

# Sistemas Operacionais I

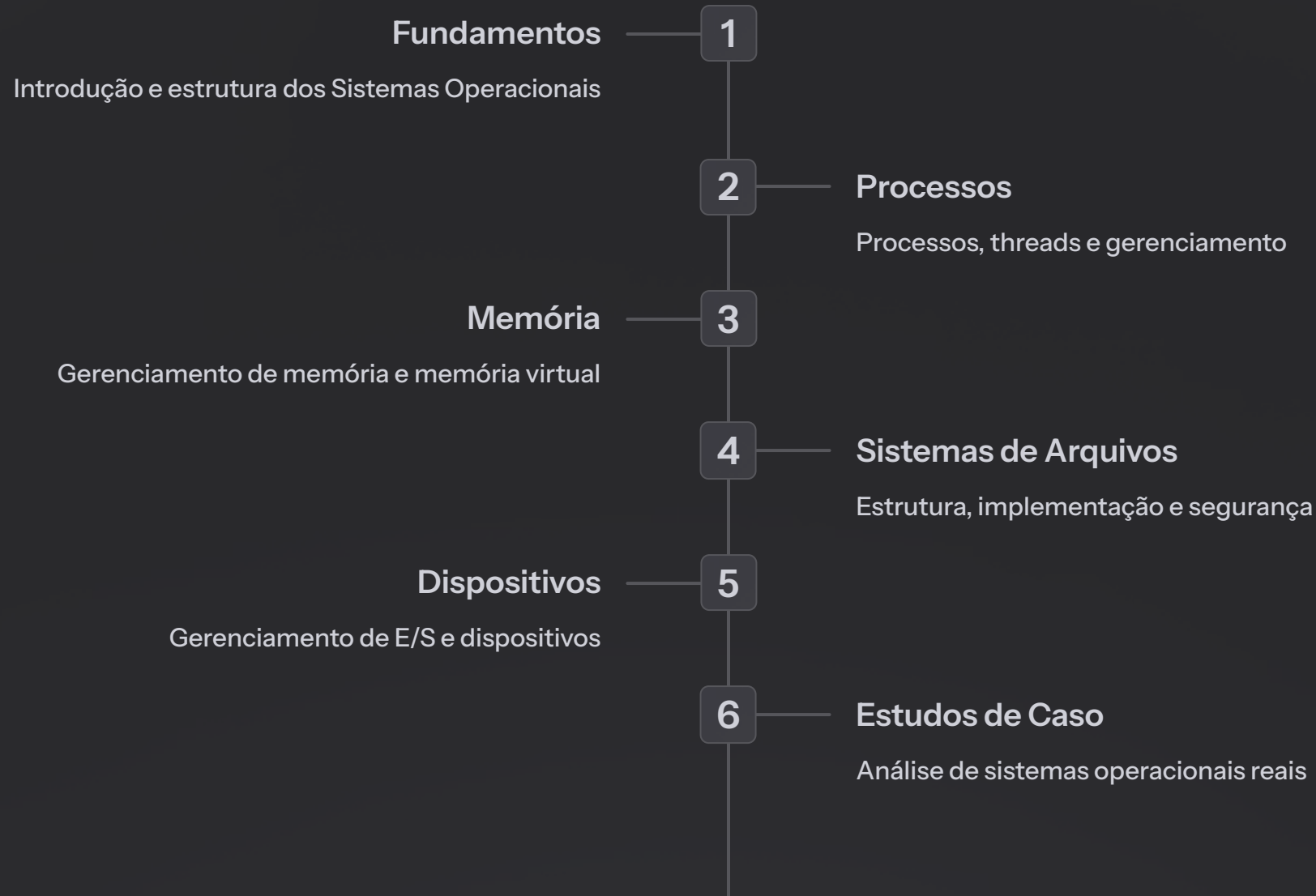
Bem-vindos à disciplina de Sistemas Operacionais !! Durante este semestre, mergulharemos no fascinante mundo dos sistemas que dão vida aos nossos computadores, desde os princípios fundamentais até estudos de casos práticos.

Prepare-se para uma jornada que transformará completamente sua visão sobre como funciona a tecnologia que utilizamos diariamente.



