

# Tecnologia em Segurança da Informação

## *Programação II*

*Prof. Dr. Adilson Ferreira da Silva*

# *Programação II*

**Objetivo:** Ser capaz de desenvolver aplicativos que se utilizem de modularização, recursividade, técnicas de busca e ordenação e estruturas de dados com a adoção de linguagem procedural e técnicas de programação estruturada.

**Ementa:** Modularização, recursividade, técnicas de busca e ordenação e estruturas de dados (registro e lista) no desenvolvimento de aplicativos por meio da abordagem estruturada e implementação com linguagem procedural.

## **Bibliografia básica:**

~~CELES, Waldemar; CERQUEIRA, Renato; RANGEL, Jose Lucas. Introdução a estrutura de dados — uma introdução com técnicas de programação em C. Rio de Janeiro: Campus, 2004.~~

~~DROZDEK, Adam. Estrutura de Dados e Algoritmos em C++. São Paulo: Thomson Pioneira, 2002.~~

SHERROD, Allen. Data Structures and Algorithms for Game Developers. Clifton Park: Charles River Media (Delmar Cengage Learning), 2007.

DEITEL, Harvey M.; DEITEL, ~~Paul J. C++ como programar. 5.ed. São Paulo: Prentice Hall Brasil, 2006.~~

FORBELLONE, André Luiz. Lógica de Programação. 3.ed. São Paulo: Prentice Hall, 2005. PEREIRA, Sílvio do Lago. Estrutura de Dados: Fundamentos, Conceitos e Aplicações. 8.ed. São Paulo: Érica, 2000.

SCHILDT, Herbert. C Completo e Total. 3.ed. São Paulo: Makron Books. 1997.

~~TENENBAUM, Aaron M.; LANGSAM, Yedidiah, AUGENSTEIN, Moshe J. Estruturas de Dados usando C. São Paulo: Makron, 1995.~~

# *Método de Ensino*

## Objetivo das Aulas:

- Fundamentação Teórica
- Atividades práticas de programação



## Forma de avaliação:

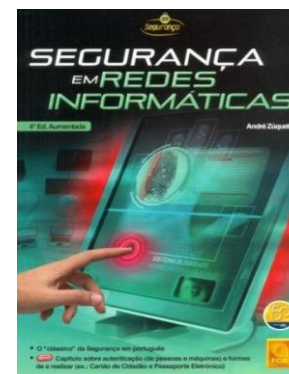
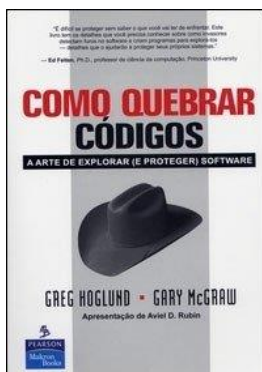
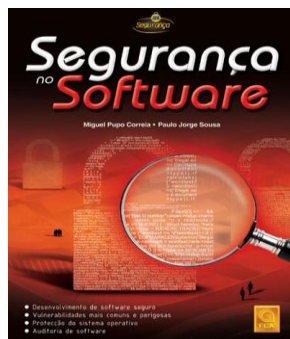
- Conforme regimento interno.
- A nota do bimestre será composta pela média das atividades (40%) e Nota da Avaliação oficial (60%)
- Caso não hajam atividades programadas pelo Professor, a média do Bimestre será equivalente a nota da prova oficial.

# *Método de Ensino*



PERGUNTAR  
NAO OFENDE!

# Segurança no Software



# *Desenvolvimento de Software*

## *Etapas*

- Levantamento e especificação de requisitos
- Análise e validação de requisitos
- Análise
- Programação
- Testes
- Preparação do ambiente
- Entrega

# *Desenvolvimento de Software*

## *Etapas*

- Levantamento e especificação de requisitos
- Análise e validação de requisitos
- Análise
- **Programação**
- Testes
- Preparação do ambiente
- Entrega

# *Segurança no Software*

## *Vulnerabilidades (algumas)*

- Buffer Overflow
- Validação de input
- Injeção de SQL
- Senhas padronizadas não modificadas pelo usuário
- Mau/Inexistente uso de Criptografia



# *Segurança no Software*

## *Vulnerabilidades (algumas)*

- **Buffer Overflow**
- **Validação de input**
- **Injeção de SQL**
- Senhas padronizadas não modificadas pelo usuário
- Mau/Inexistente uso de Criptografia

# *Programação de Computadores*

# *Características do processo de ensino aprendizagem de programação*

- *Metódico (Programar não é uma Arte)*
- *Exige disciplina dos estudantes*
- *A forma de estudar não é trivial*
- *Evolutivo: Uma etapa por vez!*

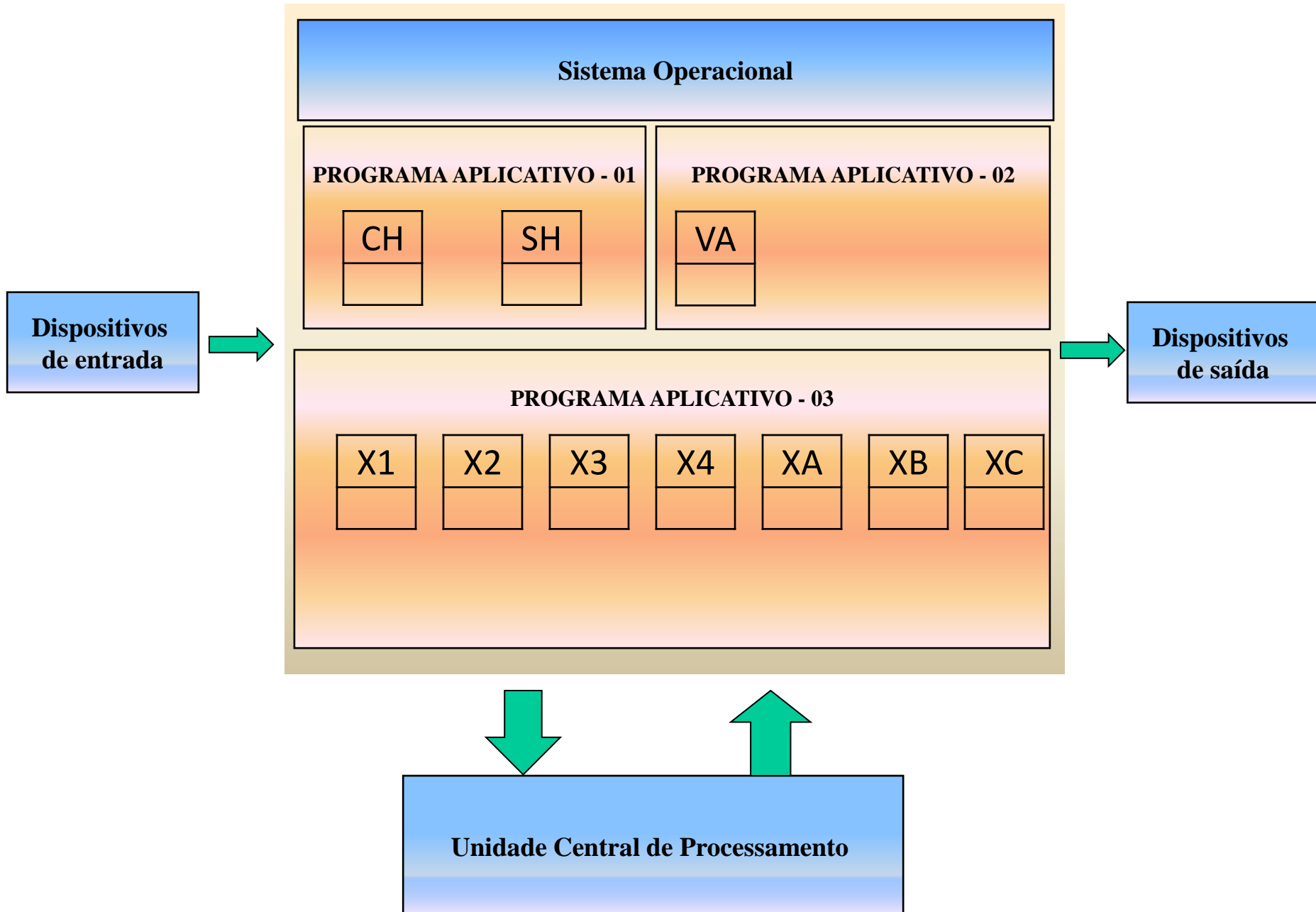
# *O que é preciso aprender para se tornar um bom programador?*

