

RELATÓRIO FINAL DO PROJETO

Projeto Simulador

O Simulador mostra o passo a passo de execução de cada instrução de um programa lido a partir de um arquivo texto.

Integrante I: GESSÉ FELIPE DE SÁ TAVEIRA

Integrante II: LUCAS LOPES CORREA NUNES

Orientador: Prof. Renato da Silva Ramalho

Resumo

Este relatório contém uma descrição das atividades realizadas no projeto simulador da disciplina Organização e Arquitetura de Computador, para tanto, o cumprimento das atividades previstas no cronograma do projeto originalmente estabelecido. Além das atividades realizadas, este relatório contém descrições de aspectos teóricos e práticos referentes aos processos realizados nas instruções da CPU.

Índice:

1. Introdução.....	2
2. Descrição das atividades realizadas.....	3
3. Testes e exemplo de uso.....	4
4. Conclusão.....	16

1. Introdução

O objetivo geral do projeto estabelecia o seguinte propósito: Implementar um simulador de CPU, que mostre o passo a passo de execução de cada instrução de um programa lido a partir de um arquivo texto.

Este relatório descreve, portanto, o que foi realizado neste projeto ao longo do mês, isto é, de outubro de 2017 a novembro de 2017. A seguir são apresentadas resumidamente as principais atividades deste período.

O integrante Gessé Felipe é formado no curso Técnico em Informática e no curso Programador de Sistemas pelo Senac São Paulo, onde obteve conhecimentos técnicos e práticos na linguagem de programação C# e na ferramenta do Visual Studio.

A escolha da linguagem de programação foi pelo fato de um dos integrantes possuir mais facilidade e domínio na linguagem. A linguagem de programação C# possibilita uma implementação fácil e usual de recursos que em outras linguagens de programação levaria um tempo para implementar. Foi utilizado a plataforma de hospedagem de código fonte com controle de versão (**GitHub**).

Conteúdo deste relatório: a próxima seção deste relatório é descrito em detalhes as tarefas realizadas no projeto; a seguir, é dada uma breve explicação sobre o computador hipotético Neander. As seções que seguem abordam aspectos teóricos importantes para o correto entendimento do projeto, e também descrevem o software.

2. Descrição das atividades realizadas

O objetivo da equipe era planejar um simulador que apresenta o passo-a-passo da instrução. Para isso, planejamos desenvolver o simulador na linguagem C# por possuir um conhecimento mais amplo na linguagem que oferecia recursos para criar o simulador com uma interface gráfica de modo fácil e rápido.

Após o planejamento junto com os integrantes, passamos para a etapa de entender como funciona o computador hipotético Neander, durante os estudos foi utilizado uma playlist disponível no YouTube sobre o Neander. Em seguida, fizemos anotações da função de cada mnemônicos.

Na sequência, após um conhecimento amplo no Neander, começamos o planejamento da interface do simulador, realizando os passo-a-passo com calma. Em primeira instância criamos uma interface com um **RichTextBox** e um **menuStrip**, a partir disso, criamos uma botão “Carregar” para carregar o arquivo com extensão “.txt” e colocar no campo RichTextBox. Após, fizemos uma função para separar os mnemônicos do arquivo txt e o código para um **RichTextBox** ao lado, no início o objetivo do RichTextBox com nome **Mnemônicos** foi criado apenas para verificarmos como estava o funcionamento da separação dos mnemônicos do arquivo, mas no final, decidimos deixar para visualização para o usuário.

Com isso, precisaríamos colocar as anotações feitas no papel e passar para o simulador, já que tínhamos as funções para separar os mnemônicos do arquivo texto e para pegar a posição de endereço após o mnemônico. A partir daí, fizemos uma estrutura de condição (**if**) para verificar qual o tipo do mnemônico da linha que o simulador estava executando e realizar a captura da posição de endereço a partir das funções criadas.

Para implementar o JMP (Jump), JZ (Jump Zero) e JN (Jump Negativo) criamos uma variável **desvio**, caso na estrutura de condição o mnemônico for JMP, JZ e JN, ele executa a função para pegar a posição do endereço do desvio e definio o desvio igual a 1, assim na próxima execução, ele implementa o desvio e altera o valor do PC para a posição captada na função.

