

CAETANO MASTROIANNI DIEGUEZ  
DANILO PROFITTI ASSIS

**SneSPY: um console de videogame com comunicação  
streaming e multiplayer**

São Paulo  
2015

CAETANO MASTROIANNI DIEGUEZ  
DANILO PROFITTI ASSIS

**SneSPY: um console de videogame com comunicação  
streaming e multiplayer**

Trabalho de Conclusão de Curso apresentado  
à Escola Politécnica da Universidade de São  
Paulo para obtenção do título de Bacharel  
em Engenharia Elétrica com ênfase em  
Sistemas Eletrônicos

Orientador: Prof. Dr. Bruno de Carvalho Al-  
bertini

São Paulo  
2015

CAETANO MASTROIANNI DIEGUEZ  
DANILO PROFITTI ASSIS

## **SneSPY: um console de videogame com comunicação streaming e multiplayer**

Trabalho de Conclusão de Curso apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do título de Bacharel em Engenharia Elétrica com ênfase em Sistemas Eletrônicos

Área de Concentração:  
Engenharia Elétrica com Ênfase em Sistemas Eletrônicos

Orientador: Prof. Dr. Bruno de Carvalho Albertini

São Paulo  
2015

# Resumo

Este trabalho busca a criação de um console de videogame portátil e versátil voltado a jogos antigos. O principal objetivo do projeto é a possibilidade de reprodução do áudio e do vídeo dos jogos em qualquer tela, tanto para um jogador único como em modo *multiplayer*, e a completa independência de conexões por cabos, permitindo assim que se jogue em locais com infraestrutura mínima e sem a necessidade de outros acessórios ou dispositivos auxiliares. O projeto se insere no mundo moderno interconectado, quando em vez de ignorar e substituir, busca integrar um novo dispositivo às tecnologias que o usuário já possui, enquanto faz uma ponte com o passado tentando resgatar os jogos clássicos. Tais jogos não são mais do interesse das grandes empresas, mas ainda encontram um grande público de entusiastas, e serviram de inspiração e motivação para grande parte do mundo dos atual dos jogos e da cultura pop.

Palavras-chaves: videogame, console, emulador, Raspberry Pi, streaming.

# Abstract

This work aims the creation of a versatile and portable videogame console focused on classic games. The main objective is the possibility of reproducing the games' audio and video in any screen, for both singleplayer and multiplayer modes, and the complete independence from cable connections, thus allowing gaming in places with minimal infrastructure and without other accessories or auxiliary devices. The project is inserted in the new interconnected world where, instead of ignoring and replacing, it integrates a new device with others that the user may own, while also connects itself with the past, trying to give new life to classic games, that no longer concerns the big companies, even though they have an huge fanbase, inspired and motivated the current gaming world and pop culture.

Keywords: videogame, console, emulator, Raspberry Pi, streaming.

## Lista de ilustrações

Figura 1 – Auto Race, o primeiro console portátil, lançado em 1977 . . . . .	15
Figura 2 – Árvore de objetivos de marketing do console . . . . .	22
Figura 3 – Decomposição funcional nível 0 - Diagrama Físico . . . . .	29
Figura 4 – Decomposição funcional nível 1 - Diagrama Lógico . . . . .	32
Figura 5 – EAP . . . . .	33
Figura 6 – Cronograma . . . . .	35
Figura 7 – Diagrama do <i>streaming</i> . . . . .	37
Figura 8 – Diagrama do <i>multiplayer</i> em rede . . . . .	38
Figura 9 – Preços . . . . .	41
Figura 10 – Sistema operacional RetroPie funcionando. . . . .	45
Figura 11 – Emulação da imagem do jogo Mighty Morphin Power Rangers: The Movie	46
Figura 12 – <i>Streaming</i> e reprodução local da imagem do jogo . . . . .	47
Figura 13 – <i>Streaming</i> da imagem do jogo . . . . .	48
Figura 14 – Consumo de bateria pelo SneSPY . . . . .	49
Figura 15 – Montagem do console com <i>power bank</i> , controle, e módulo Wi-Fi . . . . .	49
Figura 16 – Design estilo Gameboy . . . . .	50
Figura 17 – Design estilo controle de Playstation/XBOX . . . . .	50
Figura 18 – Montagem estilo console de mesa . . . . .	50

## Lista de tabelas

Tabela 1 – Requisitos de marketing . . . . .	23
Tabela 2 – Requisitos de engenharia . . . . .	23
Tabela 3 – <i>Benchmarking</i> . . . . .	24
Tabela 4 – Comparação. Quesitos: conexão sem fio, plataforma e modo de serviço <i>streaming</i> . . . . .	26
Tabela 5 – Continuação da comparação. Quesitos: energia, memória e emulador .	27

# Sumário

1	<b>VISÃO GERAL</b> . . . . .	15
1.1	Uma breve história dos consoles portáteis . . . . .	15
1.2	Nossa motivação . . . . .	16
2	<b>DECLARAÇÃO DO PROBLEMA</b> . . . . .	17
2.1	Declaração das necessidades . . . . .	17
2.2	Declaração dos objetivos . . . . .	18
3	<b>O PROJETO</b> . . . . .	19
3.1	Descrição . . . . .	19
3.2	Tecnologias relevantes . . . . .	20
3.3	Árvore de objetivos . . . . .	22
4	<b>ESPECIFICAÇÃO DE REQUISITOS DE ENGENHARIA</b> . . . . .	23
4.1	<i>Benchmarking</i> . . . . .	23
5	<b>ESTUDO E ESCOLHA DE CONCEITOS PARA O SNESPY</b> . . . . .	25
6	<b>DECOMPOSIÇÃO FUNCIONAL</b> . . . . .	29
6.1	Nível 0 . . . . .	29
6.2	Nível 1 . . . . .	30
7	<b>GERENCIAMENTO DO PROJETO</b> . . . . .	33
7.1	EAP . . . . .	33
7.2	Cronograma . . . . .	33
8	<b>PROVA DE CONCEITO</b> . . . . .	37
8.1	Método de <i>streaming</i> . . . . .	37
8.2	Método de <i>multiplayer</i> em rede . . . . .	38
9	<b>ANÁLISE DE CUSTOS</b> . . . . .	41
10	<b>ANÁLISE DE RISCOS</b> . . . . .	43
10.1	<i>Hardwares e softwares</i> . . . . .	43
10.2	Métodos . . . . .	43
10.3	Modelo de negócios . . . . .	43
11	<b>IMPLEMENTAÇÃO DO PROTÓTIPO</b> . . . . .	45

