
Sistema de controle de apresentação por meio de Raspberry Pi

Cássio Moreira Silva



UFU
UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Monte Carmelo - MG
2019

Cássio Moreira Silva

**Sistema de controle de apresentação por meio
de Raspberry Pi**

Trabalho de Conclusão de Curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau de
Bacharel em Sistemas de Informação.

Área de concentração: Ciência da Computação

Orientador: Thiago Pirola Ribeiro

Monte Carmelo - MG

2019

Dedico este trabalho primeiramente a Deus, por ser essencial em minha vida, autor de meu destino, meu guia, socorro presente na hora da angústia, ao meu pai José Moreira Cruvinel, minha mãe Rosimar de Fátima Silva Cruvinel e ao meu irmão Igor Silva Cruvinel. Dedico ao meu professor Thiago Pirola Ribeiro e aos meus colegas que me apoiaram na conclusão da monografia.

Agradecimentos

A Deus por ter me dado saúde e força para superar as dificuldades.

A esta universidade, seu corpo docente, direção e administração que deram esta oportunidade que vislumbro hoje.

Ao meu orientador Thiago Pirola Ribeiro, pelo suporte no concorrido tempo que lhe coube, pelas suas correções, incentivos e ideias.

Aos meus pais, pelo amor, incentivo e apoio incondicional.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

“Não confunda derrotas com fracasso nem vitórias com sucesso. Na vida de um campeão sempre haverá algumas derrotas, assim como na vida de um perdedor sempre haverá vitórias. A diferença é que, enquanto os campeões crescem nas derrotas, os perdedores se acomodam nas vitórias.”

(Roberto Shinyashiki)

Resumo

Por mais que as tecnologias como celular, tablets, desktops chegaram para trazer benefícios incontestáveis como praticidade, também é um fato de que torna-se um incômodo quando utilizada no momento incorreto ou de mal uso em ambientes educacionais. Pensando nesse problema, o trabalho implementa parte de uma solução para que a mesma tecnologia que possa vir atrapalhar o ensino, possa ajudar aos alunos e tornar o cotidiano do docente mais prático. A solução foi desenvolver uma aplicação web rodando em um Raspberry Pi que possibilite o professor ter mais autonomia, obtendo seus materiais de ensino em um só lugar e controlando a apresentação com qualquer dispositivo eletrônico em mãos.

Palavras-chave: Sistema de Controle, Sala Inteligente, Raspberry Pi, Comunicação.

Lista de ilustrações

Figura 1 – Esboço da ideia geral do projeto “Sala de Aula Inteligente com o uso de Raspberry Pi”	12
Figura 2 – Dispositivo Arduino com a aplicação Arduino Mega Chess	15
Figura 3 – Dispositivo Arduino com a aplicação Luzes de Led RGB nas escadas	16
Figura 4 – Raspberry Pi 3 modelo B	17
Figura 5 – Exemplo de Socket (Kurose; Ross, 2007)	20
Figura 6 – Diagrama de Atividades Gerais explicitando, parcialmente, as funções do projeto “Sala de Aula Inteligente com o uso de Raspberry Pi”. Destacado em azul as funções que serão implementadas neste trabalho.	23
Figura 7 – Caso de Uso exemplificando os processos realizados pelo Raspberry Pi.	24
Figura 8 – Diagrama de Atividades, traçando os possíveis trajetórias da usabilidade dos usuários.	25
Figura 9 – Tela de Abertura do Trabalho. Imagem modificada de (Freitas, 2017) - slide 4.	26
Figura 10 – Visão geral das atividades e processos do trabalho.	27
Figura 11 – Tela de login para acesso de alunos e professores.	29
Figura 12 – Caso de Uso dos atores (professor e aluno).	29
Figura 13 – Exemplo da tela com a listagem de pastas e arquivos.	30
Figura 14 – Tela contendo o teclado para manipulação de arquivos.	31
Figura 15 – Diagrama de Estado - Fluxo entre os Scripts	33

Lista de siglas

CSS Folha de Estilo em Cascatas - *Cascading Style Sheets*

DSI Interface Serial do Monitor - *Display Serial Interface*

DLL Biblioteca de Link Dinâmico - *Dynamic Link Library*

GPIO Entrada/saída para Fins Gerais - *General Purpose Input/Output*

HDMI Interface Multimídia de Alta Definição - *High-Definition Multimedia Interface*

HTTP Protocolo de Transferência de Hipertexto - *Hypertext Transfer Protocol*

IA Inteligência Artificial - *Artificial Intelligence*

IoT Internet das Coisas - *Internet of Things*

IP Protocolo de Internet- *Internet Protocol*

JPEG Grupo de Especialistas em Fotografia- *Joint Photographic Experts Group*

JS Arquivo JavaScript - *JavaScript File*

LAN Rede de Área Local - *Local Area Network*

LED Diodo Emissor de Luz - *Light Emitting Diode*

Mac OS Sistema Operacional Macintosh - *Macintosh Operating System*

ODF Formato de Documento de Fórmula - *Open Document Formula*

ODP Formato de Documento de Apresentação - *Open Document Presentation*

ODS Formato de Documento de Planilha - *Open Document Spreadsheet*

ODT Formato de Documento de Texto - *Open Document Text Document*

PDF Formato de Documento Portátil - *Portable Document Format*

PHP Página Pessoal - *Personal Home Page*

RGB Vermelho-Verde-Azul - *Red-Green-Blue*

SGBD Sistema de Gerenciamento de Banco de Dados - *Datadase Management System*

SO Sistema Operacional - *Operational System*

TIC Tecnologias da Informação e Comunicação - *Information and Communication Technology*

TXT Arquivo de texto simples - *Plain Text File*

USB Porta Universal - *Universal Serial Bus*

Wi-Fi Fidelidade sem fio - *Wireless Fidelity*

Sumário

1	INTRODUÇÃO	11
1.1	Objetivos	12
1.2	Organização da Monografia	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Hardware	14
2.1.1	Arduíno	14
2.1.2	Raspberry Pi	16
2.2	Softwares	18
2.2.1	Sistema Operacional	18
2.2.2	Linguagens para Programação	18
2.3	Comunicação	19
2.3.1	Script	19
2.3.2	Shell Script	20
2.3.3	Socket	20
2.3.4	Xdotool	20
3	DESENVOLVIMENTO E ANÁLISES	22
3.1	Ideia Geral do Projeto	22
3.2	Instalações e requisitos	23
3.3	Fluxo das Atividades Desenvolvidas	24
3.3.1	Iniciando o Sistema	25
3.3.2	Acesso aos Arquivos	27
3.4	Desenvolvimento	28
4	CONCLUSÃO	35
4.1	Trabalhos Futuros	35
4.2	Contribuições em Produção Bibliográfica	36

REFERÊNCIAS	37
-------------------	----

Introdução

Com a evolução computacional em diversas áreas, o cotidiano tem ficado mais tecnológico com o uso da Robótica, Inteligência Artificial (IA), Internet das Coisas (IoT) e Tecnologias da Informação e Comunicação (TIC) e, o contexto educacional vem necessitando de mudanças para acompanhar esse processo evolutivo, com o intuito de ter mais comodidade, praticidade e formas mais didáticas de ministrar aulas, assim como o cenário do mundo contemporâneo exige (Cavalcante et al., 2014).

Ao longo de vários anos, as instituições de ensino enfrentaram diversos problemas, por exemplo, políticos, culturais, psicológicos e assim por diante. Mas, atualmente, o que mais chama a atenção por ser um problema dentro das salas de aula, são as novas tecnologias, que estão presentes na rotina das pessoas com infinitas utilidades e, na maioria das vezes, tirando o foco dos estudos.

Inibir o uso de dispositivos móveis não é interessante, mas controlar o modo como os alunos usam a tecnologia pode ser uma alternativa viável. Com isso, porque não empregar essa terrível nova tecnologia a favor da educação e aprendizado tornando-a uma ótima ferramenta de aprendizado (Sutilo, 2017; Prado et al., 2017). Os alunos poderiam usar os eletrônicos para aprender em vez de outros conteúdos que fogem da sala de aula.

Na contemporaneidade, professores estão sempre buscando novas formas de estimular alunos à estudar e pensar. Nesse sentido, professores levam a tecnologia para as escolas, criando uma cultura de tecnologia e inovação (Meirinhos; Osório, 2015).

O número de projetos tecnológicos voltados para a área da educação tendem a crescer cada dia mais, visando ampliar as possibilidades de ministrar aula e a habilidade do professor ensinar e o aluno aprender. Desse modo, este trabalho tem como objetivo auxiliar os professores a ter mais comodidade, trazendo os conceitos das Tecnologias da Informação e Comunicação (TIC) para as salas de aula. Além disso, os alunos também poderão ser afetados com essas mudanças, uma vez que a tecnologia da informação aplicada a aprendizado traz bons resultados, como pode ser visto no trabalho de Oliveira (2015).

Uma forma de ter mais produtividade nas salas de aula é tornar as atividades já desempenhadas pelo professor como buscar, acessar e apresentar o material de aula, de

forma mais práticas, simples e rápidas. E isso pode ser feito utilizando a tecnologia para automatizar e facilitar o modo como o professor ministra a aula, tornando a tecnologia que antes era uma “distração” em um auxílio necessário.

