktop-integration/315
loop13      7:13     0 112.6M  1 loop /snap/ubuntu-desktop-bootstrap/413
sda         8:0      0   25G  0 disk
├─sda1      8:1      0     1M  0 part
└─sda2      8:2      0   25G  0 part
sr0        11:0     1   5.9G  0 rom  /cdrom
ubuntu@ubuntu:~$

```


2.2 Windows

A metodologia adotada se baseou em estudo de referências, análise prática e captura de fotos reais de uma BIOS/UEFI em operação. O procedimento foi feito em um notebook Samsung, acessando a tela de configuração do firmware e registrando as opções de boot, segurança e detalhes do sistema.

2.3 Android

A metodologia do Android foi baseada em referências, prints em tempo real e análise prática. O procedimento foi feito em um pc desktop e o outro foi feito por um celular com USB DEBUG para acessar a bios do Android, por conta que uma AVD não existe bios nela.

3. Comparação entre Sistemas Operacionais

Para compreender como diferentes filosofias de design afetam o funcionamento, a instalação e o processo de inicialização de cada plataforma, é fundamental comparar sistemas operacionais. Ao analisar Windows, Linux e Android, é possível notar como cada sistema aborda questões como usabilidade, adaptabilidade, segurança e eficiência, evidenciando os prós e contras que impactam usuários e programadores.

3.1 Windows

A sequência de inicialização no Windows começa com a execução do firmware da máquina (BIOS/UEFI), responsável por detectar o hardware instalado e encontrar o dispositivo de boot. Depois, o Windows Boot Manager é carregado e, a partir dele, o núcleo do sistema (ntoskrnl.exe) é inicializado, prosseguindo com a carga de drivers e serviços básicos.

3.1.1 Configurações de BIOS/UEFI

A BIOS/UEFI oferece uma interface de configuração que permite ajustar diversos parâmetros do sistema. Entre eles, a prioridade de boot, os modos de inicialização, a ativação do Secure Boot e as configurações de segurança, como senhas de acesso.

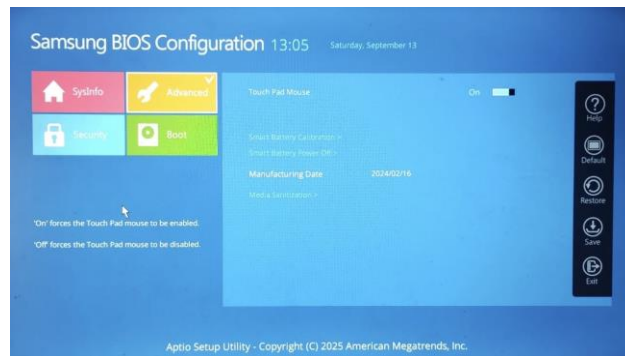
Exemplo de tela de configuração da BIOS/UEFI:



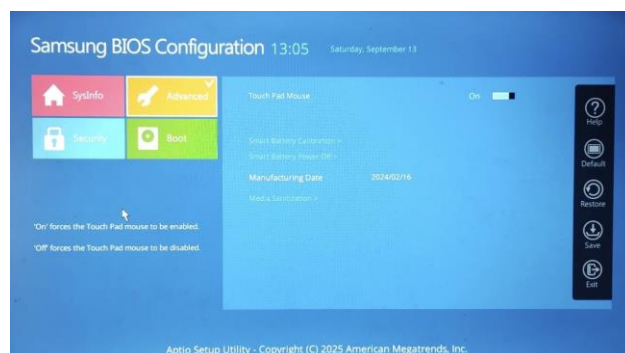
Configurações de Boot e opções de inicialização:



Tela SysInfo com informações do sistema:



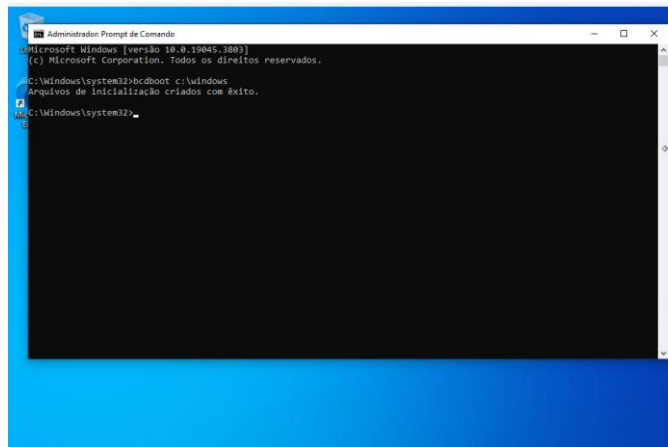
Tela Advanced com opções avançadas de hardware:



3.1.2 Acesso ao Boot em Máquinas Virtuais

Em ambientes virtuais, o acesso ao boot pode ser feito por meio de atalhos de teclado ou configurações específicas da máquina virtual. No VirtualBox, a tecla F12 abre o menu de boot, enquanto no VMware a tecla ESC é utilizada para o mesmo propósito. Também é possível configurar a ordem de boot diretamente nas opções de cada software.