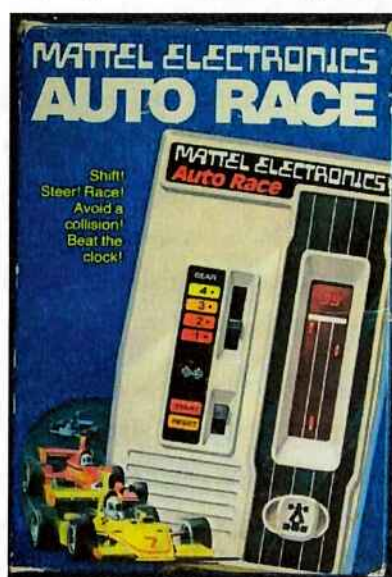


# 1 Visão geral

## 1.1 Uma breve história dos consoles portáteis

Em 1977, fabricado pela Mattel, surgiu o primeiro console portátil lançado no mercado: O Auto Race. Ele contava com um botão *on/off* e dois botões deslizantes, um para controlar a direção do “carrinho” e outro para sua velocidade. No visor, os carros eram representados por pontos de luz vermelhos e o objetivo era conseguir completar quatro voltas em até 99 segundos. Todo o sistema foi programado em apenas 512 bytes de memória, muitas vezes menor do que os jogos atuais, que já passam da casa dos gigabytes.

Figura 1: Auto Race, o primeiro console portátil, lançado em 1977



Na década de 80 surge o Game & Watch, o primeiro produto eletrônico da história da Nintendo que, até então, trabalhava apenas com baralhos e jogos de tabuleiro.

A evolução da tecnologia propiciou o surgimento de outros consoles e alguns portáteis destacaram-se no mercado, como o Game Boy de 1989 (e seus sucessores nas versões Color e Advanced) da Nintendo, além das soluções de outras empresas como o Atari Lynx, o Turbo Express, o Neo Geo Pocket, entre outros (CASANOVA, 2015).

Dos portáteis mais atuais, são destaques o PSP e o PS Vita da Sony e o Nintendo DS (cujo sucessor chama-se 3DS), todos com uma vasta lista de funcionalidades e jogos. Estes consoles trouxeram consigo uma quebra de paradigma no que diz respeito à qualidade técnica dos portáteis, que costumava ser inferior à dos consoles de mesa.

Além dos consoles portáteis, também celulares, smartphones e tablets entraram no

## 2 Declaração do problema

### 2.1 Declaração das necessidades

A atual indústria dos grandes consoles fornece máquinas caras e poderosas, com grandes capacidades de processamento e gigabytes de memória que conseguem rodar os últimos lançamentos de jogos. Estes contam com mapas abertos, realismo nos detalhes e cálculos precisos de física. Mas um dos maiores desapontamentos de um jogador é investir em uma máquina com esta capacidade e ficar limitado à nossa sala ou quarto.

Já os videogames portáteis atuais investem bastante em inovação e em maior fidelidade gráfica. Por exemplo, a Nintendo investiu na tecnologia 3D e nas telas sensíveis ao toque com o Nintendo 3DS, enquanto a Sony apostava em poder computacional com uma CPU 2GHz Quad-Core, 512MB de RAM e 128 MB na placa gráfica em seu PlayStation Vita. Também a adoção de padrões exclusivos em conectores e mídias dificulta a integração dessas tecnologias com os outros dispositivos como computadores e televisores. Em geral, as empresas de tecnologia investem em soluções tecnológicas que integram apenas os seus produtos, como no caso da Apple que por meio da AppleTV permite conectar facilmente o iPhone, o MacBook e a televisão.

Apesar de toda essa evolução tecnológica, ainda não existe um console portátil, de fácil usabilidade, capaz de reproduzir áudio e vídeo de maneira versátil em qualquer local, de acordo com a infraestrutura que o ambiente proporciona (conexão sem fio, reprodução em telas maiores como televisão, entre outros), que integre mais de um jogador no sistema para jogos *multiplayer* e que rodem jogos mais antigos, agradando os jogadores que preferem os jogos clássicos.

Uma tecnologia de console móvel também não pode depender de conexões físicas ou por cabeamento. Este tipo de tecnologia deve ser leve e fácil de carregar, portanto não há espaço para que o usuário carregue consigo cabos e conectores. Enquanto televisores e computadores apresentam hoje grande suporte a conexões físicas, como entradas USB e HDMI, além do incentivo a tecnologias chamadas *plug and play* que permitem o uso quase instantâneo sem a necessidade de longos e complicados processos de instalação. Ainda existem algumas complicações ao se utilizar tecnologias sem fio, pois estas exigem do usuário algum conhecimento sobre como se conectar em redes, como instalar os aplicativos necessários para receber o áudio e vídeo de seus jogos e como estabelecer conexões com outros usuários para jogos *multiplayer*. Dessa forma, ao se investir nesse tipo de tecnologia, deve se tomar cuidado para se desenvolver um método versátil e fácil de se estabelecer as conexões, para que o console seja acessível ao maior número de usuários.

## 3 O projeto

### 3.1 Descrição

Desde seu lançamento em 1991, o sistema conhecido como SNES (*Super Nintendo Entertainment System*) foi um dos consoles de mesa com maior sucesso na história dos videogames, tanto para os jogadores de um modo geral quanto para os entusiastas do mundo dos jogos, fazendo com que surgissem vários estudos e meios de emular tal sistema, principalmente após a descontinuidade da fabricação e suporte aos consoles existentes.

Sendo assim, tendo em vista que o SNES possui especificações técnicas completas divulgadas em meios eletrônicos de acesso livre, diferentemente da grande maioria dos consoles antigos, decidimos fazer dele a principal arquitetura alvo a ser integrada em nosso console.

Emulando o SNES em um processador embarcado ou computador de placa-única, planejamos integrar processador, memórias, processamento gráfico e comunicação sem fio em um único dispositivo que também será o controle utilizado pelo usuário. O maior desafio dessa etapa do trabalho é a integração do hardware em um único aparelho pequeno o suficiente para que o usuário possa transportá-lo facilmente e de forma que não atrapalhe o conforto do usuário durante as partidas.

