

**Universidade do Estado de Santa Catarina/Centro de Ciências Tecnológicas – UDESC/CCT**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Departamento de Ciência da Computação - DCC

Curso: TADS – Tecnologia em Análise e Desenvolvimento de Sistemas

Disciplina: SOP0002 – Sistemas Operacionais

Professor: Charles Christian Miers

Acadêmicos(as): Bryan Guilherme Konowalow

Luara Bruning

Luís Felipe da Cunha Duarte

Vitória Córdova da Silva

TÍTULO: COMPARAÇÃO DO DESEMPENHO ENTRE O RASPBERRY PI MODEL B E O RASPBERRY PI 4 MODEL B, COM FOCO NO CONSUMO DE PROCESSADOR E MEMÓRIA

OBJETIVO: Comparar o desempenho do Raspberry PI model B com o Raspberry PI 4 model B, ambos executando o Raspberry PI OS 1.4 tendo como parâmetros o consumo de processador e consumo de memória com base nos benchmarks Linpack e RAMspeed/SMP.

JUSTIFICATIVA: Com o avanço da tecnologia utilizada nos computadores, a miniaturização dos dispositivos se tornou cada vez mais latente e necessária, de tal maneira que computadores completos do tamanho de cartões de crédito são cada vez mais comuns. Portanto a comparação entre as gerações desse hardware se faz necessária a fim de atestar a melhora na capacidade de processamento e consumo de memória.

**LISTA DE FIGURAS**

**Figura 1** - Um dos primeiros protótipos de Raspberry PI 9

**Figura 2** - Raspberry Pi Zero W 11

**Figura 3**- Raspberry PI 4 11

**Figura 4** - Diretório de revisões das especificações de *Software* Livre 14

**LISTA DE TABELAS**

**Tabela 1** - Versões de Raspberry Pi e suas especificações 12

**Tabela 2**- Exemplos de *Software* Livre 14

SUMÁRIO

[1       CONCEITOS 7](#_Toc50404238)

[1.1 DEFINIÇÃO DE HARDWARE LIVRE 7](#_Toc50404239)

[1.2     HISTÓRICO 7](#_Toc50404240)

[1.3 PRINCIPAIS HARDWARES 8](#_Toc50404241)

[1.4 RASPBERRY PI 9](#_Toc50404242)

[1.4.1       HISTÓRICO 9](#_Toc50404243)

[1.4.2       VERSÕES PRINCIPAIS DA SÉRIE 10](#_Toc50404244)

[1.5 DEFINIÇÃO DE *SOFTWARE* LIVRE 13](#_Toc50404245)

[1.5.1 HISTÓRICO 14](#_Toc50404246)

[1.5.2 PRINCIPAIS VERSÕES 14](#_Toc50404247)

[1.5.3 RASPBERRY PI OS 15](#_Toc50404248)

[1.5.3.1 HISTÓRICO 15](#_Toc50404249)

[1.5.3.2 CARACTERÍSTICAS PRINCIPAIS 15](#_Toc50404250)

[1.6 BENCHMARK 15](#_Toc50404251)

[1.6.1 CONCEITO 15](#_Toc50404252)

[1.6.2 PRINCIPAIS TIPOS 16](#_Toc50404253)

[REFERÊNCIAS BIBLIOGRÁFICAS 16](#_Toc50404254)

ESCOPO DE TRABALHO

**INTRODUÇÃO**

1. **CONCEITOS**
   1. DEFINIÇÃO DE HARDWARE LIVRE
   2. HISTÓRICO
   3. PRINCIPAIS HARDWARES
   4. RASPBERRY PI
      1. HISTÓRICO
      2. VERSÕES PRINCIPAIS DA SÉRIE
   5. DEFINIÇÃO DE SOFTWARE LIVRE
      1. HISTÓRICO
      2. PRINCIPAIS VERSÕES
      3. RASPBERRY PI OS
         1. HISTÓRICO
         2. CARACTERÍSTICAS PRINCIPAIS
   6. BENCHMARK
      1. CONCEITO
      2. PRINCIPAIS TIPOS

1. **PROJETO DE EXPERIMENTO**
   1. AMBIENTE DE EXPERIMENTAÇÃO
      1. Hardwares utilizados
         1. Raspberry PI model B
         2. Raspberry PI 4 model B
      2. Softwares utilizados
         1. Raspberry PI OS 1.4
         2. Linpack
         3. RAMspeed/SMP
   2. EXPERIMENTO
      1. Bateria de Testes executando Linpack
         1. Raspberry PI model B
         2. Raspberry PI 4 model B+
      2. Bateria de Testes executando RAMspeed/SMP
         1. Raspberry PI model B
         2. Raspberry PI 4 model B
2. **ANÁLISE**
   1. Comparação da performance
      1. Comparação dos resultados executando o Linpack
      2. Comparação dos resultados executando o RAMspeed/SMP

**4 CONSIDERAÇÕES**

# 1       CONCEITOS

# 1.1 DEFINIÇÃO DE HARDWARE LIVRE

Desde tempos antigos a inovação é algo que prende a atenção, e por meio dela surgem as mais diversas tecnologias. Ao se aprofundar neste termo, podemos ver que um dos muitos conceitos formados é acerca de hardwares livres. Mais conhecido no inglês como Open Hardware, popularizado pelo sistema operacional Ubuntu de distribuição Linux o qual tem o código fonte aberto, embora seja algo físico, se espelha muito nos princípios de *Open Software (software* livre) onde o usuário com acesso ao código fonte pode modificá-lo da maneira que desejar. São hardwares de computadores, circuitos eletrônicos e design de circuitos de drivers que podem ser copiados e modificados sem restrições, já que a premissa se prende ao fato de que qualquer pessoa pode construir e modificar esse hardware de acordo com suas próprias necessidades, desde algo para uso próprio ou até algum tipo de comércio do mesmo. Com isso, se torna de muito mais fácil o acesso tanto da procura dos materiais como da parte financeira para realizar projetos próprios sem ter que depender e ficar à mercê de grandes empresas. O principal diferencial de um hardware livre é exatamente a característica de adaptabilidade, uma vez que podendo ser sempre modificado facilmente, independente do propósito, cria infindáveis possibilidades. O movimento disso vem cada ano crescendo mais, já que a própria comunidade gosta de trocar informações e experiências de projetos.

