

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ENGENHARIA ELETRÔNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA

RENAN FELIPE TOSHIAKI BIESZCZAD IKEDA

**SISTEMA DE GERENCIAMENTO DE EMPRÉSTIMOS DOS
EQUIPAMENTOS DE LABORATÓRIO DO DAELN-CM**

TRABALHO DE CONCLUSÃO DE CURSO

CAMPO MOURÃO - PR

2017

RENAN FELIPE TOSHIAKI BIESZCZAD IKEDA

**SISTEMA DE GERENCIAMENTO DE EMPRÉSTIMOS DOS
EQUIPAMENTOS DE LABORATÓRIO DO DAELN-CM**

Trabalho de Conclusão de Curso de Graduação do curso de Bacharelado em Engenharia Eletrônica do Departamento Acadêmico de Eletrônica (DAELN) da Universidade Tecnológica Federal do Paraná (UTFPR), como requisito final para obtenção do título de Engenheiro em Eletrônica.

Orientador: Prof. Me. Lucas Ricken Garcia

CAMPO MOURÃO – PR

2017



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Campo Mourão
Coordenação do Curso de Engenharia Eletrônica

TERMO DE APROVAÇÃO
DO TRABALHO DE CONCLUSÃO DE CURSO INTITULADO
SISTEMA DE GERENCIAMENTO DE EMPRÉSTIMOS DOS
EQUIPAMENTOS DE LABORATÓRIO DO DAELN-CM

por

Renan Felipe Toshiaki Bieszczad Ikeda

Trabalho de Conclusão de Curso apresentado no dia 20 de Junho de 2017 ao Curso Superior de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná, Campus Campo Mourão. O Candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho _____ (aprovado, aprovado com restrições ou reprovado).

Prof. Dr. André Luiz Regis Monteiro
(UTFPR)

Prof. Me. Reginaldo Nunes de Souza
(UTFPR)

Prof. Me. Lucas Ricken Garcia
(UTFPR)
Orientador

A Folha de Aprovação assinada encontra-se na Coordenação do Curso

AGRADECIMENTOS

Agradeço a todos aqueles que me apoiaram na realização do curso e deste trabalho. Em especial agradeço a Deus, a minha família, e a minha querida namorada que foi essencial durante praticamente todo o tempo em que eu estive na faculdade, sempre foi paciente e extremamente amorosa, mesmo nos momentos desfavoráveis. Agradeço também aos meus professores, principalmente ao meu grande amigo e orientador, que foi extremamente importante na realização deste trabalho. E por fim, agradeço a todos os meus amigos que de alguma forma contribuíram para a realização deste trabalho e também do curso.

Obrigado a todos!

RESUMO

IKEDA, RENAN F. T. B. SISTEMA DE GERENCIAMENTO DE EMPRÉSTIMOS DOS EQUIPAMENTOS DE LABORATÓRIO DO DAELN-CM. Trabalho de Conclusão de Curso – Bacharelado em Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná. Campo Mourão, 2017.

A automatização de processos é cada vez mais comum nos dias de hoje, isso faz com que todo processo manual seja passível a automatização. No caso deste trabalho, foi desenvolvido um sistema para automatizar o processo de registro dos empréstimos de equipamentos do departamento de engenharia eletrônica do campus da UTFPR de Campo Mourão. Este projeto foi desenvolvido com o auxílio de um sistema embarcado (*Raspberry PI*) e utiliza um leitor de código de barras para realizar o processo de leitura dos códigos de patrimônio do equipamento e de RA do aluno. O registro dos empréstimos é armazenado em um banco de dados e a inserção dos dados é feita através de uma interface *web*. O sistema também gera alguns relatórios contendo informações pertinentes e mantendo um histórico dos empréstimos realizados. Com o desenvolvimento desse sistema, o registro dos empréstimos tende a melhorar, isso foi comprovado através da realização de testes que validaram o seu funcionamento.

Palavras chave: banco de dados, sistema web, automatização de processos

ABSTRACT

IKEDA, RENAN F. T. B. DAELN-CM LABORATORY EQUIPMENT LOAN MANAGEMENT SYSTEM. Course Completion Work - Bachelor of Electronic Engineering, Federal Technological University of Paraná. Campo Mourão, 2017.

Process automation is becoming more common these days, so every manual process is amenable to automation. In the case of this work, a system was developed to automate the process of registering equipment loans from the electronic engineering department of the campus of UTFPR Campo Mourão. This project was developed with the aid of an embedded system (Raspberry PI) and uses a bar code reader to perform the process of reading the equipment patrimony codes and the student's RA. Loan registration is stored in a database and data entry is done through a web interface. The system also generates some reports containing pertinent information and maintains a history of the loans made. With the development of this system, loan registration tends to improve, this has been proven through the tests that validated its operation.