3.1.3 Comando



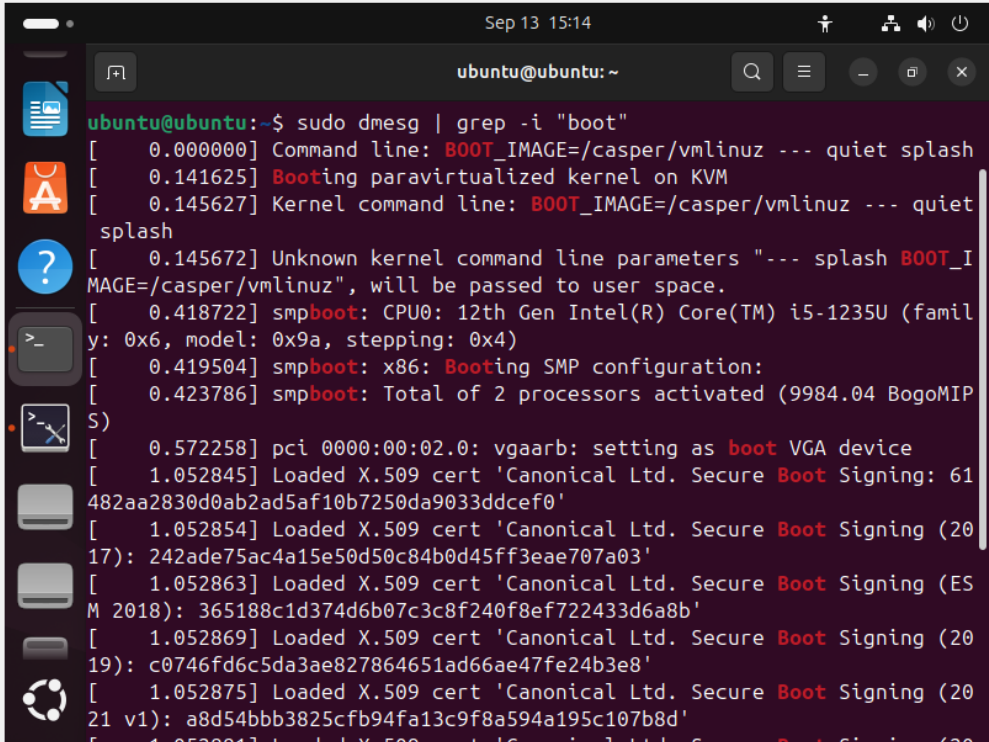
- bcdboot é uma ferramenta do Windows usada para criar ou reparar os arquivos de inicialização (boot) do sistema.
- parâmetro c:\windows indica o diretório onde o Windows está instalado, de onde ele vai copiar os arquivos necessários para inicializar.
- “Arquivos de inicialização criados com êxito” Isso quer dizer que os arquivos de boot foram gerados e configurados corretamente no computador.

Resumidamente, o Windows conseguiu restaurar ou recriar os dados de inicialização para que o sistema possa iniciar sem problemas.

3.2 Linux

Comando: `sudo dmesg | grep -i "boot"`

Função: O comando mostra as mensagens do kernel (o núcleo do sistema) que estão associadas ao processo de inicialização. Ele registra tudo o que ocorre quando o computador é ligado, filtrando apenas as linhas que incluem a palavra "boot", o que auxilia no diagnóstico de problemas durante o processo de inicialização. As mensagens exibem processos como a identificação dos discos, o carregamento do kernel e a ativação de outros serviços e dispositivos.