Nesse sentido, o projeto “Sala de Aula Inteligente com o uso de Raspberry Pi”, que está em desenvolvimento, pretende utilizar os dispositivos dos próprios usuários (celulares e/ou *tablets*) para criar a interação entre o professor, os alunos e o material didático. Em linhas gerais (Figura 1), o projeto utiliza um sistema cliente/servidor conectado a uma rede LAN e/ou Internet, disponibilizando para seus clientes (professor e aluno) por meio de wireless, o acesso à página de login e o material didático necessário para a aula que estaria sendo ministrada, além da projeção do material consequentemente. Este projeto, por ter diversos componentes foi subdividido em projetos menores.

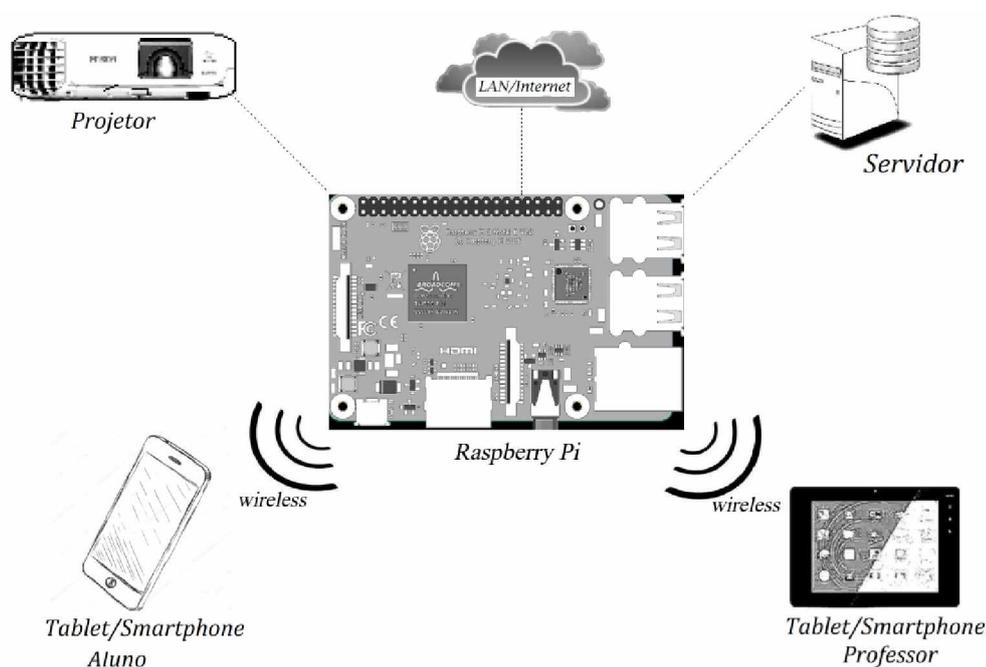


Figura 1 – Esboço da ideia geral do projeto “Sala de Aula Inteligente com o uso de Raspberry Pi”

O escopo deste trabalho será a implementação e testes da parte da interação do professor com o sistema e a parte de projeção do material resultando, principalmente, em um sistema para o gerenciamento da apresentação utilizando o Raspberry Pi.

1.1 Objetivos

O objetivo geral da monografia proposta é desenvolver uma parte de projeto maior. Esse trabalho desenvolverá o módulo específico de seleção, projeção e controle da apresentação, baseado no usuário com perfil professor referente ao projeto “Sala de Aula Inteligente com o uso de Raspberry Pi”.

Os objetivos específicos desta monografia são:

- ❑ Desenvolver um sistema de acesso ao material didático por meio de dispositivos móveis;
- ❑ Realizar a seleção e o controle da projeção do material didático por meio de dispositivo móvel;
- ❑ Desenvolver meios para o Raspberry Pi projetar os materiais selecionados remotamente.

1.2 Organização da Monografia

No Capítulo 2 abordam-se a fundamentação teórica para auxiliar na compreensão do trabalho. No Capítulo 3 é apresentado o desenvolvimento e suas análises. Finalizando com um capítulo contendo as conclusões elencando as principais contribuições obtidas com o trabalho juntamente com as publicações geradas e trabalhos futuros.

Fundamentação Teórica

Existem diversas tecnologias, técnicas e soluções para atividades vivenciadas no dia a dia, seja para estudar, trabalhar ou para entretenimento. Neste capítulo será apresentada a fundamentação teórica do trabalho, descrevendo técnicas e tecnologias utilizadas no desenvolvimento do trabalho, sendo distribuídas em:

- ❑ Hardware;
- ❑ Software;
- ❑ Comunicação.

2.1 Hardware

O projeto necessitará de uma plataforma (equipamento) que possa processar as requisições feitas pelos dispositivos móveis, além de efetuar análises e ações. O equipamento deverá ficar localizado dentro de uma sala de aula e, deve permitir fácil mobilidade do equipamento e, com isso, deseja-se que o equipamento tenha dimensões reduzidas. Pensando nisso, serão analisadas duas tecnologias que são amplamente utilizadas: o Arduíno (2.1.1) e o Raspberry Pi (2.1.2).

2.1.1 Arduíno

Uma placa Arduíno é uma plataforma eletrônica capaz de ler entradas, como luz em um sensor, câmera, e outras inúmeras atividades como sensores e atuadores. É uma plataforma de código aberto baseada em hardware e software. Tem a função de ser o cérebro de pequenos projetos até instrumentos científicos complexos. Algumas das áreas que mais tem chamado atenção em relação a sua utilização é no âmbito de aplicações para IoT, *wearable*, impressão em 3D e ambientes incorporados. A plataforma é eventualmente adaptativa para cada particularidade.

Quanto as linguagens e o Sistema Operacional (SO) utilizado no Arduino, ele possui código aberto, inclusive o software. Executado no Windows, Mac OS X e Linux. O ambiente Arduino IDE pode ser usado em qualquer placa Arduino e foi desenvolvido na linguagem Java.

A plataforma eletrônica é fácil de ser instalada e manuseada, tendo uma flexibilidade para usuários iniciantes até avançados. Utilizado por milhares de professores e alunos, para fins educativos e acadêmicos; Designers e arquitetos com intuito de criar protótipos interativos; músicos e artistas utilizam para instalações de palco e experimentos com instrumentos musicais, etc. (Fonseca, 2017).

Um exemplo de uma aplicação utilizando Arduino é o *Arduino Mega Chess* (Figura 2). O programa contém 2.000 linhas de código com tela sensível ao toque e emite som, sendo o mesmo capaz de reproduzir um jogo de xadrez (Arduino Project HUB, 2017).



Figura 2 – Dispositivo Arduino com a aplicação Arduino Mega Chess

Fonte: <https://hackster.imgix.net/uploads/attachments/404757/dsc00470_Khc0pueDH2.jpg>

Outro exemplo é o luzes de LED RGB de escada automatizada (Figura 3) que por meio de um sensor de movimento no início e no final da escada, as luzes se acendem quando detecta a presença de algo ou alguém. As luzes se alternam de cor, sendo tudo controlado pelo Arduino (Arduino Project HUBs, 2017).



Figura 3 – Dispositivo Arduíno com a aplicação Luzes de Led RGB nas escadas

Fonte: <https://hackster.imgix.net/uploads/attachments/405970/stairs_logo_BiHlwVDeBY.jpg>

2.1.2 Raspberry Pi

A fundação Raspberry Pi é uma instituição de caridade educacional com sede no Reino Unido. O Raspberry Pi é um pequeno dispositivo, um computador, capaz de inúmeras atividades e acessível para todas as idades explorarem.

De acordo com a organização Raspberry Pi, não há nada que o computador de mesa faça que o Raspberry Pi não consiga realizar (Raspberrypi, 2012). Portanto, esse pequeno dispositivo pode ser utilizado para projetos relativamente simples e pequenos quanto para projetos mais complexos.

Assim como várias empresas, a fundação Raspberry Pi possui diversas plataformas (Raspberry PI, 2017): Framboesa Pi Zero, Framboesa Pi Zero W, Raspberry Pi 1 Modelo A+, Raspberry Pi 1 Modelo B+, Raspberry Pi 2 Modelo B, Raspberry Pi 3 Modelo B, Raspberry Pi 3 Modelo A+, Raspberry Pi 3 Modelo B+ e por fim o Raspberry Pi 4.

A Figura 4 ilustra o Raspberry Pi 3 Modelo B, que será utilizado neste trabalho, sendo suas características:

- ❑ Dimensões totais: 85.6mm x 56mm x 21mm.
- ❑ Processador Broadcom BCM2837 64bit Quad Core de 1,2GHZ e 1GB RAM.
- ❑ Bluetooth 4.1 e Wi-Fi integrado.