Keywords: database, web system, process automation

LISTA DE FIGURAS

Figura 1 - Raspberry PI Model B.	15
Figura 2 - Etapas de funcionamento de uma aplicação Web.....	17
Figura 3 - Exemplo de descrição HTML.....	18
Figura 4 - Diagrama de funcionamento do AJAX.....	20
Figura 5 - Componentes de um sistema de banco de dados.	21
Figura 6 - Diagrama de blocos do sistema	23
Figura 7 - Interface do programa Win32 Disk Imager	26
Figura 8 - Fluxograma do modulo leitor de código de barras.	27
Figura 9 - Leitores de código de barra.....	28
Figura 10 - Diagrama das tabelas do banco de dados.....	29
Figura 11 - Diagrama geral da interface web.	31
Figura 12 - Diagrama da página de cadastro de alunos.....	31
Figura 13 - Diagrama da página de cadastro de equipamentos.	32
Figura 14 - Diagrama da página de empréstimos.....	33
Figura 15 - Diagrama de funcionamento da página Relatório de empréstimos para o aluno...	34
Figura 16 - Diagrama da página Relatório de empréstimos por equipamento.	35
Figura 17 - Número de patrimônio utilizado para os testes.....	36
Figura 18 - Patrimônio do equipamento conforme lido pelo módulo leitor de código de barras.	37
Figura 19 - Número de RA utilizado nos testes.....	37
Figura 20 - RA lido pelo módulo leitor de código de barras.....	38
Figura 21 - Tabela de alunos no banco de dados do sistema.....	39
Figura 22 - Tabela de equipamentos no banco de dados.....	39
Figura 23 - Tabela de empréstimos do banco de dados.....	40
Figura 24 - Página de login.	41
Figura 25 – Página inicial do sistema.....	41
Figura 26 – Falha no login.....	42
Figura 27 - Página de cadastro de alunos.	43
Figura 28 – erro no cadastro de alunos.....	43

Figura 29 - Página de cadastro de equipamentos.	44
Figura 30 - Erro no cadastro de equipamentos.	45
Figura 31 - Página de empréstimo e devolução.....	46
Figura 32 – Erro de falta de cadastro de equipamento.	47
Figura 33 - Erro de aluno não cadastrado.....	47
Figura 34 – Erro de duplicidade de empréstimos.	48
Figura 35 - Erro na devolução.	49
Figura 36 – Tela de relatório geral.	50
Figura 37 – Formulário para geração do relatório de empréstimos por aluno.	51
Figura 38 – Erro na geração do relatório de empréstimos por aluno.	51
Figura 39 – Relatório de empréstimos por aluno.	52
Figura 40 – Formulário para geração do relatório de empréstimos por equipamento.....	53
Figura 41 – Erro no relatório por equipamentos.....	53
Figura 42 – Relatório de empréstimos por equipamento.....	54

LISTA DE QUADROS

Quadro 1 – Comparativo dos modelos de Raspberry Pi.....	17
---	----

LISTA DE SIGLAS

AJAX	<i>Asynchronous Javascript and XML</i>
ANSI	<i>American National Standards Institute</i>
ARM	<i>Advanced RISC Machine</i>
CSS	<i>Cascading Style Sheets</i>
DAELN	Departamento Acadêmico de Engenharia Eletrônica
GNU/GLP	<i>General Public License</i>
GPIO	<i>General Purpose Input/Output</i>
GPS	<i>Global Positioning System</i>
HTML	<i>HyperText Markup Language</i>
ODBC	<i>Open Database Connectivity</i>
PDO	<i>PHP Data Objects</i>
PHP	<i>Hypertext Preprocessor</i>
PI	Projeto Integrador
RA	Registro Acadêmico
RAM	<i>Random Access Memory</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language</i>
TCC	Trabalho de Conclusão de Curso
USB	<i>Universal Serial Bus</i>
UTFPR	Universidade Tecnológica Federal do Paraná
W3C	<i>World Wide Web Consortium</i>
WDDX	<i>Web Distributed Data eXchange</i>
WHATWG	<i>Web Hypertext Application Technology Working Group</i>

SUMÁRIO

1. INTRODUÇÃO.....	11
1.1. TEMA.....	11
1.2. JUSTIFICATIVA.....	12
1.3. OBJETIVOS.....	13
1.3.1. Objetivos específicos	13
2. FUNDAMENTAÇÃO TEÓRICA	14
2.1. AUTOMATIZAÇÃO DE PROCESSOS	14
2.2. SISTEMAS EMBARCADOS.....	14
2.3. RASPBERRY PI.....	15
2.4. APLICAÇÃO WEB.....	16
2.4.1. HTML	17
2.4.2. PHP	18
2.4.3. AJAX	19
2.5. BANCO DE DADOS.....	20
2.5.1. SQL.....	21
2.5.2. MySQL	22
3. MATERIAIS E MÉTODOS	23
3.1. METODOLOGIA	23
3.2. INSTALAÇÃO DO SISTEMA OPERACIONAL (SO) <i>JESSIE LITE</i>	25
3.3. DESENVOLVIMENTO DO MÓDULO LEITOR DE CÓDIGO DE BARRAS.....	26
3.4. PROJETO DO BANCO DE DADOS DO SISTEMA	29
3.5. DESENVOLVIMENTO DA INTERFACE <i>WEB</i>	30
3.5.1. Cadastro de Alunos	31
3.5.2. Cadastro de equipamentos	32
3.5.3. Página de empréstimos	33
3.5.4. Relatórios	34