The image shows a terminal window on an Ubuntu system. The title bar indicates the date and time as 'Sep 13 15:14'. The terminal prompt is 'ubuntu@ubuntu: ~'. The user has entered the command 'sudo dmesg | grep -i "boot"'. The output displays various boot messages, including the kernel command line, booting on KVM, CPU identification (12th Gen Intel(R) Core(TM) i5-1235U), SMP configuration (2 processors activated), and Secure Boot signing events for various certificates.

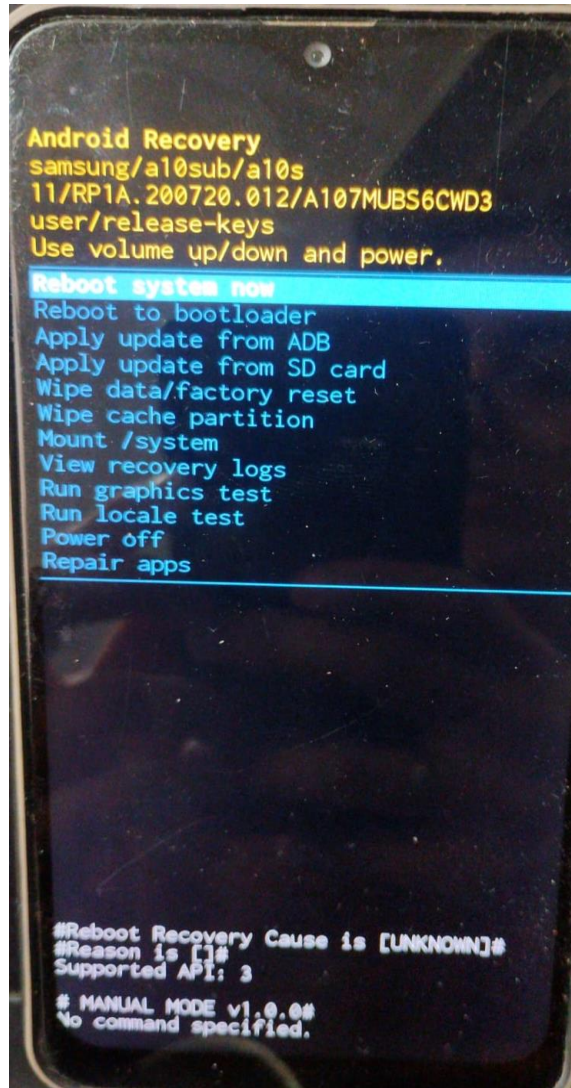
```
ubuntu@ubuntu:~$ sudo dmesg | grep -i "boot"
[ 0.000000] Command line: BOOT_IMAGE=/casper/vmlinuz --- quiet splash
[ 0.141625] Booting paravirtualized kernel on KVM
[ 0.145627] Kernel command line: BOOT_IMAGE=/casper/vmlinuz --- quiet splash
[ 0.145672] Unknown kernel command line parameters "--- splash BOOT_I
MAGE=/casper/vmlinuz", will be passed to user space.
[ 0.418722] smpboot: CPU0: 12th Gen Intel(R) Core(TM) i5-1235U (famil
y: 0x6, model: 0x9a, stepping: 0x4)
[ 0.419504] smpboot: x86: Booting SMP configuration:
[ 0.423786] smpboot: Total of 2 processors activated (9984.04 BogoMIP
S)
[ 0.572258] pci 0000:00:02.0: vgaarb: setting as boot VGA device
[ 1.052845] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing: 61
482aa2830d0ab2ad5af10b7250da9033ddcef0'
[ 1.052854] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (20
17): 242ade75ac4a15e50d50c84b0d45ff3eae707a03'
[ 1.052863] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (ES
M 2018): 365188c1d374d6b07c3c8f240f8ef722433d6a8b'
[ 1.052869] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (20
19): c0746fd6c5da3ae827864651ad66ae47fe24b3e8'
[ 1.052875] Loaded X.509 cert 'Canonical Ltd. Secure Boot Signing (20
21 v1): a8d54bbb3825cfb94fa13c9f8a594a195c107b8d'
```

3.3 Android

Ligar um celular Android é mais do que apertar um botão. Primeiro, ele dá uma pequena “espreguiçada” com o bootloader, se preparando para ligar. Depois, o kernel começa a trabalhar, como o coração do celular, fazendo tudo funcionar junto. O Zygote aparece em seguida, preparando os aplicativos para começar, enquanto o System Server liga os serviços essenciais, como se acendesse as luzes da casa. Só depois disso a tela inicial aparece, mostrando que o aparelho está pronto para você usar. Para quem segura o celular, parece simples: apertar o botão e esperar. Mas por trás disso, há todo um processo acontecendo para que o Android esteja totalmente pronto.