- ❑ 1 Slot para cartão MicroSD
- ❑ 1 Saída DSI
- ❑ Micro USB para entrada de energia. Fonte de alimentação comutada que pode lidar com até 2,5 Amperes.
- ❑ 1 Suporte para câmera.
- ❑ 1 Saída HDMI.
- ❑ 1 Saída de áudio, 3,5mm.
- ❑ 1 porta LAN.
- ❑ 4 portas USB 2.0.
- ❑ 40 pinos externos GPIO.



Figura 4 – Raspberry Pi 3 modelo B

Fonte: <<https://www.raspberrypi.org/app/uploads/2017/05/Raspberry-Pi-3-1-1619x1080.jpg>>

Para ser capaz de realizar tantas atividades, o Raspberry Pi possui um sistema operacional oficial baseado no Kernel do Linux, cujo o nome é Raspbian (Raspberrypi, 2012), porém existem outros sistemas operacionais que também podem ser instalados: Arco, Ubuntu, RISC OS e o Windows 10 para IoT CORE, sendo este último um sistema projetado para IoT. Algumas das linguagens suportadas são C/C++, Java, Python, dentre outras.

Por ser uma ferramenta completa, pode ser utilizada por qualquer pessoa independente do nível de conhecimento em hardware. O objetivo inicial da Raspberry.org é envolver as crianças com a emoção de programar, conseqüentemente, melhorando o raciocínio lógico e criando nas crianças as habilidades necessárias para o futuro. Atualmente, não somente as crianças se utilizam dessa plataforma, mas todos que se interessem em desenvolvimento e modificação de hardware. Existem muitos exemplos do uso do Raspberry Pi, como o *Countdown timer* - Mostra um cronômetro de contagem regressiva em um SATON para o Raspberry Pi (<<https://projects.raspberrypi.org/en/projects/countdown-timer>>). Diversos projetos podem ser encontrados na área de projetos do Raspberry.org (Raspberry Project, 2012).

Existem outras plataformas de prototipagem além do Raspberry Pi como o Arduíno, a BeagleBoard.org (2013), entre outras, porém foi escolhido o Raspberry Pi para este projeto por ser uma plataforma bem divulgada e com grande volume de material disponível para estudo, além de já dispor de integração e comunicação entre seus hardwares e periféricos, como por exemplo saída de vídeo HDMI, mouse, teclado, Wi-Fi, etc e, com isso, focou-se no desenvolvimento de software, não necessitando de implementações para que o hardware funcionasse.

2.2 Softwares

2.2.1 Sistema Operacional

Um Sistema Operacional (SO) é um conjunto de programas, fornecendo uma interface que permitem aos usuários interagirem com o computador. No mercado existe muitas opções de SO, por exemplo Windows, GNU/Linux, MacOS, Chrome OS sendo esses alguns dos SO para desktops/notebooks. Os dispositivos móveis possuem os seus próprios sistemas operacionais por exemplo Apple iOS, Windows Phone e o Android (Delfino, 2018).

Atualmente, a diversificação de plataformas e dispositivos eletrônicos existentes no mercado é enorme, sendo que alguns se destacam pela grande aceitação, por exemplo Android e iOS. O sistema Android, atualmente desenvolvido pela Google, é baseado no núcleo do Linux sendo o sistema operacional mais utilizado no mundo (Developers, 2019). O sistema iOS é um sistema operacional da Apple Inc[®], a mesma não permite que seu sistema seja utilizado em software de terceiros, inclusive de código fechado (Bohn Dieter e Aaron, 2013).

2.2.2 Linguagens para Programação

Linguagem de Programação é uma linguagem escrita de forma estruturada para especificar um conjunto de instruções e regras usadas para gerar programas (software). Um

software pode ser desenvolvido para rodar em um computador, dispositivo móvel ou em qualquer equipamento que permita sua execução. Há diversas linguagens de programação e algumas das mais utilizadas nos últimos anos são: JavaScript, Java, Python, C#, PHP, C++, C, TypeScript, Ruby, Swift (Forbellone; Eberspacher, 2000; Monteiro, 2018; Filipe, 2018).

Uma das linguagens de programação mais usadas nos últimos anos é o PHP, um acrônimo recursivo para *Hypertext Preprocessor* do inglês, criado em 1995 por Rasmus Lerdorf (Marra, 2009). Em dezembro de 2015 foi lançado o PHP versão 7. Os principais conjuntos de qualidades do PHP são velocidade e robustez, sendo uma linguagem estruturada, interpretada e orientada a objeto. Para seu funcionamento é necessário apenas um interpretador PHP, servidor Web (Apache, por exemplo) e um Sistema de Gerenciamento de Banco de Dados (SGBD) (MySQL, por exemplo).

Para o desenvolvimento do projeto será utilizado o PHP por ser multiplataforma, permitir implementações web rápidas, simples e eficientes, ser gratuita, código maduro, atualizações consistentes, grande banco de funções prontas e compatível com todos os dispositivos que possuam uma conexão com a internet (Nunes; Strazburger, 2009; PHP Net, 2018).

2.3 Comunicação

Em diversos momentos existe a necessidade de troca de informações entre diferentes dispositivos, estando ou não conectados na mesma rede. Entre dois sistemas em uma rede é possível a troca de informações utilizando-se diferentes linguagens e bibliotecas ou ainda por meio de soquetes (*Network Socket* do inglês). Atualmente tudo está conectado, seja por cabos ou via rede wireless, fazendo por meio dessas conexões uma rede de comunicação independente ou sincronizada, realizando diferentes atividades simultaneamente, inclusive a maioria dessas comunicações entre computadores é baseada em Protocolo de Internet (IP), portanto a maioria dos soquetes de rede são soquetes de Internet (Massetto, 2012).

2.3.1 Script

Um Script é um arquivo de texto que armazena uma sequência de comandos para execução de determinada tarefa (Jargas, 2004). Os Scripts são ótimos para automatizar tarefas repetitivas e/ou atividade de difícil execução, tanto pela complexidade ou mesmo excessiva quantidade de comandos. Por serem de tamanho reduzido e de fácil execução, são muito utilizados em servidores para automação de processo, mas podem ser usados para inúmeras finalidades em um SO. Existe uma vasta variedade de linguagens para Scripts, sendo as mais comuns: Shell Script, PHP, Perl, Python, JavaScript e muitas outras.

O agendamento de tarefas com *cron*, checagem do código HTTP, leitura do *status* do servidor, automatização de tarefas em jogos de *unit*, dentre outras atividades são alguns exemplos de uso de Scripts (Marinho, 2008).

2.3.2 Shell Script

Shell Script é uma linguagem de Script usada em vários sistemas operacionais, sendo uma linguagem interpretada, e um dos interpretadores mais conhecidos e utilizados nas distribuições GNU/Linux, tendo o nome de *Bash*.

Todo sistema *nix (Unix, Linux, Minix, etc.) são repletos de Scripts em Shell para a realização das mais diversas atividades administrativas e de manutenção do sistema. Os arquivos **.bat* do sistema operacional do Windows também são exemplos de Scripts.

A linguagem possui diversas funções, por exemplo condicionais, repetições, dentre outra menos comuns como itens de sistemas (*date*, *print*), etc. tornando-a uma ponte alternativa para desenvolvimento rápido.

2.3.3 Socket

O soquete está situado nos sistemas finais em um fluxo de comunicação bidirecional entre processos por meio de uma rede de computadores, sendo gerenciado pela camada de transporte, provendo a comunicação lógica entre emissor e receptor (Figura 5).

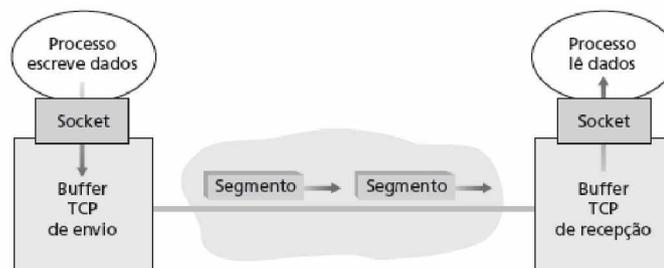


Figura 5 – Exemplo de Socket (Kurose; Ross, 2007)

O servidor soquete tem como ciclo de vida as operações: criação do soquete, associação do mesmo à um endereço local, aguardar por requisição do cliente, aceitar a conexão, ler requisição, opcionalmente enviar uma resposta e para concluir o ciclo fechar o soquete.

O cliente soquete possui um ciclo diferente: efetuar a criação do soquete, estabelecer a conexão, enviar uma requisição, opcionalmente aguardar uma resposta e por último fechar o soquete.

2.3.4 Xdotool

O movimento de mouse, o pressionamento de uma tecla ou até mesmo o redimensionamento das janelas do SO, podem ser algo simples de executar localmente, entretanto

realizar estas e outras atividades similares remotamente exigem muitas vezes um software intermediador, como o Xdotool para torna possível a comunicação esperada.

Xdotool é uma ferramenta que permite simular a entrada do teclado e mouse, mover e redimensionar janelas, ocultar e modificar propriedades da janela, etc. Isso é possível usando a extensão XTEST do X11 e outras funções do Xlib.

É uma ferramenta gratuita e possui uma comunidade muito ativa e sem fins lucrativos, a ferramenta esta disponível para sistemas Windows, Mac OS, mas é no Linux que a ferramenta atinge o auge de usuários presentes.