O único componente de hardware que não está incluso no console é o sistema de reprodução de áudio e vídeo. Através de um sistema de comunicação sem fio, o usuário poderá conectar o seu console a qualquer *smartphone*, computador ou televisor inteligente que suporte a plataforma de *streaming*, utilizando estes dispositivos para a reprodução dos jogos. Os dispositivos que serão utilizados como dispositivos de saída precisam suportar a execução de um reprodutor de *streaming* compatível com o padrão que será adotado. O desafio desta etapa é desenvolver uma comunicação estável e contínua entre o console e os equipamentos, de forma que não haja interrupção dos jogos por problemas de interferência ou perda de dados.

Através da mesma rede, outra unidade do console poderá detectar a comunicação e assim requisitar para se juntar à conexão, permitindo assim que os jogadores participem de jogos em *multiplayer*. As partidas em *multiplayer* do SNES foram concebidas para que todos os jogadores compartilhassem a mesma televisão, portanto todos os jogadores possuem acesso às mesmas informações simultaneamente. Para permitir mobilidade, nosso sistema incluirá a possibilidade de que o mesmo sinal de *streaming* seja enviado para dois ou mais dispositivos de saída diferentes, de forma que os jogadores possam utilizar cada um uma tela diferente, mas sem alterar a jogabilidade dos jogos de SNES. O desafio

- **Wi-Fi**(ALECRIM, 2015)

Wi-Fi é um conjunto de especificações para redes locais sem fio (WLAN - *Wireless Local Area Network*) baseada no padrão IEEE 802.11. O nome Wi-Fi é tido como uma abreviatura do termo inglês *Wireless Fidelity*, embora a Wi-Fi Alliance, entidade responsável principalmente pelo licenciamento de produtos baseados na tecnologia, nunca tenha afirmado tal conclusão. É comum encontrar o nome Wi-Fi grafado como Wi-Fi, Wi-fi ou até mesmo wifi. Todas estas denominações se referem à mesma tecnologia.

Com a tecnologia Wi-Fi, é possível implementar redes que conectam computadores e outros dispositivos compatíveis (*smartphones, tablets, consoles de videogame, impressoras, etc.*) que estejam próximos ao alcance do sinal. Estas redes não exigem o uso de cabos, já que efetuam a transmissão de dados por meio de radiofrequência. Este modelo oferece várias vantagens, entre elas: permite ao usuário utilizar a rede em qualquer ponto dentro dos limites de alcance da transmissão; possibilita a inserção rápida de outros computadores e dispositivos na rede; e evita que paredes ou estruturas prediais sejam furadas ou adaptadas para a passagem de fios.

A flexibilidade do Wi-Fi torna viável a implementação de redes que fazem uso desta tecnologia nos mais variados lugares, principalmente pelo fato de as vantagens citadas no parágrafo anterior muitas vezes resultarem em diminuição de custos. Assim sendo, é comum encontrar redes Wi-Fi disponíveis em hotéis, aeroportos, rodoviárias, bares, restaurantes, shoppings, escolas, universidades, escritórios, hospitais, etc. Para utilizar estas redes, basta ao usuário ter um *laptop, smartphone* ou qualquer dispositivo compatível com Wi-Fi.

- **Projeto RetroPie**(PETROCKBLOG, 2015)

O projeto RetroPie é uma coleção de projetos que têm todos o objetivo de transformar o Raspberry Pi em um console dedicado a reviver os jogos antigos. Trata-se de uma customização do sistema operacional Raspbian que integra uma série de emuladores para uma larga variedade de sistemas do passado. Além dos emuladores, também fazem parte do projeto uma interface gráfica, além de vários outros softwares. O *script* de instalação do RetroPie é a parte central do software para instalação e configuração desses componentes. Já existe uma imagem para cartão SD com a instalação completa do projeto, incluindo todos emuladores e funções. Tecnicamente, o *script* de instalação do RetroPie é um *script bash* que consiste de algumas milhares de linhas de código e é executado à partir da linha de comando. O código fonte completo está armazenado no site Github para colaboração, *feedback*, relatos de falhas e requisições.

## 4 Especificação de requisitos de engenharia

À partir dos requisitos de marketing especificados anteriormente, e listados na tabela 1, foram compilados requisitos de engenharia que os atendem, como pode ser visto na tabela 2.

Tabela 1: Requisitos de marketing

Requisitos de Marketing
a. Versatilidade
b. Conexão de áudio e vídeo sem fio
c. Conexão tradicional a/v com cabo
d. Conexão sem fio de fácil realização
e. <i>Multiplayer</i> sem fio
f. Leve
g. Ergonômico
h. Bateria recarregável
i. Boa conexão dentro de uma mesma sala

Tabela 2: Requisitos de engenharia

Requisitos de Engenharia
I. O console deve ser capaz de anunciar seu IP para possibilitar a transmissão do <i>streaming</i> para aparelhos diversos
II. O console deve ser capaz de emular todos jogos de SNES
III. O console deve ser capaz de formar uma rede <i>multiplayer</i> de no mínimo 2 jogadores
IV. A defasagem entre as transmissões para duas telas diferentes (console principal e console cliente) não pode ser maior que o tempo de meio frame
V. Boa conexão sem-fio em um raio de 6m
VI. Possuir uma bateria que dure por volta de 4 horas

### 4.1 *Benchmarking*

À partir desses requisitos, podemos comparar o comportamento desejado de nosso console com o de vários consoles, alguns famosos como o Nintendo 3DS e o PlayStation Vita e outros mais desconhecidos como o GP2X e o Dingoo.

A comparação será feita seguindo os critérios listados abaixo:

1. Conectividade;
2. Emulação de jogos antigos;

## 5 Estudo e escolha de conceitos para o SneSPY

Para o nosso projeto, houve algumas decisões não triviais a serem tomadas. Para nos ajudar, realizamos uma análise de forças e fraquezas, ponderadas à partir do ponto de vista dos requisitos estabelecidos anteriormente.

Houve dois critérios principais utilizados. Primeiro, foi a capacidade de se obter um console funcional dentro do prazo de um ano, portanto foi dado mais peso à utilização de plataformas e softwares prontos e já bem especificados que poderiam nos dar suporte na implantação das funcionalidades exigidas, em vez de gastar muito tempo no projeto destes componentes. E segundo, foi sempre buscar maior comodidade e facilidade de operação do console pelo usuário.

À partir das tabelas 4 e 5 decidimos que a tecnologia de comunicação sem fio a ser utilizada será a Wi-Fi. O grande fator de decisão foi a incapacidade do Bluetooth de fornecer banda suficiente ao *streaming* A/V.

Em relação à plataforma, foi decidido pelo desenvolvimento do projeto em um Raspberry Pi, principalmente pelo fato de que já existe um projeto de código livre, o RetroPie, bastante desenvolvido, sobre o qual temos maiores capacidades de expansão e da implementação das funcionalidades que queremos, sem precisar gastar muito tempo desenvolvendo um hardware dedicado.