Quando temos o STA, ele substitui o valor após a execução na posição do endereço de memória para o valor do acumulador.

No arquivo do projeto possui algumas instruções prontas geradas pelo Neander, durante os testes todas funcionaram corretamente. Caso exista alguma dúvida, a equipe está disponível para solucionar as dúvidas.

3. Testes e exemplos de uso

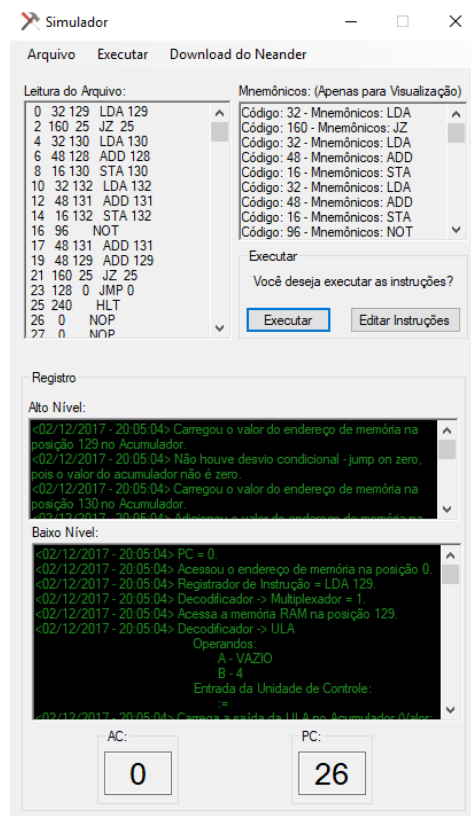
Para executar o projeto corretamente, é recomendável possuir o Visual Studio 2017 ou posterior, com esta ferramenta é possível ter acesso ao código fonte do projeto, para isso deverá seguir os seguintes passos:

- Executar a ferramenta Visual Studio 2017 ou posterior;
- Após o passo anterior, acesse o menu Arquivo >> Abrir >> Projeto/Solução... (ou CTRL+SHIFT+O). Em seguida, acesse a pasta do projeto do Simulador e abra o arquivo **Simulador.sln**, e na sequência, você terá acesso a todos os detalhes do código fonte.

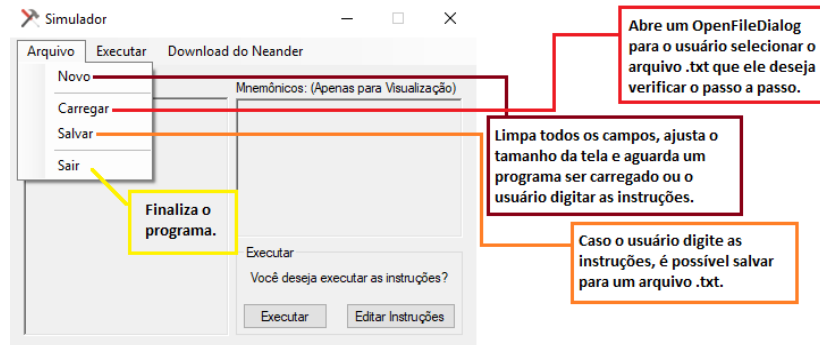
Software em Execução:

Tela Gráfica

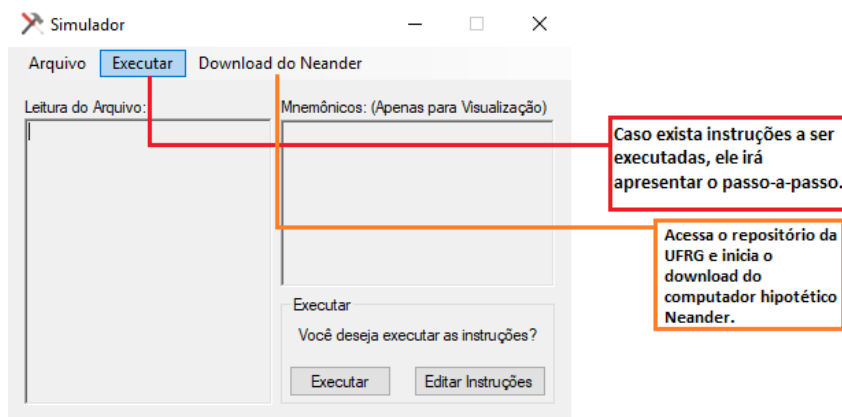
a) Tela Gráfica do Programa em execução



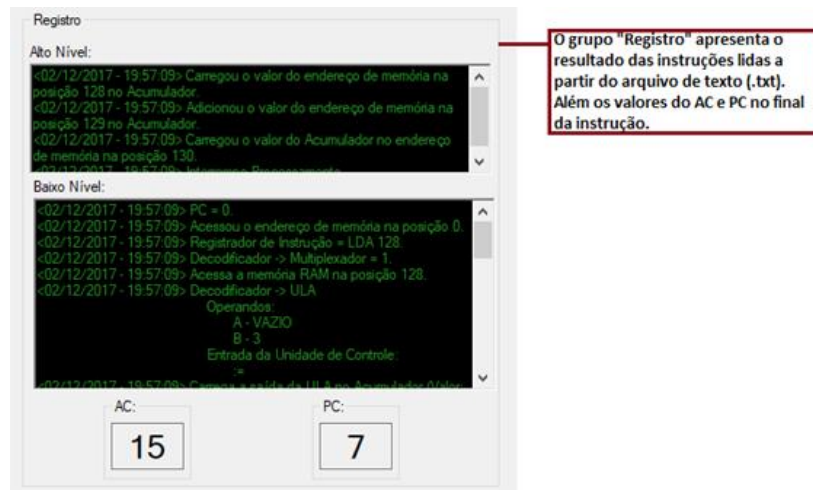
b) Explicação dos itens incluso no menu Arquivo



c) Explicação do menu Executar e Download do Neander



d) Explicação da Tela Gráfica de “Registro”, após a execução do arquivo texto.



Registro de Instruções de baixo e alto nível

Arquivo de Multiplicação

Baixo nível:

```
<02/12/2017 - 20:05:04> PC = 0.
<02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 0.
<02/12/2017 - 20:05:04> Registrador de Instrução = LDA 129.
<02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:04> Acessa a memória RAM na posição 129.
<02/12/2017 - 20:05:04> Decodificador -> ULA
    Operandos:
        A - VAZIO
        B - 4
    Entrada da Unidade de Controle:
        :=
<02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 4).
<02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 2.
<02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 2.
<02/12/2017 - 20:05:04> Registrador de Instrução = JN 25.
<02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 0.
<02/12/2017 - 20:05:04> Acessa a memória RAM na posição 25.
<02/12/2017 - 20:05:04> Decodificador -> ULA
    Operandos:
        A - 4
        B - 0
    Entrada da Unidade de Controle:
        =
<02/12/2017 - 20:05:04> Sem Desvio Condicional.
<02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 4.
<02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 4.
<02/12/2017 - 20:05:04> Registrador de Instrução = LDA 130.
<02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:04> Acessa a memória RAM na posição 130.
<02/12/2017 - 20:05:04> Decodificador -> ULA
    Operandos:
        A - VAZIO
        B - 0
    Entrada da Unidade de Controle:
        :=
<02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 0).
<02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 6.
<02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 6.
<02/12/2017 - 20:05:04> Registrador de Instrução = ADD 128.
<02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:04> Acessa a memória RAM na posição 128.
<02/12/2017 - 20:05:04> Decodificador -> ULA
    Operandos:
        A - 0
        B - 3
    Entrada da Unidade de Controle:
        +
<02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 3).
<02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 8.
```