Além de fácil instalação, a ferramenta conta com uma grande gama de comandos para serem aplicados nas janelas especificadas, sendo possível identificar os programas e controlar por meios de comandos que simulam o mouse ou teclado (Sissel, 2015).

Esta ferramenta será utilizada no projeto para realizar o controle da apresentação remotamente.

Desenvolvimento e Análises

Neste capítulo serão apresentados o Desenvolvimento e Análises obtidas neste trabalho. Inicialmente será apresentada a metodologia de cada tela ou funcionalidade por meio de diagramas, pseudocódigos e esquemas.

Para o desenvolvimento do projeto foi utilizado um Raspberry Pi 3, conforme especificado na Seção 2.1.2 com o Sistema Operacional (SO) Raspbian versão 8 Stretch 4.14.

3.1 Ideia Geral do Projeto

O projeto “Sala de Aula Inteligente com o uso de Raspberry Pi” está em desenvolvimento e, conforme já ilustrado na Figura 1 e ilustrado no esboço na Figura 6, haverá uma interação entre os alunos e o professor por meio dos próprios dispositivos móveis realizando acesso no Raspberry Pi, sendo que este será a central de acesso e comunicação.

Conforme pode ser observado no diagrama da Figura 6, inúmeras atividades são propostas para compor o projeto, porém neste trabalho será implementado o módulo do professor. Esse módulo é composto pelos itens destacados em círculo azul na Figura 6: solicitar login, mostrar a listagem dos materiais (no dispositivo e na USB) e projetar. Os demais itens fazem parte do projeto maior.

A Figura 7 retrata o diagrama de caso de uso exclusivo do Raspberry Pi. Este fluxo é usado para representar as interações do ator, que por sua vez é representado por um usuário que utiliza o sistema, porém, no diagrama de caso de uso, o ator será o sistema. Com essa mudança, a intenção é demonstrar como é o uso do Raspberry Pi e seus periféricos e por fim, visualizar o funcionamento geral do atual projeto.

O ator é um servidor (Raspberry Pi) e fornecerá opção de login devido à integração de dados com um SGBD. Um contador de tempo mínimo é iniciado assim que o aluno ou professor se logar na página e, após um tempo de inatividade, o mesmo será deslogado. O servidor relaciona-se com os dispositivos conectado, sendo eles o do professor e do aluno, realizando leitura e executando processos de acordo com ações realizadas pelos mesmos. Outra forma de interação é com os repositórios locais e unidades de armazenamentos

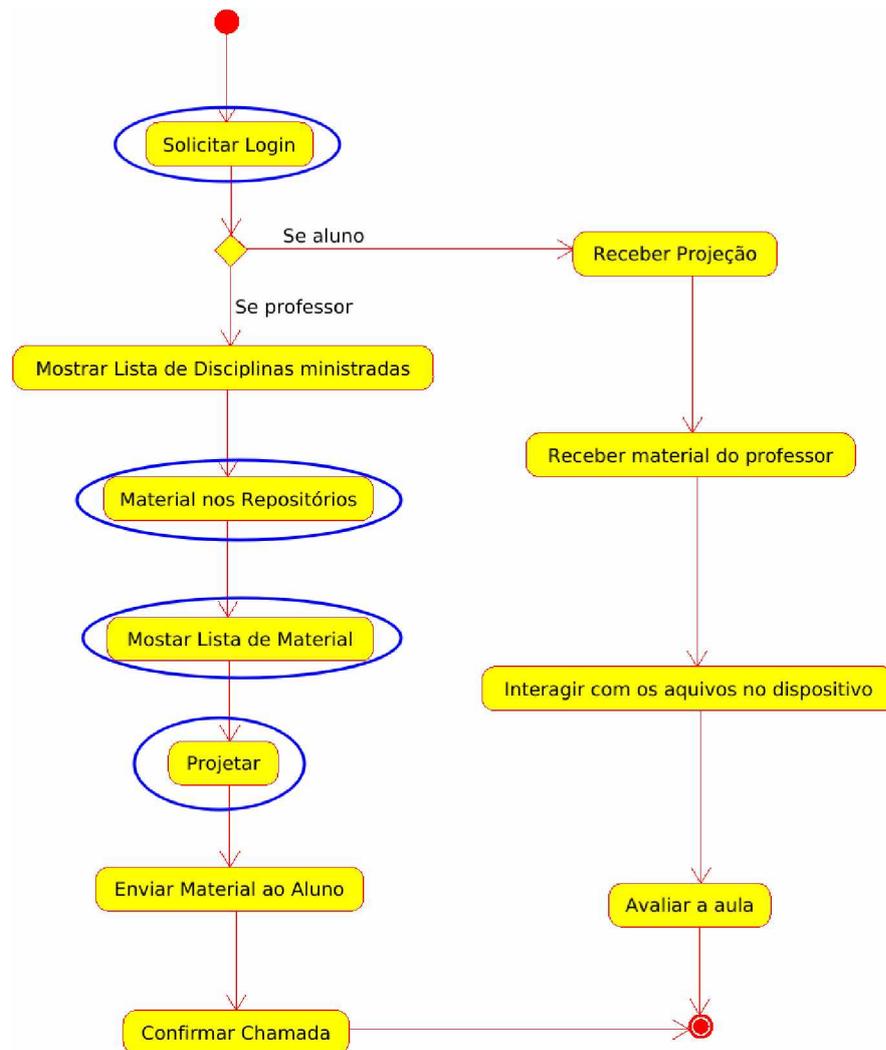


Figura 6 – Diagrama de Atividades Gerais explicitando, parcialmente, as funções do projeto “Sala de Aula Inteligente com o uso de Raspberry Pi”. Destacado em azul as funções que serão implementadas neste trabalho.

externos, sendo possível realizar leitura dos mesmos. E, por fim, realizar a projeção através da saída HDMI para o projeto de vídeo. O servidor possui acesso a rede LAN e acesso a web. Assim o Raspberry Pi desempenha o papel principal de gerenciar, realizar leitura e processar os dados para que os resultados sejam alcançados.

3.2 Instalações e requisitos

Conforme explicitado anteriormente e ilustrado na Figura 1, o desenvolvimento do trabalho teve como início a preparação do servidor (Raspberry Pi) por ser o cérebro do trabalho. Conforme já mencionado anteriormente, o Raspberry Pi possui seu próprio Sistema Operacional (SO) (Raspbian) a ser instalado. Assim que o ambiente estiver pronto e atualizado, faz-se necessário a instalação de alguns softwares, inserção de Scripts e configurações.

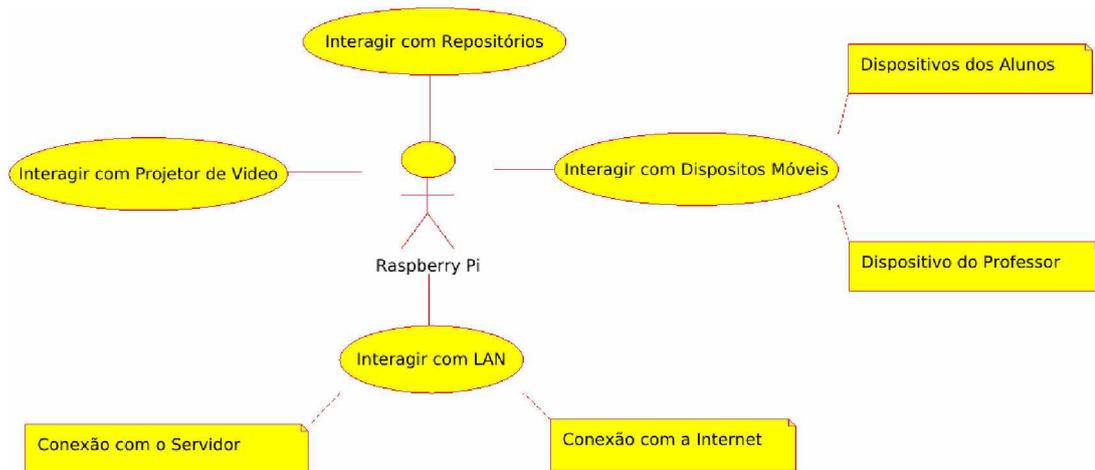


Figura 7 – Caso de Uso exemplificando os processos realizados pelo Raspberry Pi.

Para o funcionamento do projeto são necessárias algumas instalações de softwares: Servidor web Apache, PHP, o SGBD MySQL, serviços de roteamento wireless, software de visualização de imagem (pqiv), visualizador de PDF (okular), visualizador de texto (kate) e para visualização de documentos o LibreOffice. Após a instalação e configuração, o sistema estará pronto para a inserção de Scripts.

As versões de hardware e softwares utilizados nesse trabalho foram:

- ❑ Raspberry Pi 3 Modelo B;
- ❑ Sistema Operacional Raspbian versão 8 Stretch 4.14.
- ❑ Servidor web Apache versão 2.5;
- ❑ PHP7 versão 7.0.33;
- ❑ MySQL versão 5.7.26;
- ❑ Pqiv versão 2.6;
- ❑ Okular versão 1.1.3;
- ❑ Kate versão 17.04.3;
- ❑ LibreOffice versão 6.1.3.2.