4. RESULTADOS E DISCUSSÕES.....	36
4.1. MÓDULO LEITOR DE CÓDIGO DE BARRAS	36
4.2. BANCO DE DADOS	38
4.3. INTERFACE <i>WEB</i>	40
4.3.1. Página de login.....	40
4.3.2. Cadastro de alunos	42
4.3.3. Cadastro de equipamentos	44
4.3.4. Empréstimo e devolução.....	45
4.3.5. Relatórios	49
5. CONCLUSÃO.....	55
REFERÊNCIAS.....	57
Anexo 1.....	58
Anexo 2.....	67
Anexo 3.....	70

1. INTRODUÇÃO

1.1. Tema

O departamento de Engenharia Eletrônica da UTFPR de Campo Mourão-PR possui diversos equipamentos, os quais podem ser utilizados durante as aulas práticas e também podem ser emprestados para os alunos em períodos fora do horário das aulas para que sejam desenvolvidos projetos, trabalhos de conclusão de curso (TCC), projetos integradores (PI), etc. Esses equipamentos geralmente possuem um custo elevado, portanto, é preciso ter muito cuidado em relação a eles.

A utilização dos equipamentos é de extrema importância no desenvolvimento acadêmico dos alunos. Segundo Varela (2009), a educação não se resume apenas à teoria apresentada em sala de aula, ela também está diretamente relacionada aos recursos que são fornecidos para que os alunos possam desenvolver o seu potencial. A utilização dos equipamentos não deve ser restrita apenas às aulas práticas acompanhadas dos professores, mas para isso é preciso ter um controle sobre o empréstimo desses equipamentos.

Os equipamentos disponíveis ficam armazenados no almoxarifado do departamento e podem ser emprestados aos alunos mediante a apresentação do registro acadêmico (RA). Esse processo de empréstimo de equipamentos é controlado manualmente apenas por uma ficha e isso pode acabar gerando uma confusão nas anotações, por isso é necessário que seja feito um controle mais aprimorado desses empréstimos.

O controle manual consiste em anotar o RA do aluno e o patrimônio do equipamento, assim como a data da retirada e posteriormente a data da devolução. No entanto, o sistema manual as vezes acaba dando brechas para alguns problemas, como a anotação errada dos dados, a não anotação de uma devolução, entre outros. Além disso, o sistema manual não guarda um histórico do empréstimo dos equipamentos e mesmo que venha a guardar, o acesso aos dados não é tão fácil quanto num sistema automatizado.

Um sistema automatizado faria todo esse processo de anotações de maneira informatizada, guardando os dados do aluno, do equipamento e das datas em um banco de dados, o que reduziria praticamente por completo os problemas causados pela operação manual.

Além disso, o sistema guardaria um histórico de todos os empréstimos realizados para que isso possa ser rapidamente acessado caso seja necessário.

A automatização dos empréstimos pode ser feita com algum sistema já existente, como o *Aleph*, o *Scobi*, o *Pergamum*, entre outros, mas esses sistemas são desenvolvidos para o gerenciamento de bibliotecas, o que faz com que eles possuam funções que não são necessárias no caso do controle de empréstimo de equipamentos. Além dessa questão, esses sistemas geralmente possuem um alto custo de implantação e em alguns casos exigem *hardwares* adicionais, o que eleva ainda mais o custo (RODRIGUES E PRUDENCIO, 2009).

A ideia deste trabalho consiste no desenvolvimento de um sistema de baixo custo que possua as características necessárias para a automatização do processo de empréstimo de equipamentos eletrônicos pertencentes ao DAELN-CM.

1.2. Justificativa

Pelo fato do DAELN possuir equipamentos de alto custo, é necessário que seja feito um controle minimamente estruturado dos empréstimos. Para se ter esse controle, os dados do equipamento e do aluno que solicitou, devem ser armazenados de maneira correta para que caso ocorra algum problema com o equipamento, possam ser tomadas as atitudes necessárias para a resolução do mesmo.

O controle manual desses empréstimos se mostrou eficaz até certo ponto, porém pode ser que apresente alguns erros. Com um sistema automatizado se ganha agilidade, maior segurança nos dados do empréstimo e ainda mantém um histórico sempre atualizado dos empréstimos de cada equipamento.