- *Arquitetura de Computadores*
- *Lógica de programação*
- *Linguagem de programação*

# *Arquitetura de Computadores* *Sistema Operacional*

*(Visão do programador)*

# Memória RAM



*Escrever um programa para calcular e mostrar o salario semanal de um funcionário, que corresponde o número de horas trabalhadas multiplicadas pelo salario hora, dados estes que serão fornecidos pelo usuário.*

### ***Dados de entrada***

*NH → número de horas trabalhadas*

*SH → salario hora*

### ***Dados de saída***

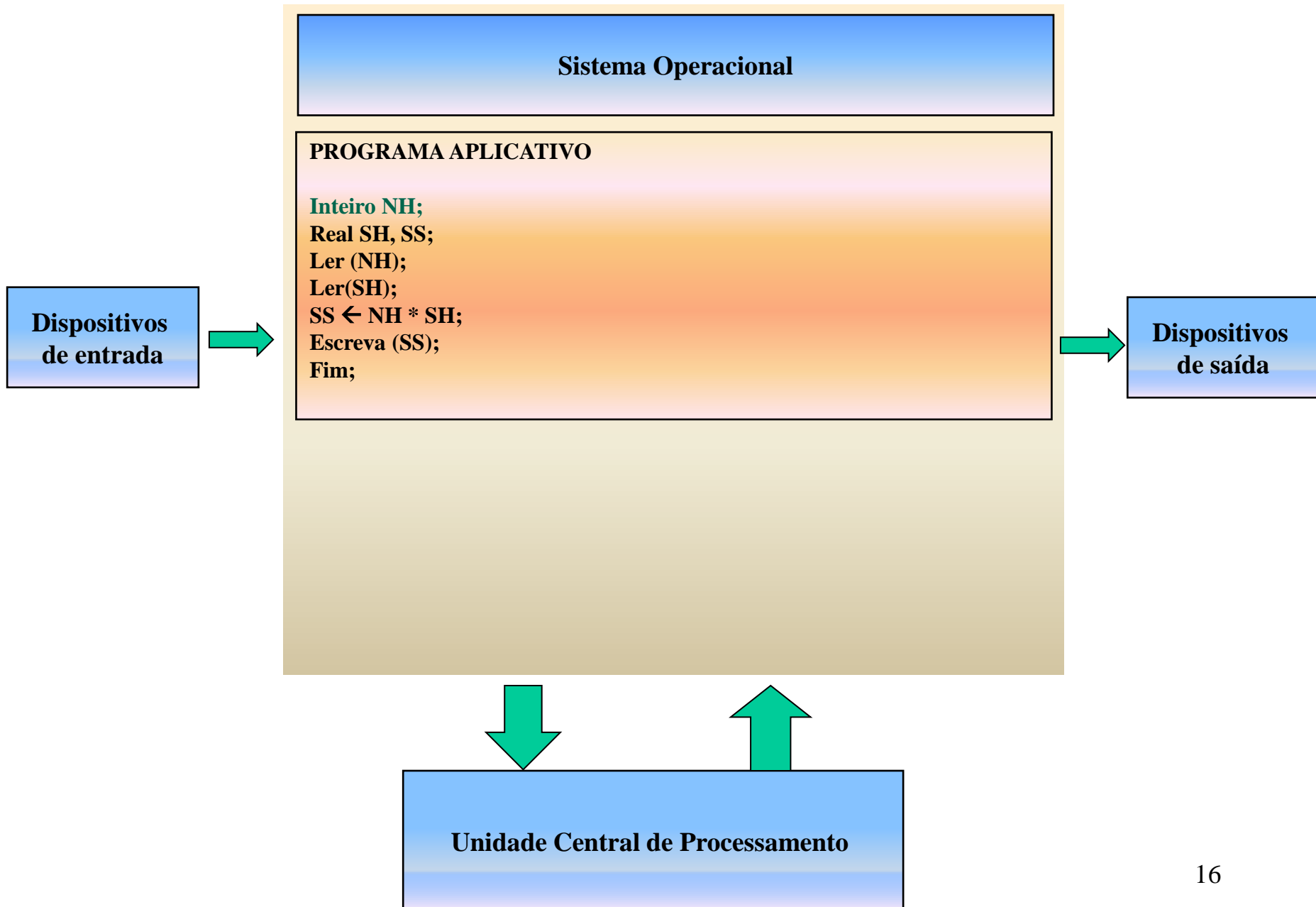
*SS → salário semanal*

***O que fazer com os dados de entrada para obter os dados de saída?***

$SS = NH * SH$

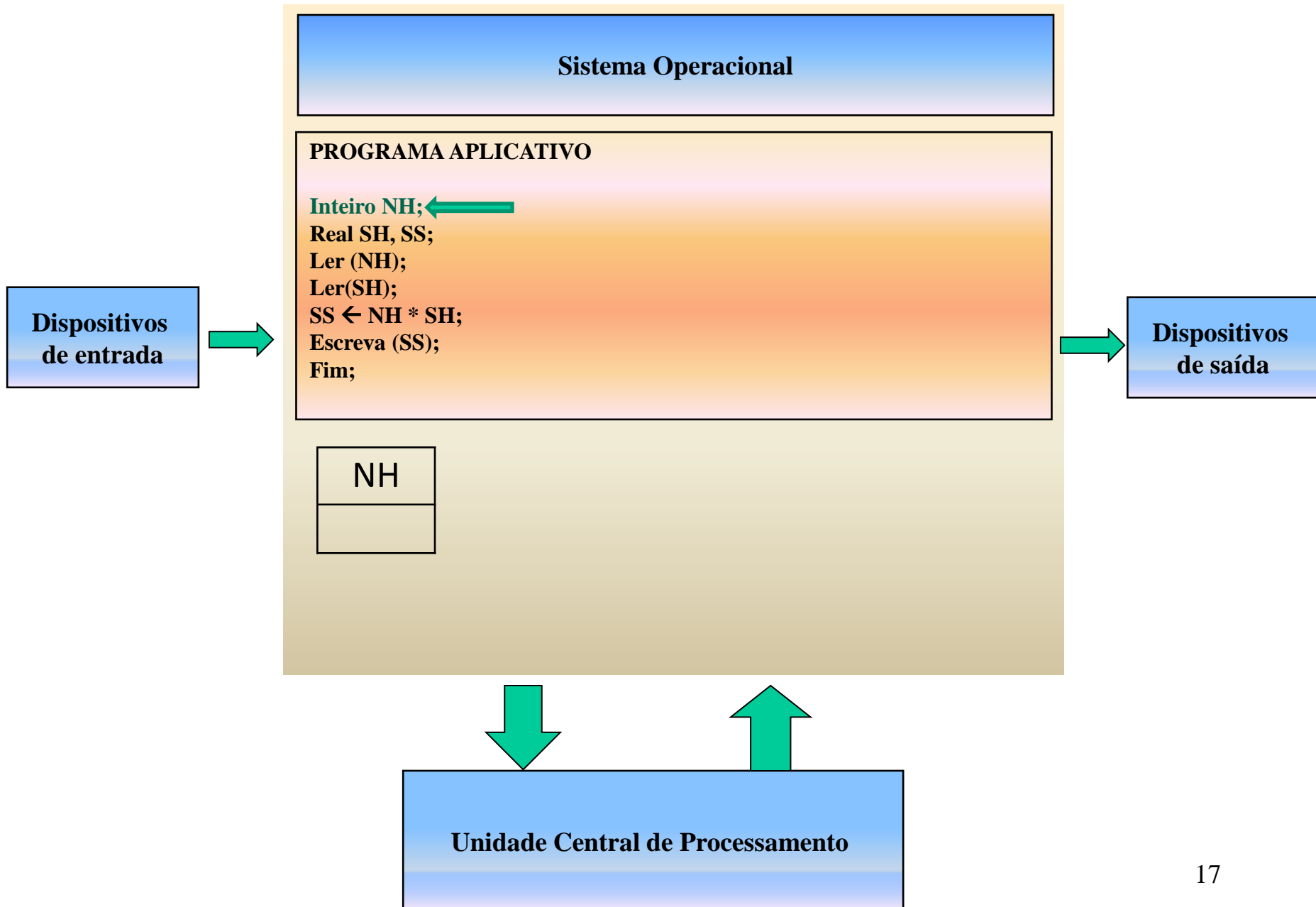
```
Inteiro NH;  
Real SH, SS;  
Ler (NH);  
Ler(SH);  
SS ← NH * SH;  
Escreva (SS);  
Fim;
```