Não se pode deixar confundir o conceito com a pirataria. Pirataria são produtos falsificados vendidos por preços abaixo da média, já na montagem de um hardware livre usa-se componentes de várias marcar, a qual o consumidor pode optar pelas mais em conta para montar o projeto desejado, mas sempre respeitando as marcas registradas.

# 1.2     HISTÓRICO

A idealização de hardware livre não é algo que ocorreu da noite para o dia, foi algo que demandou tempo e muito empenho de pessoas dispostas a trocar conhecimentos, experiência e informações, de modo a se criar uma grande comunidade interessada e realmente empenhada no assunto.

As tecnologias surgem de modo a nos auxiliar no nosso dia a dia, o uso de hardware livre tem sido aplicado em diversas áreas e projetos, como por exemplo, sistemas de computadores e componentes, câmeras, rádio, telefone, educação, máquinas e ferramentas, robótica, energia renovável, automação de casa, medicina e biotecnologia, automotiva, prototipação, teste de equipamentos e instrumentos musicais.

# 1.3 PRINCIPAIS HARDWARES

Definir tipo de hardwares livres não é uma tarefa fácil, uma vez que eles podem ser modificados por qualquer pessoa e de qualquer forma, porém existem alguns desses produtos comercializados por grandes empresas que se tornaram bem famosos, tanto pelo preço como pela facilidade de manuseio. Um deles é o Arduino, uma plataforma eletrônica, de placa única, na qual permite se ler entradas, como sensores de luz, um determinado barulho ou até um botão pressionado e transformar isso em uma saída, como por exemplo, acender um led, dar início em um motor, entre muitas outras possibilidades. Este hardware livre é um dos pioneiros dentro da área da robótica e eletrônica.

Muito conhecido e procurado, damos ênfase no Raspberry Pi, da Fundação - sem fins lucrativos - Raspberry Pi. O qual consiste também em uma placa única, sendo o principal hardware abordado neste trabalho.

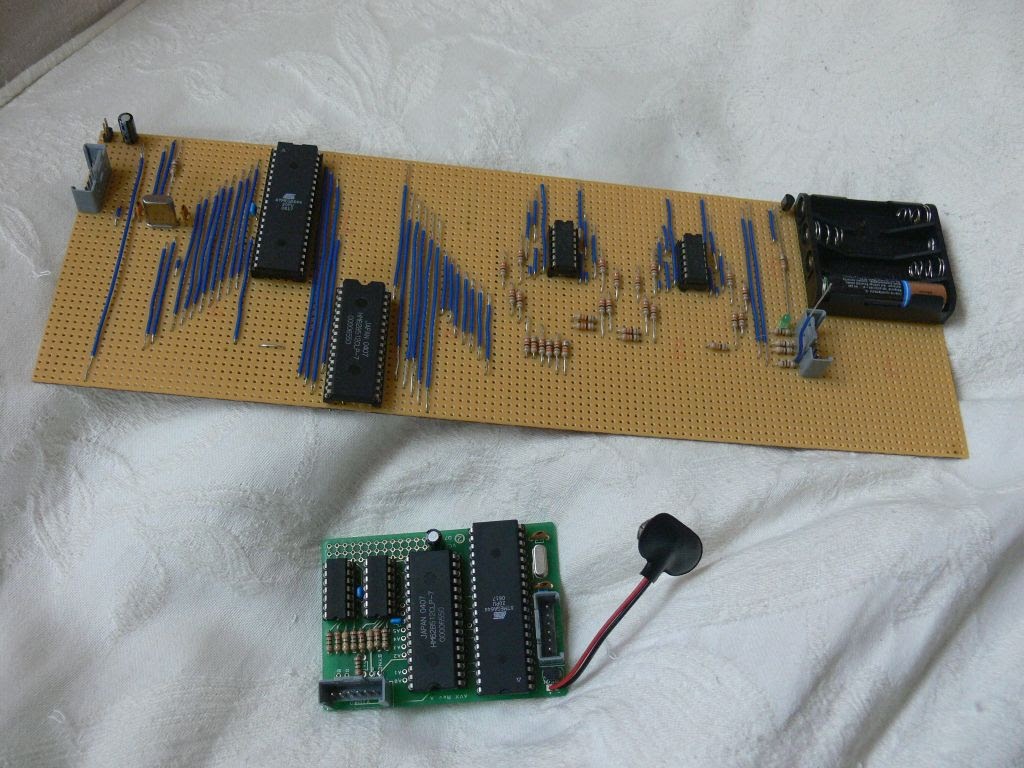
Existem ainda alguns tipos de hardware que tem grande competição com o Raspberry Pi, porém não são livres. Exemplos deles são o Rock Pi 4B, com mais desempenho de processamento e de qualidade gráfica, com RAM de até 4 GB e um processador com seis núcleos de 64 bits; o Banana Pi M2 Zero, um minicomputador com processador Alliwnner H2+ com quatro núcleos e 512 MB de RAM; o Rock 64, com processador de quatro núcleos e 4 GB de RAM com HDR, focado na parte de mídia; o Nvidia Jetson Nano, com base no SO Ubuntu, que possui um processador de quatro núcleos com velocidades de até 1.43 GHz e uma RAM de 4 GB, podendo reproduzir vídeos em 4K sem grandes problemas; o Asus Tinker Board com processador de quatro núcleos e 2 GB de RAM e o Gigabyte GA-SBCAP3350, bem robusto e de ótimo desempenho, com um processador Celeron N3350 de 1.6 GHz com até 8 GB de RAM, compatível com Windows 10.

# 1.4 RASPBERRY PI

# 1.4.1       HISTÓRICO

O projeto do microcomputador Raspberry Pi foi idealizado em 2006 no laboratório de computação da universidade de Cambridge, no Reino Unido, pelos membros Eben Upton, Rob Mullins, Jack Lang e Alan Mycroft. Eles criaram diversos protótipos, como mostrado abaixo, em cima do microcontrolador Atmel ATmega644, os quais modelos atuais são baseados até hoje.

