

**Nome:** Barbara Letícia da Silva.

**Prontuário:** CJ3029921.

**1º Semestre.**

**Professor:** Josivan Pereira da Silva.

## **Atividade de Pesquisa: Sobre Modelo *Von Neumann* e Ciclo de Execução**

### **Atividade de Pesquisa: Arquitetura de *Von Neumann* e Ciclo de Instrução**

#### **Parte - 1A.**

#### **Parte 1A - Arquitetura de *Von Neumann***

#### **1. Quem foi *John Von Neumann* e qual foi sua contribuição para a computação?**

**R:** Nascido em Budapeste em 28 de dezembro de 1903, *John Von Neumann* foi um dos mais brilhantes matemáticos e cientistas do século XX, com contribuições revolucionárias que moldaram a computação como a conhecemos hoje. De origem húngara e naturalizado americano, esse gênio multidisciplinar destacou-se não apenas na ciência da computação, mas também na física quântica, economia e engenharia nuclear até seu falecimento em 8 de fevereiro de 1957.

Sua principal herança para a computação foi a revolucionária Arquitetura de *Von Neumann*, proposta em 1945, que estabeleceu os pilares fundamentais dos computadores modernos. Esse modelo inovador introduziu três componentes essenciais: uma unidade central de processamento (*CPU*) para executar operações, um sistema de memória unificado para armazenar tanto dados quanto instruções, e barramentos de comunicação entre esses elementos. Essa abordagem representou uma mudança radical em relação aos primeiros computadores, que exigiam reconfiguração física para cada nova tarefa. A ideia de programas armazenados na memória, conceito central da arquitetura proposta por *Von Neumann*, permitiu pela

primeira vez que computadores pudessem ser reprogramados simplesmente alterando o conteúdo da memória, sem necessidade de modificações físicas. Esse princípio foi implementado no computador *EDVAC* (1949), sucessor do *ENIAC*, marcando o nascimento da programação moderna, *Von Neumann* também contribuiu significativamente para o desenvolvimento do próprio *ENIAC*, um dos primeiros computadores eletrônicos digitais.

O legado de *Von Neumann* permanece vivo na estrutura de praticamente todos os dispositivos computacionais atuais, desde smartphones até supercomputadores. Sua arquitetura continua sendo o paradigma dominante na construção de máquinas computacionais, mais de sete décadas após sua concepção. Além de revolucionar a tecnologia, suas ideias transcenderam a computação, influenciando profundamente campos como inteligência artificial, ciência de dados e modelagem econômica. Por sua genialidade visionária e contribuições multidisciplinares, *John Von Neumann* é justamente considerado um dos maiores gênios científicos da história moderna.

## **2. Quais são os cinco principais componentes da arquitetura de *Von Neumann*?**

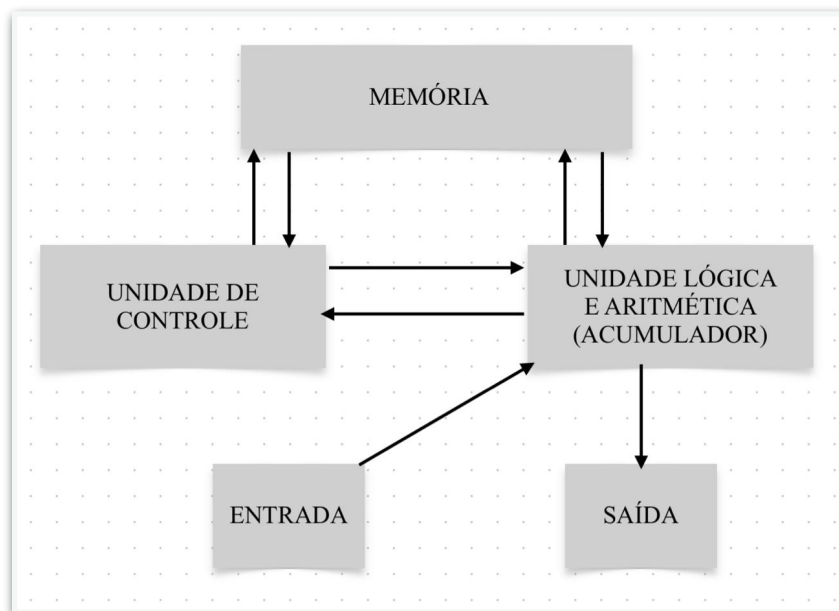
**R:** A arquitetura de *Von Neumann* (ou arquitetura de *Princeton*) é composta por cinco unidades principais, sendo elas:

- **Unidade de Memória** que é responsável por armazenar tanto os dados quanto as instruções(programas) que o computador está utilizando ou que irá executar, a memória única e compartilhada pode ser *RAM*, *ROM* ou outro tipo de memória principal.
- **Unidade de Entrada** na qual é encarregada de fazer o recebimento dos dados externos e envia-lós ao sistema, um exemplo disso são os componentes como teclado, *mouse*, *scanner*, sensores, etc.
- **Unidade Lógica Aritmética (ULA)** tem a função de executar os cálculos matemáticos utilizados para processar os dados dentro do computador, como adição, subtração, multiplicação, etc. Além de operações lógicas de comparações, *AND*, *OR*, etc. Podemos considerar que a ULA é o “cérebro matemático” do processador.

- **Unidade de Controle** é responsável por buscar, decodificar e executar as instruções armazenadas na memória, coordenando o funcionamento de todas as outras unidades do computador, além de trabalhar com registradores como o contador de programa (*PC*) e o registrador de instrução (*IR*).
- **Unidade de Saída** na qual é encarregada de enviar os resultados processados para o meio externo, um exemplo disso seria monitores, impressoras, alto-falantes, etc.

A **Figura 1** abaixo apresentará de forma gráfica como é a Arquitetura de *Von Neumann* baseada em cinco unidades principais. A **Tabela 1** apresenta o resumo gráfico da Arquitetura de *Von Neumann*.

**Figura 1.** Arquitetura de *Von Neumann*.



**Tabela 1.** Resumo Gráfico.

UNIDADE	FUNÇÃO PRINCIPAL
<b>Entrada</b>	Receber dados do mundo externo
<b>Controle</b>	Coordenar e receber instruções
<b>Lógica e Aritmética(ULA)</b>	Executar cálculos e comparações
<b>Memória</b>	Armazenar dados e instruções
<b>Saída</b>	Mostrar ou enviar resultados ao mundo externo

Fonte: **Figura 1** elaborada pelo estudante e **Tabela 1** elaborada por Prof. Dr Josivan(2025).

### **3. Explique por que a memória é considerada “única” nessa arquitetura.**

**R:** A Arquitetura de *Von Neumann* transformou a computação ao implementar uma memória unificada que armazena simultaneamente programas e dados no mesmo espaço, substituindo os sistemas anteriores que requeriam estruturas físicas separadas para cada função. Essa inovação trouxe flexibilidade inédita, permitindo que instruções fossem tratadas como dados e possibilitando a modificação dinâmica de programas durante a execução, a criação de programas geradores de código e o desenvolvimento de técnicas como compilação e interpretação. O modelo simplificou o *hardware* ao empregar um único barramento para ambos os tipos de informação, otimizando o uso da memória através de alocação dinâmica sem divisões fixas entre código e dados.

Diferente do *ENIAC*, que por sua vez necessitava de reconfiguração física para cada nova tarefa, ou da Arquitetura *Harvard* (com memórias separadas), a proposta de *Von Neumann* alcançou um equilíbrio perfeito entre eficiência e versatilidade, tornando-se base para computadores de propósito geral que executam diversas funções apenas trocando programas na memória. Apesar do "Gargalo de *Von Neumann*" (limitação no barramento compartilhado), sua simplicidade e adaptabilidade mantiveram-na como alicerce da computação moderna, desde os primeiros computadores até os atuais. Essa revolução não só possibilitou a programação contemporânea como também estabeleceu os princípios fundamentais que ainda direcionam a evolução tecnológica.