# Memória RAM

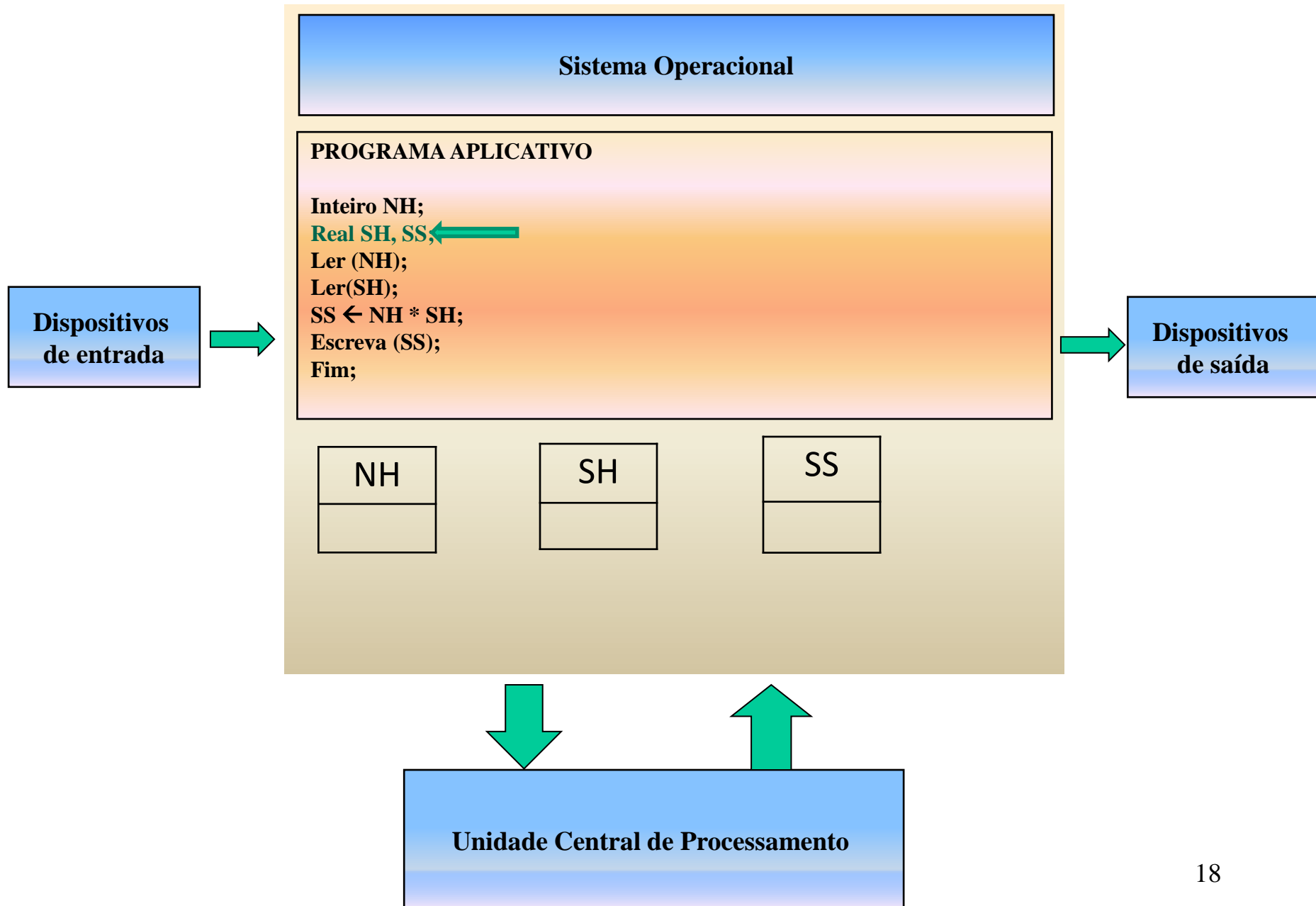




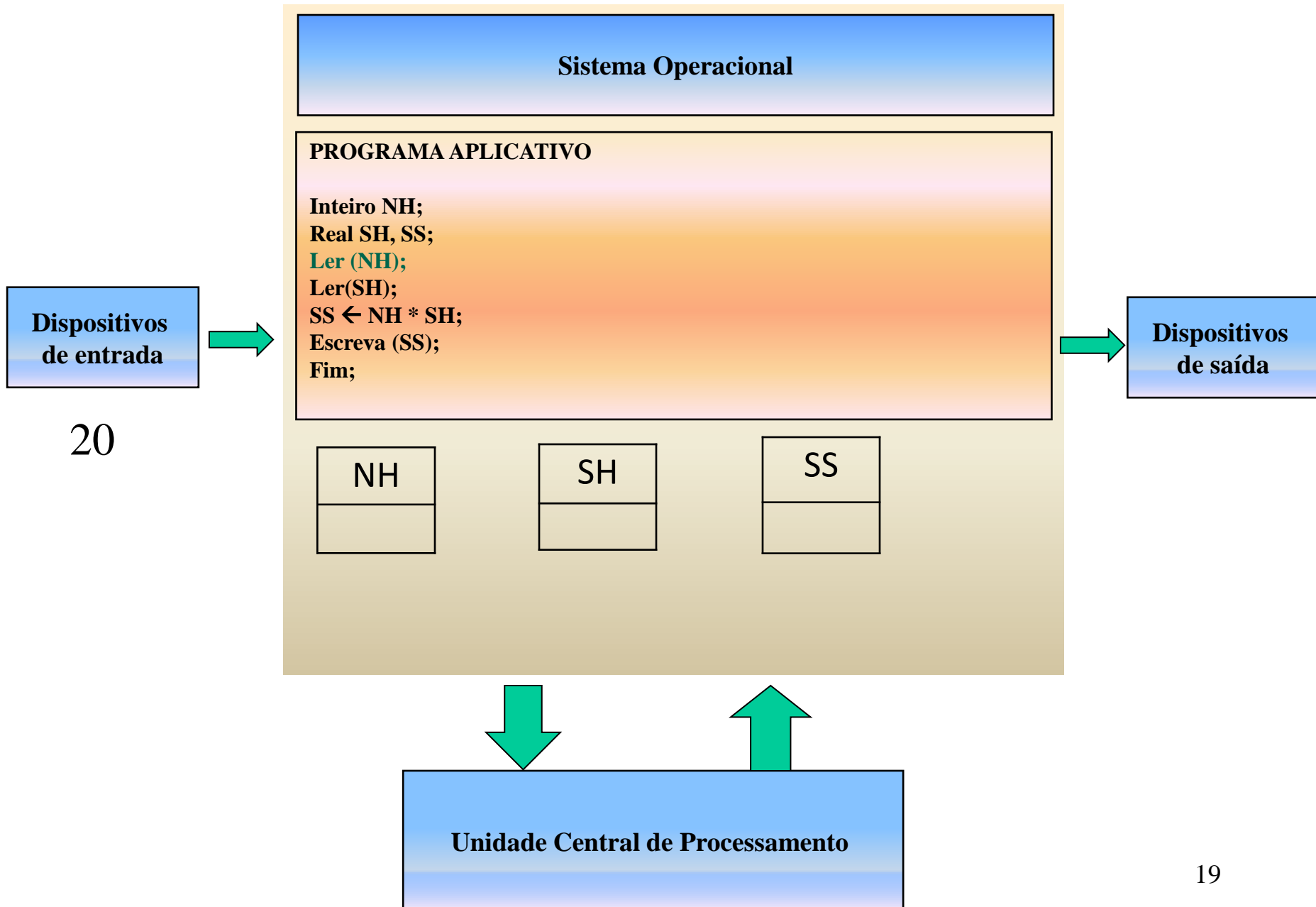
# Memória RAM



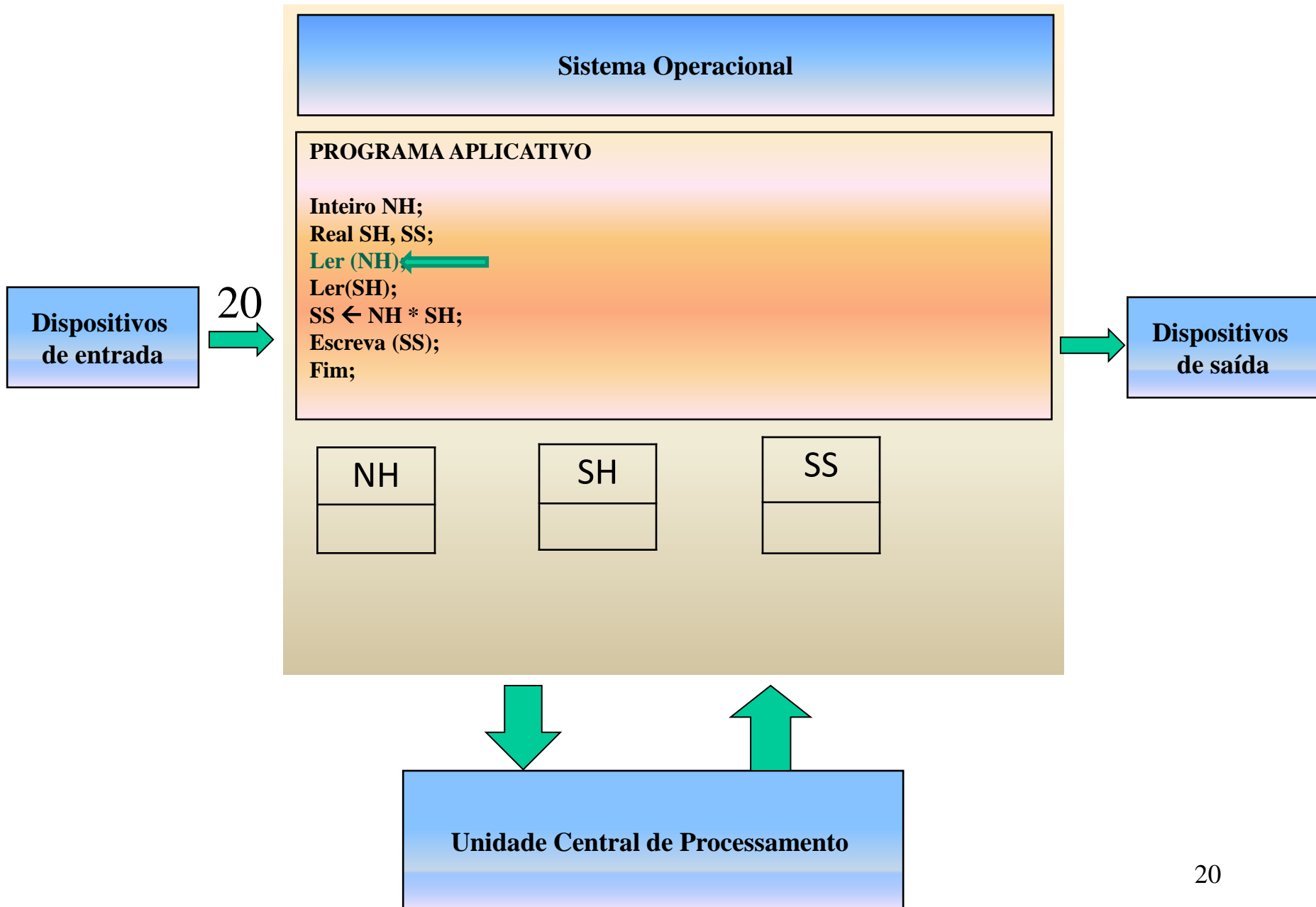