**Figura 1** - Um dos primeiros protótipos de Raspberry PI



Fonte: Divulgação/Fundação Raspberry Pi

A figura mostra uma placa ainda de tamanho grande comparado com as dos dias de hoje, porém foi apenas um dos primeiros protótipos criados. A premissa do projeto eram focadas em dois fatores, um era criar um dispositivo super pequeno, comparado a um cartão de crédito, que pudesse ser plugado a uma televisão e fazer a maioria das coisas que um computador comum realizava e o outro era a preocupação com o ensino de tecnologia da informação, de modo a conseguir ajudar crianças a aprender programação, um dos motivos do custo do produto girar em torno de U$35. Como em 2008 começou a ficar mais frequente o uso de dispositivos móveis com boa capacidade de processamento, os custos consequentemente se tornam mais baixos, sendo possível dar mais um passo no projeto, porém apenas 6 anos depois da ideia idealizada que ocorreu a primeira comercialização do produto, em 2012.

Para começar a utilizar um microprocessador Raspberry Pi, deve ser levado em consideração alguns periféricos mínimos necessários, como fonte micro USB de 5V e 700mA ao menos, um cabo HDMI para poder conectar com o monitor, cartão SD de no mínimo 4GB e classe 4, cabo de internet, um teclado e um mouse.

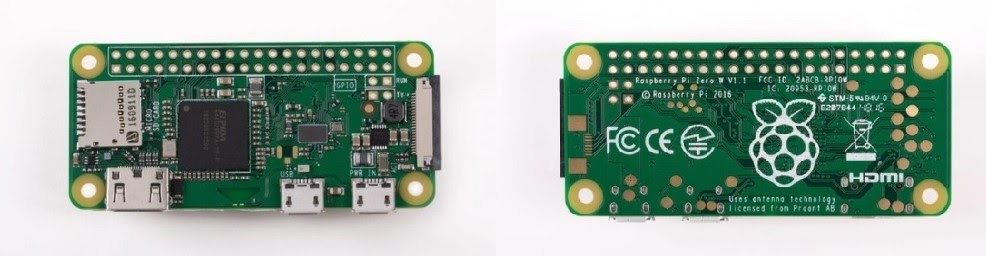
Por curiosidade, Raspberry significa uma amora, uma fruta comum em países de clima temperado e Pi remete a linguagem Phyton utilizada.

# 1.4.2       VERSÕES PRINCIPAIS DA SÉRIE

Atualmente existem 4 gerações no mercado, e a linha cronológica de lançamentos segue basicamente com atualizações todos os anos. Na nomenclatura, classificar um modelo com a letra B (média de U$35), significa dizer que ele é o mais completo da série, já com a letra A (média de U$25), uma versão reduzida, como por exemplo a lite nos smartphones. Um modelo sendo aprimorado comparado com sua versão padrão, é atribuído um sinal de + (plus), como sinônimo de incremento. Uma exceção é a versão Zero, a qual possui uma variante também denominada Zero W (média de U$10), onde é incluído wi-fi e bluetooth nas suas características, este modelo é o mais econômico e compacto encontrado à venda.

No ano de 2012 foi lançado a primeira geração de Raspberry Pi, o modelo 1B e em 2013, o modelo 1A. Logo no ano seguinte, 2014, foram lançadas as versões melhoradas das anteriores, o Raspberry Pi 1B+ com processador Broadcom 700 MHz, 512 MB de RAM e ethernet e o Raspberry Pi 1A+ com processador Broadcom 700 MHz e 256 MB de RAM. Em 2015 entrou em cena a segunda geração desses microcomputadores, sendo disponível o modelo Raspberry Pi 2B com processador Broadcom 900 MHz quad-core, 1 GB de RAM e ethernet e o Raspberry Pi Zero, marcado pela redução considerável de tamanho, com processador Broadcom 1 GHz e 512 MB de RAM. No ano de 2016 se iniciou a terceira geração, o Raspberry Pi 3B com processador Broadcom 1,2 GHz quad-core, 1 GB de RAM, bluetooth 4.1, ethernet e Wi-fi 4,2 GHz. Vale ressaltar que em 2017 houve uma exceção e foi lançada uma versão melhorada do modelo Zero, o Raspberry Pi Zero W, como mostra a figura abaixo, a qual incluiu nas suas características Wi-fi 2,4 GHz e bluetooth 4.1, porém fazendo parte ainda da segunda geração.

**Figura 2** - Raspberry Pi Zero W

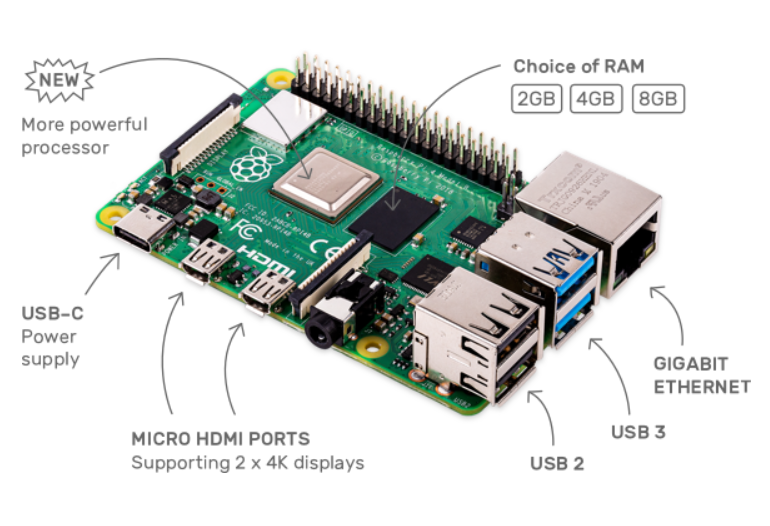


Fonte: https://www.raspberrypi.org/products/raspberry-pi-zero-w/

Nessa figura, é apresentado o menor dos modelos existentes da Raspberry Pi, do tamanho similar à um cartão de crédito.

