

UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACOM - FACULDADE DE COMPUTAÇÃO  
SISTEMA DE INFORMAÇÃO

---

LUCAS ALBINO MARTINS

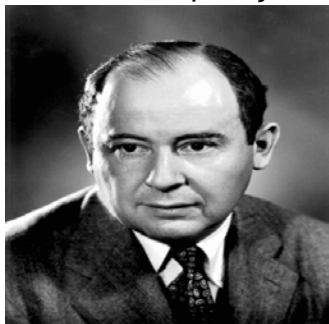
Matrícula: 12011ECP022

**TRABALHO EXTRA:**  
**ARQUITETURA DE VON NEUMANN, modelo e funcionamento.**

Disciplina: Arquitetura e Organização de Computadores.

Uberlândia  
2020

## O modelo de VON NEUMANN.



A computação no seu sentido histórico não havia o conceito de programa armazenado, ou seja, não existia um computador com diversos programas apenas uma máquina projetada para executar funções específicas. A partir das necessidades de armazenamento e execução de mais de uma instrução ao mesmo tempo de um determinado programa que foi desenvolvida as arquiteturas que possibilitavam essas necessidades. A arquitetura de Von Neumann surgiu a partir de 1946, quando o matemático húngaro John Von Neumann que viveu a maior parte de sua vida nos Estados Unidos, ele e sua equipe desenvolveram um projeto de “máquina(computador) de programa armazenado”. Projetado pela IAS, este computador foi largamente difundido, influenciando muitos projetos subsequentes de outras máquinas. Essa arquitetura que formalizou esse projeto lógico de um computador, basicamente é uma arquitetura de computador que se caracteriza pela possibilidade de uma máquina digital armazenar seus programas no mesmo espaço de memória que os dados, podendo assim manipular tais programas, isso foi possível através de uma unidade de processamento (CPU) e uma de armazenamento (MEMÓRIA) para comportar respectivamente as instruções e dados. Essa máquina proposta por Von Neumann reúne os seguintes componentes: uma memória, uma unidade aritmética e lógica (ULA) que tem por função efetiva execução das instruções e uma unidade central (UC) que tem por função a busca, interpretação e controle de execução das instruções, e o controle dos demais componentes da máquina(computador), cuja função é a semelhante da tabela de controle da famosa máquina de Alan Turing, basicamente buscar um programa na memória, instrução por instrução, e executá-la sobre os dados de entrada. Todos os elementos dessa arquitetura são alinhados da estrutura hardware do CPU, assim o sistema pode realizar todas as suas atividades sem apresentar erros no desempenho.

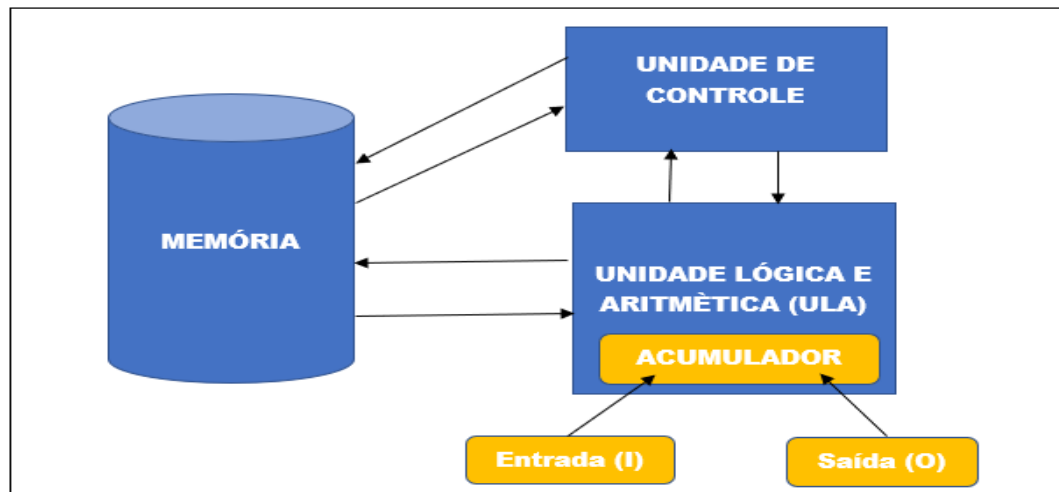


Figura 1 – Ilustração representa à arquitetura de Von Neumann.