<02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 8.
 <02/12/2017 - 20:05:04> Registrador de Instrução = STA 130.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 0.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 130.
 <02/12/2017 - 20:05:04> Carrega o valor do Acumulador no endereço de memória na posição acessada anteriormente (Posição: 130).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 10.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 10.
 <02/12/2017 - 20:05:04> Registrador de Instrução = LDA 132.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 132.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - VAZIO
 B - 0
 Entrada da Unidade de Controle:
 :=
 <02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 0).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 12.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 12.
 <02/12/2017 - 20:05:04> Registrador de Instrução = ADD 131.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 131.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - 0
 B - 1
 Entrada da Unidade de Controle:
 +
 <02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 1).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 14.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 14.
 <02/12/2017 - 20:05:04> Registrador de Instrução = STA 132.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 0.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 132.
 <02/12/2017 - 20:05:04> Carrega o valor do Acumulador no endereço de memória na posição acessada anteriormente (Posição: 132).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 16.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 16.
 <02/12/2017 - 20:05:04> PC = 17.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 17.
 <02/12/2017 - 20:05:04> Registrador de Instrução = ADD 131.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 131.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - 254
 B - 1
 Entrada da Unidade de Controle:
 +
 <02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 1).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 19.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 19.
 <02/12/2017 - 20:05:04> Registrador de Instrução = ADD 129.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 129.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - 255
 B - 4
 Entrada da Unidade de Controle:
 +

<02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 4).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 21.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 21.
 <02/12/2017 - 20:05:04> Registrador de Instrução = JN 25.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 0.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 25.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - 3
 B - 0
 Entrada da Unidade de Controle:
 =

<02/12/2017 - 20:05:04> Sem Desvio Condicional.
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 23.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 23.
 <02/12/2017 - 20:05:04> Registrador de Instrução = JMP 0.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> O valor do PC foi alterado para a posição do desvio (Posição: 0).

<02/12/2017 - 20:05:04> PC = 0.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 0.
 <02/12/2017 - 20:05:04> Registrador de Instrução = LDA 129.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 129.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - VAZIO
 B - 4
 Entrada da Unidade de Controle:
 :=

<02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 4).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 2.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 2.
 <02/12/2017 - 20:05:04> Registrador de Instrução = JN 25.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 0.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 25.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - 4
 B - 0
 Entrada da Unidade de Controle:
 =

<02/12/2017 - 20:05:04> Sem Desvio Condicional.
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 4.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 4.
 <02/12/2017 - 20:05:04> Registrador de Instrução = LDA 130.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 130.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - VAZIO
 B - 3
 Entrada da Unidade de Controle:
 :=

<02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 3).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 6.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 6.
 <02/12/2017 - 20:05:04> Registrador de Instrução = ADD 128.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 128.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:

A - 3
 B - 3
 Entrada da Unidade de Controle:
 +
 <02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 3).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

 <02/12/2017 - 20:05:04> PC = 8.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 8.
 <02/12/2017 - 20:05:04> Registrador de Instrução = STA 130.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 0.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 130.
 <02/12/2017 - 20:05:04> Carrega o valor do Acumulador no endereço de memória na posição acessada anteriormente (Posição: 130).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

 <02/12/2017 - 20:05:04> PC = 10.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 10.
 <02/12/2017 - 20:05:04> Registrador de Instrução = LDA 132.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 132.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - VAZIO
 B - 1
 Entrada da Unidade de Controle:
 :=
 <02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 1).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

 <02/12/2017 - 20:05:04> PC = 12.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 12.
 <02/12/2017 - 20:05:04> Registrador de Instrução = ADD 131.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 131.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - 1
 B - 1
 Entrada da Unidade de Controle:
 +
 <02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 1).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

 <02/12/2017 - 20:05:04> PC = 14.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 14.
 <02/12/2017 - 20:05:04> Registrador de Instrução = STA 132.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 0.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 132.
 <02/12/2017 - 20:05:04> Carrega o valor do Acumulador no endereço de memória na posição acessada anteriormente (Posição: 132).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

 <02/12/2017 - 20:05:04> PC = 16.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 16.
 <02/12/2017 - 20:05:04> PC = 17.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 17.
 <02/12/2017 - 20:05:04> Registrador de Instrução = ADD 131.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 131.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - 253
 B - 1
 Entrada da Unidade de Controle:
 +
 <02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 1).
 <02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