### **4. Por que a arquitetura de *Von Neumann* pode chegar o chamado “gargalo de *Von Neumann*”?**

**R:** Basicamente, a via de transmissão de dados entre a *CPU* e a memória limita de certa forma a velocidade do processamento de um computador. Os barramentos têm esta função e a troca de dados entre o processador e a memória fica limitada pela taxa de transferência de dados que esses barramentos são capazes de proporcionar, que em geral são bem menores que a capacidade dos processadores, sendo um fator limitador da velocidade atingida no processamento das informações. Esse problema aumenta a cada nova geração e o desenvolvimento da tecnologia com maior número de barramentos é uma das soluções adotadas pelos fabricantes da tecnologia.

## Parte 2A - Ciclo de Instrução

### 5. Quais são as três etapas principais do ciclo de instrução?

**R:** O ciclo de instrução é um processo cíclico que a *CPU* executa para desempenhar cada uma das instruções de um programa, ele ocorre constantemente milhões ou até mesmo bilhões de vezes por segundo, durante o tempo em que o computador está ligado. As três etapas principais são: **Busca, Decodificação e Execução.**

- **Busca(*Fetch*):** A *CPU* irá localizar a próxima instrução na memória (endereço adequado pelo contador de programa - *PC*). Essa instrução é reproduzida da memória para o registrador de instrução (IR).
- **Decodificação(*Decode*):** A Unidade de Controle compreende a instrução como lida, reconhece o que deve ser feito e quais são os componentes da *CPU* que devem ser operados.
- **Execução(*Execute*):** A *CPU* irá executar uma determinada ação, se:
  1. For um cálculo a *ULA* é ativada.
  2. For uma leitura ou escrita ela irá acessar a memória.
  3. For para exibir algo, enviará para o dispositivo de saída.

### 6. O que acontece em cada uma dessas etapas (resuma em uma frase para cada)?

**R: Busca** - “ Irá buscar a próxima instrução para ser executada.”

**Decodificação** - “ Compreender o que a instrução irá mandar fazer.”

**Execução** - “ Tem de realizar a ação de somar, mover, comparar, exibir, etc.”

### 7. Para que serve o contador de programa (*PC*) nesse processo?

**R:** O Contador de Programa (*PC*) é o guia essencial que mantém a *CPU* no caminho certo durante a execução de um determinado programa. Ele opera como um marcador inteligente que sempre sabe qual é a próxima instrução a ser executada, armazenando seu endereço na memória. Geralmente, ele avança automaticamente de uma instrução para outra em sequência, mas irá mostrar a sua verdadeira utilidade quando o programa precisa tomar decisões, saltando para diferentes partes do código em caso de condicionais, loops ou chamadas de funções.

O *PC* também atuará como um "salvador de progresso", na qual irá guardar sua posição atual quando o sistema necessitar de lidar com interrupções ou intercalar

entre as tarefas, onde possibilitará a voltar exatamente de onde parou. Essa combinação de sequenciamento previsível e flexibilidade para saltos controlados é o que torna possível desde programas simples até sistemas operacionais complexos. Sem esse componente fundamental, a *CPU* não teria como coordenar o fluxo de execução, ficando perdida em meio às instruções do programa.

### **8. O que a Unidade de Controle faz durante o ciclo de instrução?**

**R:** A Unidade de Controle (UC) é considerada o cérebro da *CPU* na qual comanda todo o ciclo de instruções. Inicia-se buscando a próxima instrução na memória utilizando do Contador de Programas (*PC*) e armazena no Registrador de Instrução (IR). Imediatamente, decodifica essa instrução para compreender que operação fazer (como somar, mover dados ou saltar para outro código).

