



CEUNSP
Centro Universitário
N. Sra. do Patrocínio

Sistemas Operacionais

Aula 01

Agenda

- ☐ Apresentação
- ☐ Conteúdo programático
- ☐ Estratégia de ensino
- ☐ Avaliação
- ☐ Bibliografia

Um pouco sobre mim:

3

- ❑ Engenharia Cartográfica
- ❑ Mestrado em Geofísica
- ❑ Ciência da Computação
- ❑ Esp. em Docência do Ensino Superior
- ❑ MBA em Gestão de TI

Um pouco sobre mim:

4

- ❑ Mais de 9 anos trabalhando em geofísica
- ❑ Dev. Java – Delogic – Fev/2020 – 45 dias
- ❑ Analista de BI – Mkt Now – Jul/2020 – Jul/2021
- ❑ Estágio – DevSecOps – IBM – Dez/2020 – Ago/2021
- ❑ Application Developer – DevSecOPS – IBM – Set/2021 - Ago/2022
- ❑ Dev Full Stack – Senior - Ago/2022 – Atual – Software Engineering II
 - ❑ Java – Back e Angular – Front

Um pouco sobre mim:

5

- ❑ Gosto de cachorros
- ❑ Engenheiros do Hawaii
- ❑ Corinthians mano...
- ❑ Fiz intercâmbio na Austrália
- ❑ Morei na Colômbia
- ❑ Moro em Itu com minha namorada
- ❑ Álvaro Itu – simplesmente Itu

Maggie

6



Contatos

7

- ❑ [linkedin.com/alvaro-augusto-pereira](https://www.linkedin.com/alvaro-augusto-pereira)
- ❑ Face – Instagram – Twiter
- ❑ @ituoficial
- ❑ alvaro.pereira@ceunsp.edu.br

Apresentação dos Alunos

8

Você trabalha? Se sim, na área de TI?

Qual área de TI você mais se identifica?

Estratégias de Ensino e avaliação:

9

Estratégia

- ☐ Aulas expositivas
- ☐ Atividades práticas

Avaliação

- ☐ A2 – 5,0
Trabalhos
- ☐ A1 – 5,0
Avaliação
- ☐ Nota $\geq 6,0$ – parabéns
- ☐ Nota $< 6,0$ – AF

Ementa e referências:

Pilares do pensamento computacional

11

- ❑ **Decomposição:** divisão de um problema em partes menores.
- ❑ **Reconhecimento de padrões:** após decompor um problema, é comum notarmos que tarefas semelhantes ou idênticas. Tais similaridades auxiliam na reutilização de soluções com base em experiências anteriores.
- ❑ **Abstração:** eliminação de detalhes não relevantes para a solução do problema, auxiliando a dar foco nas informações realmente importantes.
- ❑ **Algoritmos:** conjunto de instruções claras e finitas necessárias para a solução de um problema.

Abstrações

12

- ❑ A abstração é um conceito essencial para o pensamento computacional, capaz de criar representações simplificadas de qualquer ser vivo ou objeto.
- ❑ Quando em excesso, os detalhes sobre a entidade podem desviar o foco da solução ou trazer complexidade extra para o processamento.
- ❑ Modelos abstratos são muito utilizados em várias áreas do conhecimento.

Abstrações

13

- ❑ Abstrair é uma habilidade do ser humano, que consiste na capacidade de expressar algo de maneira concisa, eliminando detalhes desnecessários.
- ❑ Saber trabalhar com diferentes camadas de abstração é um requisito para o profissional da computação, e ao mesmo tempo um constante desafio

Conceito de abstração

14

- ❑ É um conceito chave tanto da computação quanto do pensamento computacional, chegando a descrever a ciência da computação como a “automação da abstração”
- ❑ Nesse contexto, o computador é considerado uma máquina que não pensa realmente, mas interpreta o mundo a sua volta, somente executando tarefas com base na descrição de mundo realizada pelo cientista da computação.