Em 2018 ocorreu os lançamentos do Raspberry Pi 3B+ com processador Broadcom 1,4 GHz quad-core, 1 GB de RAM, bluetooth 4.2, ethernet e Wi-fi Dual Band e do Raspberry Pi 3A+ com processador Broadcom 1,4 GHz quad-core, 512 MB de RAM, bluetooth 4.2 e Wi-fi Dual Band, mesmo não tendo sido comercializado o modelo 3A nos anos anteriores. Em 2019 surge a quarta e mais nova geração desses microcomputadores, trazendo o Raspberry Pi 4B com processador Broadcom 1,5 GHz quad-core, de 2 até 8 GB de RAM, bluetooth 5.0, ethernet e Wi-fi Dual Band, sendo o modelo mais recente encontrado no mercado nos dias de hoje. Na figura 3 é apresentada sua estrutura:

**Figura 3**- Raspberry PI 4



Fonte: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>

Como já está explícito, esse modelo possui três portas USBs, sendo uma tipo C, duas portas micro HDMI e uma entrada ethernet. A tabela 1 apresenta de forma organizada todas informações dos modelos citados anteriormente.

**Tabela 1** - Versões de Raspberry Pi e suas especificações

| **Ano** | ***Modelo***  ***Raspberry Pi*** | ***CPU***  ***(Broadcom)*** | ***RAM*** | ***Bluetooth*** | ***Internet*** |
| --- | --- | --- | --- | --- | --- |
| 2014 | **1B+** | 700 MHz | 512 MB | Não | Ethernet |
| **1A+** | 700 MHz | 256 MB | Não | Não |
| 2015 | **2B** | 900 MHz  quad-core | 1 GB | Não | Ethernet |
| **Zero** | 1 GHz | 512 MB | Não | Não |
| 2015 | **3B** | 1,2 GHz  quad-core | 1 GB | 4.1 | Ethernet e  Wi-Fi 2,4 GHz |
| 2017 | **Zero w** | 1 GHz | 512 MB | 4.1 | Wi-Fi 2,4 GHz |
| 2018 | **3B+** | 1,4 GHz  quad-core | 1 GB | 4.2 | Ethernet e  Wi-fi Dual Band |
| **3A+** | 1,4 GHz  quad-core | 512 MB | 4.2 | Wi-fi Dual Band |
| 2019 | **4B** | 1,5 GHz  quad-core | até 8 GB | 5.0 | Ethernet e  Wi-fi Dual Band |

Fonte: dados obtidos da Raspberry Pi Foundation

# 1.5 DEFINIÇÃO DE *SOFTWARE* LIVRE

O conceito de *Software* Livre foi desenvolvido inicialmente por Richard Stallman na década de 80. A ideia surgiu na época em que o programador trabalhava no Laboratório de Inteligência Artificial do MIT e precisou solicitar a liberação de um código-fonte ao fabricante de uma das impressoras utilizadas no Instituto, a qual havia quebrado. Após ter sua solicitação negada, Stallman empenhou-se em encontrar formas de facilitar a acessibilidade à códigos-fonte de programas, a partir daí surgiu a ideia de *Software* Livre.

Entende-se por *Software* Livre, um *software* onde os usuários possuem a liberdade de executar, estudar, copiar, distribuir e modificar esse s*oftware* sem qualquer restrição*.* Pelo termo “livre” não deve ser subentendido que o *software* seja grátis, mas sim que o mesmo, uma vez em posse do usuário, estará disponível para quaisquer eventuais modificações e formas de uso. Fundada por Stallman, a FSF - *Free Software Foundation* definiu quatro liberdades essenciais para um *software* ser considerado livre, sendo elas:

- Liberdade 0: A liberdade de executar o programa, para qualquer propósito.

- Liberdade 1: A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades.

- Liberdade 2: A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo.

- Liberdade 3: A liberdade de aperfeiçoar o programa, e liberar seus aperfeiçoamentos, de modo que toda a comunidade se beneficie.

Para que as liberdades 1 e 3 sejam devidamente atendidas, é imprescindível o acesso ao código-fonte do programa.

Para que um *software* seja considerado um *Software* Livre, o mesmo deve ser licenciado através de uma licença de *software* livre. Existem diversas licenças de *software* livre e embora esta prática não seja recomendada, nada impede que cada interessado crie sua licença, desde que a mesma atenda às quatro liberdades básicas. As licenças livres mais populares são: GLP – *General Public License* ou GNU, Licença BSD, *Mozilla Public License* e *Apache License*.

Atualmente milhares de softwares livres utilizam a licença GPL criada pela FSF. A GLP trata-se de uma licença que utiliza os princípios do direito autoral para proteger um *software* livre e assegurar que ninguém possa torná-lo proprietário. Dentro da licença GLP existe o conceito de *Copyleft*, o qual assegura-se de utilizar o inverso ao que se refere o *Copyright* - restringe o direito da cópia -, permitindo assim a liberdade de copiar. O *Copyleft*, cuja tradução aproximada seria “cópia permitida” ou “deixamos copiar”, é uma extensão das quatro liberdades essenciais, a qual exige que todas as versões modificadas ou extensões do programa também sejam livres.

É importante também ressaltar que há diferença entre *Software* Livre e Código Aberto, também conhecido como *Open Source*. As quatro liberdades essenciais para um software ser considerado livre também atendem à definição de Código Aberto, assim pode-se dizer que na maioria dos casos quando um *software* é livre, ele é também de código aberto, e vice-versa; O Código Aberto por sua vez possui mais seis premissas a serem seguidas. A real diferença entre *Software* Livre e *Open Source* está em seus objetivos e filosofias.