### **Parte 3A - Reflexão final**

### **9. Por que entender essa arquitetura ainda é importante hoje, mesmo com computadores modernos muito mais potentes?**

**R:** A arquitetura de *Von Neumann* tem um papel fundamental na computação moderna porque determinou os princípios estruturais que continuam conduzindo o desenvolvimento de sistemas computacionais, mesmo sete décadas após sua concepção. Seu modelo unificado, onde programas e dados compartilham a mesma memória e são processados sucessivamente através do ciclo *fetch-decode-execute* tornou-se a base invisível por trás de todos os dispositivos que usamos hoje, desde *smartphones* até supercomputadores.

O engenho dessa arquitetura está em sua simplicidade organizacional ao separar claramente as funções da *CPU*, memória e barramentos, criou um paradigma escalável que concedeu a evolução exponencial da tecnologia. Processadores modernos implementam sofisticadas otimizações como *pipelines* profundos, múltiplos núcleos e hierarquias de *cache*, mas todos mantêm a essência do modelo original. Mesmo arquiteturas alternativas, como a *Harvard* utilizada em microcontroladores, frequentemente disputam características de *Von Neumann* para manter compatibilidade com sistemas de *software* já existentes.

Por fim, o legado de *Von Neumann* vai além da história da computação, seus fundamentos tornaram-se tão essenciais quanto as leis da física para a engenharia. Enquanto houver evolução nos computadores, a arquitetura que ele criou continuará sendo o princípio constitutivo da tecnologia digital, uma prova do impacto de ideias genuinamente transformadoras.

**10. Você acha que essa arquitetura ainda influencia os computadores atuais? Justifique com um exemplo.**

**R:** A arquitetura de *Von Neumann* ainda é o alicerce oculto por trás de toda computação moderna, mesmo nos sistemas mais sofisticados. Seu projeto inovador, desenvolvido na década de 1940, ainda molda diretamente a estrutura dos processadores atuais, desde os chips de celulares até os maiores supercomputadores.

Um exemplo marcante dessa influência duradoura são os processadores *multicore*, como o *Intel Core i9* ou o *AMD Ryzen*. Cada núcleo desses processadores ainda segue rigorosamente o ciclo básico de busca-decodificação-execução estabelecido por *Von Neumann*. Quando você abre um aplicativo como o *Photoshop*, o processador recupera as instruções da *RAM* (que, seguindo o modelo original, armazena dados e programas juntos), interpreta esses comandos e os executa em sequência, exatamente como nos primeiros computadores.

Mesmo com avanços modernos, como *pipelines* de 14 estágios, execução fora de ordem e sistemas de *cache* hierárquico (L1, L2, L3), a arquitetura fundamental permanece inalterada. Até o conhecido "gargalo de *Von Neumann*", o limite físico na troca de dados entre *CPU* e memória ainda é um obstáculo atual, enfrentado com tecnologias como memórias *DDR5* ultrarrápidas e *caches* mais amplas.

Surpreendentemente, mesmo arquiteturas paralelas, como *GPUs*, quando usadas em sistemas convencionais, precisam se ajustar ao modelo de memória unificada de *Von Neumann* para garantir compatibilidade. Isso demonstra como seu padrão se tornou onipresente: toda a indústria de *software* foi erguida sobre esses fundamentos.

A prova final da relevância dessa arquitetura está no seu próprio celular quando você alterna entre aplicativos, o processador gerencia múltiplas tarefas salvando e restaurando estados, exatamente como um computador dos anos 1950 faria, só que

em velocidades bilhões de vezes maiores. A arquitetura de *Von Neumann* não apenas influencia os computadores atuais, mas os define. Seu legado persiste porque, apesar de todas as inovações, nenhum modelo alternativo ofereceu uma combinação tão eficiente de simplicidade, versatilidade e desempenho para a computação de modo geral.

**Atividade de Pesquisa: Máquinas que Seguem ou Não o Modelo de *Von Neumann***  
**Parte - 1B.**

**Objetivo:**

Pesquisar e analisar 6 dispositivos tecnológicos populares, identificando quais seguem o modelo de arquitetura de *Von Neumann* e quais não seguem, com base nas 5 unidades fundamentais dessa arquitetura.