 <02/12/2017 - 20:05:04> PC = 19.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 19.
 <02/12/2017 - 20:05:04> Registrador de Instrução = ADD 129.

```

<02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:04> Acessa a memória RAM na posição 129.
<02/12/2017 - 20:05:04> Decodificador -> ULA
    Operandos:
        A - 254
        B - 4
    Entrada da Unidade de Controle:
        +
<02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 4).
<02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 21.
<02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 21.
<02/12/2017 - 20:05:04> Registrador de Instrução = JN 25.
<02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 0.
<02/12/2017 - 20:05:04> Acessa a memória RAM na posição 25.
<02/12/2017 - 20:05:04> Decodificador -> ULA
    Operandos:
        A - 2
        B - 0
    Entrada da Unidade de Controle:
        =
<02/12/2017 - 20:05:04> Sem Desvio Condicional.
<02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 23.
<02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 23.
<02/12/2017 - 20:05:04> Registrador de Instrução = JMP 0.
<02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:04> O valor do PC foi alterado para a posição do desvio (Posição: 0).

<02/12/2017 - 20:05:04> PC = 0.
<02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 0.
<02/12/2017 - 20:05:04> Registrador de Instrução = LDA 129.
<02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:04> Acessa a memória RAM na posição 129.
<02/12/2017 - 20:05:04> Decodificador -> ULA
    Operandos:
        A - VAZIO
        B - 4
    Entrada da Unidade de Controle:
        :=
<02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 4).
<02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 2.
<02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 2.
<02/12/2017 - 20:05:04> Registrador de Instrução = JN 25.
<02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 0.
<02/12/2017 - 20:05:04> Acessa a memória RAM na posição 25.
<02/12/2017 - 20:05:04> Decodificador -> ULA
    Operandos:
        A - 4
        B - 0
    Entrada da Unidade de Controle:
        =
<02/12/2017 - 20:05:04> Sem Desvio Condicional.
<02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:04> PC = 4.
<02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 4.
<02/12/2017 - 20:05:04> Registrador de Instrução = LDA 130.
<02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:04> Acessa a memória RAM na posição 130.
<02/12/2017 - 20:05:04> Decodificador -> ULA
    Operandos:
        A - VAZIO
        B - 6
    Entrada da Unidade de Controle:
        :=
<02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 6).
<02/12/2017 - 20:05:04> Incrementa o valor do PC para a próxima instrução.

```

<02/12/2017 - 20:05:04> PC = 6.
 <02/12/2017 - 20:05:04> Acessou o endereço de memória na posição 6.
 <02/12/2017 - 20:05:04> Registrador de Instrução = ADD 128.
 <02/12/2017 - 20:05:04> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:04> Acessa a memória RAM na posição 128.
 <02/12/2017 - 20:05:04> Decodificador -> ULA
 Operandos:
 A - 6
 B - 3
 Entrada da Unidade de Controle:
 +
 <02/12/2017 - 20:05:04> Carrega a saída da ULA no Acumulador (Valor: 3).
 <02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 8.
 <02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 8.
 <02/12/2017 - 20:05:05> Registrador de Instrução = STA 130.
 <02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 0.
 <02/12/2017 - 20:05:05> Acessa a memória RAM na posição 130.
 <02/12/2017 - 20:05:05> Carrega o valor do Acumulador no endereço de memória na posição acessada anteriormente (Posição: 130).
 <02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 10.
 <02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 10.
 <02/12/2017 - 20:05:05> Registrador de Instrução = LDA 132.
 <02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:05> Acessa a memória RAM na posição 132.
 <02/12/2017 - 20:05:05> Decodificador -> ULA
 Operandos:
 A - VAZIO
 B - 2
 Entrada da Unidade de Controle:
 :=
 <02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 2).
 <02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 12.
 <02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 12.
 <02/12/2017 - 20:05:05> Registrador de Instrução = ADD 131.
 <02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:05> Acessa a memória RAM na posição 131.
 <02/12/2017 - 20:05:05> Decodificador -> ULA
 Operandos:
 A - 2
 B - 1
 Entrada da Unidade de Controle:
 +
 <02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 1).
 <02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 14.
 <02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 14.
 <02/12/2017 - 20:05:05> Registrador de Instrução = STA 132.
 <02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 0.
 <02/12/2017 - 20:05:05> Acessa a memória RAM na posição 132.
 <02/12/2017 - 20:05:05> Carrega o valor do Acumulador no endereço de memória na posição acessada anteriormente (Posição: 132).
 <02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 16.
 <02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 16.
 <02/12/2017 - 20:05:05> PC = 17.
 <02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 17.
 <02/12/2017 - 20:05:05> Registrador de Instrução = ADD 131.
 <02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
 <02/12/2017 - 20:05:05> Acessa a memória RAM na posição 131.
 <02/12/2017 - 20:05:05> Decodificador -> ULA
 Operandos:
 A - 252
 B - 1