Conceito de abstração

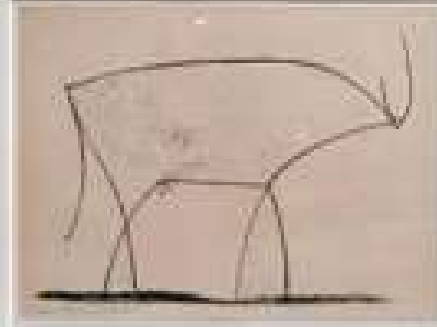
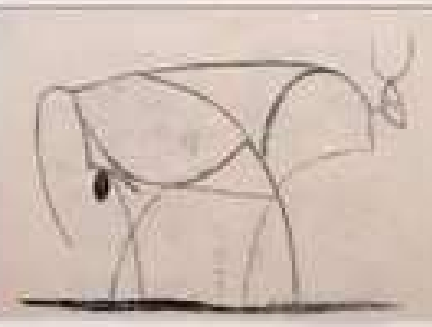
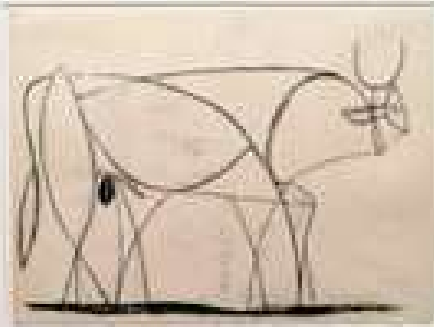
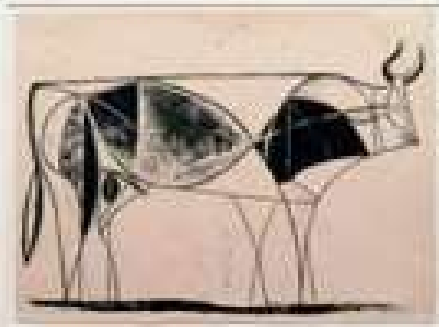
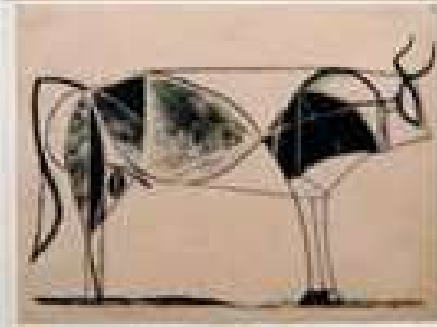
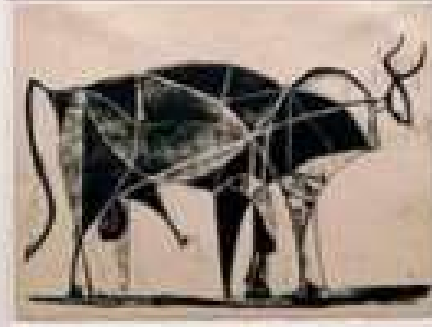
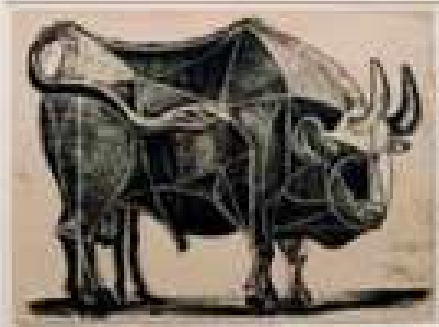
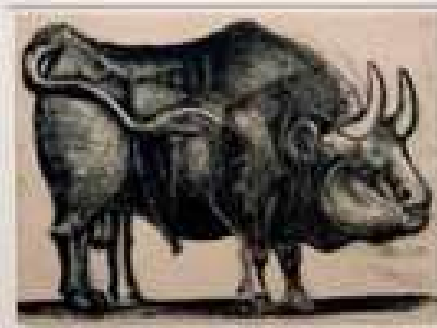
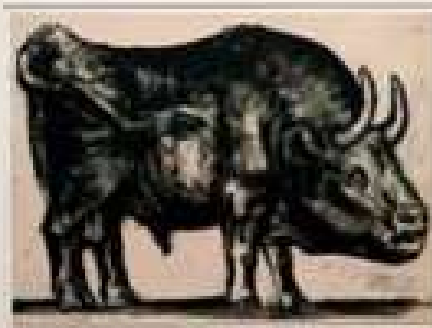
15

- ❑ Quando precisamos que o computador execute uma tarefa, devemos explicar somente as informações necessárias para que ele consiga executar, pois se não houver um filtro podemos perder poder de processamento com excesso de informações e desviar do problema principal
- ❑ Ex: Média dos alunos

Conceito de abstração

16

- ❑ Esse processo de selecionar quais as informações são mais ou menos relevantes é o que chamamos de abstração.
- ❑ A essência da abstração é manter as informações relevante de determinado contexto e omitir todo o resto.