Buscando o uso do console em qualquer tipo de local, optamos por um método de conexão *ad hoc peer-to-peer*, na qual duas unidades do console poderão estabelecer uma comunicação diretamente entre si, sem depender de um servidor central que necessitasse de uma infraestrutura de conexão com a internet, já que a falta desta poderia se tornar impeditiva para o uso do videogame.

Foi decidido que o sistema será energizado por uma bateria Li-ion, tecnologia hoje presente em diversos celulares e outros dispositivos, com a vantagem de ser recarregável e de maior capacidade.

A decisão sobre o local onde armazenar os jogos, já que o Raspberry Pi não possui uma memória interna voltada para o armazenamento de dados, foi feita principalmente pelo fato que o cartão SD é muito mais discreto do que um pendrive. Mas vale notar que o console possuirá uma entrada USB disponível para o usuário poder carregar novos jogos sem precisar desligar o sistema.

Por fim, iremos utilizar o emulador Snes9x no console, pelos mesmos motivos observados em relação à plataforma: já é um programa bem estabelecido e com uma boa

Tabela 5: Continuação da comparação. Quesitos: energia, memória e emulador

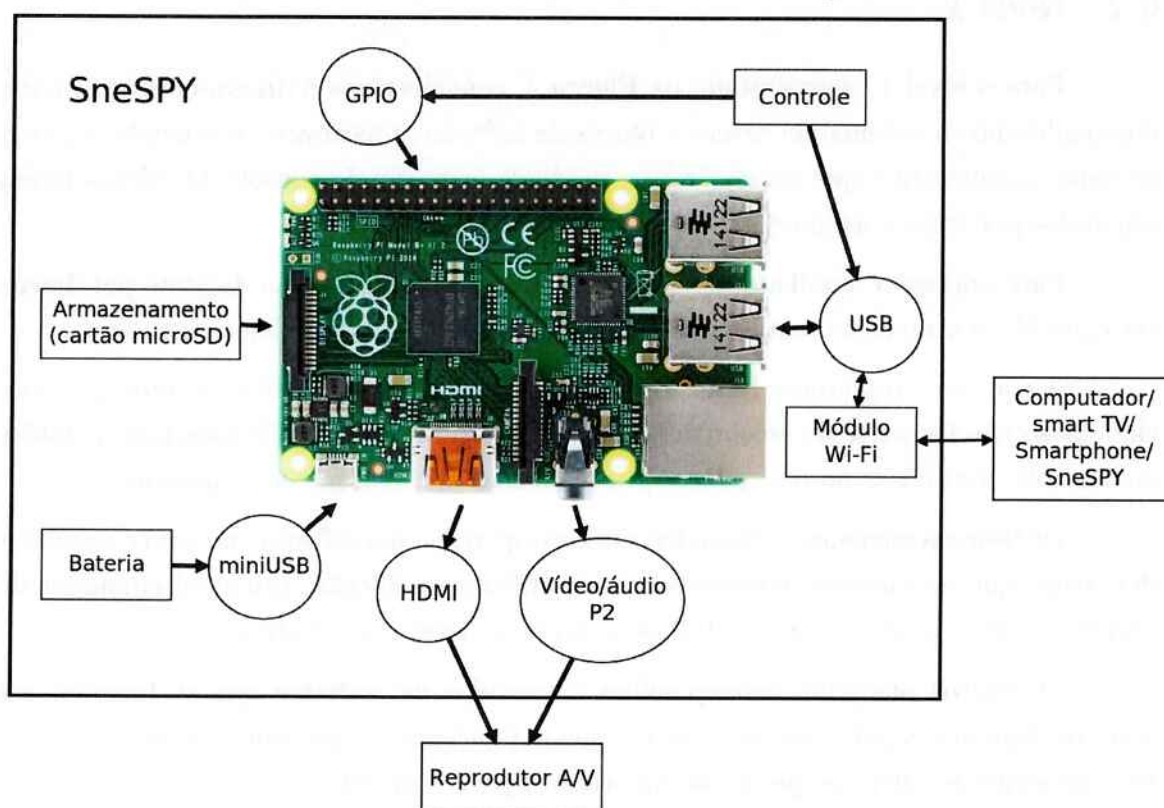
Quesitos	Forças	Fraquezas	Avaliação
<b>Energia</b>			
Bateria Li-ion	Alta capacidade + Fácil recarga +	Maior custo -	1
Pilha AAA	Maior facilidade de reposição +	Baixíssima capacidade - - -	-2
Bateria 9V	Maior facilidade de reposição +	Baixa capacidade - -	-1
Tomada	Energia ilimitada ++	Depende de conexão física - - -	-1
<b>Memória</b>			
Pendrive USB	Taxas de transferência levemente maiores + Alta capacidade de armazenamento +	O tamanho pode atrapalhar o uso do console pelo usuário -	1
Cartão SD	Tamanho extremamente pequeno e discreto ++ Alta capacidade de armazenamento +	Parte da memória é ocupada pelo sistema operacional -	2
<b>Emulador</b>			
Snes9x	Emulador mais famoso e bem conhecido + Possui maiores informações divulgadas para análise +	Não possui fraquezas relevantes	2
Fabricação própria	Maior liberdade de projeto ++ Não depende de conexão com a internet ++	Ainda precisa ser desenvolvido - -	0
RetroArch	Emulador bem conhecido +	Emulador genérico com características que não serão usadas e que podem complicar o projeto -	0

## 6 Decomposição Funcional

### 6.1 Nível 0

Nosso projeto envolve uma integração de hardware e software, logo, para um melhor entendimento de nosso projeto, podemos entender o nível 0 da decomposição funcional como o diagrama das conexões físicas que o compõe. Assim como vemos na Figura 3, o console SneSPY será composto por cinco blocos de hardware.

Figura 3: Decomposição funcional nível 0 - Diagrama Físico



- Um conjunto de botões, por meio dos quais o usuário irá interagir com os jogos e os menus de opções, que representamos pelo bloco 'Controle'. O botões poderão estar ligados nos pinos GPIO ou à uma das entradas USB da placa Raspberry Pi, que detectará quando o usuário pressiona cada botão;
- Uma bateria recarregável que alimentará o console através da entrada miniUSB;
- Um cartão microSD que guardará o sistema operacional e funcionará como memória;



No diagrama observa-se três grandes blocos. Sendo o principal corresponde ao SneSPY do primeiro jogador, onde será realizado, em especial, o processamento do jogo e a captura do áudio e vídeo, de forma a permitir a reprodução destes em uma tela. O segundo bloco, 'SneSPY 2' representa o funcionamento do console no terceiro modo. Abaixo vemos um bloco 'Outros jogadores', representando a possibilidade de que mais de dois jogadores estejam na mesma rede. O último grande bloco é o 'Receptor do streaming A/V sem fio', que aqui representa um SneSPY no quarto modo, mas que genericamente também pode ser um computador, um celular ou uma *smart TV*.