A implementação de um sistema eletrônico para fazer o controle dos empréstimos é necessária para que os dados sejam registrados de maneira correta, e isso acabará evitando transtornos no futuro. A manutenção de um histórico dos empréstimos é importante para evitar o uso excessivo de um mesmo equipamento, ou seja, não emprestar sempre o mesmo equipamento para evitar o seu desgaste por uso excessivo.

Como já foi falado anteriormente, até existem sistemas que podem ser utilizados nessa aplicação, mas eles não são desenvolvidos especialmente para isso, possuindo funções excedentes e custo elevado. Dessa forma, foi desenvolvido um sistema para automatizar o processo de empréstimos dos equipamentos e armazenar o histórico dos mesmos.

1.3. Objetivos

O objetivo geral deste trabalho é o desenvolvimento de um sistema para controlar o empréstimo de equipamentos do DAELN-CM. Este sistema obtém, de maneira informatizada, os dados do equipamento (patrimônio) e do aluno (RA), além de manter um histórico atualizado dos empréstimos de cada equipamento.

1.3.1. Objetivos específicos

Os objetivos específicos desse projeto foram divididos em alguns tópicos, os quais estão listados nos tópicos a seguir:

- Desenvolver um *software* na linguagem C para fazer a leitura dos códigos de barra do RA e do patrimônio do equipamento por meio de um leitor conectado a *Raspberry Pi*;
- Modelar um sistema de banco de dados, o qual irá armazenar os dados dos empréstimos;
- Desenvolver uma interface *web* para fazer o controle dos empréstimos de maneira automatizada. Através dessa interface, o usuário poderá controlar todo o sistema e terá acesso a todos os relatórios desejados;
- Desenvolver uma forma de integração entre a interface *web* e o *software* que faz as leituras dos códigos de barra;
- Fazer as verificações necessárias no sistema e por fim validar o seu funcionamento;
- Com o sistema funcionando, desenvolver um *script* para a sua instalação;
- Desenvolver um manual de operação do sistema para fornecer para o usuário.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. Automatização de processos

Automatizar um processo consiste em defini-los e otimizá-los para em seguida executá-los sobre uma plataforma informatizada (CAPIOTTI, 2012). Para a automação é necessário inicialmente elaborar a chamada arquitetura de processos. Essa arquitetura nada mais é do que a estrutura geral de um sistema de processos combinada com o conceito de automatização aplicável a ela (DAWIS, 2001).

No meio industrial, a preocupação com produtividade, redução do risco operacional e qualidade, leva à implantação de sistemas de automatização. Esses sistemas visam melhorar os processos industriais e também auxiliam na identificação de indicadores de desempenho do processo, o que permite aperfeiçoamento constante das atividades dos processos (SGANDERLA, 2013).

2.2. Sistemas embarcados

Um sistema embarcado consiste em uma união de *hardware*, *software* e em alguns casos, elementos mecânicos, que são utilizados para realizar uma determinada tarefa (STADZISZ; RENAUX, 2008). Com o advento da eletrônica e a redução dos custos, a utilização dos sistemas embarcados aumentou significativamente nos últimos anos. Para se ter uma noção, 99% de toda a produção mundial de microprocessadores é utilizada em sistemas embarcados e apenas 1% é destinada a computadores (BARR; MASSA, 1999).

Sistemas embarcados podem ser utilizados em diversas aplicações. Isso é possível devido à enorme variedade de processadores disponíveis no mercado, o que leva ao desenvolvimento de vários sistemas, os quais vão desde uma simples escova de dentes eletrônica, até o painel de controle de um avião. Apesar dessa variedade, o desenvolvimento de sistemas embarcados pode se tornar complicado devido às restrições de cada um.

Essas restrições, de acordo com Stadzisz e Renaux (2008) implicam numa grande diferença entre os sistemas embarcados e os computadores convencionais. Dentre elas pode-se citar as chamadas restrições dimensionais, as quais envolvem tamanho e peso, sendo de extrema

importância em equipamentos de pequeno porte como um celular. Uma outra restrição é a de consumo de energia, a qual é de extrema importância em equipamentos móveis e que são alimentados por meio de baterias, como o caso de um aparelho de GPS. As restrições de recursos, como memória e processamento geram um impacto na construção do *software*, ou seja, é preciso ter um *software* eficiente para que o funcionamento do sistema não apresente problemas. Uma outra restrição que pode ser citada é a de tempo de execução. Ela é pertinente pois existem diversas aplicações que precisam ser executadas em um limite de tempo bem específico.