Abstrações em camadas

18

- ❑ Essas diferenças entre níveis de abstração é conhecida como camadas de abstração.
- ❑ Cada camada represente um nível de abstração
- ❑ Aumentar detalhes – diminuir abstração
- ❑ Diminuir detalhes – aumentar abstração

Abstração no contexto computacional

- ❑ Linguagens de programação
- ❑ Antigamente a programação era feita diretamente na linguagens das máquinas, operadores booleanos (zeros e uns).
- ❑ Muitas vezes feitas diretamente nos hardwares.
- ❑ Quanto passamos a substituir longas instruções em linguagem de maquina para instruções mais curtas e compreensíveis, mais parecidas com linguagem natural, os detalhes de hardwares ficaram ocultas aos programadores.
- ❑ Quanto mais alto nível da linguagem mais abstrata ela se torna.

Abstração no contexto computacional

20

- ❑ Outro exemplo
- ❑ Quanto acionamos uma tecla, não sabemos todos os processos que acontecem na máquina.
- ❑ Estamos apenas interessados em ter a resposta na tela do que foi digitado.

Abstração no contexto computacional

21

- ❑ Diversas áreas empregam camadas de abstrações
- ❑ Modelagem de banco de dados : modelo entidade relacionamento que utiliza camadas de abstrações diferentes, relacionado ao nível de maturidade da implementação
- ❑ Analise de sistemas : são utilizados diagramas e documentos com diversos níveis de abstração, ou seja, um diagrama seria com alta abstração, já o documento seria com baixa abstração.

Abstração no contexto computacional

- ❑ Programação : orientada a objetos, existem entidades que são representadas por uma interface por exemplo.
- ❑ Webdesign : vai alterando os níveis de abstração conforme vai evoluindo seu pensamento de um logo por exemplo
- ❑ Hardwares e redes : na construção, vai mudando o nível de abstração nas representações de acordo com o nível do processo.
- ❑ Não existe um melhor nível de abstração e sim o nível de abstração necessário para resolver determinado tema

Introdução

23

- ▶ Um computador moderno consiste em um ou mais processadores, alguma memória principal, discos, impressoras, um teclado, um mouse, um monitor, interfaces de rede e vários outros dispositivos de entrada e saída.
- ▶ Como um todo, trata-se de um sistema complexo. Se todo programador de aplicativos tivesse de compreender como todas essas partes funcionam em detalhe, nenhum código jamais seria escrito
- ▶ Além disso, gerenciar todos esses componentes e usá-los de maneira otimizada é um trabalho extremamente desafiador.

Introdução

24

- ▶ Por essa razão, computadores são equipados com um dispositivo de software chamado de sistema operacional, cuja função é fornecer aos programas do usuário um modelo do computador melhor, mais simples e mais limpo, assim como lidar com o gerenciamento de todos os recursos mencionados.
- ▶ Todos que possuem alguma experiência com computadores já usaram um sistema operacional como Windows, Linux, OS X, mas as aparências podem ser enganadoras.
- ▶ O programa com o qual os usuários interagem, normalmente chamado de shell (ou interpretador de comandos) quando ele é baseado em texto e de GUI (Graphical User Interface) quando ele usa ícones, na realidade não é parte do sistema operacional, embora use esse sistema para realizar o seu trabalho

Introdução

25

- ▶ A maioria dos computadores tem dois modos de operação: modo núcleo e modo usuário. O sistema operacional, a peça mais fundamental de software, opera em modo núcleo (também chamado modo supervisor).
- ▶ Nesse modo ele tem acesso completo a todo o hardware e pode executar qualquer instrução que a máquina for capaz de executar.

Introdução

26

- ▶ O resto do software opera em modo usuário, no qual apenas um subconjunto das instruções da máquina está disponível.
- ▶ Em particular, aquelas instruções que afetam o controle da máquina ou realizam E/S (Entrada/Saída) são proibidas para programas de modo usuário.