O bloco 'Controle usuário' merece atenção especial. Esse bloco iniciará antes da execução dos jogos e definirá o modo de funcionamento do console. Os modos podem ser jogo em uma pessoa, jogo multiplayer, buscar uma partida ou receptor de streaming A/V. No primeiro modo, o 'Servidor *input*' apenas irá detectar os dados do controle do próprio console e o emulador realizará partidas com apenas um jogador. No segundo modo, o console irá esperar a conexão de outros jogadores para a realização de partidas *multiplayer*. No terceiro modo, o console não irá executar jogos e tentará buscar na rede um console que esteja esperando a conexão de outros jogadores para uma partida *multiplayer*. Caso o usuário deseje, ele também pode receber os dados do áudio e vídeo do console principal para a reprodução do jogo em uma outra tela. No último modo, o console apenas irá receber dados via Wi-Fi de áudio e vídeo a serem reproduzidos.

O fluxo inicia-se com a interação física do usuário com o controle. Essa interação será convertida em sinais elétricos que serão detectados por um bloco de software chamado 'Servidor *input*'. Esse bloco será responsável por detectar os comando emitidos pelo jogador 1 e, se estiver operando no segundo modo, os comandos dados por outros jogadores que estejam conectados na mesma rede e enviá-los ao emulador 'Snes9x' que estará executando o jogo. O bloco 'Captura A/V' captura os dados de áudio e vídeo gerados pelo emulador e formata-os de modo adequado para o streaming. O 'Servidor *streaming*' é responsável por enviar esses dados via Wi-Fi ou via cabo.

O console no terceiro modo e no quarto modo apresentam o bloco 'Cliente *streaming*' que recebe os dados A/V via Wi-Fi possibilitando a sua reprodução local. É importante salientar que um console no terceiro modo pode ainda transmitir os dados a um console no quarto modo de forma a reproduzir o áudio e o vídeo sem a utilização de cabos.

O último bloco a ser analisado é o bloco 'Snes9x falso'. Este bloco irá transmitir via Wi-Fi os comandos dados pelo jogador local ao 'Servidor *input*' do console principal. O objetivo deste bloco é que localmente ele seja visto pelos demais blocos como um emulador real, que recebe os dados do controle e supostamente estaria atuando nos jogos.

Podemos perceber que por ser um sistema coeso, obteremos uma maior facilidade na depuração em busca de erros, conseguindo identificar qual a etapa do projeto que está apresentando defeito. A divisão por funcionalidades diferentes também permite que haja

## 7 Gerenciamento do projeto

### 7.1 EAP

Figura 5: EAP



### 7.2 Cronograma

Dividimos o projeto em quatro atividades independentes. A primeira corresponde ao ‘*Streaming A/V*’ e corresponde a enviar e reproduzir com sucesso o áudio e vídeo de um jogo em outro dispositivo. Essa atividade depende da elaboração de dois blocos de software, o ‘*Cliente streaming*’ e o ‘*Servidor streaming*’. Este último bloco depende da elaboração do bloco ‘*Captura A/V*’ para que ele possua dados a serem transmitidos.

A segunda corresponde à elaboração do bloco ‘*Controle usuário*’ que definirá o modo de operação do console.

A terceira corresponde ao funcionamento básico e tradicional de um console. Conseguir emular um jogo com apenas um jogador, e transmitir seus dados a uma TV via cabo. Esse bloco depende da instalação do projeto RetroPie, que por sua vez depende da instalação do sistema operacional no Raspberry Pi.

A última atividade corresponde a conseguir emular jogos em modo *multiplayer* via conexão Wi-Fi. Essa atividade depende da elaboração dos blocos ‘*Servidor input*’ e

Figura 6: Cronograma

Nº	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	X										
2	X	X									
3		X	X								
4		X	X	X							
5		X	X								
6				X							
7				X	X						
8			X	X	X						
9		X		X	X	X					
10			X	X	X						
11					X	X					
12					X	X	X				
13				X	X	X					
14				X	X	X	X				
15							X	X	X		
16							X	X	X	X	
17	X	X	X	X	X	X	X	X	X	X	X
18		X	X						X	X	X

## 8 Prova de conceito

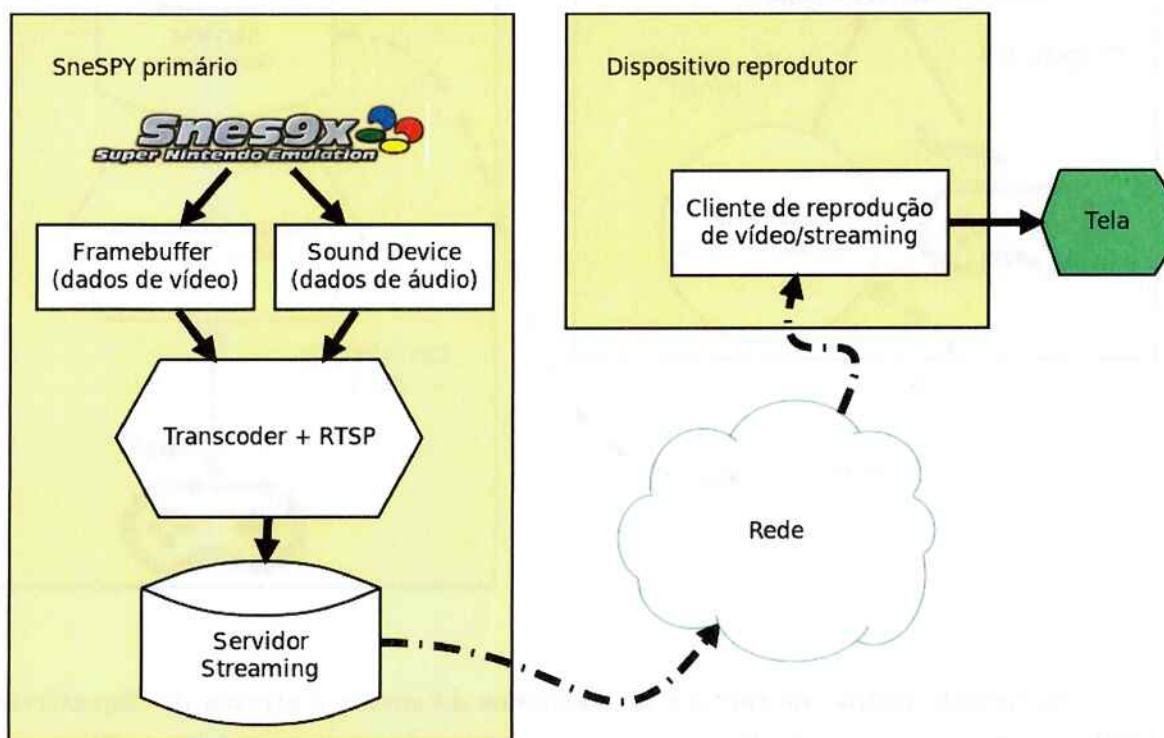
A prova de conceito de nosso projeto envolve provar o funcionamento das atividades que definimos como diferenciais em relação aos produtos no mercado e de maior peso na nossa árvore de decisão, que são o *streaming* de dados e o *multiplayer* em rede.