O projeto de um sistema embarcado é extremamente ligado à sua aplicação, portanto não é possível descrever uma rotina genérica para o desenvolvimento de um projeto. Uma opção muito usada por desenvolvedores é a utilização de placas de desenvolvimento, como a *Raspberry Pi*, a *Beaglebone*, o *Arduino*, etc. Essas placas possuem vários periféricos e são utilizadas para facilitar e agilizar o processo de prototipação. Posteriormente é possível otimizar os recursos e dispensar os periféricos desnecessários e projetar um *hardware* específico para a aplicação desejada.

2.3. *Raspberry PI*

A *Raspberry PI* é um projeto desenvolvido por um grupo de estudos do Reino Unido que foi iniciado com o objetivo de incentivar o ensino de programação nas escolas de ensino básico (ELINUX, 2017b). Na Figura 1 é possível visualizar uma *Raspberry PI Model B*.



Figura 1 – Raspberry PI Model B.
Fonte: Autoria própria.

Atualmente existem três modelos de *Raspberry Pi* no mercado, a *Raspberry Pi zero*, a *Raspberry Pi model A* e a *Raspberry Pi model B/B+*. Esses modelos possuem características particulares, mas todos são baseados na arquitetura ARMv6. A diferença está em detalhes como a quantidade de portas de entrada e saída (*GPIO*), memória RAM, periféricos de rede, vídeo, portas USB, etc. O Quadro 1 traz um resumo das especificações de cada um dos modelos citados anteriormente.

Quadro 1 – Comparativo dos modelos de Raspberry Pi

Modelo	Raspberry Pi 3	Raspberry Pi Zero W	Raspberry Pi Zero	Raspberry Pi 2	Raspberry Pi Model B	Raspberry Pi Model A+
Processador	Cortex-A53 64-bit 1.2GHz	ARM1176JZ F-S 1GHz	ARM1176JZ F-S 1GHz	Cortex-A7 900MHz	ARM1176JZ F-S 700MHz	ARM1176JZ F-S 700MHz
RAM	1GB	512MB	512MB	1GB	512MB (256 Model A)	256MB
GPIO	40	40	40	40	26	40
Portas USB	4	Micro & micro OTG	Micro & micro OTG	4	2	1
Ethernet	Sim	Não	Não	Sim	Sim	Não
WI-FI	Sim	Sim	Não	Não	Não	Não
Bluetooth	Sim	Sim	Não	Não	Não	Não

Fonte: Adaptado de Raspberry Pi Foundation (2014).

Por ser baseado na arquitetura ARM, é possível utilizar qualquer linguagem de programação compilável nessa arquitetura para o desenvolvimento de *softwares* e isso faz com que a *Raspberry* seja utilizada em diversas aplicações, tais como projetos eletrônicos, servidores *web* e até mesmo como central multimídia (RASPBerry PI FOUNDATION, 2014).

2.4. Aplicação WEB

Uma aplicação *web* pode ser definida como uma aplicação que é acessada através do navegador (NATIONS, 2016). Ela executa tarefas em um servidor e faz a interface com o usuário através de uma página *web* (PALMEIRA, 2013). Essas aplicações são desenvolvidas com o auxílio de diversas tecnologias, como linguagens de programação (*PHP*, *Javascript*, etc), elementos de interface gráfica (HTML, CSS) e também técnicas como o AJAX (*Asynchronous JavaScript And XML*).

As aplicações *web* se diferenciam das aplicações ‘*desktop*’ pois não é necessário que se desenvolva um programa específico para cada *hardware* ou sistema operacional, basta acessá-las via navegador *web*. Isso é possível pois todo o processamento de funções e instruções é feito no servidor *web* e o navegador funciona apenas como uma ‘interface’ da aplicação (NATIONS, 2016).

O funcionamento de uma aplicação *web* pode ser melhor compreendido na Figura 2, nela é possível observar um exemplo básico das etapas de execução de uma aplicação *Web*.