**Máquinas para Pesquisar:**

**1. *iPhone 13* - Segue o modelo de arquitetura de *Von Neumann***

**R:** O chip A15 *Bionic* do *iPhone 13* exemplifica a permanência da arquitetura de *Von Neumann* na computação moderna. Seus núcleos de processamento mantêm o ciclo *fetch-decode-execute* clássico, enquanto a memória unificada *LPDDR4X* preserva o princípio de armazenamento conjunto de dados e instruções. Embora incorpore avanços como co-processadores especializados (*GPU/NPU*), *cache* hierárquico e paralelismo, a arquitetura fundamental permanece inalterada. Até a *Neural Engine* opera dentro deste paradigma, demonstrando sua adaptabilidade. O persistente gargalo de *Von Neumann* é mitigado por essas inovações, comprovando que mesmo em um *chip* com 15 bilhões de transistores, os princípios concebidos décadas atrás continuam fundamentais, combinando simplicidade, flexibilidade e eficiência de forma insuperável.



## **2. *Arduino Uno* (microcontrolador) - Não segue o modelo de arquitetura de *Von Neumann***

**R:** O *Arduino Uno*, baseado no microcontrolador *ATmega328P*, exemplifica a eficácia da arquitetura *Harvard* modificada para sistemas embarcados. Diferente da arquitetura *Von Neumann*, seu design separa fisicamente a memória *Flash* (programas) da *SRAM* (dados), garantindo maior previsibilidade e eficiência na execução de tarefas únicas por *ciclo* de *clock*. Essa simplicidade intencional, que dispensa sistemas operacionais complexos e paralelismo avançado, onde resulta em maior confiabilidade e determinismo, características essenciais para automação e prototipagem. A versão modificada da arquitetura *Harvard*, com barramento único mas memórias independentes, equilibra desempenho e simplicidade. O *Arduino Uno* demonstra que, em contextos específicos, soluções especializadas e limitadas podem superar sistemas mais versáteis, destacando o valor da otimização para fins particulares.

Executa uma única tarefa por vez, sem separar claramente memória e programa.

## **3. *Smartphone Xiaomi Redmi Note 10* - Segue o modelo de arquitetura de *Von Neumann***

**R:** O *Xiaomi Redmi Note 10*, com seu processador *Qualcomm Snapdragon 678*, exemplifica a relevância contínua da arquitetura *Von Neumann* na computação moderna. Seus núcleos *ARM Cortex-A76* e *A55* executam o clássico ciclo *fetch-decode-execute*, enquanto a memória unificada *LPDDR4X* mantém o princípio de armazenamento conjunto de dados e instruções.

Notavelmente, o dispositivo demonstra como esses fundamentos se adaptam às exigências atuais: o sistema *Android* implementa multitarefa avançada sobre essa base arquitetônica tradicional. Assim como em *smartphones premium*, essa solução acessível prova que a arquitetura *Von Neumann* permanece essencial, mostrando que a inovação eficaz muitas vezes reside na evolução de conceitos estabelecidos, não em sua substituição.

O *Redmi Note 10* personifica a durabilidade de princípios computacionais fundamentais, comprovando que ideias verdadeiramente revolucionárias transcendem gerações tecnológicas, mesmo em dispositivos do dia a dia.

#### **4. *FPGA* (como *Xilinx Spartan-6*) - Não segue o modelo de arquitetura de *Von Neumann***

**R:** As *FPGAs*, como a *Xilinx Spartan-6*, introduzem uma transformação radical em relação ao modelo clássico de *Von Neumann*. A essência disso está na capacidade de ajustar o *hardware* em tempo real onde facilitará a customização para cada tarefa específica. Ao contrário dos *CPUs* tradicionais, dispensam o ciclo de busca, decodificação e execução, bem como a hierarquia de memória padronizada, dados e operações se materializam como arranjos físicos de portas lógicas programáveis.