# Agenda do Curso

Durante o semestre, cobriremos os seguintes tópicos fundamentais:



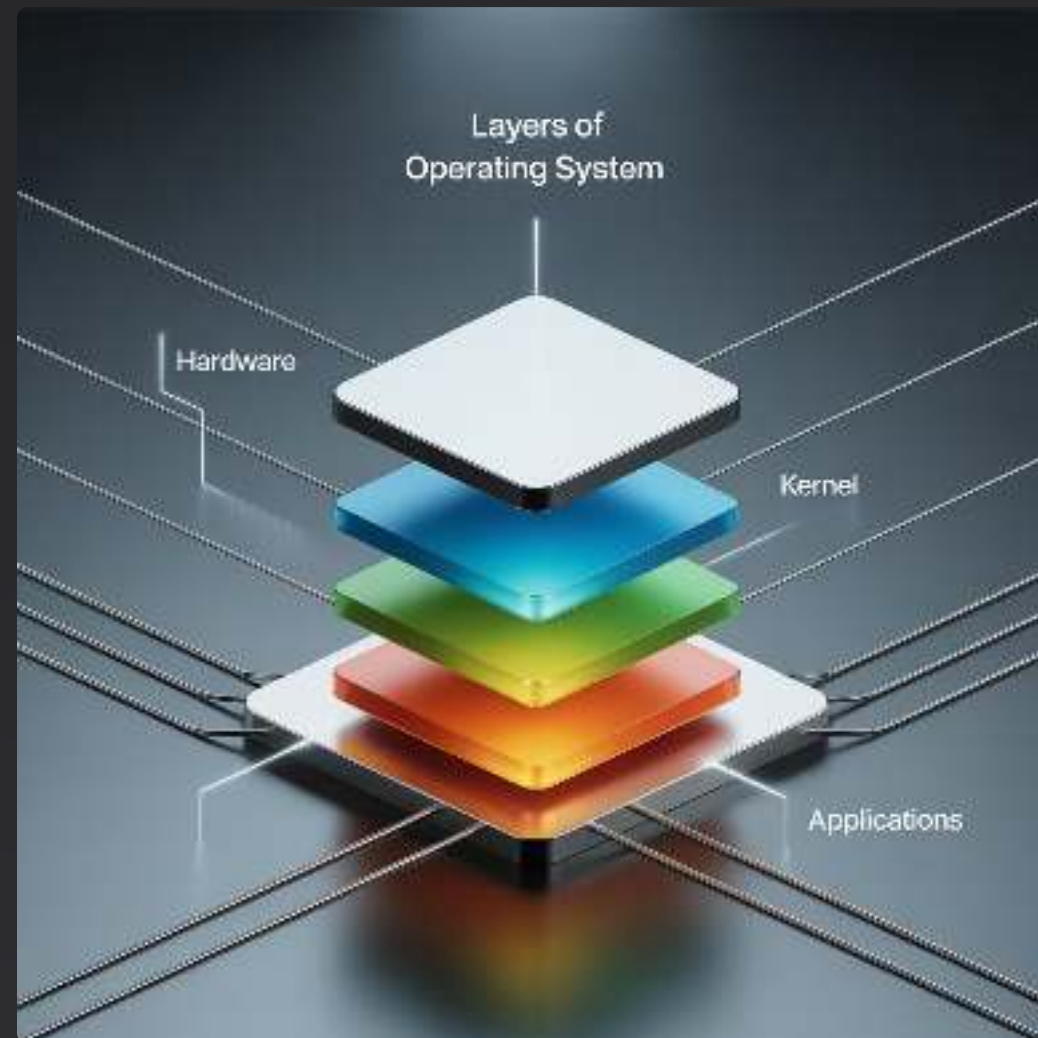
Carga horária total: 80 aulas

# O que é um Sistema Operacional?

Um Sistema Operacional (SO) é o software fundamental que gerencia recursos de hardware e fornece serviços para programas de aplicativos. É a camada intermediária entre o hardware e os aplicativos.

O SO é responsável por:

- Gerenciar recursos como CPU, memória e dispositivos
- Fornecer uma interface entre usuários e o computador
- Permitir que múltiplos programas funcionem simultaneamente
- Garantir a segurança e integridade do sistema



# Por que estudar Sistemas Operacionais?

## Base para qualquer desenvolvedor

Compreender como o SO funciona permite desenvolver software mais eficiente e otimizado.

## Fundamento para outras disciplinas

Redes, segurança, computação distribuída e desenvolvimento web dependem de conceitos de SO.