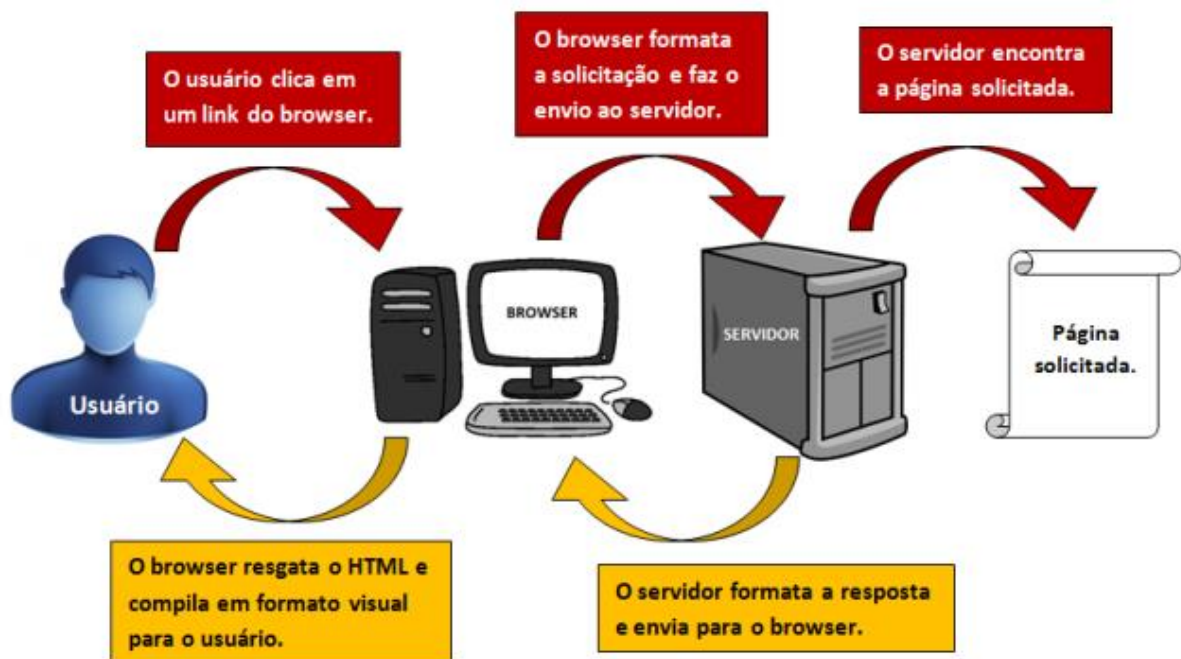


Figura 2 – Etapas de funcionamento de uma aplicação Web.
Fonte: Palmeira (2013).

2.4.1. HTML

O HTML (*HyperText Markup Language*) é uma linguagem de marcação para descrever documentos *web* (W3Schools, 2017). Ela é utilizada para descrever a estrutura e o conteúdo semântico de uma página *web*, sendo que praticamente todas elas são descritas por essa linguagem, portanto quase tudo que podemos visualizar via navegador *web* é feito em HTML (MOZILA DEVELOPER NETWORK, 2017).

A linguagem HTML é um padrão internacional que possui especificações definidas pela W3C (*World Wide Web Consortium*) e pela WHATWG (*Web Hypertext Application*

Technology Working Group). O padrão atual do HTML é o HTML5 que teve sua versão publicada no ano de 2008 pela W3C.

O padrão HTML é utilizado para descrição do conteúdo e do seu significado. A formatação e o estilo em que esse conteúdo é exibido é definido pelas folhas de estilo de cascadeamento de utilização (CSS). Um exemplo de descrição HTML, juntamente com a sua exibição no navegador pode ser visualizada na Figura 3.



Figura 3 – Exemplo de descrição HTML.
Fonte: Adaptado de escolaw3 (2017).

2.4.2. PHP

PHP, ou *Hypertext Preprocessor*, é uma linguagem de *script open source* de uso geral. Ela foi desenvolvida especialmente para ser utilizada em aplicações *web* e pode ser combinada com HTML (PHP.NET, 2017).

Os scripts do PHP são enviados para um servidor via navegador *web*. Esse servidor interpreta os códigos, envia uma resposta para o navegador que transforma essa resposta em uma página *web* que possa ser interpretada pelo usuário (PHP.NET, 2017).

Por ser interpretado por um servidor, o PHP pode ser utilizado nas mais diversas plataformas e sistemas operacionais, como UNIX, *Windows*, *MAC OS*, *Risc OS*, entre outros. O PHP também é compatível com diversos servidores *web*, dentre os principais é possível destacar o Apache e o ISS.

A linguagem PHP não é limitada apenas a geração de arquivos HTML, ela também pode gerar arquivos PDF e *Flash* (com o auxílio de extensões). É possível também gerar outros

arquivos de texto como XML e armazená-los no sistema de arquivos, o que permite a criação de um *cache* do sistema no lado do servidor.

Uma das características mais marcantes da linguagem PHP é o seu suporte a diversos sistemas de banco de dados, como MySQL, SQLite, PostgreSQL, etc. No PHP é extremamente simples fazer operações que envolvem um banco de dados. Essas operações são executadas através de extensões específicas, através uma camada de abstração como o PDO (*PHP Data Objects*), ou se conectando a qualquer banco de dados que suporte o padrão “*Open Database Connection*” usando a extensão ODBC (*Open Database Connectivity*).

O PHP possui suporte para diversos protocolos de comunicação como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM e vários outros. Também é possível fazer o uso de *sockets* de rede para interação entre quaisquer protocolos. Ele também suporta a troca de dados complexos WDDX (*Web Distributed Data eXchange*), que é utilizada virtualmente em todas as linguagens de programação para *web*.

Devido às suas características e relativa simplicidade, o PHP é utilizado por muitos desenvolvedores, desde iniciantes, até mesmo profissionais com experiência, portanto é comum encontrar diversas páginas *web* que foram desenvolvidas com o PHP (PHP.NET, 2017).