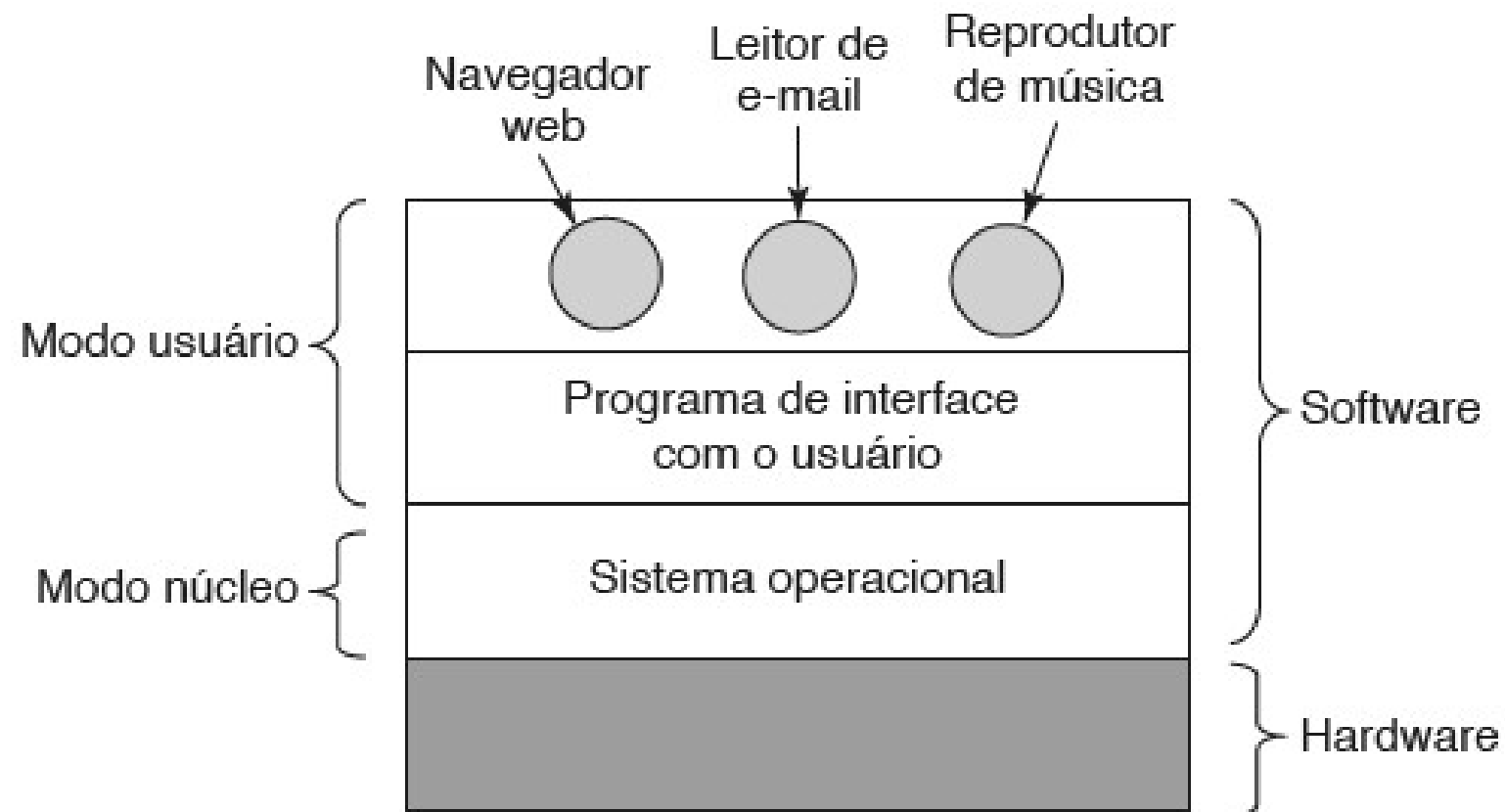
Introdução

27

- ▶ O programa de interface com o usuário, shell ou GUI, é o nível mais inferior de software de modo usuário, e permite que ele inicie outros programas, como um navegador web, leitor de e-mail, ou reproduutor de música.
- ▶ Esses programas, também, utilizam bastante o sistema operacional.
- ▶ O posicionamento do sistema operacional é mostrado na Figura. Ele opera diretamente sobre o hardware e proporciona a base para todos os outros softwares.

