

# Gabarito Oficial Explicado

Telefone sem Fio: Arquitetura de Computadores

## Instruções para a Correção

Este documento não serve apenas para ver se o desenho está "certo". Ele serve para **ensinar o conceito**. Na fase de correção, leiam a **Explicação Didática** em voz alta para o grupo e discutam se o desenho final conseguiu capturar essa ideia.

### CARTA 01: ARQUITETURA DE VON NEUMANN

#### O que deve aparecer no desenho:

- 3 blocos: CPU, Memória e Entrada/Saída.
- Todos ligados por um caminho central (Barramento).

#### Explicação Didática (O Conceito):

- Antes de Von Neumann, computadores eram reprogramados mudando fios de lugar.
- A grande sacada aqui é o **Programa Armazenado**: as instruções do programa e os dados (variáveis) ficam guardados no **mesmo lugar** (a Memória), e a CPU busca um por um para executar. Se o desenho mostra a memória "alimentando" a CPU, o conceito está correto.

### CARTA 02: SISTEMA DE BARRAMENTOS

#### O que deve aparecer no desenho:

- **Endereço (Address Bus)**: Seta SÓ IDA da CPU para a Memória.
- **Dados (Data Bus)**: Seta IDA E VOLTA.
- **Controle (Control Bus)**: Seta saindo da CPU (mandando ler/escrever).

#### Explicação Didática (O Conceito):

- Pense na CPU como alguém pedindo pizza.
- O **Endereço** é unidirecional porque a CPU diz "Quero o dado da casa 10". A casa 10 nunca diz "Quero a CPU".
- O **Dado** é bidirecional porque a informação pode ir para a memória (salvar) ou vir dela (ler).

## CARTA 03: ORGANIZAÇÃO INTERNA DA CPU

### O que deve aparecer no desenho:

- **ULA:** Fazendo contas.
- **UC:** Dando ordens.
- **Registradores:** Guardando valores.

### Explicação Didática (O Conceito):

- A CPU é como uma cozinha.
- A **UC (Unidade de Controle)** é o Chefe de Cozinha: ele não põe a mão na massa, mas diz o que fazer.
- A **ULA (Unidade Lógica e Aritmética)** é o Cozinheiro: é quem realmente corta e frita (soma e subtrai).
- Os **Registradores** são a bancada: onde os ingredientes ficam à mão para serem usados agora.

## CARTA 04: PROGRAM COUNTER (PC)

### O que deve aparecer no desenho:

- Uma lista de instruções numeradas (10, 11, 12).
- O PC apontando para a **próxima** (ex: acabou a 10, aponta para a 11).

### Explicação Didática (O Conceito):

- O computador é "burro": ele só sabe fazer uma coisa de cada vez.
- O **PC** é como o dedo indicador de alguém lendo um livro. Ele marca onde você está para você não se perder. Sem o PC, a CPU não saberia qual é a próxima linha de código a executar.

## CARTA 05: CICLO DE BUSCA (FETCH)

### O que deve aparecer no desenho:

- Seta saindo da Memória com uma "caixinha"(instrução).
- Essa caixinha entrando no **RI (Registrador de Instrução)**.

### Explicação Didática (O Conceito):

- Antes de executar um comando (ex: SOMAR), a CPU precisa descobrir qual é o comando.
- O "Fetch"é a busca. É o ato de ir até a memória, pegar o código e trazer para dentro para ser lido. O **RI** é o "óculos"que a CPU usa para ler e entender o que precisa ser feito.

## CARTA 06: E/S MAPEADA EM MEMÓRIA

### O que deve aparecer no desenho:

- Um único "prédio" de memória.
- Endereços misturados: Variáveis e Periféricos no mesmo mapa.

### Explicação Didática (O Conceito):

- Imagine um prédio de apartamentos.
- No apto 101 mora um Dado. No apto 102 mora o Teclado.
- Para a CPU, é tudo "endereço". Ela usa o mesmo comando *LOAD* para ler o dado ou ler a tecla digitada. Não existe distinção especial para o hardware.

## CARTA 07: E/S ISOLADA

### O que deve aparecer no desenho:

- Caminhos separados ou comandos diferentes para Memória e Periféricos.

### Explicação Didática (O Conceito):

- Diferente do anterior, aqui são dois prédios separados.
- O apto 101 do prédio "Memória" não tem nada a ver com a sala 101 do prédio "Periféricos".
- A CPU precisa usar comandos diferentes (ex: *IN/OUT* para periféricos vs *LOAD/STORE* para memória) para saber em qual prédio entrar.

## CARTA 08: FLAGS DA ULA

### O que deve aparecer no desenho:

- Uma conta acontecendo e luzes/bandeiras acendendo.
- Flags: Zero, Negativo, Overflow.

### Explicação Didática (O Conceito):

- A ULA faz a conta, mas também dá um "relatório" do resultado.
- As Flags são esse relatório. Exemplo: Se você subtrair 5 - 5, o resultado é 0. A ULA acende a luzinha **ZERO** para avisar o resto do processador: "Ei, a última conta deu zero!". Isso é usado para decisões (ex: IF/ELSE).

## CARTA 09: GARGALO DE VON NEUMANN

### O que deve aparecer no desenho:

- Contraste: CPU rápida vs Memória lenta.
- Barramento engarrafado.

### Explicação Didática (O Conceito):

- Imagine uma Ferrari (CPU) presa no trânsito de uma estrada de terra (Barramento) atrás de um trator (Memória).
- Não adianta ter um motor superpotente se você não consegue receber o combustível rápido o suficiente. O desempenho do computador fica limitado pela velocidade da transferência de dados.

## CARTA 10: INTERRUPÇÃO DE HARDWARE

### O que deve aparecer no desenho:

- CPU parando o trabalho por causa de um sinal externo.
- Ação de "Salvar Contexto" (guardar o que estava fazendo).

### Explicação Didática (O Conceito):

- Imagine que você está lendo um livro e o telefone toca.
- 1. O toque é a **Interrupção**.
- 2. Antes de atender, você coloca um marcador de página (Salva o Contexto/Registros).
- 3. Você atende o telefone (Trata a Interrupção).
- 4. Você volta a ler exatamente de onde parou (Restaura o Contexto).
- Sem salvar o contexto, a CPU "perderia a página" e o programa travaria.