3.3.1 Bios Android

Na Bios do Android podemos realizar múltiplas tarefas como resetar o celular, o fast boot, ver as partições de memória, resetar a cache, fazer testes gráficos e locais, reparar programas e ver as logs de recuperação. Como na print abaixo:



3.3.2 Acessando a Bios no Android Studio

No Android Studio não é possível acessar a bios, as únicas coisas possíveis pelo Android Studio é fazer um fast boot e um bootloader. Tanto que bootloader e fast boot é uma configuração de ordem de boot.

3.3.3 Partições de memória

Para acessar as partições de memória exige um celular em modo de desenvolvedor com a depuração por usb ligado, os comandos utilizados foram:

1. Primeiro, eu baixei o arquivo platform-tools-latest-windows.zip do site oficial do Android, que contém os arquivos do ADB.
2. Como eu já tinha o WinRAR instalado, cliquei com o botão direito no ZIP e escolhi "Extrair para platform-tools\". A pasta foi criada direto no meu disco C: com o nome platform-tools.
3. No meu celular Samsung Galaxy A10s, fui em Configurações > Sobre o telefone > Informações do software e toquei várias vezes em Número de compilação até ativar o Modo Desenvolvedor.
4. Depois, entrei nas Opções do desenvolvedor e ativei a Depuração USB.
5. Conectei o celular ao computador com um cabo USB de dados e selecionei o modo Transferência de arquivos (MTP). Quando apareceu a mensagem "Permitir depuração USB?", toquei em Permitir.
6. No computador, abri o Prompt de Comando e digitei: `cd C:\platform-tools`
7. Testei a conexão com: `adb devices`
8. No começo não aparecia nada, mas depois de ajustar o cabo e permitir a depuração, o dispositivo foi reconhecido.
9. Entrei no shell do Android com: `adb shell`
10. Já dentro do sistema do celular, digitei: **`df -h`**, esse comando me mostrou o uso de espaço nas partições do Android, como `/system`, `/data`, `/cache`, entre outras.

Print das partições abaixo:

```
C:\WINDOWS\system32\cmd. x + v
C:\platform-tools>adb devices
List of devices attached

C:\platform-tools>adb devices
List of devices attached

C:\platform-tools>adb devices
List of devices attached
R9QN301CSPR    device

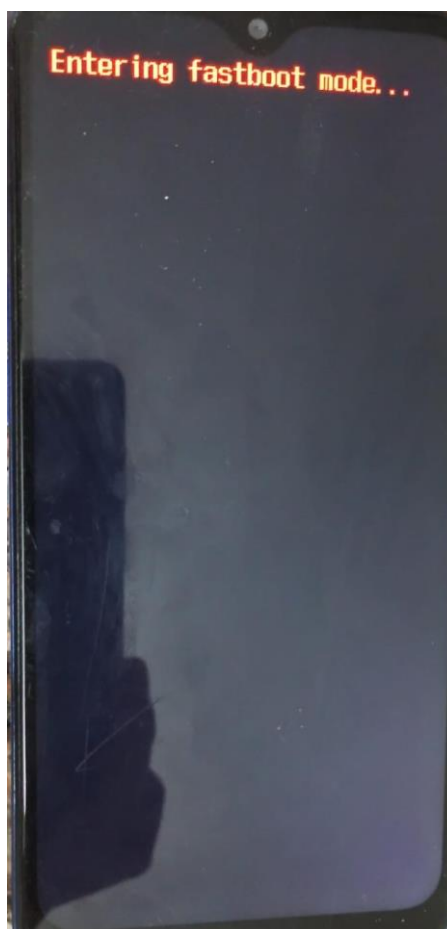
C:\platform-tools>adb shell
al0s:/ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            895M  2.3M  893M   1% /dev
tmpfs            895M   0  895M   0% /mnt
/dev/block/dm-1  3.3G  3.1G  164M  96% /
/dev/block/dm-2  418M   89M  321M  22% /product
/dev/block/dm-3  434M  322M  103M  76% /vendor
tmpfs            895M   0  895M   0% /apex
/dev/block/mmcblk0p37 398M   8.9M  273M   3% /cache
tmpfs            895M   0  895M   0% /mnt/sde
/dev/block/mmcblk0p6  3.7M  404K   3.1M  12% /efs
/dev/block/mmcblk0p38  43M   24K   42M   1% /omr
/dev/block/mmcblk0p39  24G   5.5G   18G  24% /data
/dev/fuse        24G   5.6G   18G  25% /storage/emulated
al0s:/ $
```

3.3.4 FastBoot

Para Fazer o fast boot exige um celular em modo de desenvolvedor com a depuração por usb ligado, os comandos utilizados foram:

1. Primeiro, eu baixei o arquivo platform-tools-latest-windows.zip do site oficial do Android, que contém os arquivos do ADB.