3.3 Fluxo das Atividades Desenvolvidas

Nos Capítulos e Seções anteriores, foram ilustrados, em vias gerais, o funcionamento do hardware (Figura 1), as interações do Raspberry Pi (Figura 7) e, na Figura 8 o funcionamento da lógica de desenvolvimento para facilitar e exibir o fluxo de como o software está em execução, tendo como principais características um ponto de início e um ponto

final de execução, essa representação é outra maneira de exemplificar o funcionamento do trabalho. Vale ressaltar que serão explicitados os desenvolvimentos e resultados da implementação do módulo do professor.

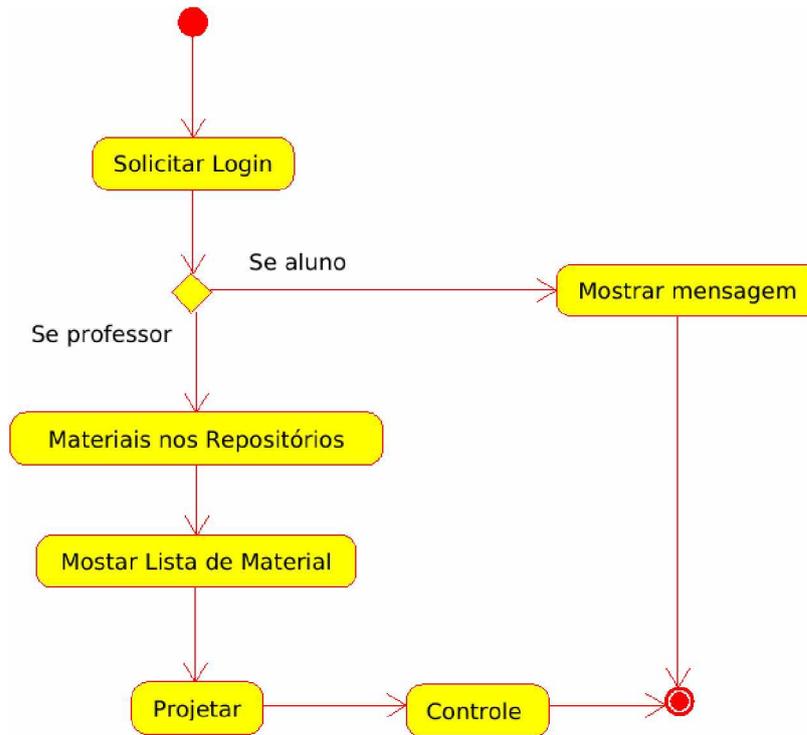


Figura 8 – Diagrama de Atividades, trançando os possíveis trajetórias da usabilidade dos usuários.

3.3.1 Iniciando o Sistema

Inicialmente, quando o Raspberry Pi é ligado, após o carregamento do Sistema Operacional (SO), é iniciado o navegador Chromium, para isso foi inserido um agendamento de tarefas no *cron*, ou seja, um script que possui instruções para abrir o navegador Chromium em tela cheia carregando uma imagem de fundo. A imagem apresentada contendo a logotipo do trabalho (Figura 9). Este processo é feito com o propósito de prover um visual mais amigável para os alunos que estarão visualizando as informações no projetor, preservando as funcionalidades por trás da plataforma.

Uma das configurações realizadas no SO, mais especificamente em Scripts contendo configuração de rede de internet, foram modificados para que os objetivos do trabalho fossem atingidos e, assim, as alterações tornaram possível: o servidor (Raspberry Pi) através de uma conexão LAN terá acesso à web, e ao mesmo tempo realizará o roteamento, tornando o servidor um ponto de acesso Wi-Fi. Entretanto, foi configurado para que o servidor tenha acesso a internet, mas não permita acesso a web por meio da sua conexão Wi-Fi disponibilizada, deste modo, é possível que somente o servidor acesse arquivos na internet, mas os usuários conectados só troquem informações com o servidor.



Figura 9 – Tela de Abertura do Trabalho. Imagem modificada de (Freitas, 2017) - slide 4.

Junto ao sistema operacional também foi introduzido um Script em PHP, cujo o nome é “servidor_socket.php”, que tem o objetivo ser o servidor soquete, que ficará responsável por fazer a comunicação entre os clientes (alunos e professor) e o servidor (Raspberry Pi). O Script entra em execução na inicialização do SO e permanecendo ativo até o desligamento do mesmo. Esse Script faz a interpretação e comunicação com o terminal de comandos do núcleo do SO, e toda a ação é executada em segundo plano. A Figura 10 mostra a ligação existente entre esses arquivos.

Algoritmo 1: SERVIDOR SOCKET

Entrada: host = "127.0.0.1", port = 25003, set_time_limit(0)

Saída: Servidor Socket Ativo

início

Dados: socket = socket_create(AF_INET, SOCK_STREAM, 0);

Dados: result = socket_bind(socket, host, port);

repita

Dados: result = socket_listen(socket, 3);

Dados: spawn = socket_accept(socket);

Dados: input = socket_read(spawn, 1024);

Dados: res = exec(input);

Dados: input = trim(input);

Dados: socket_write(spawn, input, strlen (input));

até Verdade;

Dados: socket_close(spawn);

fim

retorna *Sucesso*

Como pode ser observado no algoritmo 1, inicialmente é definido o *host* e porta, e

o `Socket_create` cria uma instância de socket, e o `socket_bind` cria o `bind` soquete, logo após o serviço de soquete espera por uma conexão, isso é feito adicionando um cliente, lendo a requisição, executando pela função `exec`, em seguida escreve uma resposta que é a saída do `exec` e, por fim, fecha o soquete.

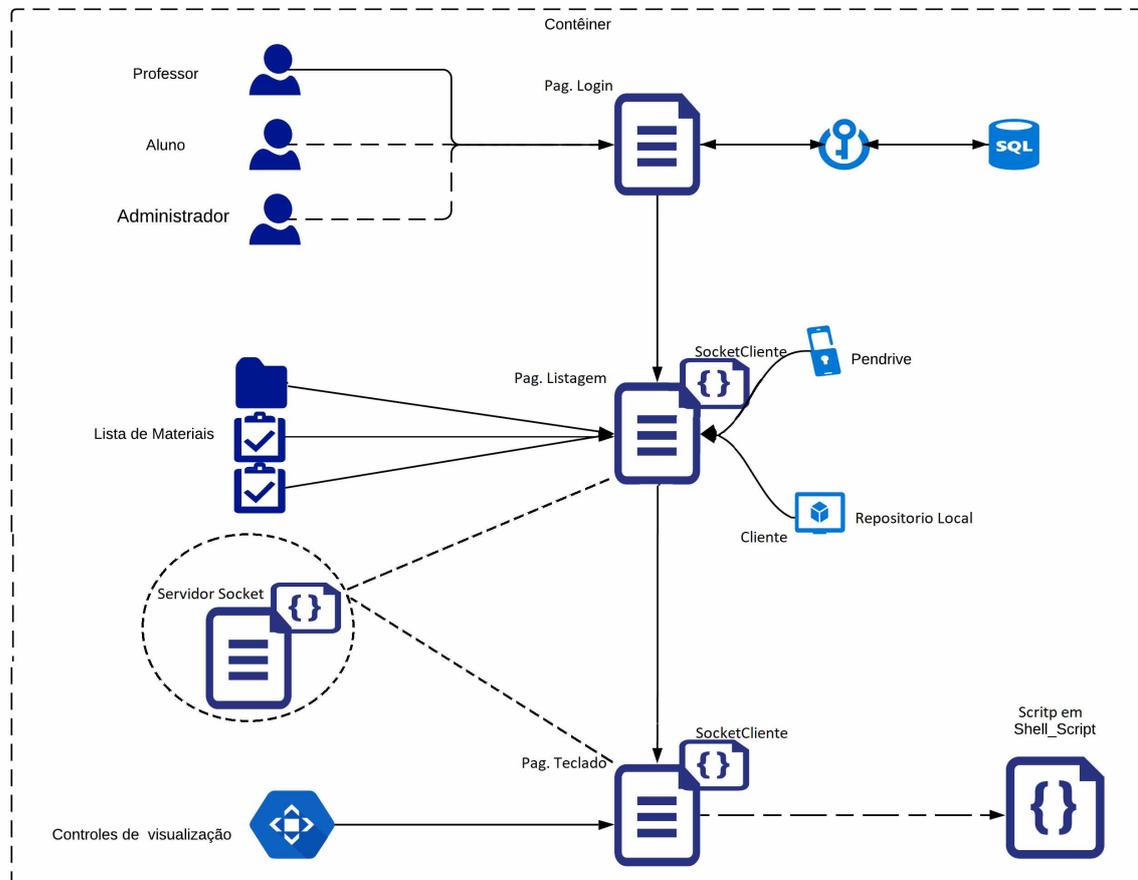


Figura 10 Visão geral das atividades e processos do trabalho.