# Memória RAM



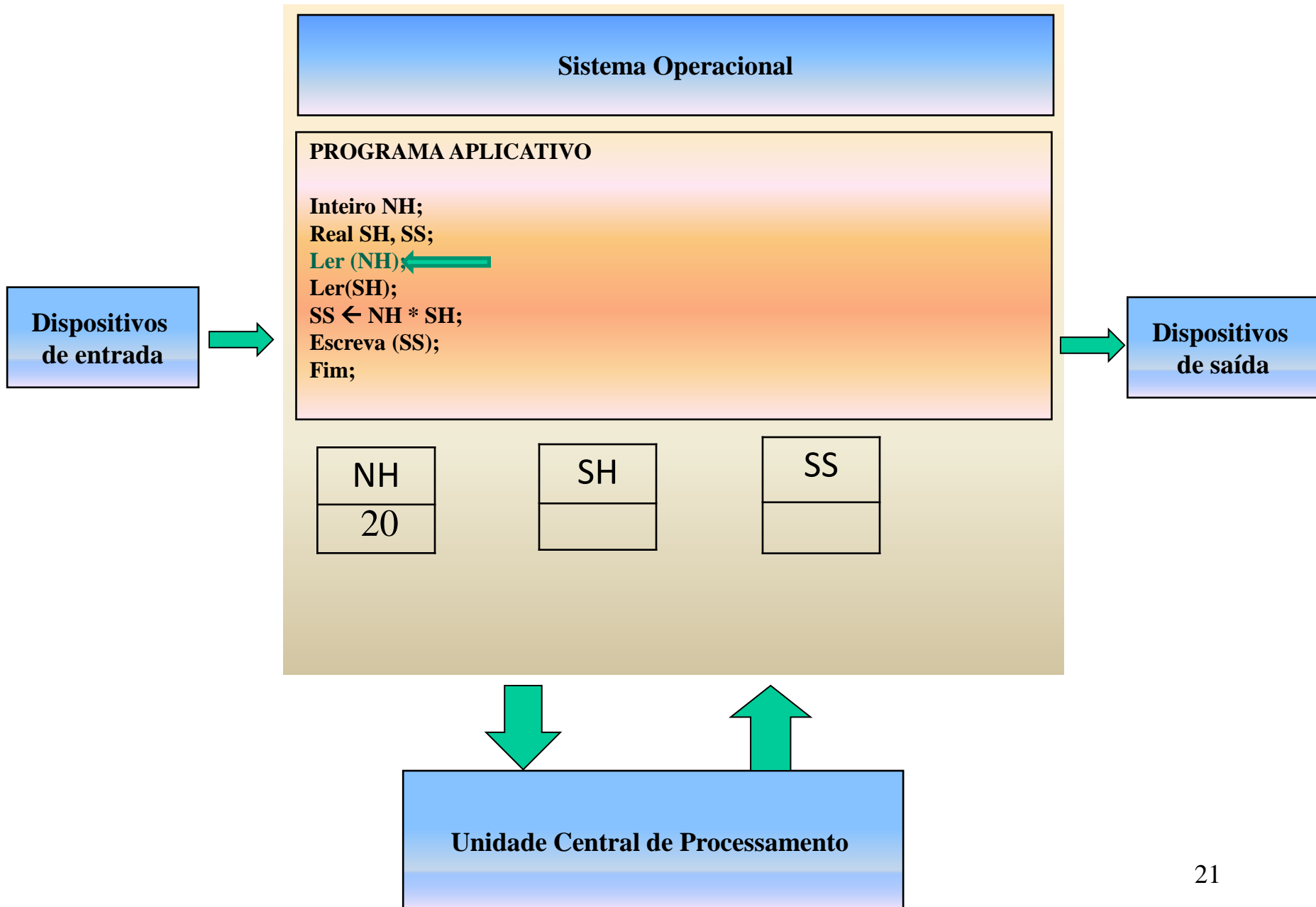
# Memória RAM



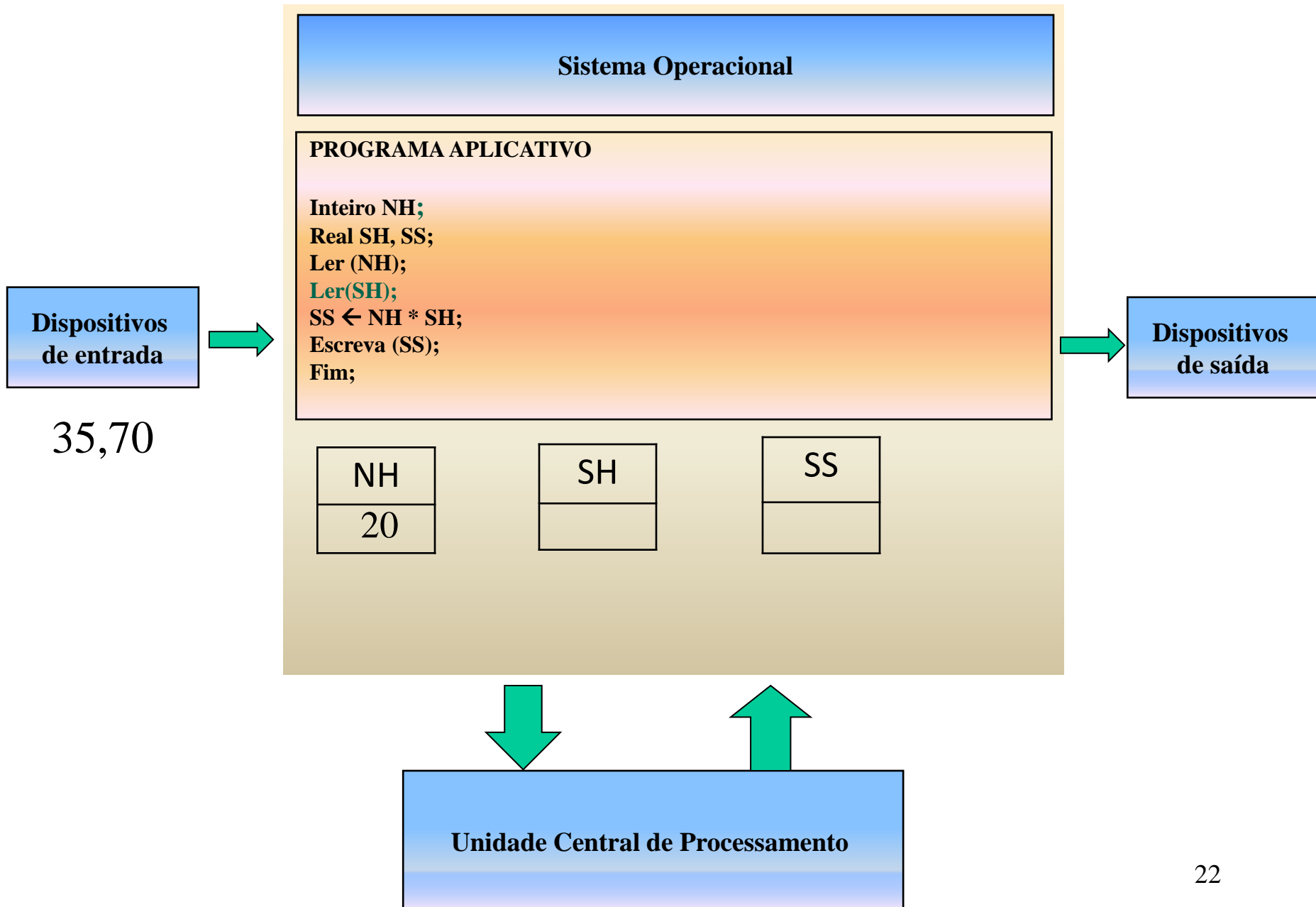
# Memória RAM