Entrada da Unidade de Controle:
+

<02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 1).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 19.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 19.
<02/12/2017 - 20:05:05> Registrador de Instrução = ADD 129.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 129.
<02/12/2017 - 20:05:05> Decodificador -> ULA
Operandos:
A - 253
B - 4

Entrada da Unidade de Controle:
+

<02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 4).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 21.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 21.
<02/12/2017 - 20:05:05> Registrador de Instrução = JN 25.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 0.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 25.
<02/12/2017 - 20:05:05> Decodificador -> ULA
Operandos:
A - 1
B - 0

Entrada da Unidade de Controle:
=

<02/12/2017 - 20:05:05> Sem Desvio Condicional.
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 23.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 23.
<02/12/2017 - 20:05:05> Registrador de Instrução = JMP 0.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:05> O valor do PC foi alterado para a posição do desvio (Posição: 0).

<02/12/2017 - 20:05:05> PC = 0.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 0.
<02/12/2017 - 20:05:05> Registrador de Instrução = LDA 129.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 129.
<02/12/2017 - 20:05:05> Decodificador -> ULA
Operandos:
A - VAZIO
B - 4

Entrada da Unidade de Controle:
:=

<02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 4).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 2.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 2.
<02/12/2017 - 20:05:05> Registrador de Instrução = JN 25.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 0.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 25.
<02/12/2017 - 20:05:05> Decodificador -> ULA
Operandos:
A - 4
B - 0

Entrada da Unidade de Controle:
=

<02/12/2017 - 20:05:05> Sem Desvio Condicional.
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 4.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 4.
<02/12/2017 - 20:05:05> Registrador de Instrução = LDA 130.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 130.

```

<02/12/2017 - 20:05:05> Decodificador -> ULA
    Operandos:
        A - VAZIO
        B - 9
    Entrada da Unidade de Controle:
        :=
<02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 9).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 6.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 6.
<02/12/2017 - 20:05:05> Registrador de Instrução = ADD 128.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 128.
<02/12/2017 - 20:05:05> Decodificador -> ULA
    Operandos:
        A - 9
        B - 3
    Entrada da Unidade de Controle:
        +
<02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 3).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 8.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 8.
<02/12/2017 - 20:05:05> Registrador de Instrução = STA 130.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 0.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 130.
<02/12/2017 - 20:05:05> Carrega o valor do Acumulador no endereço de memória na posição acessada anteriormente (Posição: 130).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 10.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 10.
<02/12/2017 - 20:05:05> Registrador de Instrução = LDA 132.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 132.
<02/12/2017 - 20:05:05> Decodificador -> ULA
    Operandos:
        A - VAZIO
        B - 3
    Entrada da Unidade de Controle:
        :=
<02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 3).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 12.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 12.
<02/12/2017 - 20:05:05> Registrador de Instrução = ADD 131.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 131.
<02/12/2017 - 20:05:05> Decodificador -> ULA
    Operandos:
        A - 3
        B - 1
    Entrada da Unidade de Controle:
        +
<02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 1).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 14.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 14.
<02/12/2017 - 20:05:05> Registrador de Instrução = STA 132.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 0.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 132.
<02/12/2017 - 20:05:05> Carrega o valor do Acumulador no endereço de memória na posição acessada anteriormente (Posição: 132).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 16.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 16.
<02/12/2017 - 20:05:05> PC = 17.