## Mercado de trabalho

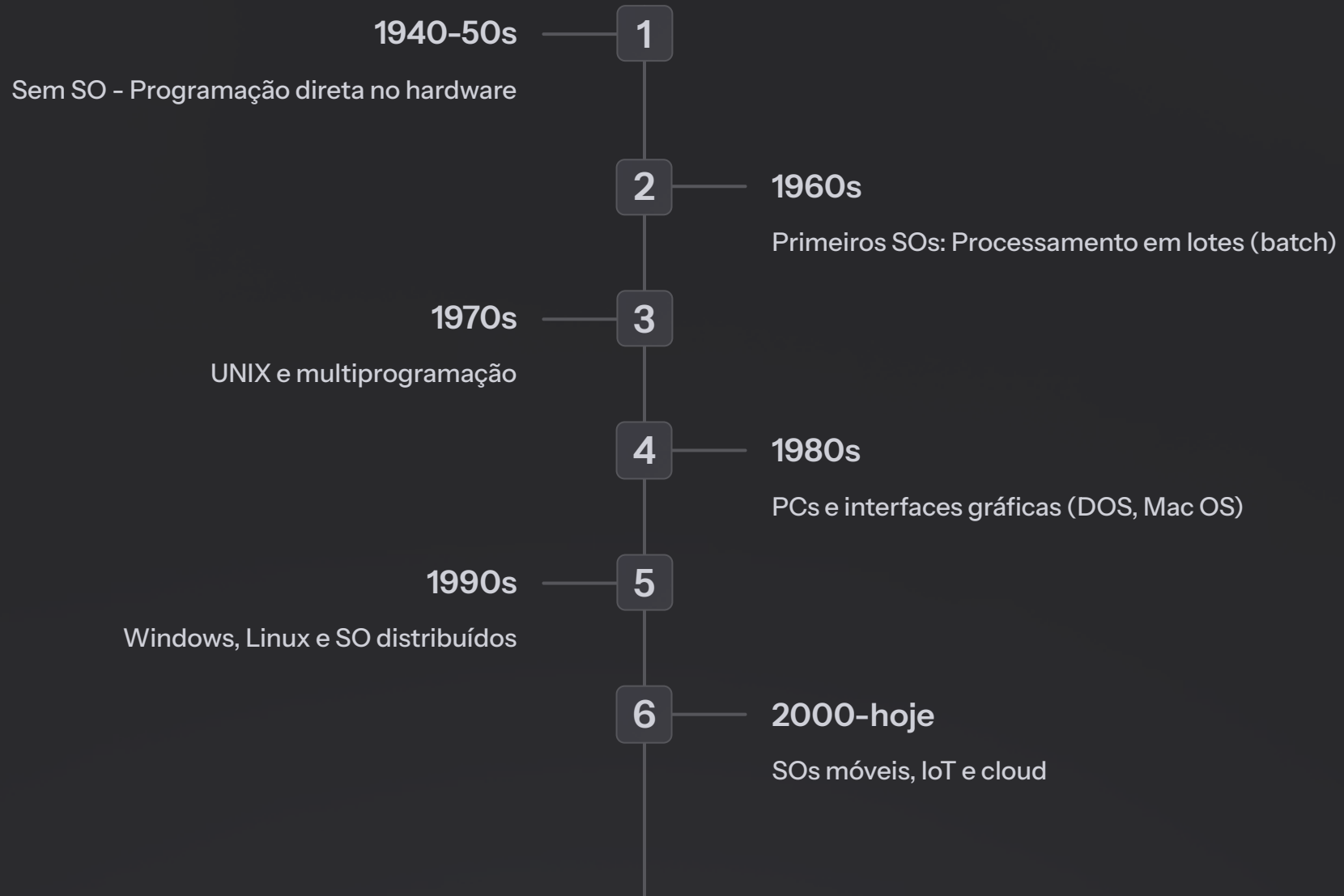
Profissionais com conhecimento sólido em SO são altamente valorizados no mercado.

## Resolução de problemas

Entender o funcionamento do SO ajuda a diagnosticar e resolver problemas de desempenho.

Este conhecimento será um diferencial na sua carreira como desenvolvedor de sistemas!

# História dos Sistemas Operacionais



A evolução dos sistemas operacionais acompanha o desenvolvimento do hardware e as necessidades crescentes dos usuários.



# Tipos de Sistemas Operacionais

## Por dispositivo

- Desktop (Windows, macOS, Linux)
- Móveis (Android, iOS)
- Embarcados (RTOS, FreeRTOS)
- Servidores (Windows Server, Linux)
- Mainframes (z/OS)

## Por arquitetura

- Monolíticos (Unix tradicional)
- Em camadas (THE)
- Microkernel (Minix, QNX)
- Híbridos (Windows, macOS)
- Exokernels (MIT)

Cada tipo de SO foi projetado para atender necessidades específicas, equilibrando fatores como desempenho, segurança, usabilidade e consumo de recursos.