Introdução

28



Introdução

29

- ▶ Os sistemas operacionais diferem de programas de usuário (isto é, de aplicativos) de outras maneiras além de onde estão localizados.
- ▶ Em particular, eles são enormes, complexos e têm vida longa. O código-fonte do coração de um sistema operacional como Linux ou Windows tem cerca de cinco milhões de linhas

Introdução

30

- ▶ Deve estar claro agora por que sistemas operacionais têm uma longa vida — eles são difíceis de escrever, e tendo escrito um, o proprietário reluta em jogá-lo fora e começar de novo.
- ▶ Em vez disso, esses sistemas evoluem por longos períodos de tempo.
- ▶ O Windows 95/98/Me era basicamente um sistema operacional e o Windows NT/2000/XP/Vista/Windows 7 é outro.
- ▶ Eles são parecidos para os usuários porque a Microsoft tomou todo o cuidado para que a interface com o usuário do Windows 2000/XP/Vista/Windows 7 fosse bastante parecida com a do sistema que ele estava substituindo, majoritariamente o Windows 98.

O que é um sistema operacional

31

- ▶ O que é um sistema operacional?
- ▶ É difícil dizer com absoluta precisão o que é um sistema operacional, além de ele ser o **software que opera em modo núcleo** — e mesmo isso nem sempre é verdade.
- ▶ Sistemas operacionais realizam duas funções essencialmente não relacionadas:
 - ▶ fornecer a programadores de aplicativos (e programas aplicativos, claro) um conjunto de recursos abstratos limpo em vez de recursos confusos de hardware,
 - ▶ e gerenciar esses recursos de hardware.
- ▶ Dependendo de quem fala, você poderá ouvir mais a respeito de uma função do que de outra. Examinemos as duas então.

O sistema operacional como uma máquina estendida

- ▶ A arquitetura (conjunto de instruções, organização de memória, E/S e estrutura de barramento) da maioria dos computadores em nível de linguagem de máquina é primitiva e complicada de programar, especialmente para entrada/saída.
- ▶ Considere os discos rígidos usados na maioria dos computadores, teria que se especializar muito para saber armazenar a informação. Em vez disso, um software, chamado driver de disco, lida com o hardware e fornece uma interface para ler e escrever blocos de dados, sem entrar nos detalhes.

O sistema operacional como uma máquina estendida

33

- ▶ Sistemas operacionais contêm muitos drivers para controlar dispositivos de E/S. Mas mesmo esse nível é baixo demais para a maioria dos aplicativos. Por essa razão, todos os sistemas operacionais fornecem mais um nível de abstração para se utilizarem discos: arquivos.
- ▶ Essa abstração é a chave para gerenciar toda essa complexidade.
- ▶ Boas abstrações transformam uma tarefa praticamente impossível em duas tarefas gerenciáveis.

O sistema operacional como uma máquina estendida

- ▶ A primeira é definir e implementar as abstrações.
- ▶ A segunda é utilizá-las para solucionar o problema à mão.
- ▶ Uma abstração que quase todo usuário de computadores compreende é o arquivo.
- ▶ Trata-se de um fragmento de informação útil, como uma foto digital, uma mensagem de e-mail, música ou página da web salvas.
- ▶ A função dos sistemas operacionais é criar boas abstrações e então implementar e gerenciar os objetos abstratos criados desse modo.