## Ciclo de VON NEUMANN.

O ciclo de von Neumann seguindo a ideia de um início, meio e fim basicamente se resume em buscar, decodificar e executar, ou seja, o processo começa com a busca pela instrução, depois ocorre a decodificação ou interpretação (a instrução é codificada em cadeias de uns e zeros, conhecida como instrução binária), então agora podem ser armazenadas em uma memória, quando feito o armazenamento das instruções e de todas as informações necessárias para a execução de uma determinada tarefa o programa é executado.

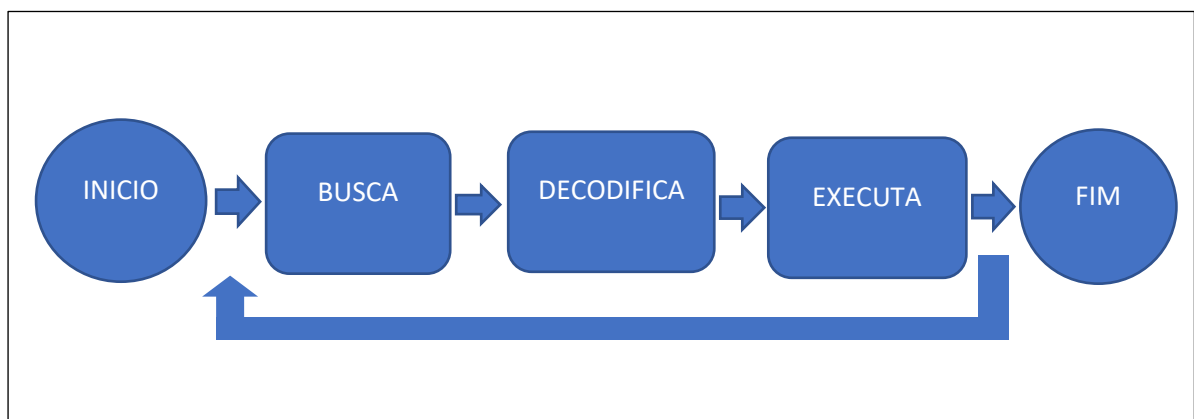


Figura 2 – Ciclo de Von Neumann.

Analisando e comparando com as primeiras máquinas de computação que tinham programas fixos, alguns computadores muito simples ainda usam

este projeto, quer para fins de simplicidade ou de formação. Por exemplo, uma calculadora de mesa (em princípio) é um programa de computador fixo. Ele pode fazer a matemática básica, mas não pode ser usado como um processador de texto ou um console de jogos. Alterar o programa de uma máquina de programa fixo, exige uma religação, reestruturação ou reprogramar a máquina. Os primeiros computadores não eram tão "programados", como eles foram "desenhados". Era um processo trabalhoso, começando com fluxogramas e cédulas de papel, seguido de desenhos detalhados de engenharia e, em seguida o processo muitas vezes penoso fisicamente de religação e reconstrução da máquina. Podendo levar três semanas para criar um programa no ENIAC e começar a trabalhar. A ideia do computador de programa armazenado, mudou tudo isso: um computador que pelo projeto inclui um conjunto de instruções e pode armazenar na memória um conjunto de instruções (programa) que detalha o cálculo. Um projeto de programa armazenado também permite que os programas possam se modificar durante a execução. Uma motivação precoce para uma instalação desse tipo foi a necessidade de um programa para incrementar ou modificar a porção do endereço das instruções, o que tinha que ser feito manualmente em projetos adiantados. Isto tornou-se menos importante quando registradores de índice e endereçamento indireto foram as características usuais da arquitetura da máquina. Código de automodificação foi amplamente caído em desuso, já que normalmente é difícil de entender e depurar, bem como sendo ineficiente em pipelining processador moderno, e esquemas de cache. Em grande escala, a capacidade de tratar as instruções como os dados é o que faz montadores, compiladores e outras ferramentas de programação automatizada possíveis. Pode-se "escrever programas que escrevem programas." Em uma escala menor, instruções de I / O da máquina intensiva, como o BitBlt primitivos usados para modificar imagens em um display bitmap. Demonstrado posteriormente que estas instruções podem ser implementadas de forma eficiente por "na compilação fly" ("just-in-time de compilação) de tecnologia, por exemplo, geração de código de programas, uma forma de código de automodificação que se manteve popular. Há desvantagens para a concepção de von Neumann. A desvantagem mais conhecida é o gargalo de von Neumann, basicamente ele ocorre no canal de transmissão de dados entre a CPU e a memória, a troca de dados limitada (taxa de transferência) entre

a CPU e a memória em relação à quantidade de memória, ou seja, caracterizado pela maior velocidade de conexão entre o processamento do processador em relação ao que a memória pode servir a ele criando assim um gargalo.