```

```

<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 17.
<02/12/2017 - 20:05:05> Registrador de Instrução = ADD 131.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 131.
<02/12/2017 - 20:05:05> Decodificador -> ULA
    Operandos:
        A - 251
        B - 1
    Entrada da Unidade de Controle:
        +
<02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 1).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 19.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 19.
<02/12/2017 - 20:05:05> Registrador de Instrução = ADD 129.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 1.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 129.
<02/12/2017 - 20:05:05> Decodificador -> ULA
    Operandos:
        A - 252
        B - 4
    Entrada da Unidade de Controle:
        +
<02/12/2017 - 20:05:05> Carrega a saída da ULA no Acumulador (Valor: 4).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 21.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 21.
<02/12/2017 - 20:05:05> Registrador de Instrução = JN 25.
<02/12/2017 - 20:05:05> Decodificador -> Multiplexador = 0.
<02/12/2017 - 20:05:05> Acessa a memória RAM na posição 25.
<02/12/2017 - 20:05:05> Decodificador -> ULA
    Operandos:
        A - 0
        B - 0
    Entrada da Unidade de Controle:
        =
<02/12/2017 - 20:05:05> Desvio Condicional (Posição: 25).
<02/12/2017 - 20:05:05> Incrementa o valor do PC para a próxima instrução.

<02/12/2017 - 20:05:05> PC = 25.
<02/12/2017 - 20:05:05> Acessou o endereço de memória na posição 25.
<02/12/2017 - 20:05:05> Registrador de Instrução = HLT.
<02/12/2017 - 20:05:05> Processo Interrompido.

```

Alto nível:

```

<02/12/2017 - 20:05:04> Carregou o valor do endereço de memória na posição 129 no Acumulador.
<02/12/2017 - 20:05:04> Não houve desvio condicional - jump on zero, pois o valor do acumulador não é zero.
<02/12/2017 - 20:05:04> Carregou o valor do endereço de memória na posição 130 no Acumulador.
<02/12/2017 - 20:05:04> Adicionou o valor do endereço de memória na posição 128 no Acumulador.
<02/12/2017 - 20:05:04> Carregou o valor do Acumulador no endereço de memória na posição 130.
<02/12/2017 - 20:05:04> Carregou o valor do endereço de memória na posição 132 no Acumulador.
<02/12/2017 - 20:05:04> Adicionou o valor do endereço de memória na posição 131 no Acumulador.
<02/12/2017 - 20:05:04> Carregou o valor do Acumulador no endereço de memória na posição 132.
<02/12/2017 - 20:05:04> Comando NOT executado.
<02/12/2017 - 20:05:04> Adicionou o valor do endereço de memória na posição 131 no Acumulador.
<02/12/2017 - 20:05:04> Adicionou o valor do endereço de memória na posição 129 no Acumulador.
<02/12/2017 - 20:05:04> Não houve desvio condicional - jump on zero, pois o valor do acumulador não é zero.
<02/12/2017 - 20:05:04> Houve desvio incondicional - jump para a posição 0.
<02/12/2017 - 20:05:04> Carregou o valor do endereço de memória na posição 129 no Acumulador.
<02/12/2017 - 20:05:04> Não houve desvio condicional - jump on zero, pois o valor do acumulador não é zero.