3.3.2 Acesso aos Arquivos

Uma das propostas do trabalho é permitir o professor acessar os arquivos em diferentes repositórios, por exemplo USB, Dropbox, Google Drive, One Drive, dentre outros. Neste trabalho foi disponibilizado arquivos locais e a entrada USB. Foi configurado o sistema, criando um atalho da unidade USB para um repositório local no servidor e, por fim, fazer a montagem automática da unidade USB. Sempre que for introduzido um pendrive ou disco rígido, o mesmo é acessível pelo trabalho desenvolvido.

Supondo que não exista uma unidade de armazenamento externa conectada ao servidor (Raspberry Pi), a pasta continua visível para o usuário, mas vazia até que seja conectado um dispositivo de armazenamento. Todo esse fluxo está ilustrado no pseudo-código 2.

Algoritmo 2: FLUXO DE ACESSO USB**Entrada:** Usuário, Senha, Disciplina, Materiais**Saída:** Atividade cumprida**início** **Dados:** Cria-se uma pasta 'Pendrive' **Dados:** Comando: mkdir /var/www/html/files/Pendrive **Dados:** Cria um atalho do diretório de dispositivos USBs para a pasta pendrive **Dados:** Comando: ln -s /media/pi/ Pendrive **Dados:** Adicionar permissão de acesso aos repositórios USB **Dados:** Comando: sudo chmod a+x /media/pi/ **Dados:** Comando: sudo chmod a+r /media/pi/ **repita** **Dados:** A pasta mantém vazia **se** *USB conectado* **então** **Dados:** A pasta 'Pendrive' não esta mais vazia **fim** **senão se** *USB removido* **então** **Dados:** A pasta 'Pendrive' não esta mais com repositório USB **fim** **senão** **Dados:** A pasta mantém vazia **fim** **até** *Repete enquanto verdadeiro;***fim****retorna** *Sucesso*

3.4 Desenvolvimento

Todos os Scripts em PHP, Shell Script, pastas, arquivos devem ser copiados para a pasta do servidor Web, localizada em `/var/www/html/`. Esta pasta deve conter os arquivos de configuração das páginas, como Folha de Estilo em Cascatas (CSS), Arquivo JavaScript (JS), imagens e os próprios Scripts em PHP das páginas Web, além de um atalho feito pela configuração de montagem de uma unidade USB, uma pasta de repositório local e os Scripts com permissão de execução, sendo servidor soquete e o Script gerenciador em PHP.

Após a preparação e instalação de todas as atualizações, programas, inserção de arquivos, Scripts e configurações, o próximo passo é o usuário se conectar à rede Wi-Fi disponibilizada pelo Raspberry Pi ("Sala_Inteligente"). A Figura 11 ilustra a página de login (*index.php*). Esta página é um Script que faz conexão com o banco de dados para validar o acesso utilizando login e senha. O sistema identifica se é um aluno ou professor através de uma consulta ao SGBD e, de acordo com o perfil do usuário, será redirecionado para a devida página de conteúdo.

O login de acesso é realizado por meio de uma consulta ao SGBD, e caso o login e senha sejam inválidos, o acesso a próxima página é negado, mas caso a tentativa de acesso



Figura 11 – Tela de login para acesso de alunos e professores.

esteja correta o sistema encaminhará o usuário para sua devida página. Por exemplo, caso seja um aluno, uma mensagem de boas vindas será apresentada e, se o usuário que está se logando for um professor, então ele será encaminhado para uma página de listagem de material.

A Figura 12 ilustra o caso de uso dos atores (professor e aluno), o ato de acessar a página de login é a primeira interação dos usuários.

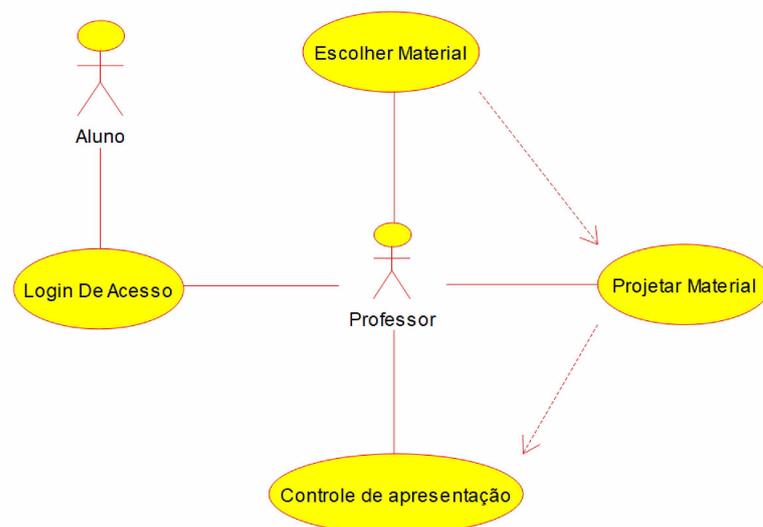


Figura 12 – Caso de Uso dos atores (professor e aluno).

A página *listagem.php* é aberta assim que o professor se autentica com sucesso na tela de login e, nessa tela pode ser escolhido entre os arquivos no pendrive, se existir algum conectado ou arquivos no repositório local. Como ilustra a Figura 13, é visível um ícone representado o tipo de arquivo, o nome e sua respectiva extensão para que possa identificar o arquivo pretendido, quanto ao “Pendrive” segue a mesma estrutura.



Figura 13 – Exemplo da tela com a listagem de pastas e arquivos.

Para obter os arquivos e exibir a listagem para o professor, na página *listagem.php*, utilizando recursos e estruturas da linguagem PHP, foi realizado uma varredura no diretório especificado, um repositório local, no caso */var/www/html/files/*. Em seguida, foi filtrado tipos específicos de arquivos, como PDF, TXT, JPEG, ODS, ODF, ODT, ODP e diretórios válidos, ou seja, que fazem parte dos arquivos do pendrive ou local. Foi feita esta filtragem para evitar eventuais tentativas de corromper o sistema com Scripts e arquivos corrompidos ou materiais não voltados à aula, como por exemplo, arquivos EXE, INI, LNK, DLL, CFG, DAT e etc.

Todo o conteúdo retornado da busca realizada pelo Script *listagem.php* é colocado em uma lista, começando com o diretório “Pendrive”, seguidos dos outros diretórios e por fim os arquivos. Tudo é mostrado de forma ordenada, primeiro os diretórios seguido dos arquivos. Vale lembrar, que o diretório “Pendrive” só será exibido se existir uma unidade USB conectada ao servidor (Raspberry Pi), caso a unidade USB esteja conectada mas, não tenha arquivos, o diretório é visível, mas será exibido ao professor sem arquivos e, caso a unidade USB esteja conectada e com arquivos, então a unidade será visível e, terá arquivos visíveis se os mesmos forem tipos de arquivos aceitos como já mencionado anteriormente, ou seja, são 3 estados diferentes para exibição do “Pendrive”.

Caso o professor abra um diretório local ou o diretório “Pendrive”, será apresentada a mesma estrutura anterior, uma lista de arquivos e diretórios, se existirem, tornando a estrutura recursiva.

É válido lembrar do caso de uso (Figura 12) que representa de forma simples e estruturada as ações do professor. Nesse cenário, logo após o professor com seu login e senha se autenticar na rede, o professor é direcionado para a segunda página *listagem.php*.

Assim que o professor abrir algum arquivo por meio de um clique, será redirecionado para a página *teclado.php*, controlando a apresentação aberta. A Figura 14 ilustra a única página de interação com os diferentes tipos de arquivos.

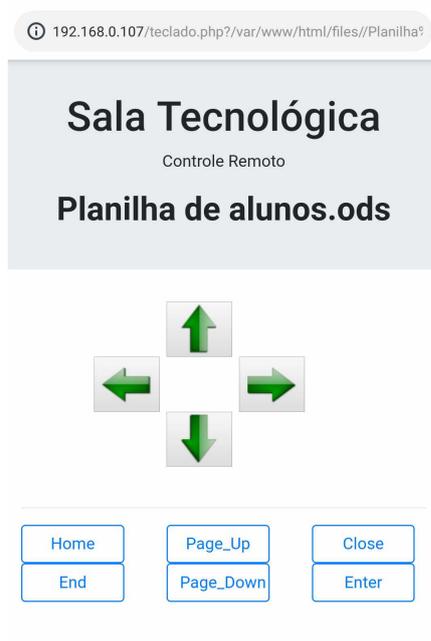


Figura 14 – Tela contendo o teclado para manipulação de arquivos.

Assim que o Script *teclado.php* é requisitado, este recebe do Script *listagem.php* o nome do arquivo que foi selecionado pelo usuário e exibe para o professor.

O processo de abrir o arquivo, não apenas faz o direcionamento de Script como também abre o arquivo que o professor requisitou. Este processo de abrir o documento é realizado por meio da comunicação de soquetes entre o cliente e o servidor. O soquete cliente está localizado na *listagem.php*, sendo passado para o servidor soquete (*servidor_socket.php*) por meio de uma mensagem que contém parâmetros para realizar a ação de abrir o arquivo. Os parâmetros enviados são: o nome e o endereço do arquivo à ser aberto, a extensão do arquivo e o tipo de interação que está sendo realizada, nesse caso, abrir o arquivo.