2. Como eu já tinha o WinRAR instalado, cliquei com o botão direito no ZIP e escolhi “Extrair para platform-tools\”. A pasta foi criada direto no meu disco C: com o nome platform-tools.
3. No meu celular Samsung Galaxy A10s, fui em Configurações > Sobre o telefone > Informações do software e toquei várias vezes em Número de compilação até ativar o Modo Desenvolvedor.
4. Depois, entrei nas Opções do desenvolvedor e ativei a Depuração USB.
5. Conectei o celular ao computador com um cabo USB de dados e selecionei o modo Transferência de arquivos (MTP). Quando apareceu a mensagem “Permitir depuração USB?”, toquei em Permitir.
6. No computador, abri o Prompt de Comando e digitei: cd C:\platform-tools.
7. Testei a conexão com: adb devices. No começo não aparecia nada, mas depois de ajustar o cabo e permitir a depuração, o dispositivo foi reconhecido.
8. Entrei no shell do Android com: adb shell.

9. Depois foi utilizado o comando: `adb reboot bootloader` entrando em modo fast boot.



3.4 Comparação Crítica

Há várias diferenças que afetam as filosofias de design do Windows, Linux e Android, influenciando os processos de inicialização e instalação de cada um. No Windows, o foco está em oferecer uma solução simples e de fácil uso para todos, com um sistema de inicialização unificado que opere em diferentes tipos de computadores. Por outro lado, o Linux valoriza a liberdade de escolha, permitindo que você utilize diversos programas para iniciar o sistema, como o GRUB, e personalize todos os aspectos durante a instalação. Embora possa ser complicado para iniciantes, isso o torna ideal para servidores e sistemas críticos. O Android, construído sobre o Linux, simplifica as coisas para celulares e tablets, com a inicialização controlada pelas fabricantes e pouca liberdade na instalação, focando na segurança e no bom funcionamento. Desse modo, enquanto o Windows facilita a vida do usuário comum e o Android protege o

sistema, o Linux se destaca pela sua capacidade de adaptação, mas exige mais conhecimento para lidar com a instalação e o início do sistema.

3.4.1 Tabela Comparativo do Processo de Boot, Inicialização e Instalação: Linux, Windows e Android:

ETAPA	LINUX	WINDOWS	ANDROID
Criação do Mídia de Instalação	Baixar ISO da distribuição (Ubuntu, Debian, Fedora etc.); Criar pendrive ou disco de boot.	Criar pendrive/DVD de boot com Rufus ou Media Creation Tool	Firmware pre-instalado pelo fabricante ou download do pacote oficial OTA/ROM
Boot pelo Instalador	Configurar BIOS/UEFI para iniciar pelo pendrive/disco; Carregar instalador na memória (Live environment).	Configurar BIOS/UEFI para iniciar pelo pendrive/DVD; Carregar instalador na memória (Windows Setup).	Bootloader entra em ação ao ligar; Em atualizações OTA, bootloader entra em modo recovery.