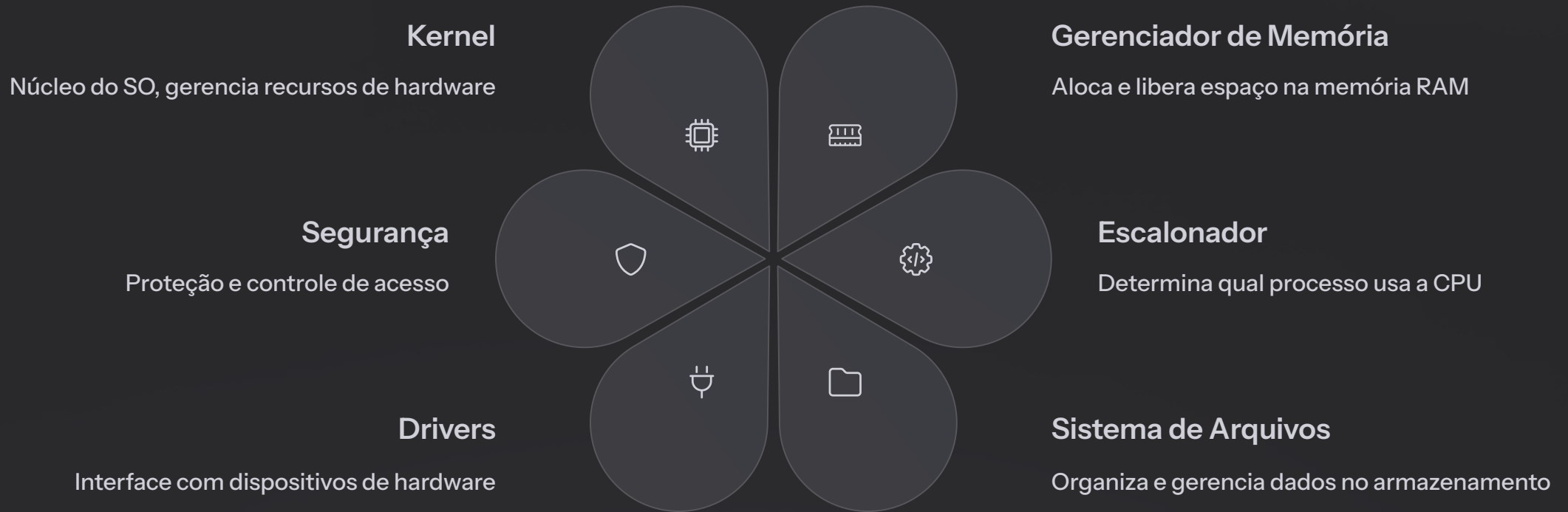
# Módulo 1

## Estrutura dos Sistemas Operacionais

Vamos começar nossa jornada entendendo como os SOs são organizados internamente.



# Componentes de um Sistema Operacional



Estes componentes trabalham juntos para fornecer um ambiente estável e eficiente para a execução de aplicações.



# Processos e Threads

## Processo

Um programa em execução com:

- Código (texto)
- Dados
- Pilha
- Heap
- Contador de programa
- Recursos alocados

## Thread

Unidade de execução dentro de um processo:

- Compartilha recursos do processo
- Tem sua própria pilha e registradores
- Execução independente
- Criação e comunicação mais eficientes

A compreensão de processos e threads é fundamental para entender como programas são executados e como o SO gerencia múltiplas tarefas.



# Estados de um Processo



## Novo

O processo está sendo criado



## Executando

Instruções estão sendo executadas



## Pronto

Aguardando para ser executado



## Bloqueado

Esperando por evento ou recurso



## Terminado

Execução concluída

O SO constantemente muda os processos entre esses estados para realizar multitarefa e otimizar o uso dos recursos do sistema.