2.4.3. AJAX

O AJAX, ou *Asynchronous JavaScript And XML*, não é uma linguagem de programação e sim uma combinação de linguagens e técnicas de desenvolvimento *web* como *Javascript*, *XML*, *CSS* e *DOM*. Ele é utilizado para tornar as páginas *web* mais responsivas e permite que sejam feitas atualizações em partes da página sem que haja a necessidade de uma atualização completa da página (W3SCHOOLS, 2017).

O funcionamento do AJAX se resume à utilização da função *XMLHttpRequest*, que é nativa dos navegadores *web* que faz a solicitação de informações para o servidor, e a funções *Javascript* e HTML DOM, que exibem no navegador as informações obtidas (W3SCHOOLS, 2017).

Para compreender melhor o funcionamento do AJAX, pode-se visualizar o diagrama da Figura 4.

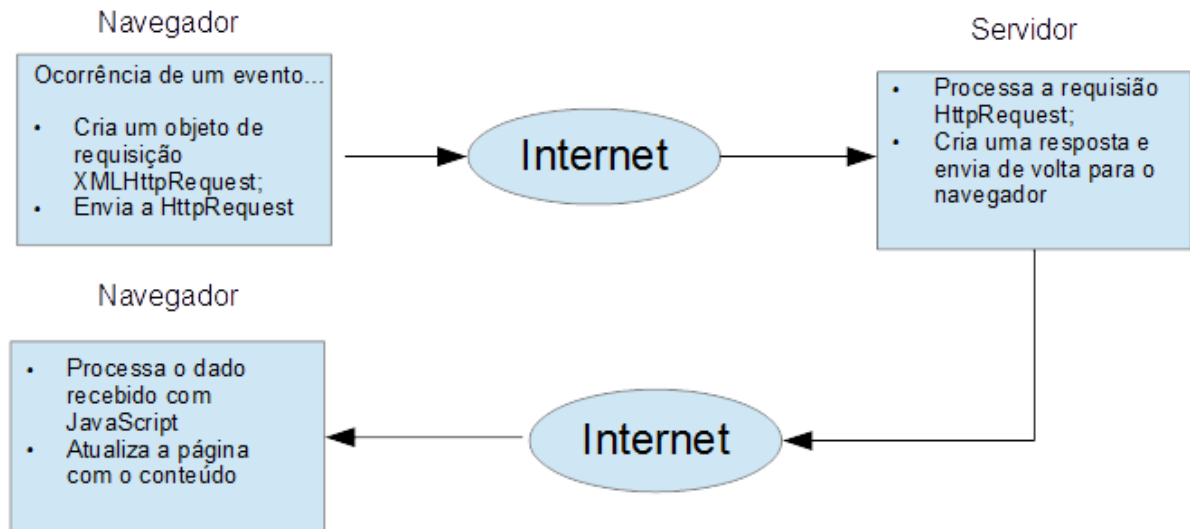


Figura 4 – Diagrama de funcionamento do AJAX.
Fonte: Adaptado de w3schools (2017).

2.5. Banco de dados

Um banco de dados pode ser definido como um conjunto de dados inter-relacionados que representam informações sobre um domínio específico, ou seja, sempre que for possível agrupar informações que se relacionam, existe um banco de dados (KORTH, 1994). Como exemplo de banco de dados podemos citar um sistema de bibliotecas, uma agenda telefônica, um cadastro de clientes, etc.

Para fazer o gerenciamento do banco de dados, são utilizados os sistemas de gerenciamento de banco de dados (SGBD). Esses sistemas nada mais são do que um *software* que é capaz de manipular as informações do banco de dados e fazer a interação com o usuário. Existem vários SGBD's no mercado, como o MySQL, o SQL *Server*, o PostgreSQL, entre outros (REZENDE, 2006).

Um sistema de banco de dados pode ser definido como um conjunto de quatro componentes básicos: os dados, o *hardware*, o *software* e os usuários. Para compreender melhor esses componentes, apresenta-se o diagrama da Figura 5.