Configuração Inicial	Escolha de idioma, layout de teclado e fuso horário; Particionamento do disco (manual ou automático).	Escolha de idioma, layout de teclado, fuso horário e chave de ativação; Particionamento do disco.	Verificação de integridade do firmware (assinatura digital); Preparação das partições do dispositivo.
Instalação do Sistema Base	Copiar arquivos do sistema (kernel, bibliotecas, utilitários); Configurar usuário, senha e root; Instalar bootloader (GRUB/systemd-boot).	Copiar arquivos do sistema (kernel, bibliotecas, utilitários); Criar conta de usuario e senha; Instalar bootloader (Windows Boot Manager).	Copiar arquivos do sistema (kernel, bibliotecas, framework Android) para partições; Configuração de usuários e contas iniciais.
Configuração de Pacotes e Serviços	Instalar pacotes essenciais (apt/yum/dnf) e drivers; Configuração de rede e serviços básicos.	Instalar drivers essenciais; Configuração inicial de rede e atualizações do sistema.	Instalar aplicativos essenciais do sistema (Google Apps ou fabricante); Configuracao de rede, contas e serviços basicos.
Reinicialização e Boot Final	Bootloader carrega kernel + initramfs - systemd - login do usuário	Bootloader carrega kernel - inicializa serviços do Windows - tela de login	Bootloader carrega kernel - init - Zygote - framework Android - interface do usuario

4. Análise Crítica

4.1 Linux

A filosofia de design da Linux é conhecida como Filosofia Unix, ela oferece um método de design que divide um sistema complexo em partes menores, que são autônomas e independentes, chamadas de módulos, também disponibiliza clareza, simplicidade, código aberto e colaborativo e até mesmo criação de programas que realizam uma única tarefa de forma eficiente. A filosofia deles são baseadas em “faça cada programa uma ferramenta que faça uma coisa e que faça bem”.

4.2 Windows

A forma como o Windows "arranca" não envolve uma única ação, mas sim duas ideias diferentes: uma é o começo do computador em si (o "bootstrapping" real), onde as peças físicas e o software básico (BIOS/UEFI) começam a carregar o sistema operacional, algo que acontece tanto em máquinas Windows quanto em outras.

4.3 Android

Quando ligamos um aparelho Android, o sistema vai acordando aos poucos, quase como nós ao despertar. Primeiro, o carregador de inicialização faz uma análise rápida, como um check-up matinal para garantir que tudo está ok. Depois, o núcleo do sistema ativa as funções básicas, como a memória, o processador e os drivers. O processo "init" começa a funcionar, colocando tudo em ordem, tal qual arrumamos a casa antes do café. O Zygote, então, prepara o terreno para que os apps funcionem bem. Os serviços do sistema atuam nos bastidores para harmonizar tudo e evitar problemas. Por fim, a tela inicial e os aplicativos aparecem, prontos para serem usados.

5. Conclusão

Ao entender a forma como os sistemas operacionais Linux, Windows e Android são iniciados, percebemos que, embora distintos, todos visam o mesmo: preparar o ambiente para que o usuário possa interagir com o sistema de forma estável e funcional.

Com o Linux, o uso de máquinas virtuais possibilitou entender, na prática, como a organização das partições, o GRUB e os registros do kernel organizam a inicialização. No Windows, ficou claro o papel crucial da BIOS/UEFI, tanto para definir a ordem de boot quanto para ajustar as configurações de hardware e segurança, essenciais para o sistema. No Android, a análise revelou como cada fase do bootloader ao kernel, até a ativação dos serviços essenciais, é vital para assegurar que o dispositivo proporcione uma experiência estável e confiável ao usuário. Em suma, a comparação entre os três sistemas mostrou que entender o processo de inicialização é mais do que teoria; é ver a base essencial que impulsiona as ferramentas que usamos diariamente.

6. Autoavaliação

Gabriela: Neste relatório, o maior desafio não foi instalar as máquinas virtuais, pois essa fase já tinha sido cumprida no relatório anterior. A atenção, agora, foi na investigação da partição MBR e GPT, além da configuração do GRUB2, percebendo como ele organiza as partições no Linux.

Me empenhei para entender a ligação do GRUB2 com as partições, auxiliando na construção do relatório. Ajudei os colegas nas reuniões também, cuidando do GRUB e escrevi a conclusão e a metodologia.