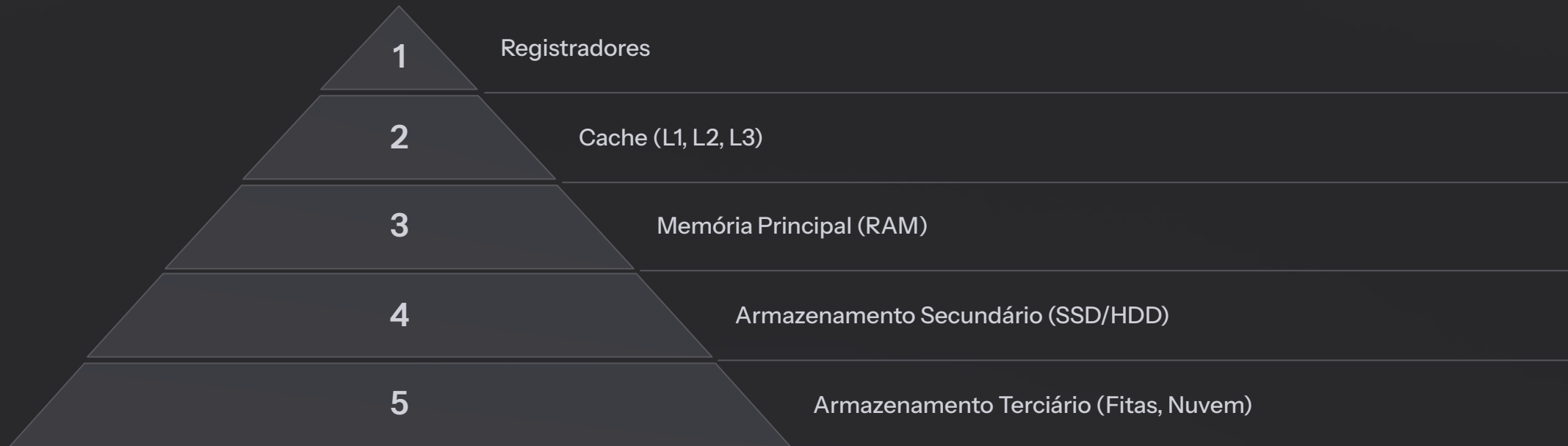


# Módulo 2

## Gerenciamento de Memória

A memória é um recurso precioso. Vamos entender como o SO a administra.

# Hierarquia de Memória



À medida que descemos na hierarquia, aumenta a capacidade de armazenamento e diminui a velocidade de acesso. O SO precisa balancear o uso de diferentes níveis para otimizar o desempenho.

Essa hierarquia influencia diretamente o desempenho das aplicações e como o SO toma decisões sobre alocação de recursos.





# Estratégias de Gerenciamento de Memória

## Particionamento

- Fixo: memória dividida em blocos de tamanho fixo
- Dinâmico: tamanho de partições varia conforme necessidade

## Paginação

Divide memória física e lógica em páginas de tamanho fixo

## Segmentação

Divide memória em segmentos lógicos de tamanho variável (código, dados, pilha)

## Memória Virtual

Usa disco para estender capacidade da RAM, com técnicas de paginação e swapping

Estas estratégias evoluíram para lidar com problemas como fragmentação, proteção e compartilhamento de memória entre processos.



# Memória Virtual

## Conceito

Técnica que cria ilusão de memória maior que a RAM física disponível, utilizando espaço em disco para complementar.