# 1.5.1 HISTÓRICO

A primeira versão das especificações e definições detalhadas de *Software* Livre foi publicada em 2001, desde então de tempos em tempos tais definições são revisitadas a fim de esclarecer eventuais dúvidas que venham a surgir por desenvolvedores ou usuários. A última revisão foi publicada em julho de 2019 e refere-se a versão 1.168, conforme apresentado na Figura 4:

**Figura 4** - Diretório de revisões das especificações de Software Livre



Fonte: Histórico obtido no site do GNU

# 1.5.2 PRINCIPAIS VERSÕES

A Tabela 2 apresenta alguns exemplos de Software Livre empregados em diferentes seções:

**Tabela 2**- Exemplos de Software Livre

| ***Desenvolvimento*** | ***Linguagens*** | ***Navegador*** | ***Servidor*** | ***S.O*** | ***Outros*** |
| --- | --- | --- | --- | --- | --- |
| Eclipse | Java | Konqueror | Apache | Linux | GIMP |
| GCC | PHP | Mozila Firefox | - | OpenSolaris | OpenOffice |
| MySQL | Python | - | - | - | VLC Media Player |
| NetBeans | Ruby | - | - | - | Worldpress |
| PostgreSQL | **-** | - | - | - | - |

Fonte: dados obtidos

# 1.5.3 RASPBERRY PI OS

# 1.5.3.1 HISTÓRICO

O sistema operacional *Raspbian* foi desenvolvido em 2012, ano em que o *Raspberry PI* original foi lançado e recentemente renomeado para *Raspberry PI OS*. Em conjunto à alteração do nome e implementações significativas no sistema operacional, foi lançada também uma versão de 64 bits do mesmo, a qual não utiliza o *software* do projeto *Raspbian*.

# 1.5.3.2 CARACTERÍSTICAS PRINCIPAIS

O *Raspberry PI OS* é o sistema operacional mais popular para execução de todos os modelos do *Raspberry PI*. É um sistema operacional de código aberto e variante de uma distribuição Debian do Linux. Caracteriza-se por ser uma versão mais simples e de fácil utilização recomendada para iniciantes que desejam executar o *Raspberry PI*. Além do sistema operacional, o *Raspberry PI OS* vem com mais de 35 mil pacotes pré-compilados e em um formato prático para fácil instalação no *Raspberry PI*.

O *Raspberry PI OS* possui uma interface gráfica de usuário chamada LXDE. Essa interface leve foi desenvolvida para funcionar bem em computadores lentos, de baixo desempenho e sistemas simplificados como netbooks e outros computadores e dispositivos de porte menor, como exemplo o *Raspberry PI*, o qual possui recursos limitados.

# 1.6 BENCHMARK

# 1.6.1 CONCEITO

O termo e técnicas de *Benchmark* são aplicados tanto no mundo corporativo quanto no tecnológico, salvas as diferenças de aplicação entre os mesmos. No ambiente corporativo o *benchmark* trata-se da análise por comparação de diferentes ecossistemas de uma empresa, sejam eles operacional, comercial e desenvolvimento estratégico.

Em decorrência da evolução constante das arquiteturas de computadores, tornou-se cada vez mais difícil comparar o desempenho de sistemas de computação validando somente as especificações citadas pelo seu desenvolvedor. O *Benchmark* no ambiente tecnológico é utilizado visando aferir o desempenho entre dispositivos a partir de testes padrões, permitindo assim a comparação dos resultados mesmo que os produtos envolvidos nos testes tenham arquiteturas diferentes; O intuito dos testes é basicamente medir o quão rápido um dado sistema computacional efetua um determinado conjunto de tarefas.

Existem dois tipos de testes de *benchmark*: os Sintéticos e os de Aplicação. Os *benchmarks* sintéticos consistem em dados gerados artificialmente em ambientes de execução de programas reais a fim de terem comportamentos ou ações desejados pelo usuário, possibilitando assim a inserção de características específicas de modo a explorar deficiências encontradas ou pressentidas nos sistemas a serem avaliados. Por sua vez, os *benchmarks* de Aplicação executam programas do “mundo real” em situações reais; O intuito dos *benchmarks* de Aplicação é executar as funções do sistema no máximo de sua capacidade a fim de extrair todo o potencial e validar com maior precisão até onde a funcionalidade pode chegar.

# 1.6.2 PRINCIPAIS TIPOS

Dada a diversidade de domínios de aplicações, os quais possuem diferentes características de execução, estão disponíveis no mercado diversos tipos de *benchmarks* para diferentes programas e sistemas a serem analisados, dentre eles *benchmarks* para análises de CPU, *hardware*, banco de dados, para celulares, entre outros.

O *benchmark* Whetstone, desenvolvido em Algol 60 no ano de 1972 pelo National Physical Laboratory na Inglaterra foi um dos primeiros grandes programas escritos para *benchmark* e continua entre os principais tipos utilizados atualmente. Trata-se de um *benchmark* sintético projetado para medir o comportamento de programas científicos onde seu objetivo principal é medir o desempenho da aritmética de inteiros e de ponto flutuante.

Desenvolvido em 1984, o *benchmark* Dhrystone tem como objetivo a avaliação de operações aritméticas simples, operações com *string*, decisões lógicas e acessos de memória com o intuito de refletir as atividades da CPU nas aplicações de computação. A principal diferente entre os *benchmarks* Whetstone e Dhrystone é a ênfase que o Whetstone têm em operações numéricas, enquanto o Dhrystone têm maior ênfase em operações de *string*.

Tratando de benchmarks para avaliar o desempenho de aparelhos celulares, o aplicativo mais utilizado é o AnTuTu. O aplicativo está disponível gratuitamente na App Store e Google Play e permite ao usuário avaliar a capacidade do seu hardware em testes sintéticos e de aplicação, além de comparar a pontuação obtida com outros dispositivos ao redor do mundo. O aplicativo também fornece diversos recursos de monitoramento que permitem avaliar as temperaturas dos circuitos, saúde da bateria e até mesmo descobrir se o celular é original.

Para a execução da bateria de testes do seguinte trabalho serão utilizados os *benchmarks* Linpack e RAMspeed/SMP, os quais serão descritos com maior precisão no tópico 2.1.2.2 e 2.1.2.3.