```

<02/12/2017 - 20:05:04> Carregou o valor do endereço de memória na posição 130 no Acumulador.
<02/12/2017 - 20:05:04> Adicionou o valor do endereço de memória na posição 128 no Acumulador.
<02/12/2017 - 20:05:04> Carregou o valor do Acumulador no endereço de memória na posição 130.
<02/12/2017 - 20:05:04> Carregou o valor do endereço de memória na posição 132 no Acumulador.
<02/12/2017 - 20:05:04> Adicionou o valor do endereço de memória na posição 131 no Acumulador.
<02/12/2017 - 20:05:04> Carregou o valor do Acumulador no endereço de memória na posição 132.
<02/12/2017 - 20:05:04> Comando NOT executado.
<02/12/2017 - 20:05:04> Adicionou o valor do endereço de memória na posição 131 no Acumulador.
<02/12/2017 - 20:05:04> Adicionou o valor do endereço de memória na posição 129 no Acumulador.
<02/12/2017 - 20:05:04> Não houve desvio condicional - jump on zero, pois o valor do acumulador não é zero.
<02/12/2017 - 20:05:04> Houve desvio incondicional - jump para a posição 0.
<02/12/2017 - 20:05:04> Carregou o valor do endereço de memória na posição 129 no Acumulador.
<02/12/2017 - 20:05:04> Não houve desvio condicional - jump on zero, pois o valor do acumulador não é zero.
<02/12/2017 - 20:05:04> Carregou o valor do endereço de memória na posição 130 no Acumulador.
<02/12/2017 - 20:05:04> Adicionou o valor do endereço de memória na posição 128 no Acumulador.
<02/12/2017 - 20:05:05> Carregou o valor do Acumulador no endereço de memória na posição 130.
<02/12/2017 - 20:05:05> Carregou o valor do endereço de memória na posição 132 no Acumulador.
<02/12/2017 - 20:05:05> Adicionou o valor do endereço de memória na posição 131 no Acumulador.
<02/12/2017 - 20:05:05> Carregou o valor do Acumulador no endereço de memória na posição 132.
<02/12/2017 - 20:05:05> Comando NOT executado.
<02/12/2017 - 20:05:05> Adicionou o valor do endereço de memória na posição 131 no Acumulador.
<02/12/2017 - 20:05:05> Adicionou o valor do endereço de memória na posição 129 no Acumulador.
<02/12/2017 - 20:05:05> Não houve desvio condicional - jump on zero, pois o valor do acumulador não é zero.
<02/12/2017 - 20:05:05> Houve desvio incondicional - jump para a posição 0.
<02/12/2017 - 20:05:05> Carregou o valor do endereço de memória na posição 129 no Acumulador.
<02/12/2017 - 20:05:05> Não houve desvio condicional - jump on zero, pois o valor do acumulador não é zero.
<02/12/2017 - 20:05:05> Carregou o valor do endereço de memória na posição 130 no Acumulador.
<02/12/2017 - 20:05:05> Adicionou o valor do endereço de memória na posição 128 no Acumulador.
<02/12/2017 - 20:05:05> Carregou o valor do Acumulador no endereço de memória na posição 130.
<02/12/2017 - 20:05:05> Carregou o valor do endereço de memória na posição 132 no Acumulador.
<02/12/2017 - 20:05:05> Adicionou o valor do endereço de memória na posição 131 no Acumulador.
<02/12/2017 - 20:05:05> Carregou o valor do Acumulador no endereço de memória na posição 132.
<02/12/2017 - 20:05:05> Comando NOT executado.
<02/12/2017 - 20:05:05> Adicionou o valor do endereço de memória na posição 131 no Acumulador.
<02/12/2017 - 20:05:05> Adicionou o valor do endereço de memória na posição 129 no Acumulador.
<02/12/2017 - 20:05:05> Houve desvio condicional - jump on zero para a posição 25, pois o valor do acumulador é zero.
<02/12/2017 - 20:05:05> Interrompe Processamento.

4. Conclusão

O resultado foi muito satisfatório para a equipe, durante o desenvolvimento do projeto foi possível conhecer detalhadamente o funcionamento do computador hipotético Neander e de cada instruções que são executada no computador no Neander.

Houve algumas dificuldades para realizar algumas funções na programação do simulador por falta de informações disponíveis na internet. Uma das dificuldades do grupo foi a implementação do JMP (Jump), JZ (Jump Zero) e o JN (Jump Negativo), mas após conseguir resolver um dos casos, o restante é somente realizar pequenos ajustes.

Para realizar as instruções de baixo nível, foi utilizado o passo-a-passo que ocorre no arquivo **CountProgram.swf**, disponível pelo professor Renato Ramalho no Ambiente Virtual de Aprendizagem.

Foi muito interessante a finalidade do projeto, possibilitou ampliar o conhecimento do passo-a-passo da execução de uma instrução no processador e foi possível haver uma interação com o grupo para o planejamento de tarefas entre a equipe, assim, sendo fundamental durante o mercado de trabalho.

REFERÊNCIAS

Playlist Arquitetura de Computadores. Disponível em: <<http://bit.ly/2AqC9Bi>>. Acesso em: 02 Dez. 2017.

O Computador Neander. Disponível em: <<http://bit.ly/2BEAath>>. Acesso em: 02 Dez. 2017.

Repositório do Simulador. Disponível em: < <http://bit.ly/2idycb6> >. Acesso em 02 Dez. 2017.