Essa estrutura proporciona concorrência própria por meio de circuitos independentes, opondo-se frontalmente ao processamento serializado dos sistemas convencionais. A *Spartan-6* ilustra uma metodologia orientada por fluxo de dados, desafiando as delimitações entre lógica programável e *hardware* fixo.

Embora não representem o fim dos paradigmas estabelecidos, as *FPGAs* destacam-se em contextos que exigem precisão temporal, desempenho acelerado e otimização dedicada (como manipulação de sinais digitais ou sistemas embarcados críticos). Sua ascensão comprova que o progresso da computação pode trilhar vias distintas, ampliando o espectro de soluções com estratégias inovadoras que divergem das abordagens tradicionais.

*Hardware* reconfigurável — sem ciclo fixo de busca-decodificação-execução.

#### ***Google TPU (Tensor Processing Unit)* - Não segue o modelo de arquitetura de *Von Neumann***

**R:** A *Tensor Processing Unit (TPU)* do *Google* representa uma ruptura radical com a arquitetura *Von Neumann*, redefinindo os princípios computacionais para a era da *IA*. Especializada em cálculos tensoriais, a *TPU* assume o lugar do ciclo tradicional *fetch-decode-execute* por um *design* altamente paralelizado otimizado para multiplicações matriciais, operação fundamental em redes neurais. Sua arquitetura de memória inovadora elimina o modelo unificado, utilizando *buffers* locais estrategicamente

posicionados para máximo desempenho. Cada componente é pré-configurado para fluxos de dados neurais, eliminando a sobrecarga de decodificação de instruções.

Mais que um acelerador, a *TPU* encarna um novo paradigma computacional, um modelo que é baseado em fluxo de dados onde irá priorizar a especialização sobre generalidade. Enquanto processadores convencionais evoluem incrementalmente, a *TPU* demonstra que os avanços mais significativos em *IA* exigem repensar os fundamentos da computação, provando que as maiores inovações surgem quando ousamos questionar até os princípios mais estabelecidos.

Usado em *IA* e *machine learning*; não segue o modelo sequencial tradicional.

### **5. PlayStation 5 - Segue o modelo de arquitetura de Von Neumann**

**R:** O *PlayStation 5* exemplifica a adaptabilidade da arquitetura *Von Neumann* na era moderna, combinando componentes de última geração com princípios computacionais fundamentais. Seu sistema baseado na arquitetura *AMD Zen 2 (CPU)* e *RDNA 2 (GPU)* mantém o ciclo clássico *fetch-decode-execute*, utilizando memória unificada *GDDR6* que preserva o conceito original de *Von Neumann*.

Apesar de seus avanços tecnológicos, incluindo capacidades avançadas de *ray tracing* e um *SSD* ultrarrápido, o *PS5* demonstra como inovações radicais podem ser integradas dentro do paradigma tradicional. O *console* não rejeita os fundamentos, mas os estende, mostrando que a arquitetura *Von Neumann* permanece suficientemente flexível para suportar até mesmo as demandas mais exigentes da computação gráfica moderna.

Esta implementação prova que os princípios estabelecidos podem evoluir harmonicamente com novas tecnologias, oferecendo desempenho de ponta sem rupturas arquitetônicas. O *PS5* representa a maturidade da computação contemporânea, onde tradição e inovação coexistem para criar sistemas poderosos e eficientes.

### **6. Nintendo Switch (modelo 1) - Segue o modelo de arquitetura de Von Neumann**

**R:** A *Nintendo Switch* exemplifica a versatilidade atemporal da arquitetura *Von Neumann* em um contexto moderno e inovador. Seu *chip Tegra X1*, com núcleos *ARM Cortex-A57*, mantém o clássico ciclo *fetch-decode-execute*, enquanto os 4GB de

memória *LPDDR4* unificada preservam o princípio fundamental de armazenamento conjunto de dados e instruções.