As outras atividades como a emulação dos jogos e a reprodução de áudio e do vídeo via cabo HDMI ou vídeo composto já são padrão de vários consoles e não é necessário provar que podem ser realizadas, já que estamos utilizando softwares e hardwares bem conhecidos.

### 8.1 Método de *streaming*

O objetivo do *streaming* é transmitir os dados de áudio e vídeo dos jogos aos dispositivos reprodutores, através da rede. Para isso, optamos por disponibilizar esses dados em um servidor de *streaming* dentro do SneSPY, de forma que clientes possam acessá-lo para a receber os dados. A Figura 7 mostra o diagrama que representa a nossa solução.

Figura 7: Diagrama do *streaming*



Cada emulador gera e armazena os dados de áudio e vídeo dos jogos de formas

---

forma que se realize uma conexão USB virtual entre o controle no console secundário e o emulador no console primário.

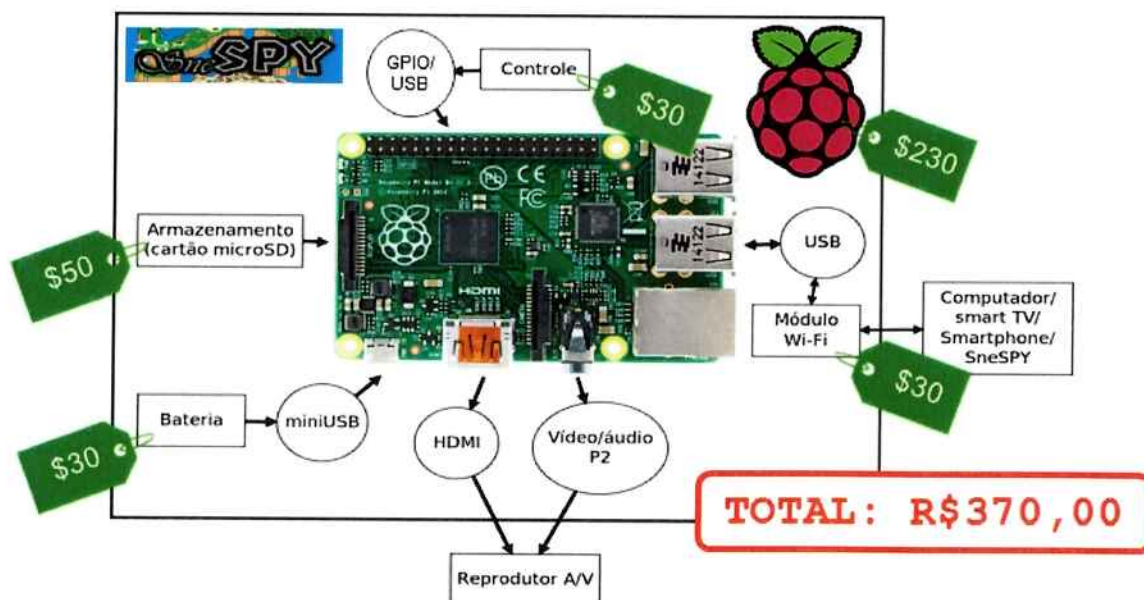
## 9 Análise de custos

Os custos principais de nosso projeto estão relacionados às compras do *hardware* necessário para a montagem do console:

- Raspberry Pi B+ – R\$ 230,00
- Controle com conexão USB – R\$ 30,00
- Cartão microSD 8GB – R\$ 50,00
- Módulo Wi-Fi USB – R\$ 30,00
- Bateria Li-Ion + *Hardware* de controle de recarga (power bank) – R\$ 30,00
- Total – R\$ 370,00

A Figura 9 ilustra a divisão de custos baseada no diagrama físico montado anteriormente.

Figura 9: Preços



## 10 Análise de riscos

A análise de risco do nosso projeto pode ser feita pela análise de três pontos principais: *hardwares* e *softwares* envolvidos, métodos e modelo de negócios.

### 10.1 *Hardwares e softwares*

Nosso projeto envolve as seguintes tecnologias de *hardware*: Raspberry Pi B+, conexão USB, módulo Wi-Fi USB, conexões HDMI e P2 e cartão microSD. E também as seguintes tecnologias de *software*: OS Raspbian, snes9x, VLC media player. Todas estas tecnologias são maduras, testadas e de ampla utilização em projetos pessoais, industriais ou no desenvolvimento tecnológico (PAUL, 2015). Diante da grande quantidade de informações, análises e manuais presentes na literatura e na internet, capazes de nos fornecer as bases para a implementação das nossas soluções, inferimos que há risco muito baixo ou nulo de nosso projeto falhar devido a alguma dessas tecnologias ser incapaz de atender às nossas necessidades.

### 10.2 Métodos

Os principais métodos envolvidos no nosso projeto são os pontos chaves que formam a nossa prova de conceito. Apesar da integração, feita em nosso console, dos métodos de *streaming* de áudio e vídeo e compartilhamento de USB não ser tão comum e demandar análise e pesquisa para conseguirmos implementá-la, estes métodos também já são conhecidos e usuais em outras atividades. Existem diversos sites dedicados exclusivamente ao *streaming*, como o YouTube e o Twitch (FIGUEIRA, 2015) e também já existem alguns projetos dedicados ao desenvolvimento de um sistema genérico de compartilhamento de USB através da rede, como o projeto USB/IP. Portanto, concluímos que o risco nesta área é médio-baixo.