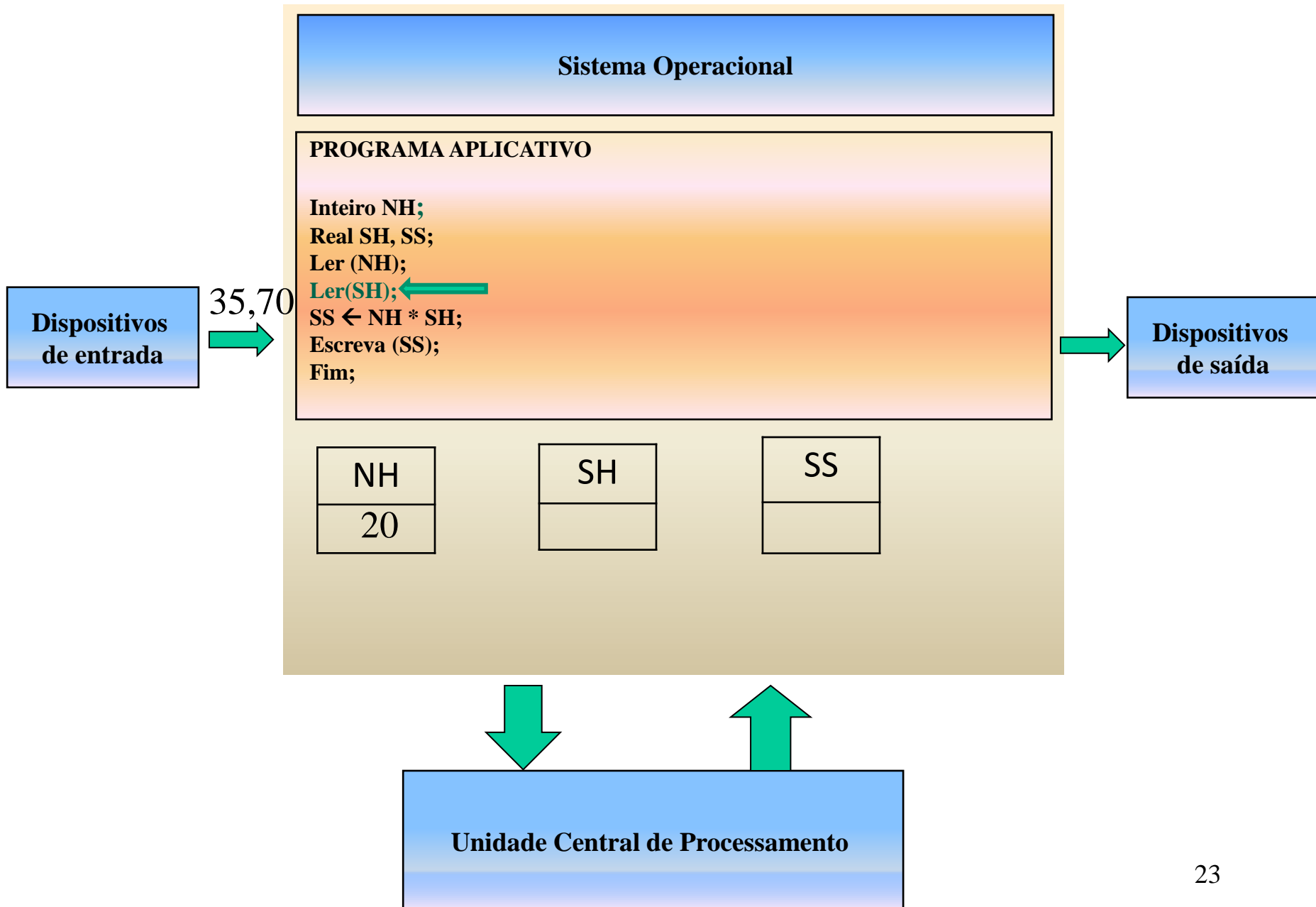
# Memória RAM



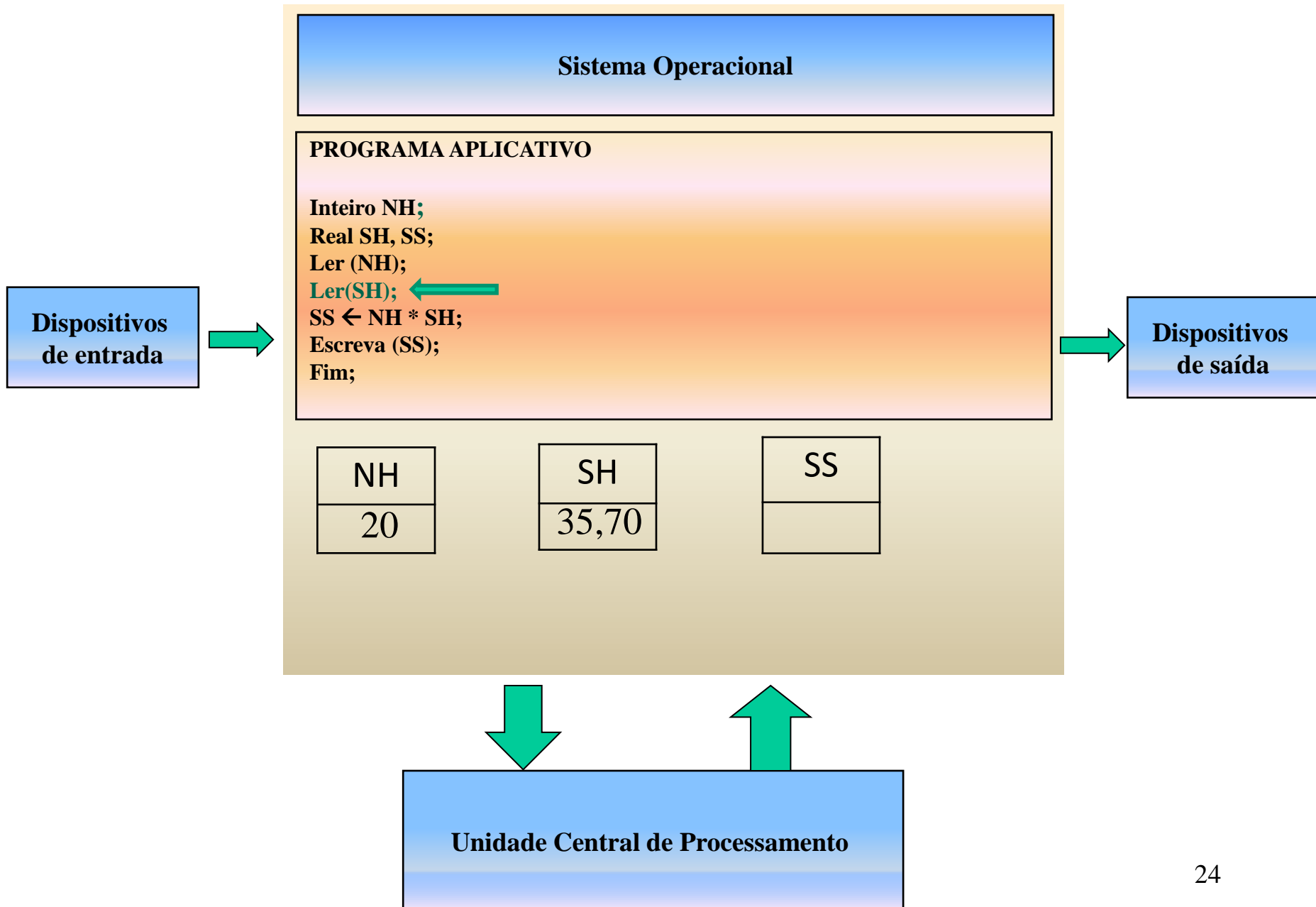
# Memória RAM



# Memória RAM

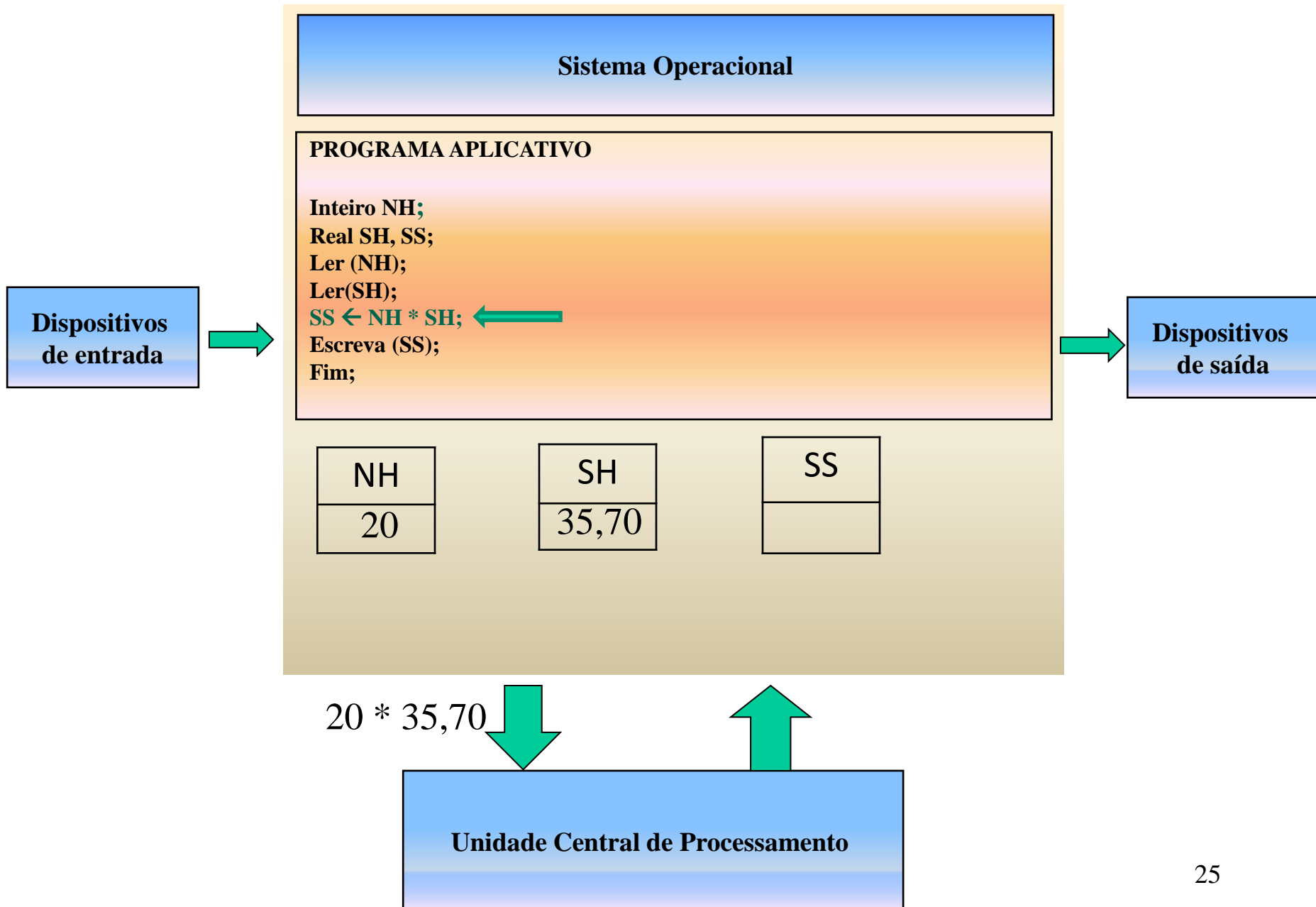