# 2 PROJETO DE EXPERIMENTO

A fim de comparar o desempenho de duas gerações do Raspberry PI, sendo elas Raspberry PI model B e Raspberry PI 4 model B+, foi desenvolvido um projeto de experimento. Esse projeto consiste em executar dois *benchmarcks que testaram o desempenho do processador e da memória.*

***2.1* AMBIENTE DE EXPERIMENTAÇÃO**

**2.1.1 HARDWARES UTILIZADOS**

**2.1.1.1 RASPBERRY PI MODEL B**

Descrição do hardware e periféricos

**2.1.1.2 RASPBERRY PI 4 MODEL B**

O experimento utiliza um RASPBERRY PI 4 MODEL B com uma fonte recomendada pelo fabricante de 5 volts e 3000mAh, periféricos: mouse, teclado e um adaptador wireless para USB.

Figura 5: Fonte de energia Raspberry PI 4 B



*Fonte: Próprio Autor(2020)*

Figura 6: Raspberry PI 4 B em um case, com adaptador wireless USB



Fonte: Próprio Autor(2020)

**2.1.2 SOFTWARES UTILIZADOS**

**2.1.2.1 Raspberry PI OS 1.4**

Como citado anteriormente na seção 1.5.3, o Raspberry PI OS é um Sistema Operacional largamente utilizado nos dispositivos criados pela organização que os desenvolve. Por este motivo foi o S.O. escolhido para rodar os softwares de benchmark sendo o S.O. que mais se aproxima da realidade de uso comum. Foi feita uma instalação limpa do S.O., com vistas de não influenciar na execução dos benchmarks com outros aplicativos ou serviços sendo executados em background.

**2.1.2.2 Linpack**

O pacote Linpack é uma coleção de sub-rotinas Fortran que analisam e resolvem equações lineares e problemas de mínimos quadrados lineares. O Linpack foi projetado para supercomputadores em uso nos anos de 1970 e inícios dos anos 1980. O Linpack foi amplamente substituído pelo Lapack, que foi projetado para funcionar com eficiência em supercomputadores vetoriais com memória compartilhada. (DONGARRA, LUSZCZEK, PETITET, 2002)

O pacote Linpack foi baseado em outro pacote, chamado de *Level 1 Basic Linear Algebra Subroutines* (BLAS). A maior parte do trabalho de ponto flutuante dentro dos algoritmos Linpack é realizada pelo BLAS, que torna possível tirar proveito de hardware de computadores especiais sem ter que modificar o algoritmo subjacente.

Para as baterias de testes foi adicionado ao código do Linpack escrito em Linguagem C a funcionalidade de exportar o resultado para um arquivo CSV como demonstrado nas imagens 7.1 e 7.2.

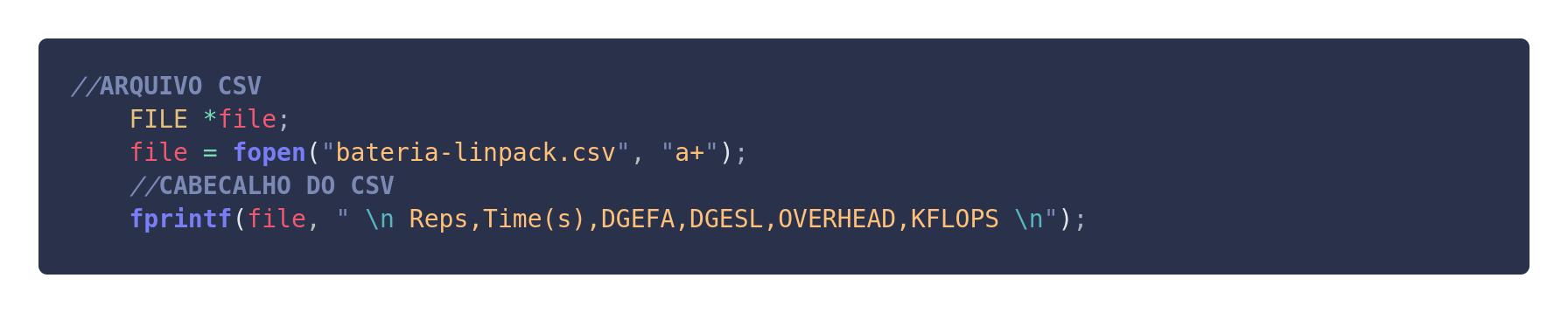
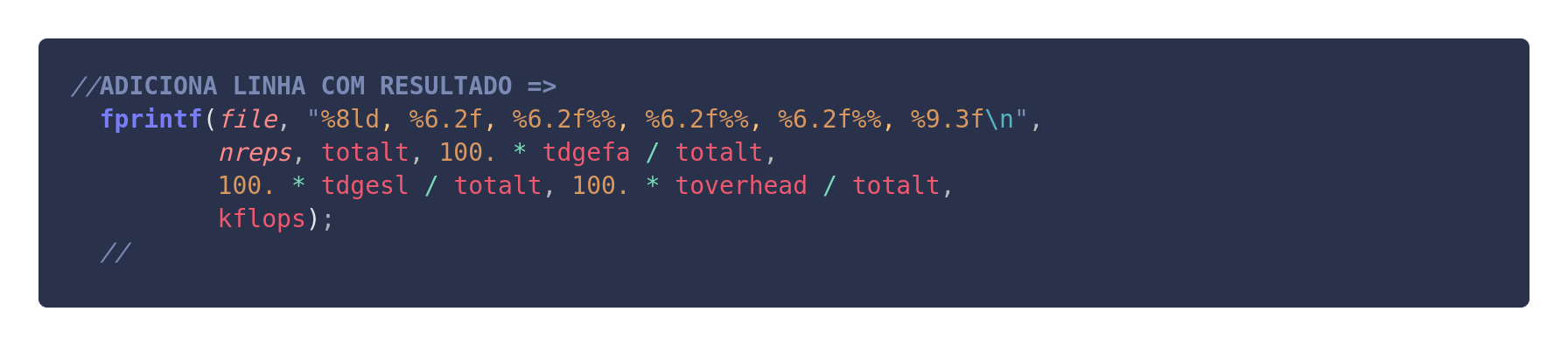
Figura 7.1: Criação de um Ponteiro para um arquivo e a escrita do cabeçalhoFonte: Próprio Autor(2020)