## Vantagens

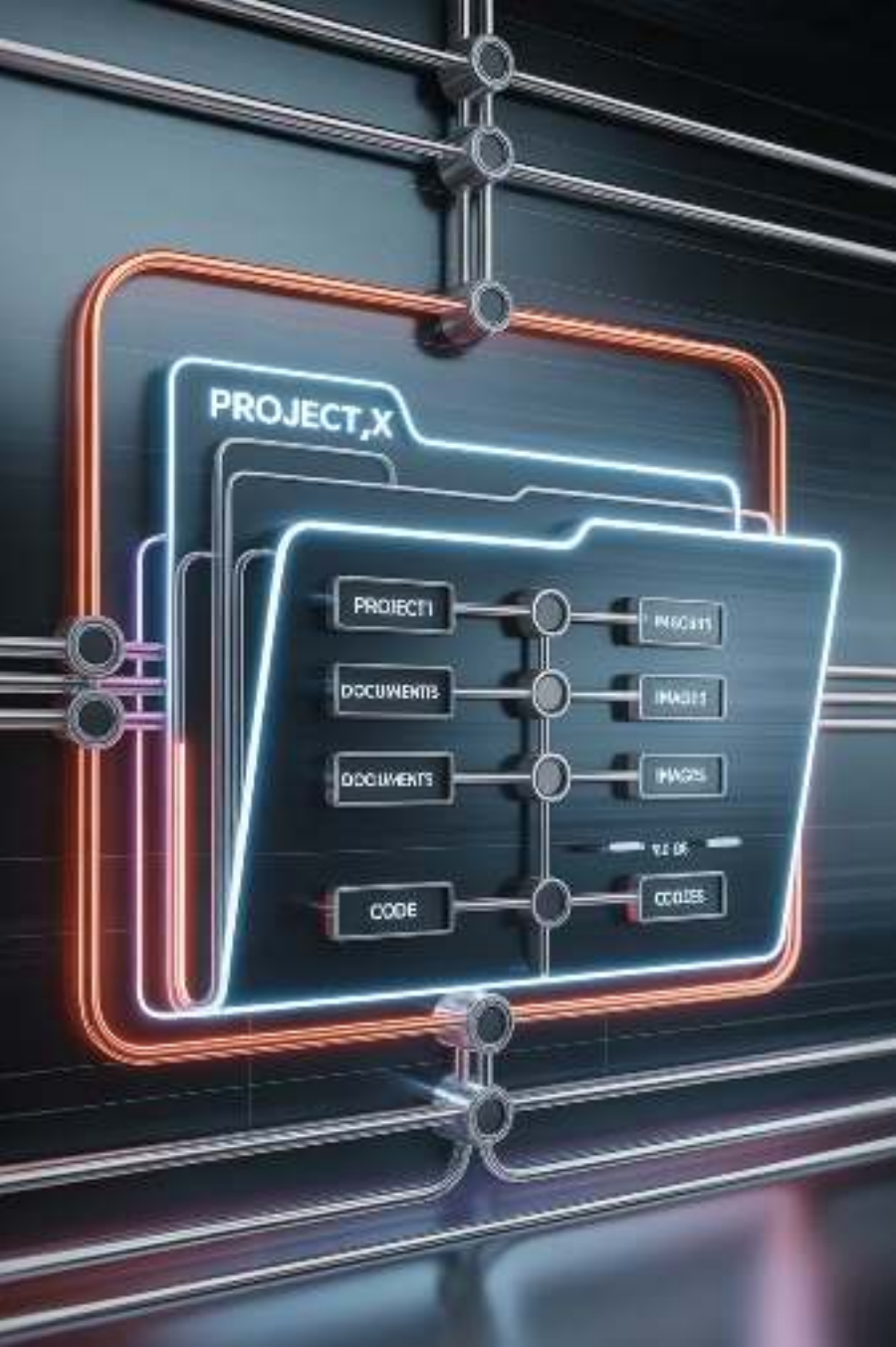
- Execução de programas maiores que a RAM física
- Simplifica programação (endereçamento virtual)
- Permite isolamento e proteção entre processos

## Mecanismos

- Tabelas de páginas
- TLB (Translation Lookaside Buffer)
- Algoritmos de substituição de páginas

A memória virtual é um dos conceitos mais importantes e revolucionários na história dos sistemas operacionais.





# Módulo 3

## Sistemas de Arquivos

Como o SO organiza e gerencia os dados armazenados.

# Conceitos de Sistemas de Arquivos

## Arquivo

Unidade lógica de informação com:

- Nome
- Tipo
- Tamanho
- Atributos (data, permissões)
- Conteúdo

O sistema de arquivos fornece abstração que permite aos usuários armazenar e recuperar dados sem preocupação com detalhes físicos do armazenamento.

## Diretório

Estrutura para organizar arquivos:

- Hierarquia em árvore
- Nível único
- Caminhos absolutos e relativos
- Metadados



# Tipos de Sistemas de Arquivos



## Windows

FAT32, NTFS, ReFS

- NTFS: Journaling, compressão, criptografia, permissões
- ReFS: Maior confiabilidade para servidores



## Linux

Ext2/3/4, XFS, Btrfs, ZFS

- Ext4: Sistema padrão em muitas distribuições
- ZFS: Foco em integridade de dados e RAID