# Memória RAM

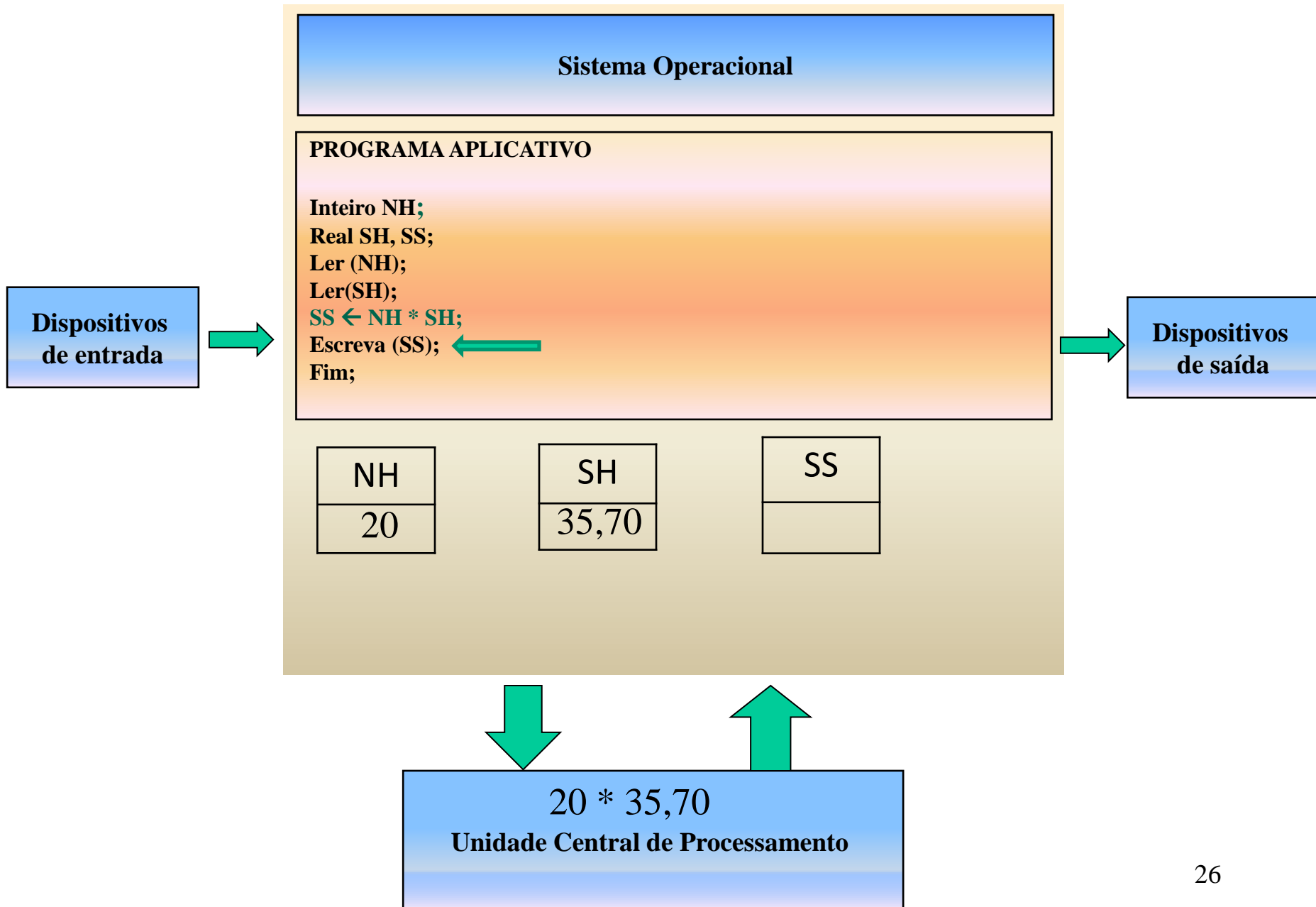




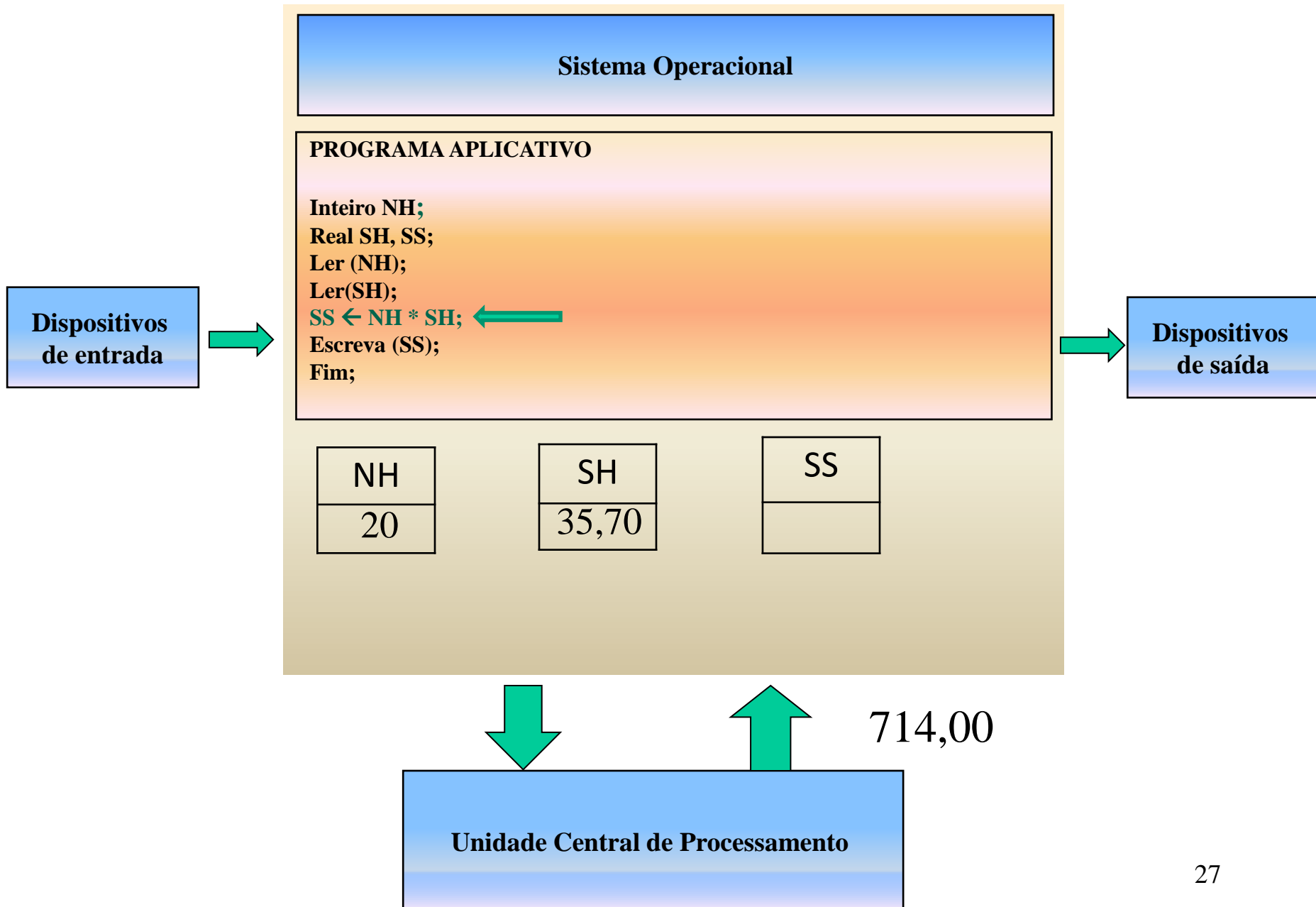
# Memória RAM



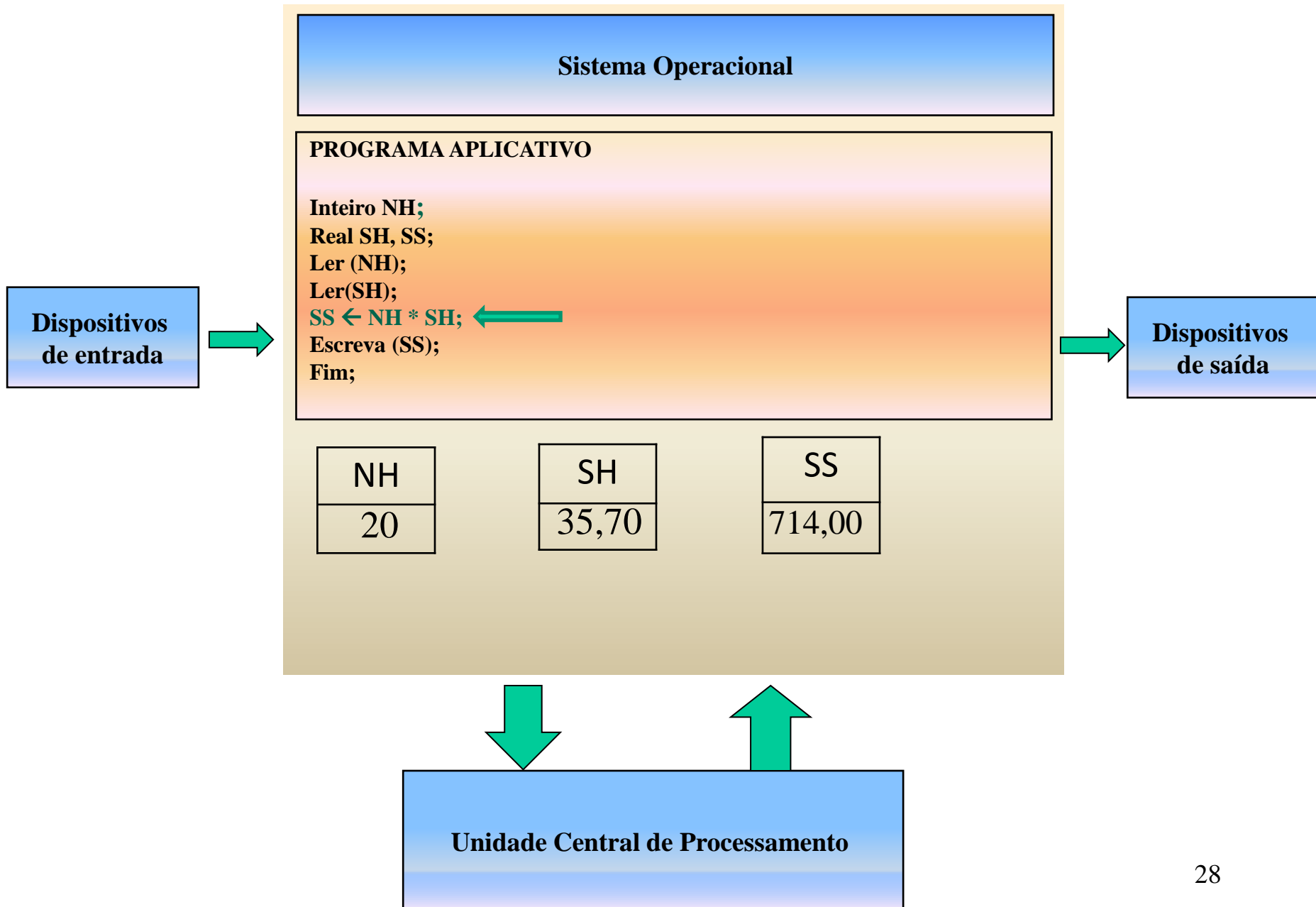