Figura 7.2: Escrita do resultado da execução dentro do arquivo CSV

Fonte: Próprio Autor(2020)

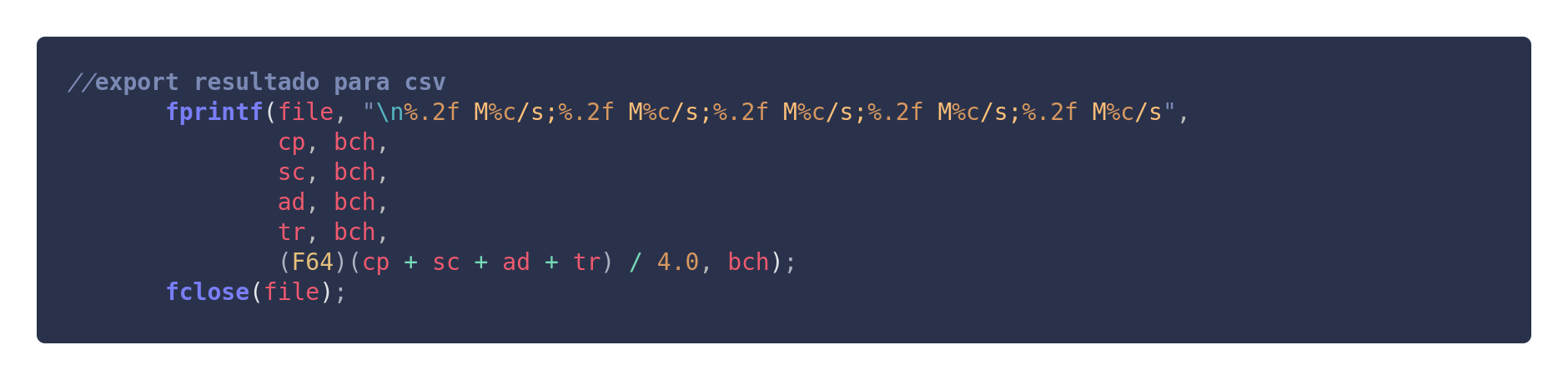
**2.1.2.3 RAMSpeed/SMP**

RAMSpeed é um utilitário de linha de comando que mede a largura de banda efetiva do cache e da memória principal. Foi escrito inteiramente em C embora tenha versões em outras linguagens de programação. Este benchmark teste o desempenho de memória (RAM) do sistema através de cálculos com números inteiros e reais. A unidade de medida utilizada é MB/s (Megabytes por segundo).

Para as baterias de teste foi adicionada à funcionalidade de exportar o resultado para um arquivo CSV, diretamente no código fonte do benchmark como demonstrado nas figuras 8.1 e 8.2.

Figura 8.1: Criação de um Ponteiro para um arquivo em modo de adiçãoFonte: Próprio Autor (2020)



Figura 8.2: Escrita do resultado de uma execução no arquivo CSVFonte: Próprio Autor (2020)

**2.2 EXPERIMENTO**

O experimento foi realizado executando os dois benchmarks de forma totalmente independente e individual. Foi utilizado um shell script para cada benchmark, figuras 9.1 e 9.2.

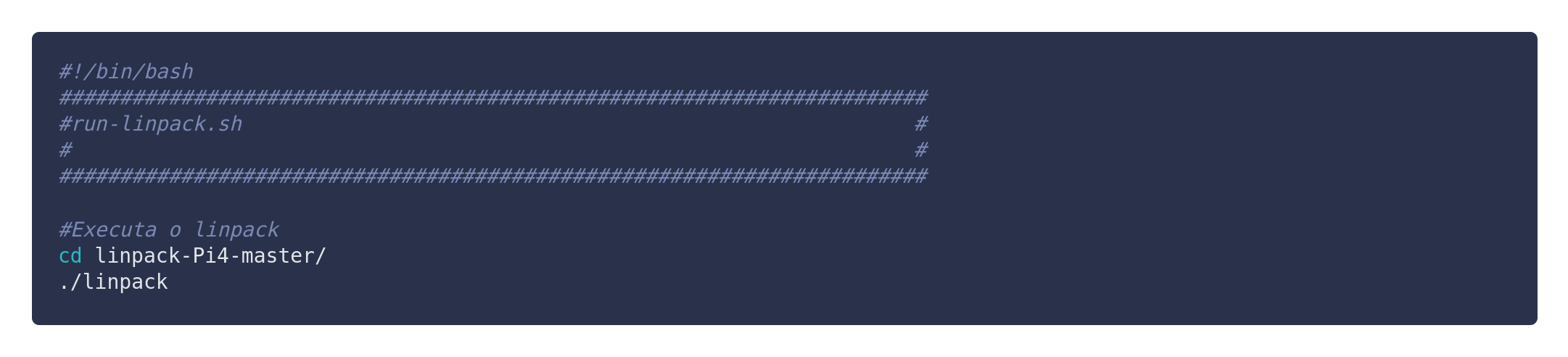
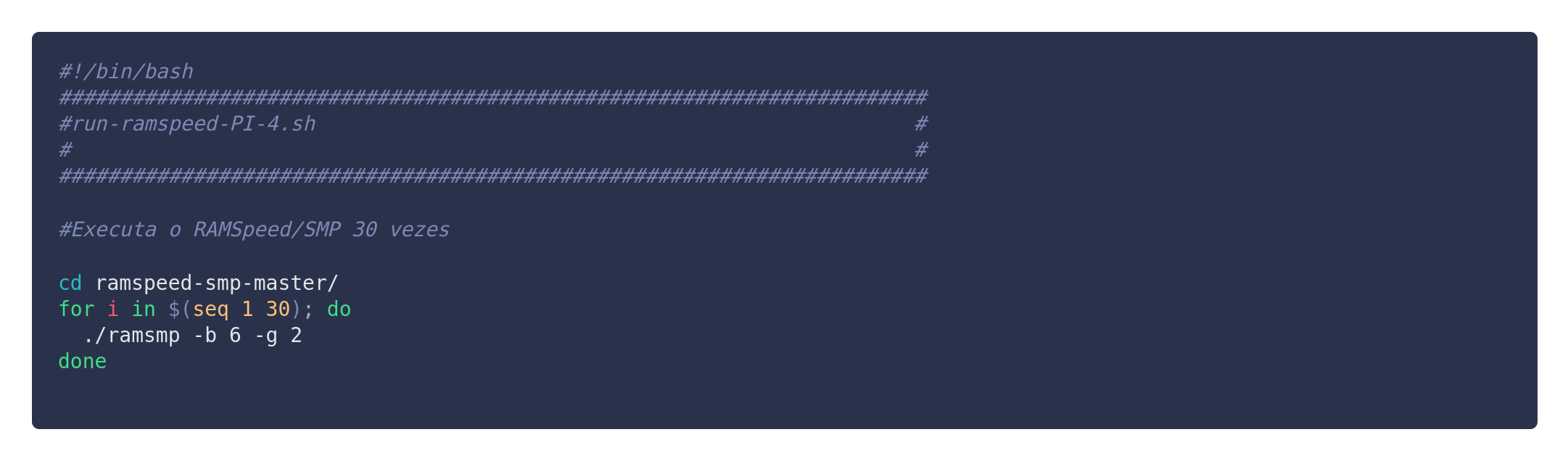
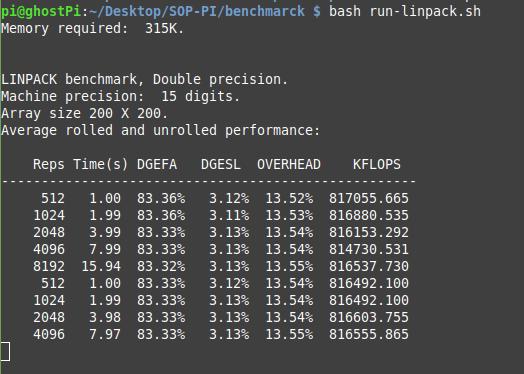
Figura 9.1: Shell Script que executa o LinpackFonte: Próprio Autor (2020)