O Script *script_gerenciar.sh* localizado no servidor (Raspberry Pi) realiza a interpretação dos outros comandos contidos na mensagem e os executa. Esses comandos são: nome do arquivo, a extensão do arquivo e um valor numérico que indica o tipo de ação a ser executada, que neste caso é abrir um documento. Utilizando os recursos do PHP, a

ação do servidor soquete ao receber a mensagem é executar, no núcleo do Linux, a mensagem que recebeu usando a função da linguagem PHP *exec()*. Essa mensagem contém o endereço e os parâmetros que o Script *script_gerenciar.sh* precisa para abrir o arquivo. Ao ser executado, este interpreta qual ação é para executar, no caso abrir arquivo e por fim, utilizando como último parâmetro a extensão do arquivo é realizado o processo de abrir, tratando cada tipo de arquivo como deve ser aberto, exemplo em tela cheia, em modo apresentação, em modo leitura, etc.

A ação do professor em abrir um arquivo gera duas ações: a de ser redirecionado para a página *teclado.php* e a comunicação entre o cliente e o servidor para transmitir a mensagem que executará o Script *script_gerenciar.sh*, que de fato interpretará e abrirá no servidor o arquivo selecionado, seja documento, apresentação, imagem, etc.

A estrutura da página *teclado.php* é ilustrada na Figura 14. Visivelmente tem-se o nome do documento aberto (*Planilha de alunos.ods*), logo abaixo um conjunto de botões em setas para movimentar os arquivos e, logo abaixo os botões *Home*, *End*, *Page_Up*, *Page_Down*, *Close*, *Enter*, simulando o teclado físico.

No momento que o professor realizar qualquer interação, clicando em algum dos botões disponíveis para o determinado arquivo aberto, será mapeado qual tecla, o nome, o endereço e a extensão do arquivo aberto, colocado-os em uma mensagem e enviando do soquete cliente para soquete servidor, que por sua vez executa a mensagem no servidor. Com a função do PHP *exec* a Mensagem é executada no núcleo do sistema e, o processo de abrir o Script *script_gerenciar.sh* é realizado e os parâmetros passados. O *script_gerenciar.sh* interpreta que é uma interação de documento, qual documento e qual ação tomar para tal documento.

O processo de abrir e interagir com os arquivos é similar, difere o modo como o *script_gerenciar.sh* trata cada parâmetro que recebe. O script, por meio da ferramenta Xdotool, simula as entradas de comando no software em questão, seja para colocar em primeiro plano, se minimizado então maximizar a janela, próximo, anterior, fechar, enfim, todas as ações possíveis que o teclado virtual criado no *teclado.php* deve realizar.

Assim que a apresentação for concluída e fechada o professor é redirecionado para a página de início, na qual visualizará seus arquivos novamente.

Todo o fluxo explicado está ilustrado na Figura 10. Para representar e exemplificar o trabalho, o Pseudo-código 3 tem o intuito de resumir o funcionamento e atividades de cada ator (professor e aluno).

Um diagrama de estado (Figura 15) foi criado com o intuito de facilitar o entendimento de transição entre os diversos Scripts criados. Quando o SO é iniciado, vários Scripts do próprio SO são executados, junto a eles foram inseridos mais alguns que fazem parte do trabalho: script de inicialização do servidor Apache, o servidor soquete que aguarda uma requisição, o script de ponto de acesso Wireless e o script que torna possível o navegador Chromium iniciar em tela cheia com uma imagem de fundo.

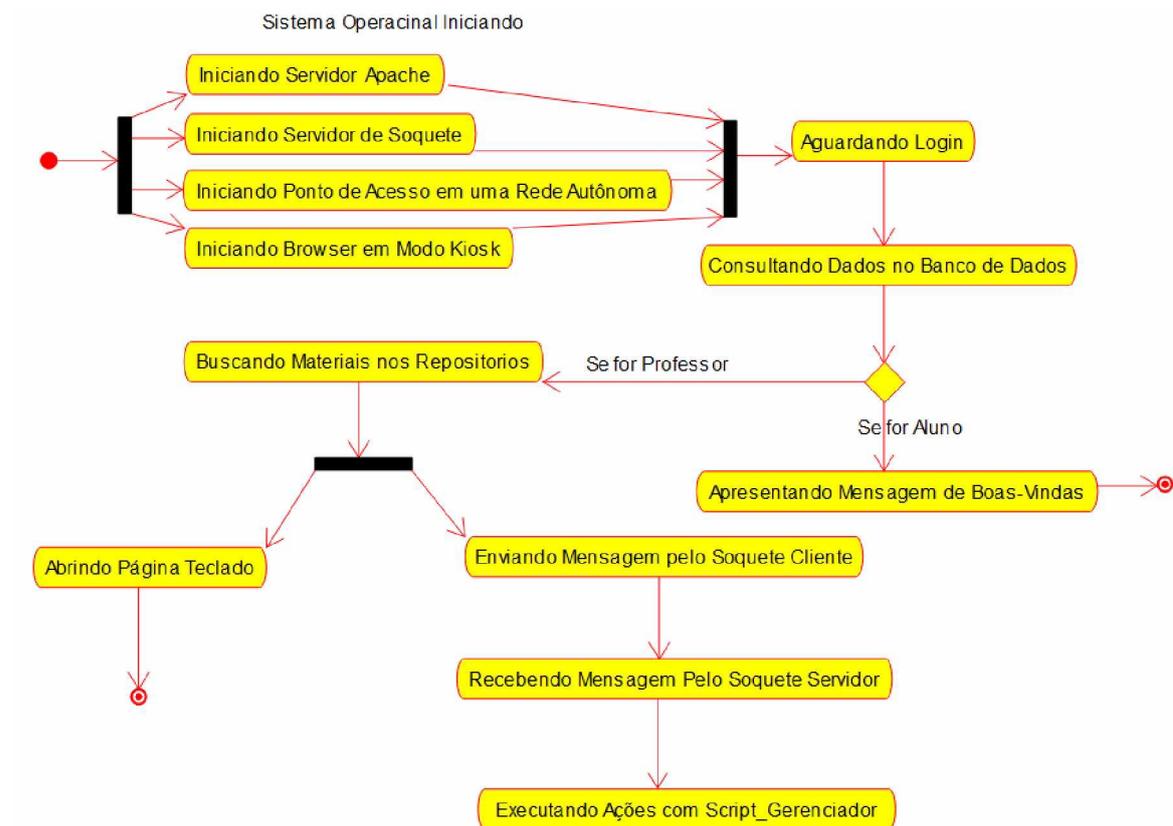


Figura 15 – Diagrama de Estado - Fluxo entre os Scripts

Assim que todo o SO é iniciado, o usuário pode se conectar a rede Wi-Fi e realizar o login, nesse momento, quando o usuário colocar os dados, será feita uma validação no banco de dados para verificar a existência de usuário. Caso seja um aluno, uma mensagem é apresentada. Caso seja um professor, um novo estado é apresentado, que será listando os materiais nos repositórios, um script faz a busca e exibe para o professor.

Logo em seguida, quando o professor escolher algum dos materiais listados dois estados são iniciados: o primeiro estado é de abrir um novo Script *teclado.php* e, o segundo estado acionado é o soquete cliente enviar uma mensagem para o soquete servidor contendo informações do pedido do professor. A partir do momento que o soquete servidor recebe a mensagem do soquete cliente, este executa uma função do PHP *exec* no núcleo do SO passando a mensagem que é uma chamada de script que contem parâmetros que são informações do pedido que pode ser abrir ou uma função do teclado a pedido do professor.

Algoritmo 3: FLUXO DE ATIVIDADE**Entrada:** Usuário, Senha, Disciplina, Materiais**Saída:** Atividade comprida

início

Dados: Usuário conecta-se ao Wi-Fi se *Login e Senha = corretos* então se *Usuário = Professor* então **Dados:** Será exibido 2 opções (Pendrive, Diretório-Local) se *Pendrive* então **Resultado:** Uma lista de arquivos no pendrive será exibida se *Arquivo = Y* então **Resultado:** O arquivo será aberto **Dados:** O usuário sera redirecionado a uma página de controle do material

fim

 senão se *Diretório = X2* então **Resultado:** Será exibido uma lista de materiais

repita

| Loop:

 até *enquanto* haver *subdiretórios*;

fim

 se *(Diretório = Vazio) e/ou (Arquivos = Inválidos)* então **Resultado:** Uma mensagem de repositório vazio e exibida

fim

fim

 se *Diretório-Local* então **Resultado:** Uma lista de arquivos do Diretório será exibido se *Arquivo = Y* então **Resultado:** O arquivo será aberto **Dados:** Usuário redirecionado a página teclado

fim

 senão se *Diretório = X2* então **Resultado:** Será exibido uma lista de materiais

repita

| Infinito:

 até *enquanto* haver *subdiretórios*;

fim

 senão se *(Diretório = Vazio) e/ou (Arquivos = Inválidos)* então **Resultado:** Uma mensagem de repositório vazio é exibida

fim

fim

fim

 se *Aluno ou Administrador* então **Resultado:** Uma mensagem é exibida

fim

fim

senão

Resultado: Erro de login

fim

fim

retorna *Sucesso*

Conclusão

Ao longo de vários anos, as instituições de ensino enfrentaram diversos problemas, por exemplo, políticos, culturais, psicológicos e assim por diante. Mas atualmente, o que mais chama a atenção por ser um problema dentro das salas de aula, são as novas tecnologias, que estão presentes no dia a dia das pessoas com infinitas utilidades e, na maioria das vezes, tirando o foco dos estudos.