O *console* demonstra como esses conceitos tradicionais se adaptam a um formato híbrido único, suportando desde um sistema operacional completo até jogos AAA em modo portátil. A *Switch* prova que a inovação radical em *design* industrial e experiência de usuário pode coexistir com arquitetura computacional estabelecida.

Mais que um *console*, a *Switch* representa um caso de estudo sobre como princípios fundamentais da computação podem ser recontextualizados com sucesso, transformando uma arquitetura clássica no coração de uma experiência de jogo revolucionária e singular no mercado.

## **Parte 1B – Instruções**

Você deve:

1. Pesquisar o funcionamento de cada máquina listada.
2. Verificar se ela possui (ou simula) as **5 unidades da Arquitetura de Von Neumann**:
  - **Unidade de Entrada**
  - **Unidade de Controle**
  - **Unidade Lógica e Aritmética**
  - **Unidade de Memória**
  - **Unidade de Saída**
3. Para **cada máquina**, responda:
  - **Ela segue o modelo de Von Neumann? Por quê?**
  - **Quais das 5 unidades ela possui (ou simula)?**
4. Por fim, **escolha uma máquina de cada grupo** (1 que segue e 1 que não segue) e justifique sua escolha.

## R: Análise profunda dos dispositivos:

### 1. **iPhone 13** (*Apple A15 Bionic*)

Segue a arquitetura de *Von Neumann*? (X) Sim / () Não

Justificativa:

- **Unidade de Entrada:** Sensores, microfone, touchscreen.
- **Unidade de Saída:** Tela, alto-falantes, vibração.
- **Unidade de Memória:** RAM LPDDR4X (unificada para dados e programas).
- **Unidade de Controle:** CPU Apple Avalanche/Blizzard (gerencia instruções).
- **ULA:** Presente nos núcleos da CPU e GPU.

**Conclusão:** Segue *Von Neumann* porque possui todas as 5 unidades integradas em um sistema de memória unificada e execução sequencial de instruções.

### 2. **Arduino Uno** (*ATmega328P*)

Segue a arquitetura de *Von Neumann*? () Sim / (X) Não

Justificativa:

- **Unidade de Entrada:** Portas digitais/analógicas.
- **Unidade de Saída:** LED, PWM, comunicação serial.
- **Unidade de Memória:** Separada (*Flash* para código, *SRAM* para dados).
- **Unidade de Controle:** Presente, mas limitada a tarefas simples.
- **ULA:** Presente, mas sem *pipeline* complexo.

**Conclusão:** Não segue *Von Neumann* devido à separação física entre memória de programa (*Flash*) e dados (*SRAM*), característica da arquitetura *Harvard*.

### 3. **Xiaomi Redmi Note 10** (*Snapdragon 678*)

Segue a arquitetura de *Von Neumann*? (X) Sim / () Não

Justificativa:

- **Unidade de Entrada/Saída:** Touchscreen, câmera, alto-falantes.
- **Unidade de Memória:** RAM LPDDR4X unificada.

- **Unidade de Controle:** *CPU ARM Cortex-A76/A55*.
- **ULA:** Presente na *CPU* e *GPU Adreno 612*.

**Conclusão:** Segue *Von Neumann*, assim como o *iPhone*, com memória unificada e execução sequencial.

#### 4. **FPGA** (*Xilinx Spartan-6*)

**Segue a arquitetura de *Von Neumann*? ( ) Sim / (X) Não**

Justificativa:

- **Unidade de Entrada/Saída:** Configurável por pinos *GPIO*.
- **"Memória":** Blocos de *RAM* distribuídos, sem separação clara entre dados/código.
- **Controle/ULA:** Não possui um ciclo fixo, a lógica é reprogramável em *hardware*.

**Conclusão:** Não segue *Von Neumann*, pois opera como circuito reconfigurável, sem *fetch-decode-execute* sequencial.

#### 5. **Google TPU** (*Tensor Processing Unit*)

**Segue a arquitetura de *Von Neumann*? ( ) Sim / (X) Não**

Justificativa:

- **Entrada/Saída:** Dados de redes neurais (matrizes).
- **Memória:** *Buffers* dedicados (não *RAM* unificada).
- **Controle/ULA:** Não decodifica instruções, executa operações matriciais fixas.