Figura 9.2: Shell Script que executa o RamSpeed/SMPFonte: Próprio Autor (2020)

Para fins estatísticos foi necessário executar os benchmarks 30 vezes cada um, para cada execução do benchmark é gerada uma saída no arquivo CSV do respectivo benchmark de modo que cada execução tenha seus dados persistidos para posterior análise.

Figura 10: Exemplo da saída no console executando o Linpack no Raspberry PI 4 B 

Fonte: Próprio Autor (2020)

Figura 11: Exemplo de saída no console executando o RamSpeed/SMP no Raspberry PI 4 B

<#######Incluir imagem do RPI 4 B executando o RamSpeed#################>

Fonte: Próprio Autor (2020)

# REFERÊNCIAS BIBLIOGRÁFICAS

**ANTUTU Benchmark**. Disponível em: <<http://www.antutu.com/en/index.htm>>. Acesso em: 05 set. 2020.

**ARDUINO**. 2020. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 05 set. 2020

DONGARRA, Jack; LUSZCZEK, Piotr; PETITET, Antoine. **The LINPACK Benchmark: Past, Present, and Future**. Disponível em: <<http://www.netlib.org/utk/people/JackDongarra/PAPERS/hplpaper.pdf>> Acesso em: 25 ago. 2020

DONGARRA, Jack. **LINPACK.** Disponível em: <<http://www.netlib.org/linpack/index.html>>

Acesso em: 25 ago. 2020

EDWARDS, C. **Not-so-humble raspberry pi gets big ideas**. Engineering Technology, v. 8, n. 3, p. 30-33, 2013. ISSN 1750-9637.

**GNU - General Public License**. Disponível em: <<https://www.gnu.org/licenses/gpl-3.0.html>>. Acesso em: 04 set. 2020.

HARRINGTON, William. **Learning Raspbian**. Birmingham: Packt Publishing Ltd, 2015. Disponível em: <https://bityli.com/r4Jay>. Acesso em 07 set. 2020.

HOLLANDER, Rhett; BOLOTOFF, Paul. **RAMspeed, a cache and memory benchmarking tool.** Disponível em: <<http://www.alasir.com/software/ramspeed/>> Acesso em: 25 de ago. 2020

NEMETH, Evi; SNYDER, Garth; HEIN, Trent R. **Manual completo do linux**. 2. ed. Trad. Carlos Schafranski e Edsib Furmankiewicz. São Paulo: Prentice Hall, 2007. 684p.

**Raspbian**. Disponível em: <<https://www.raspbian.org/>>. Acesso em: 06 set. 2020.

THE RASPBERRY FOUNDATION. **Raspberry PI**. 2020. Disponível em: <<https://www.raspberrypi.org/>>. Acesso em: 25 ago. 2020

THE RASPBERRY FOUNDATION. **Raspberry PI 4 Model B**. 2020. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>>. Acesso em: 05 set. 2020

**What is free software?**. Disponível em: <<https://www.gnu.org/philosophy/free-sw.pt-br.html>>. Acesso em 06 set. 2020.

**Whetstones**. Disponível em: <<https://www.keil.com/benchmarks/whetstone.asp>>. Acesso em: 05 set. 2020.

XUN, Y. et al. **A platform for system-on-a-chip design prototyping**. In: ASIC, 2001. Proceedings. 4th International Conference on. [S.l.: s.n.], 2001. p. 781-784.

**CRONOGRAMA**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***ATIVIDADE*** | ***Agosto*** | ***Setembro*** | ***Outubro*** | | | |
| ***Prev*** | ***Real*** | ***Prev*** | ***Real*** | ***Prev*** | ***Real*** |
| Entrega do Projeto | 25 |  |  |  |  |  |
| Entrega Capítulo 1 |  |  | 07 |  |  |  |
| Entrega Capítulo 2 |  |  | 14 |  |  |  |
| Entrega Capítulo 3 |  |  | 21 |  |  |  |
| Entrega Considerações/Introdução |  |  | 28 |  |  |  |
| Entrega TE1 Completo |  |  |  |  | 03 |  |
| Entrega da Apresentação |  |  |  |  | 03 |  |
| Apresentação |  |  |  |  | 05 |  |