# Memória RAM



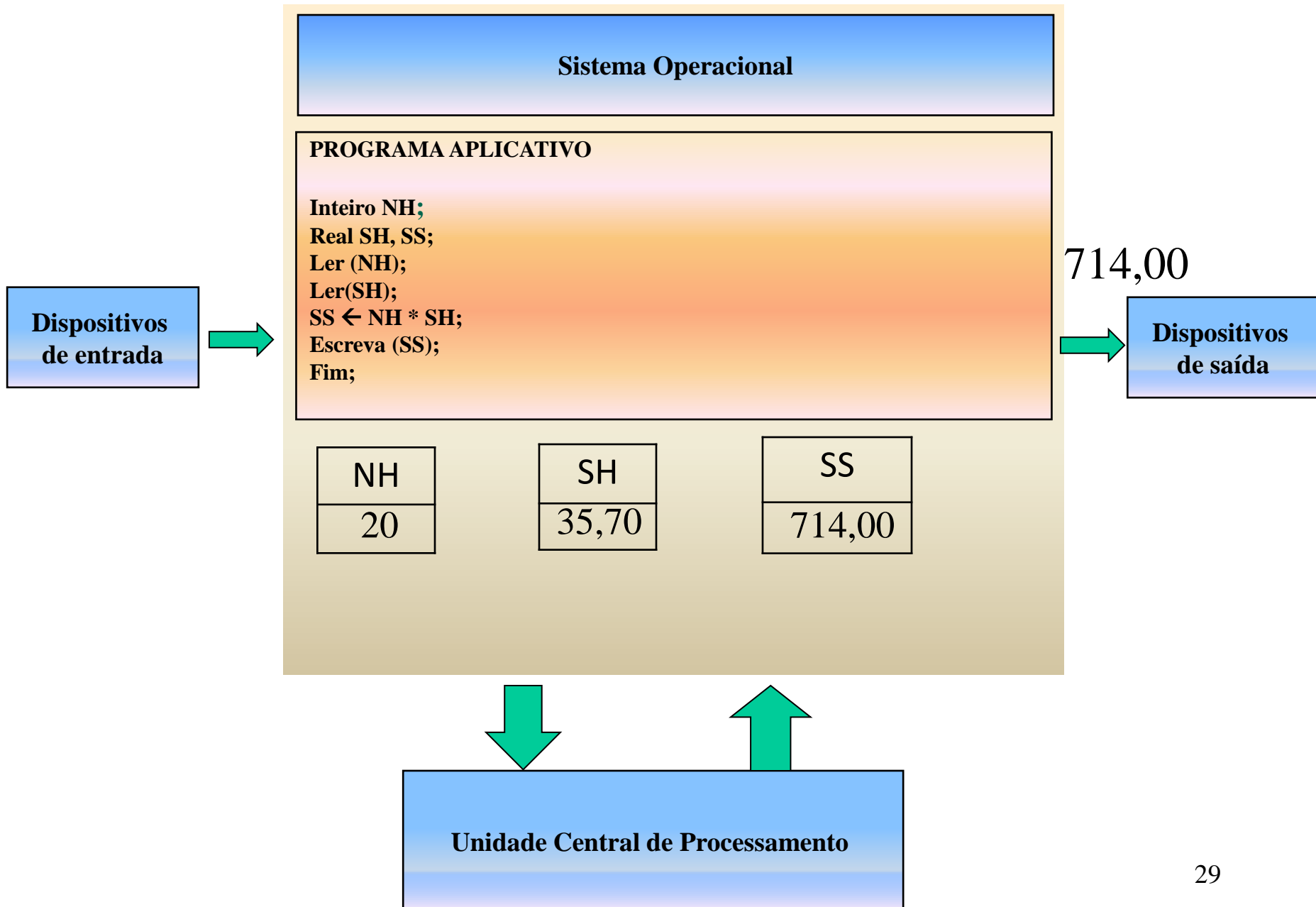
# Memória RAM



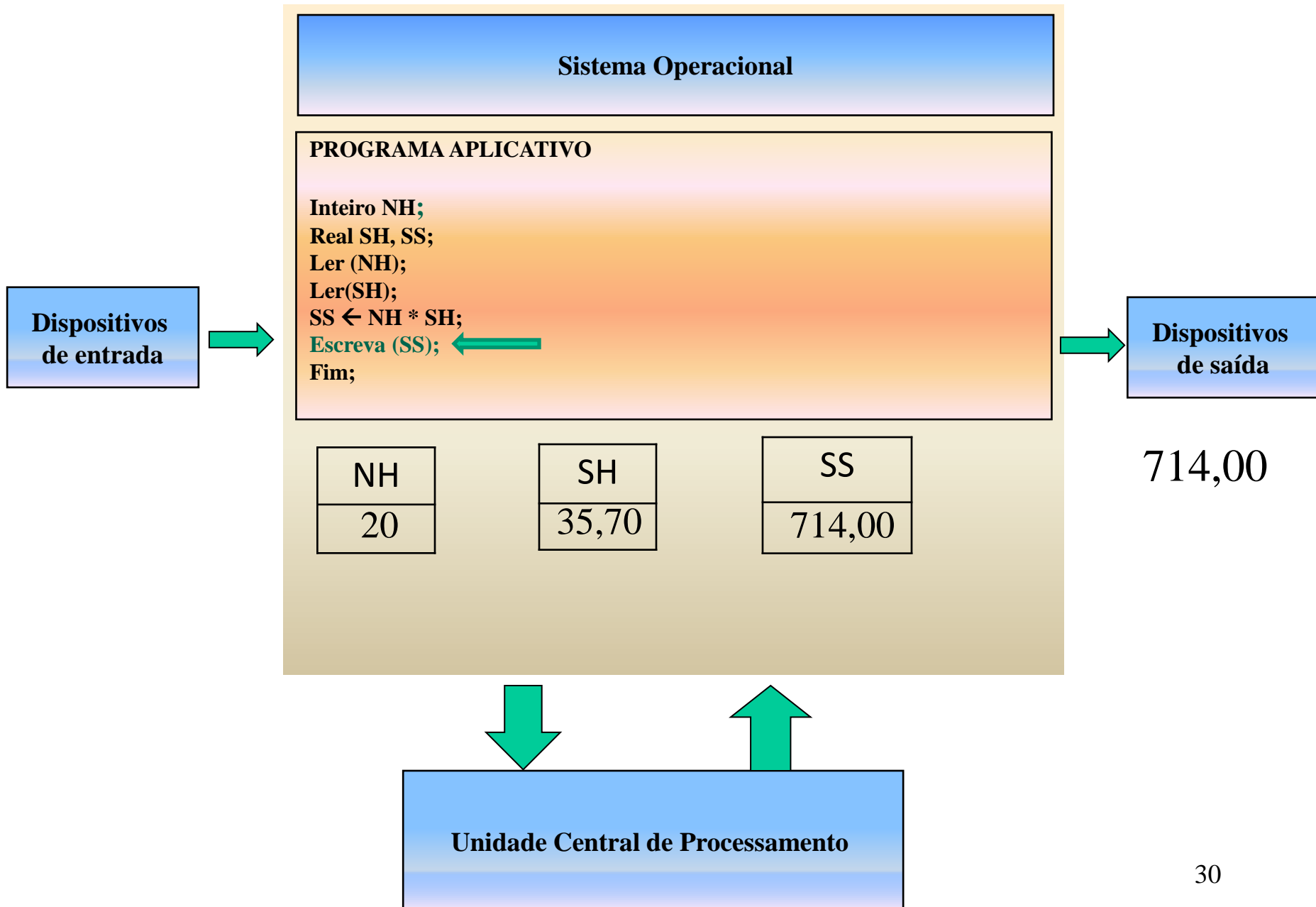
# Memória RAM



# Memória RAM



# Memória RAM



# *Codificando em Python*

```
Inteiro NH;  
Real SH, SS;  
Ler (NH);  
Ler(SH);  
SS  $\leftarrow$  NH * SH;  
Escreva (SS);  
Fim;
```