**Conclusão:** Acelerador de *IA* especializado, sem ciclo de instruções tradicional.

#### 6. **PlayStation 5** (*AMD Zen 2 + RDNA 2*)

**Segue a arquitetura de *Von Neumann*? (X) Sim / ( ) Não**

Justificativa:

- **Entrada/Saída:** Controle, *HDMI*, *SSD*.

- **Memória:** *GDDR6* unificada (*CPU/GPU*).
- **Controle/ULA:** *CPU* com *fetch-decode-execute* e ULA moderna.

**Conclusão:** *Von Neumann* com otimizações (como *SSD* de alta velocidade).

7.     *Nintendo Switch* (*Tegra X1*)

Segue a arquitetura de *Von Neumann*? (X) Sim / () Não

Justificativa:

- **Entrada/Saída:** *Joy-Cons*, *dock*, tela.
- **Memória:** *4GB LPDDR4* unificada.
- **Controle/ULA:** *CPU ARM Cortex-A57* com ULA integrada.

**Conclusão:** Segue o modelo tradicional, mesmo sendo um híbrido.

**Reflexão Final: Comparação entre Arquiteturas**

*PlayStation 5* (Segue a arquitetura de *Von Neumann*)

vs.

*Google TPU* (Não segue a arquitetura de *Von Neumann*)

Característica	<i>PlayStation 5</i>	<i>Google TPU</i>
<b>Memória</b>	Unificada ( <i>GDDR6</i> para <i>CPU/GPU</i> )	<i>Buffers</i> dedicados para matrizes
<b>Ciclo de instruções</b>	<i>Fetch-decode-execute</i> sequencial	Execução paralela de operações fixas
<b>Flexibilidade</b>	Multiuso (jogos, <i>apps</i> , sistema operacional)	Especializado em multiplicação matricial
<b>Exemplo de uso</b>	<i>Rodar God of War Ragnarök</i>	Treinar redes neurais no <i>Google Cloud</i>

## Por que são diferentes?

- O **PS5** segue *Von Neumann* para ser generalista, executando qualquer tipo de *software* mediante instruções sequenciais.
- A **TPU** é especializada, ignorando o modelo tradicional para ganhar eficiência em tarefas específicas (*IA*), onde o paralelismo é mais importante que a flexibilidade.

## Conclusão:

A arquitetura de *Von Neumann* domina dispositivos de uso geral, enquanto soluções como *TPUs* e *FPGAs* abandonam-na para otimizar desempenho em tarefas específicas.



## Referências Bibliográficas

Informática Básica: Arquitetura de *John Von Neumann* de forma simples!.

Disponível em: <https://www.youtube.com/embed/tZ5W2LpdcEw> .

Acesso em 12 de Agosto de 2025.

Arquitetura de *Von Neumann* em 6 Minutos (SEM ENROLAÇÃO).

Disponível em: <https://www.youtube.com/embed/V5qE-u6jGo4> .

Acesso em 12 de Agosto de 2025.

Arquitetura de Computadores CJOARQ - Prof. Dr Josivan.

Disponível em : [https://suap.ifsp.edu.br/media/private-media/edu/material\\_aula/eaeb9cc3b190-7b113d1c105e4083af017ac1abf80cec.pdf?st=yt07p2nkijMqcGA7WsW77g&e=1755173638](https://suap.ifsp.edu.br/media/private-media/edu/material_aula/eaeb9cc3b190-7b113d1c105e4083af017ac1abf80cec.pdf?st=yt07p2nkijMqcGA7WsW77g&e=1755173638) .

Acesso em 13 de Agosto de 2025.

O que é a arquitetura de *Von Neumann*?

Disponível em: <https://canaltech.com.br/hardware/o-que-e-arquitetura-de-von-neumann/> .

Acesso em 13 de Agosto de 2025.