### 10.3 Modelo de negócios

Nesta área se concentram os maiores riscos de nosso projeto. O primeiro ponto a se analisar é que nós não possuímos os direitos sobre nenhum jogo de SNES da Nintendo e portanto não podemos vender ou distribuir os jogos que são o foco de nosso projeto. Apesar de ser possível que o usuário compre os jogos em lojas oficiais, como eles são muito antigos, as alternativas de acesso são muito limitadas, restringindo assim o uso de nosso produto. Uma alternativa é o uso de nosso console com jogos de fabricação própria ou de

# 11 Implementação do protótipo

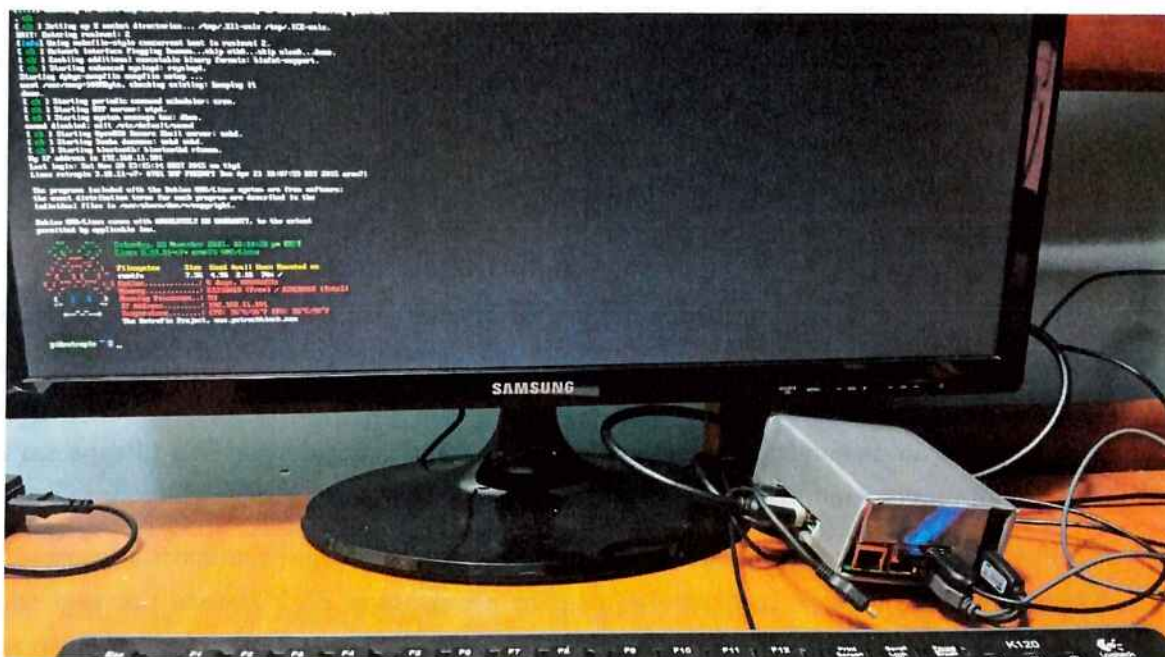
Os testes da implementação do projeto são divididos em testes específicos de cada funcionalidade implementada. A divisão foi feita em instalação e configuração do sistema operacional, conexão wi-fi, *streaming*, compartilhamento remoto de USB, uso de bateria e design do controle.

## 11.1 Instalação e Configuração do Sistema Operacional

O RetroPie, sistema operacional escolhido para o projeto, possui instruções de instalação e configuração dos controles e emuladores descritas na página oficial dos desenvolvedores na rede. Nesta fase, realizamos a adequação do sistema às nossas necessidades e também colocamos os jogos de SNES que possuíamos no cartão de memória do dispositivo.

Podemos ver na Figura 10 que a instalação do sistema operacional foi feita com sucesso e na Figura 11 a emulação da imagem do jogo Mighty Morphin Power Rangers: The Movie que um dos integrantes possui.

Figura 10: Sistema operacional RetroPie funcionando.





o emulador em um terminal do X11 e realizar a captura da janela do emulador apenas e não do framebuffer inteiro. Utilizando assim a biblioteca do *gststreamer*, conseguimos fazer o encode desta captura no padrão de compressão H264 e o transmitimos através de uma conexão UDP.

Após um problema em que o vídeo transmitido apresentava estranhas faixas coloridas na borda inferior da tela, descobrimos que o *encode* em *hardware* do Raspberry Pi só consegue gerar vídeos a algumas resoluções específicas, múltiplas de 240x180. Como os jogos de Super Nintendo utilizados para testes apresentam resolução padrão de 320x240, fizemos o *encode* na resolução de 480x360 para que não houvesse perda de pixels da imagem.

Podemos ver nas imagens 12 e 13, o *streaming* do vídeo do SneSPY para um notebook. No monitor com resolução de 1920x1080 podemos ver o jogo sendo reproduzido localmente à resolução de 320x240 e seu *streaming* a 480x360 no notebook. Todas as transmissões estão sendo realizadas através de conexão Wi-Fi por um roteador TP-LINK.

Figura 12: *Streaming* e reprodução local da imagem do jogo



Para o *streaming* do áudio, realizamos a montagem de um driver ALSA em *loopback* que recebia o áudio do emulador e o redirecionava a um falso driver de captura de áudio. Realizamos a leitura deste áudio normalmente, como se fosse o áudio captado por um microfone, passando por um encode em formato L16 (PCM linear de 16 bits) e transmitindo também por uma conexão UDP.

Ao executar ambos os *streamings* simultaneamente, obtemos finalmente a transmissão via rede da imagem e áudio do jogo.

US\$ 5,00 em lojas nos EUA, os custos de importação ou mesmo os preços praticados em lojas locais tornavam esta opção menos interessante.

Buscando melhores preços e implementações conhecidas e confiáveis, optamos pelo uso de um *power bank* comercial de 2800 mAh. Com ele, conseguimos uma autonomia de aproximadamente 5 horas para o uso do produto com *streaming* e emulação de jogos, atendendo totalmente nosso requisito de 4 horas de autonomia.

Testamos, com a montagem da Figura 15, o consumo de bateria em dois modos. No modo 1 apenas ligamos o sistema, sem executar nenhum programa. Já no modo 2 estavam ligados os programas emuladores e uma aplicação básica de *streaming*.