O desenvolvimento do projeto "Sala de Aula Inteligente com o uso de Raspberry Pi" poderá auxiliar na amenização desse problema. Seu desenvolvimento partiu da hipótese de que retirar ou inibir o uso de dispositivos eletrônicos não era a melhor alternativa para ter um aprendizado ideal, porque isso traria revolta e descontentamento por parte dos alunos.

Diante disso, o desenvolvimento deste trabalho teve como foco a parte do professor, criando um ambiente para que possa selecionar, projetar e controlar o material didático a ser apresentado durante as aulas, tudo isso baseado na tecnologia que ele já utiliza.

Constata-se que o objetivo geral e específicos foram atingidos, pois conseguiu-se desenvolver um sistema para acesso ao material didático e realizar a seleção e o controle da projeção deste material por meio de dispositivo móvel, além da projeção remota do material por meio do Raspberry Pi.

Diante da metodologia proposta, percebe-se que o trabalho poderia ser realizado com um pesquisa mais ampla na bibliografia para pesquisar as ferramentas de interação com os serviços da LibreOffice, onde foi encontrado dificuldade para realizar a atividade de interagir com os documentos suportados pela mesma.

4.1 Trabalhos Futuros

Para trabalhos futuros devem ser agregados ao projeto a obtenção de materiais da plataforma de aprendizado Moodle, Dropbox, Google Drive, One Drive, dentre outras plataformas.

A utilização do Moodle poderá realizar a autenticação por meio da base de dados de login e senha, além de gerar um maior controle de presença dos alunos.

Os controles de acesso podem ser modificados para específicos tipos de arquivos, por exemplo, para uma apresentação é necessário apenas o controle de próximo, anterior e talvez fim do arquivo e início, não sendo necessário mostrar ao usuário a opção de confirmar (*Enter*).

4.2 Contribuições em Produção Bibliográfica

Como contribuições, parte do trabalho já foi publicado e apresentado em:

SILVA, Cássio Moreira ; RIBEIRO, Thiago Pirola. Sala de Aula Tecnológica Baseada em IoT. In: **TECHnoMONTE 2018**, 2018, Monte Carmelo - MG. Anais da TECHnoMONTE 2018, 2018.

Referências

- Arduino Project HUB. **Arduino Mega Chess**. 2017. Disponível em: <https://create.arduino.cc/projecthub/Sergey_Urusov/arduino-mega-chess-d54383?ref=platform&ref_id=424_trending____&offset=3>. Acesso em: 16 de Janeiro de 2018. Citado na página 15.
- Arduino Project HUBs. **Luzes LED RGB de escadaria automatizada**. 2017. Disponível em: <https://create.arduino.cc/projecthub/notenoughtech-com/automated-staircase-rgb-led-lights-7d26d5?ref=platform&ref_id=424_trending____&offset=2>. Acesso em: 16 de Janeiro de 2018. Citado na página 15.
- BeagleBoard. **Sobre o BeagleBoard.org e a Fundação BeagleBoard.org**. 2013. Disponível em: <<http://beagleboard.org/about>>. Acesso em: 28 de Março de 2018. Citado na página 18.
- Bohn Dieter e Aaron, S. ios: A visual history. 2013. Disponível em: <<https://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad>>. Citado na página 18.
- Cavalcante, M. M. et al. A plataforma arduino para fins didáticos: Estudo de caso com recolhimento de dados a partir do plx-daq. In: **XXXIV Congresso da Sociedade Brasileira de Computação (CSBC 2014), Brasília, DF**. [S.l.: s.n.], 2014. Citado na página 11.
- Delfino, P. **Sistemas operacionais - conheça os melhores do mercado e tenha outras opções**. 2018. Disponível em: <<https://e-tinet.com/linux/sistemas-operacionais-os-melhores/>>. Acesso em: 23 de junho de 2019. Citado na página 18.
- Developers, A. **About the platform**. Android Developers, 2019. Disponível em: <<http://developer.android.com/about>>. Citado na página 18.
- Filipe. **Top 10 linguagens de programação mais usadas no mercado**. 2018. Disponível em: <<https://www.devmedia.com.br/top-10-linguagens-de-programacao-mais-usadas-no-mercado/39635>>. Acesso em: 23 de junho de 2019. Citado na página 19.
- Fonseca, B. **Arduino e suas múltiplas aplicações**. 2017. Disponível em: <<https://www.redbull.com/br-pt/arduino-e-suas-multiplas-aplicacaoe>>. Acesso em: 23 de junho de 2019. Citado na página 15.

Forbellone, A. L. V.; Eberspacher, H. F. **Lógica de programação**. [S.l.]: Makron Books, 2000. 13 p. Citado na página 19.

Freitas, A. B. de. **Tecnologia mb exame**. 2017. Disponível em: <<https://pt.scribd.com/presentation/391196858/tecnologia-mb-exame-pptx>>. Citado 2 vezes nas páginas 6 e 26.

Jargas, A. M. **Introdução ao Shell Script**. [S.l.]: aurelio.net, 2004. Citado na página 19.

Kurose, J.; Ross, K. **Redes de computadores e a Internet: uma abordagem top-down**. ADDISON WESLEY BRA, 2007. ISBN 9788588639188. Disponível em: <<https://books.google.com.br/books?id=i5WwAAAACAAJ>>. Citado 2 vezes nas páginas 6 e 20.

Marinho, J. A. **Shell script profissional**. [S.l.]: Novatec Editora, 2008. Citado na página 20.

Marra, G. M. Aplicações php compiladas utilizando o roadsend studio for php. **Revista EVS-Revista de Ciências Ambientais e Saúde**, v. 36, n. 3, p. 653–662, 2009. Citado na página 19.

Massetto, P. D. F. I. **Introdução: Sockets**. 2012. Disponível em: <<http://professor.ufabc.edu.br/francisco.massetto/sd/03-Aula3-Sockets.pdf>>. Acesso em: 20 de Março de 2019. Citado na página 19.

Meirinhos, M.; Osório, A. Projeto professores inovadores com tic. In: UNIVERSIDADE DO MINHO. **Atas da IX Conferência internacional de TIC na Educação**. [S.l.], 2015. p. 1174–1181. Citado na página 11.

Monteiro, L. P. **O que é linguagem de programação**. 2018. Disponível em: <<https://universidadedatecnologia.com.br/o-que-e-linguagem-de-programacao/>>. Acesso em: 23 de junho de 2019. Citado na página 19.

Nunes, L. dias; Strazburger, R. Disciplina linguagem de programação-sistema de informação-faculdades integradas de taquara (faccat). 2009. Disponível em: <https://fit.faccat.br/~jonis/Artigo_mySQL.pdf>. Acesso em: 08 de Fevereiro de 2018. Citado na página 19.

Oliveira, C. de. Tics na educação a utilização das tecnologias da informação e comunicação na aprendizagem do aluno. **Pedagogia em Ação**, v. 7, n. 1, 2015. Citado na página 11.

PHP Net. **php.net**. 2018. Disponível em: <www.php.net>. Acesso em: 26 de Março de 2018. Citado na página 19.

Prado, C. A. d. et al. As políticas públicas e o perfil do aluno do século xxi frente à inserção da tecnologia na educação. **REVISTA INTERSABERES**, v. 12, n. 25, p. 178–187, 2017. Citado na página 11.

Raspberrypi. **Forum Raspberry PI**. 2012. Disponível em: <<https://www.raspberrypi.org/forums/>>. Acesso em: 16 de Janeiro de 2018. Citado na página 17.

Raspberry PI. **Compre um Raspberry PI**. 2017. Disponível em: <<https://www.raspberrypi.org/products/>>. Acesso em: 29 de Março de 2018. Citado na página 16.

Raspberry Project. **Olá! O que você gostaria de fazer hoje**. 2012. Disponível em: <<https://projects.raspberrypi.org/en/>>. Acesso em: 08 de Fevereiro de 2018. Citado na página 18.

Raspberrypi. **What is a raspberry pi**. 2012. Disponível em: <<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>>. Acesso em: 16 de Janeiro de 2018. Citado na página 16.

Sissel, J. **Semicomplete - Xdotool**. 2015. Disponível em: <<https://www.semicomplete.com/categories/projects/>>. Citado na página 21.

Sutilo, S. N. F. Tecnologia e educação. **Cadernos de Gestão e Empreendedorismo**, v. 4, n. 1, p. 50–69, 2017. Citado na página 11.