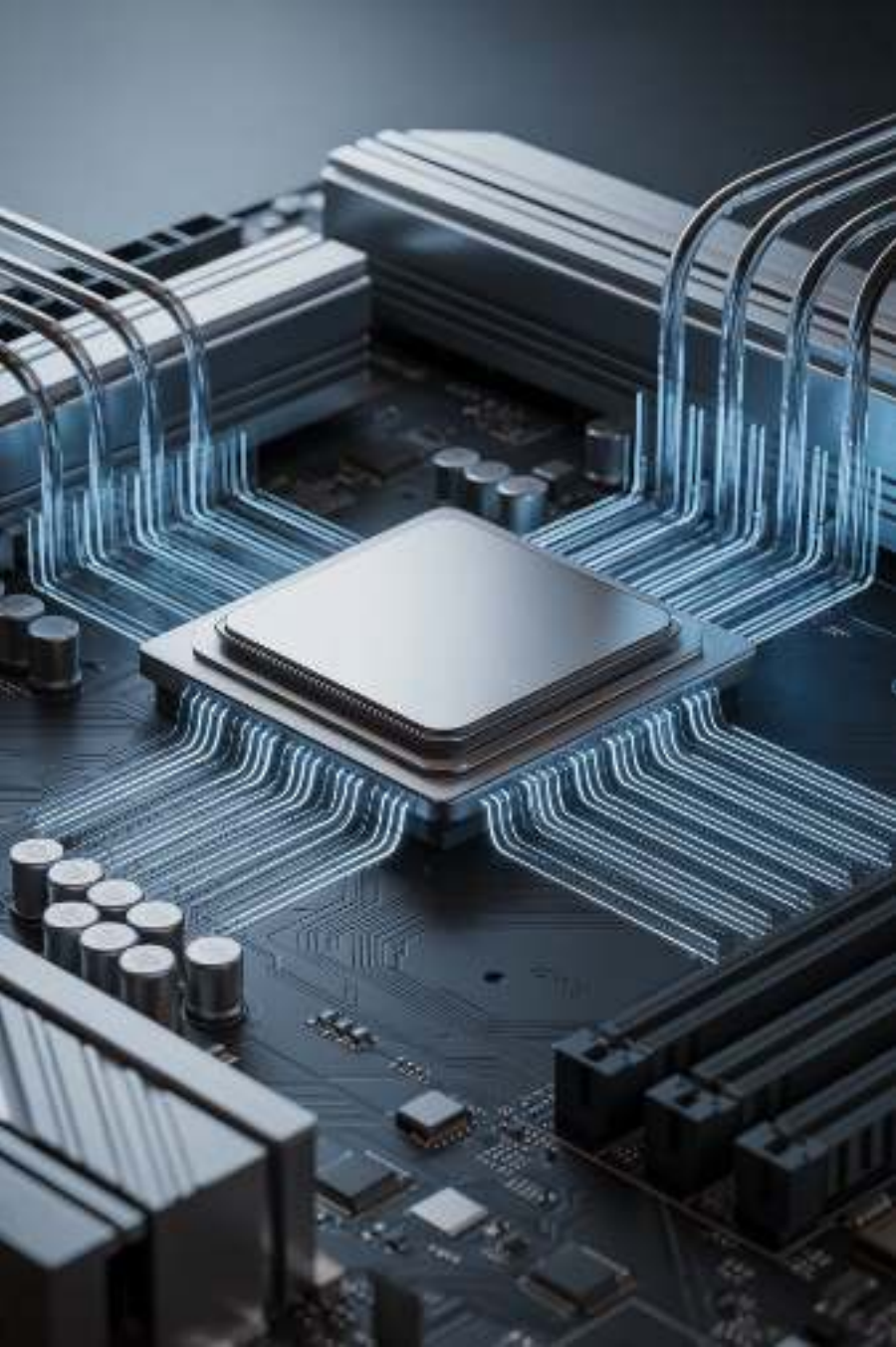
## macOS

HFS+, APFS

- APFS: Otimizado para SSDs e criptografia
- Suporte a snapshots e clones

Cada sistema de arquivos foi projetado com foco em diferentes necessidades: desempenho, confiabilidade, compatibilidade ou recursos específicos.