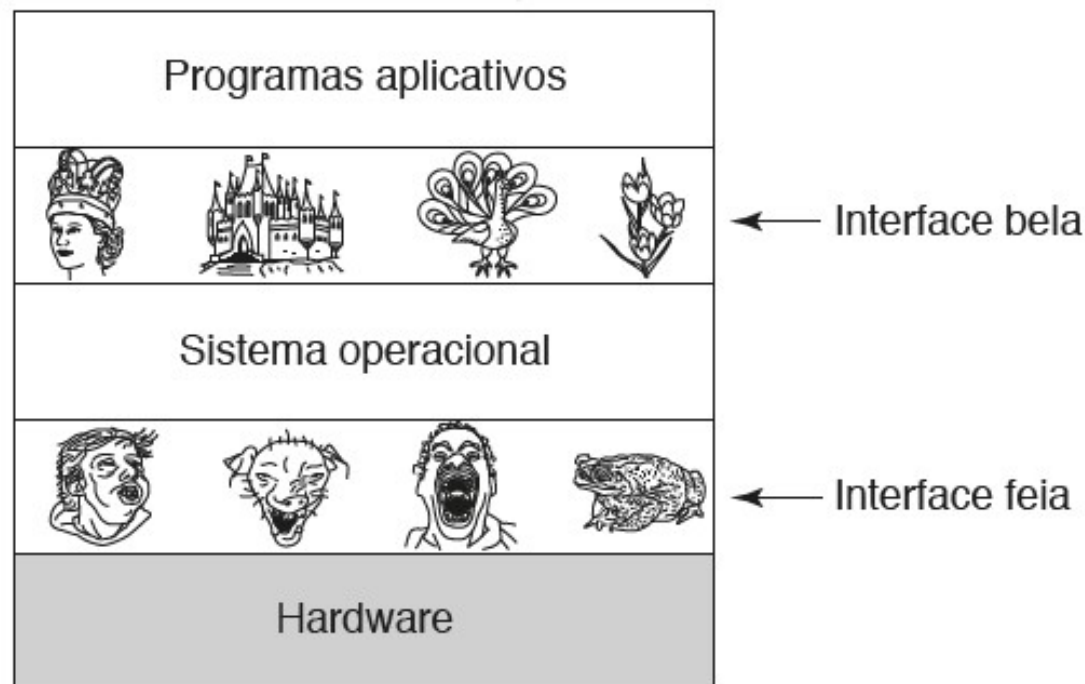
O sistema operacional como uma máquina estendida

- ▶ Processadores reais, memórias, discos e outros dispositivos são muito complicados e apresentam interfaces difíceis, desajeitadas, e inconsistentes para as pessoas que têm de escrever softwares para elas utilizarem.
- ▶ Uma das principais tarefas dos sistemas operacionais é esconder o hardware e em vez disso apresentar programas (e seus programadores) com abstrações de qualidade, limpas, elegantes e consistentes com as quais trabalhar.
- ▶ Sistemas operacionais transformam o feio em belo, como mostrado na Figura.

O sistema operacional como uma máquina estendida

36

FIGURA 1.2 Sistemas operacionais transformam hardwares feios em belas abstrações.



O sistema operacional como um gerenciador de recurso

- ▶ O conceito de um sistema operacional como fundamentalmente fornecendo abstrações para programas aplicativos é uma visão top-down (abstração de cima para baixo).
- ▶ Uma visão alternativa, bottom-up (abstração de baixo para cima), sustenta que o sistema operacional está ali para gerenciar todas as partes de um sistema complexo.
- ▶ Computadores modernos consistem de processadores, memórias, temporizadores, discos, dispositivos apontadores do tipo mouse, interfaces de rede, impressoras e uma ampla gama de outros dispositivos.
- ▶ Na visão bottom-up, a função do sistema operacional é fornecer uma alocação ordenada e controlada dos processadores, memórias e dispositivos de E/S entre os vários programas competindo por eles

O sistema operacional como um gerenciador de recurso

- ▶ Sistemas operacionais modernos permitem que múltiplos programas estejam na memória e sejam executados ao mesmo tempo.
- ▶ Imagine o que aconteceria se três programas executados em um determinado computador tentassem todos imprimir sua saída simultaneamente na mesma impressora.
- ▶ As primeiras linhas de impressão poderiam ser do programa 1, as seguintes do programa 2, então algumas do programa 3 e assim por diante.
- ▶ Quando um computador (ou uma rede) tem mais de um usuário, a necessidade de gerenciar e proteger a memória, dispositivos de E/S e outros recursos é ainda maior, tendo em vista que os usuários poderiam interferir um com o outro de outra maneira.

O sistema operacional como um gerenciador de recurso

39

- ▶ Além disso, usuários muitas vezes precisam compartilhar não apenas o hardware, mas a informação (arquivos, bancos de dados etc.) também.
- ▶ Resumindo, essa visão do sistema operacional sustenta que a sua principal função é manter um controle sobre quais programas estão usando qual recurso, conceder recursos requisitados, contabilizar o seu uso, assim como mediar requisições conflitantes de diferentes programas e usuários.