Figura 14: Consumo de bateria pelo SneSPY



Figura 15: Montagem do console com *power bank*, controle, e módulo Wi-Fi



## 11.6 Design do controle

O design do controle surgiu como um item de primeira expansão do projeto, tendo em vista que nosso principal desafio era desenvolver uma boa técnica que garantisse a transmissão do *streaming* pela conexão sem fio.

## 12 Resultados pessoais

Apesar de iniciarmos este projeto com uma boa base de programação em C e C++, possuíamos pouco ou nenhum conhecimento sobre o sistemas Linux. Desde o uso básico de comandos do Linux, até sobre o sistema de distribuição de pacotes e a compilação do kernel.

Muito do nosso projeto ocorreu no desenvolvimento de habilidades pessoais que nos permitissem integrar todos os componentes que discutimos nos capítulos anteriores.

Para começar, precisamos aprender sobre a compilação do kernel, das bibliotecas e dos programas essenciais ao desenvolvimento deste projeto e realizamos diversos tutorias sobre como criar programas utilizando as funções disponibilizadas por estas bibliotecas. E então descobrimos como compilar os programas utilizando parâmetros diferentes e como realizar o link com as bibliotecas.

Então, para conseguir configurar a conexão Wi-Fi, precisamos entender mais sobre redes e como configurar o *wpa\_supplicant* para realizar o login automaticamente nas redes com segurança WPA.

Precisamos novamente realizar um trabalho de instalação dos programas e drivers adequados para a versão correta do kernel utilizado para conseguir realizar o compartilhamento do USB, além de precisar entender melhor como o sistema reconhece e interpreta uma conexão USB.

Também precisamos entender melhor sobre o funcionamento e o formato do frame-buffer, do X11 e de dispositivos ALSA para poder fazer a captura do áudio e do vídeo dos jogos, além de aprender um pouco sobre formatos de codificação de áudio e de vídeo.

Por fim, a modelagem também apresentou outro problema pois foi necessário encontrar um bom *software* gratuito e uma máquina com capacidade de processamento suficiente para conseguir executar o programa, além de aprender a utilizar o programa para modelar o controle.

## 13 Conclusão

Podemos concluir que o projeto atingiu os requisitos necessários definidos anteriormente.

Através da conexão Wi-Fi, atingimos um raio de operação de 6 metros e conseguimos realizar o *streaming* para outras máquinas. Utilizando uma conexão UDP, também conseguimos um atraso aceitável que não impossibilita o usuário de jogar.

O emulador também se mostrou capaz de emular todos os jogos que possuíamos e muitos outros segundo especificado no site dos desenvolvedores.

Com o compartilhamento de dispositivos pela rede, pudemos formar redes *multi-player* com pelo menos dois jogadores, seja com apenas jogadores locais ou remotos, ou uma combinação das duas possibilidades.

E com o *power bank* capaz de alimentar o console por 5 horas, cumprimos todos os requisitos de engenharia, e conseqüentemente os de marketing.

### 13.1 Possibilidades de expansão

Acreditamos que o projeto ainda possua grande capacidade de expansão, mas que, devido à grande quantidade de assuntos que precisamos aprender durante este ano para realizá-lo, não pudemos implementar.

Nosso projeto, apesar de funcional, faltou algumas implementações de interface para o usuário que facilitassem a seleção de jogos e a conexão com outras máquinas.

Também não foi implementado um método de criação de redes remotas que pudesse conectar dois usuários que estivessem conectados a redes diferentes.

Observando o desempenho do Raspberry Pi, pudemos notar que ele estava em seu limite na configuração atual. Para conseguirmos executar emuladores ou jogos mais pesados ou talvez até conseguir um *streaming* em qualidade HD, seria interessante buscar máquinas mais potentes.

Outro ponto interessante seria desenvolver um design próprio completo, desde o estudo de ergonomia até a modelagem em 3D.

## Referências

- ALECRIM, E. *O que é Wi-Fi (IEEE 802.11)?*. 2015. Disponível em: <<http://www.infowester.com/wifi.php>>. Citado na página 21.
- ARELLANO, M. A. Preservação de documentos digitais. *Ci. Inf.*, IBBD, v. 33, n. 2, p. 15–27, 2004. Citado na página 16.
- CASANOVA, P. C. *Retrospectiva dos consoles portáteis*. 2015. Disponível em: <<http://www.cogumelando.com.br/episodio-208-retrospectiva-dos-portateis/>>. Citado na página 15.
- FIGUEIRA, M. *E3 2015 was watched by over 21 million people on Twitch*. 2015. Disponível em: <<http://www.lazygamer.net/e3-2015/e3-2015-was-watched-by-over-21-million-people-on-twitch/>>. Citado na página 43.
- GUTTENBRUNNER, M. et al. Evaluating strategies for the preservation of console video games. *Proceedings of the Fifth International Conference on Preservation of Digital Objects*, iPress 2008, p. 115–121, 2008. Citado na página 16.
- HINDY, J. *12 best emulators for Android*. 2015. Disponível em: <<http://www.androidauthority.com/best-emulators-for-android-315958/>>. Citado na página 16.
- MCWHERTOR, M. *Final Fantasy 7 remake coming to PlayStation 4*. 2015. Disponível em: <<http://www.polygon.com/2015/6/15/8785689/final-fantasy-7-remake-ps4-square-enix>>. Citado na página 44.
- PAUL, I. *10 insanely innovative, incredibly cool Raspberry Pi projects*. 2015. Disponível em: <<http://www.pcworld.com/article/2895874/10-insanely-innovative-incredibly-cool-raspberry-pi-projects.html>>. Citado na página 43.
- PETROCKBLOG. *What is the RetroPie Project?*. 2015. Disponível em: <<http://blog.petrockblock.com/retropie/>>. Citado na página 21.
- THE RASPBERRY PI FOUNDATION. *What is a Raspberry Pi?*. 2015. Disponível em: <<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>>. Citado na página 20.
- UOL JOGOS. *Clássico 'Final Fantasy Tactics' ganha versão mobile para Android*. 2015. Disponível em: <<http://jogos.uol.com.br/ultimas-noticias/2015/06/05/classico-final-fantasy-tactics-ganha-versao-mobile-para-android.htm>>. Citado na página 44.
- WINGET, M. A. Videogame preservation and massively multiplayer online role-playing games: A review of the literature. *J. Am. Soc. Inf. Sci. Technol.*, v. 62, n. 10, p. 1869–1883, 2011. Citado na página 16.
-