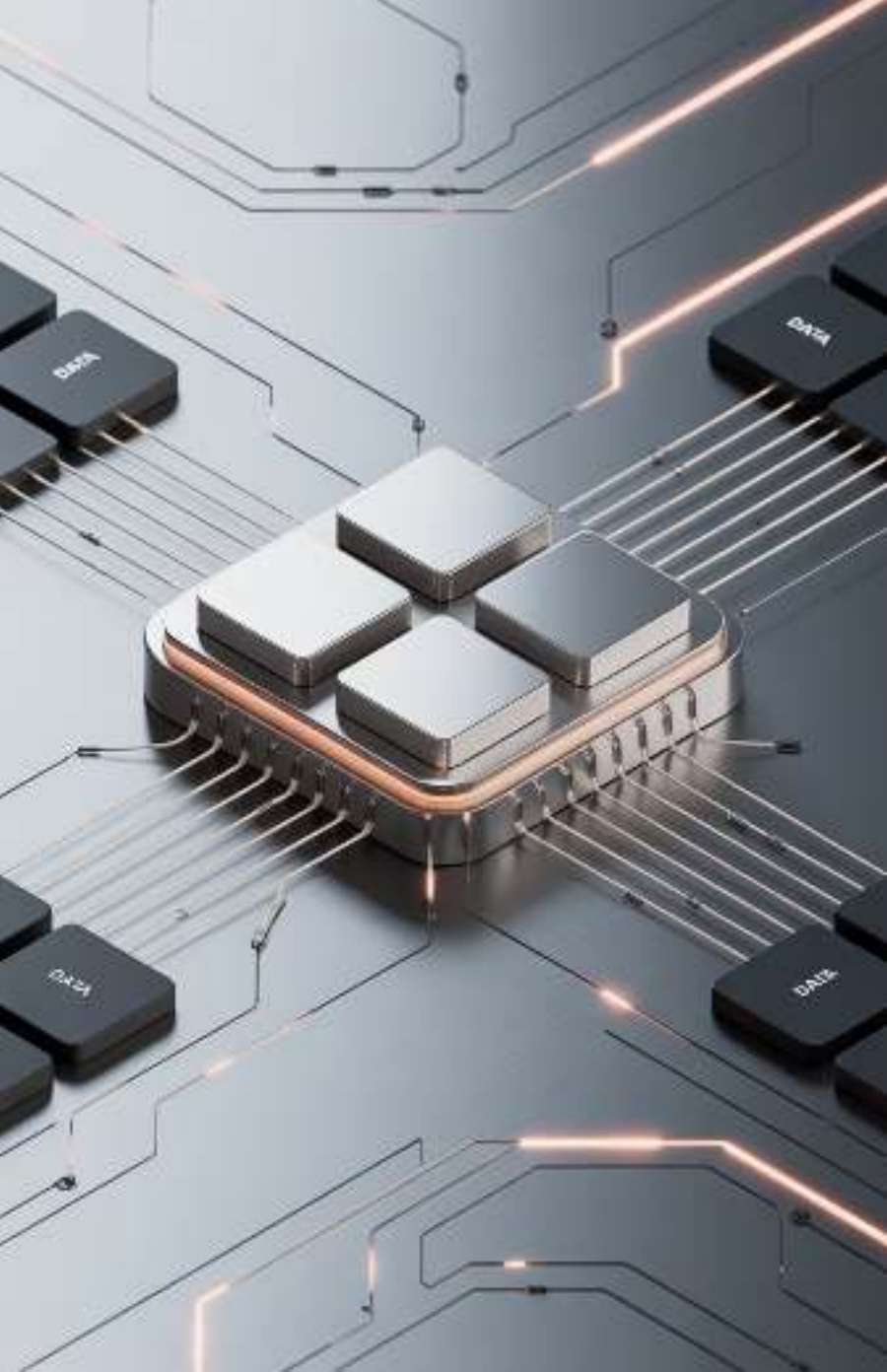


# Módulo 4

## Gerenciamento de Dispositivos

Como o SO interage com o hardware para entrada e saída de dados.





# Subsistema de E/S (Input/Output)

## Controladores de dispositivos

Hardware que conecta dispositivos periféricos ao barramento do computador

## Drivers de dispositivos

Software que permite ao SO se comunicar com controladores de hardware

## Buffers

Áreas de memória para armazenar temporariamente dados de E/S

## Spooling

Técnica para gerenciar dispositivos de uso exclusivo como impressoras

O gerenciamento eficiente de E/S é crucial para o desempenho do sistema, já que operações de E/S são tipicamente muito mais lentas que operações de CPU.



# O que esperar desta disciplina?

## Metodologia

- Aulas teóricas e práticas
- Laboratórios e experimentos
- Projetos em grupo
- Estudos de caso de SOs reais

## Bibliografia principal:

- OLIVEIRA, R S; CARISSIMI, A S; TOSCANI, S S. **Sistemas Operacionais**. Livros Didáticos 11. Bookman, 2008.
- TANENBAUM, A. S. **Sistemas Operacionais Modernos**. Prentice Hall (Pearson), 2007.

## Habilidades desenvolvidas

- Compreensão dos fundamentos de SOs
- Capacidade de análise de desempenho
- Pensamento crítico sobre design de sistemas
- Conhecimento para otimização de aplicações
- Base para disciplinas avançadas

Esta disciplina será fundamental para sua formação como desenvolvedor de sistemas, fornecendo bases teóricas e práticas essenciais para sua carreira.