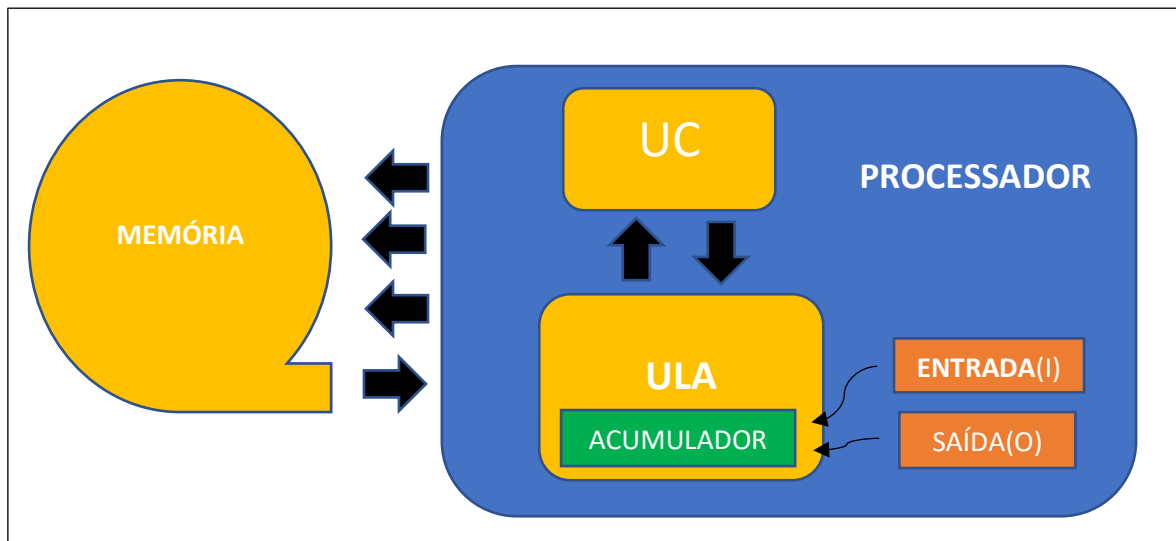


Figura 3 – Gargalo de von Neumann.

Para minimizar esse gargalo, é necessário como o processador é mais rápido que a memória, uma solução adotada há um bom tempo é o uso de memórias cache, as quais mantêm as instruções e dados mais acessados, evitando ter que buscar da memória RAM (o que seria mais lento). E, para melhorar ainda mais, as memórias cache começaram a separar os dados das instruções, aplicando o conceito da Arquitetura de Harvard (que é um melhoramento da Arquitetura de von Neumann).

A arquitetura de von Neumann vista até hoje como referência nos estudos sobre microprocessadores e arquiteturas computacionais. Mesmo essa arquitetura não seja utilizada atualmente dessa forma, ela foi precursora de todas as arquiteturas atuais.

## Bibliografia:

SEBESTA, Robert W. Concepts of programming languages. Boston: Pearson, 2012.

Amaral, Allan Francisco Forzza. Arquitetura de computadores: Curso Técnico em Informática / Allan Francisco Forzza Amaral. – Colatina: CEAD / Ifes, 2010.

Lucas, Marcio. Arquitetura de Von Neumann: Uma breve explicação sem complicação. 2019. Disponível: <<https://medium.com/trainingcenter/a-arquitetura-de-von-neumann-121489873fd4>>. Acesso em 14 de agosto de 2020.

Arquitetura de Von Neumann. 2011. Disponível: <[http://roboiv.re/uploads/documentos/arquiteturas-de-computadores/vonNeumann-2011-08-08\\_1\\_1343264233.pdf](http://roboiv.re/uploads/documentos/arquiteturas-de-computadores/vonNeumann-2011-08-08_1_1343264233.pdf)>. Acesso em 14 de agosto de 2020.