Hugo: Através desse projeto pude me aprofundar e entender os conceitos de bootstrapping, a inicialização e o funcionamento passo a passo de um sistema operacional de um dispositivo.

Busquei contribuir da melhor maneira na pesquisa do Sistema Operacional Android, juntamente com o integrante Kaio que teve participação muito efetiva. Além disso, participei das reuniões e na realização, organização do relatório, mediante as normas ABNT.

Juliana: Nesse projeto consegui entender sobre o bootstrapping e o processo de instalação, sua inicialização através de boots e instalação dos Sistemas Operacionais.

Colaborei com o grupo participando de reuniões, editando o relatório na Norma ABNT, ajudando com dúvidas sobre os sistemas operacionais, criação da tabela comparativa, focando mais na instalação e inicialização do Linux, realizei a comparação entre os sistemas operacionais e análise crítica do Linux.

Kaio: Usou a AVD para o procedimento da fast boot e boot loader, resultando em um erro no pc, que teve que desinstalar o Android Studio, para realizar o processo novamente do zero.

Lucas: A produção deste trabalho foi muito proveitosa, pois me permitiu ter contato direto com a interface de configuração da BIOS/UEFI. Acredito que consegui apresentar os conceitos de forma clara, utilizando imagens para melhor compreensão. Contribuí também nas partes relacionadas aos sistemas Android e Linux, porém foquei principalmente no sistema Windows.

7. Referências

DESENVOLVE CURSOS. Bootstrapping: o que é e como funciona? 21 ago. 2024.

Disponível em: <https://www.desenvolvecursos.com/blog/bootstrapping-o-que-e-e-como-funciona>. Acesso em: 12 set. 2025.

KEVILYN, Antonia. *Apostila de Sistemas Operacionais: Bootstrapping e Processos de Instalação*. [PDF]. São Paulo, 2025. Disponível apenas para os alunos da disciplina de Sistemas Operacionais.

LENOVO. O que é um dispositivo de inicialização e por que ele é importante? 2025.

Disponível em: <https://www.lenovo.com/br/pt/glossary/what-is-boot-device/#:~:text=O%20firmware%20BIOS%20ou%20UEFI%20desempenha%20um,d,e%20inicializa%C3%A7%C3%A3o%20do%20dispositivo%20de%20inicializa%C3%A7%C3%A3o%20designado> . Acesso em: 12 set. 2025.

HOW TO ACCESS VIRTUALBOX BIOS FOR YOUR VIRTUAL MACHINE (VM). *How to Access VirtualBox BIOS for your Virtual Machine (VM)*. 2025. Disponível em:

<https://youtu.be/W181UQduJjw>. Acesso em: 13 set. 2025.

VIRTUALBOX. *UEFI and BIOS in VirtualBox*. 2025. Disponível em:

<https://www.youtube.com/watch?v=QG1N4ZhLZDE>. Acesso em: 13 set. 2025.

COMO FORÇAR O MODO RECUPERAÇÃO E FAST BOOT COM UM CLIQUE:

<https://www.youtube.com/watch?v=Tep2NjPpbHo> Acessado em: 13/09/2025

ATUALIZAR COM FAST BOOT:

[https://source.android.com/docs/setup/test/running?hl=pt-br#:~:text=Para%20realizar%20uma%20atualização%20flash%20em%20um,fastboot%20\(Fastboot%20mode\)%2C%20execute:%20fastboot%20flashall%20-w](https://source.android.com/docs/setup/test/running?hl=pt-br#:~:text=Para%20realizar%20uma%20atualização%20flash%20em%20um,fastboot%20(Fastboot%20mode)%2C%20execute:%20fastboot%20flashall%20-w)

Acessado em :13/09/2025

ANTONIA, Kevilyn. *Apostila de Sistemas Operacionais: Bootstrapping e Processos de Instalação*. 2025.

MICROSOFT. Documentação Oficial do Windows. Disponível em: <https://learn.microsoft.com/>.

SILBERSCHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg. Fundamentos de

Sistemas Operacionais. 9. ed. Rio de Janeiro: LTC, 2016.
Vídeos educacionais no YouTube sobre processo de boot e configuração de BIOS/UEFI.