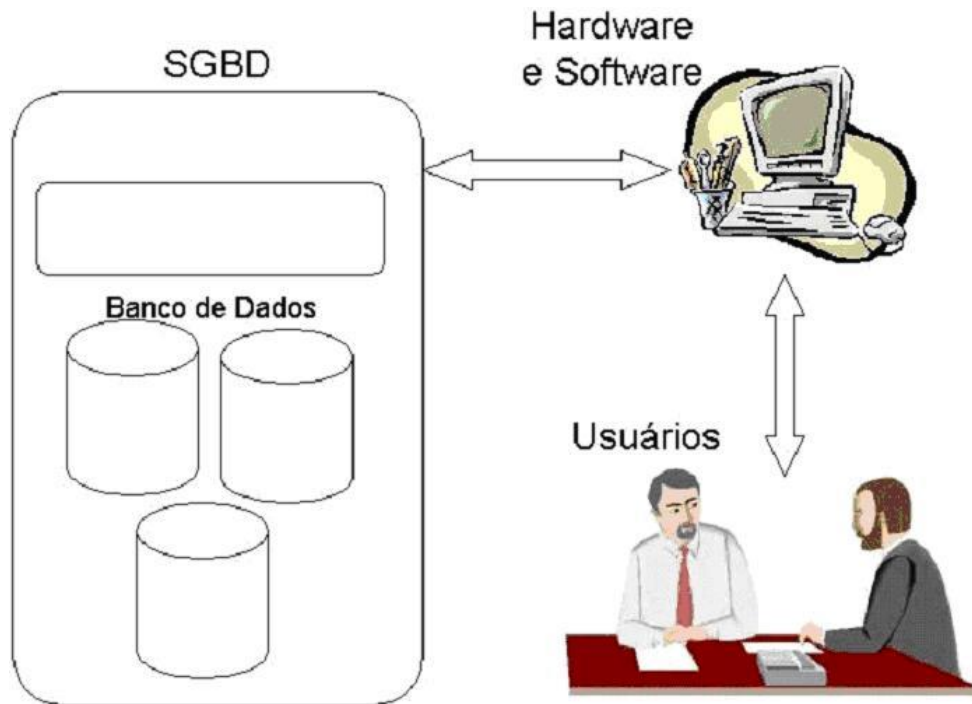


Figura 5 – Componentes de um sistema de banco de dados.
Fonte: devmedia (2017).

Os sistemas de banco de dados possuem certas vantagens em relação ao sistema tradicional de armazenamento de arquivos. Eles são implementados com a função de isolar os detalhes internos do banco de dados do usuário, ou seja, promover a abstração desses dados e também de permitir que exista uma independência dos dados em relação às aplicações que os acessam. Outro fator importante é a questão da segurança e integridade dos dados, pois geralmente eles são criptografados e não podem ser acessados tão facilmente. No entanto, a implantação de um sistema de banco de dados é mais cara e nem sempre é necessário que ele seja utilizado.

Para realizar consultas, inserção, edição e relacionar os dados armazenados no banco de dados, utiliza-se uma linguagem baseada em consultas estruturadas chamada SQL (*Structured Query Language*).

2.5.1. SQL

A linguagem SQL foi criada por E. F. Codd em meados dos anos 70 em um laboratório de pesquisas da IBM. Ao final dos anos 70 foi lançado um sistema baseado em SQL. O sucesso

dele foi tão grande, que a ANSI (*American National Standards Institute*) foi obrigada a criar uma padronização para as implementações da linguagem.

2.5.2. MySQL

O MySQL é um SGBD que utiliza a linguagem SQL para fazer a interface com o banco de dados. Esse sistema possui diversas características que fazem dele um dos sistemas mais populares da atualidade.

Um dos fatores que contribui para essa popularidade é a sua compatibilidade entre diversos sistemas, como *Windows, Unix, Linux e Mac OS*. Isso é possível por ele ter sido desenvolvido nas linguagens C e C++. Ele também é portátil entre as mais variadas linguagens de programação e compiladores. Existem interfaces que facilitam o desenvolvimento de sistemas em linguagens como PHP, *Java, Python, Ruby*, etc (TEIXEIRA, 2017).

Além da sua compatibilidade, o MySQL é um software *Open Source*, ou seja, tem seu código fonte disponibilizado para que o usuário faça as alterações que julgar necessárias para sua aplicação (ORACLE, 2017). A distribuição do MySQL é feita com a licença GNU/GLP (*General Public License*), que permite a sua utilização por qualquer usuário, e que sua distribuição seja livre (GNU, 2017).

O armazenamento de dados no MySQL pode ser feito de diversas maneiras, sendo possível adaptar o SGBD para que ele se comporte da melhor forma possível para a aplicação desejada. Por exemplo, existem sistemas onde a prioridade é a velocidade e não a quantidade de dados, já em outros é preciso priorizar a quantidade de dados, o MySQL permite que as duas situações sejam implementadas (TEIXEIRA, 2017).

Essas características fazem do MySQL um sistema extremamente confiável o que faz com que ele seja um dos mais difundidos na atualidade, estima-se que ele possua cerca de 10 milhões de instalações pelo mundo, dentre as quais pode-se citar a NASA, o Banco Bradesco, a HP, o exército dos Estados Unidos, etc (WELLING, THOMSON, FURMANKIEWICZ; 2005).

3. MATERIAIS E MÉTODOS

3.1. Metodologia

O desenvolvimento do trabalho foi dividido em etapas para que o sistema fosse desenvolvido de maneira modular. Esta forma de desenvolvimento teve como objetivo a redução de falhas no processo de programação. As dez etapas do desenvolvimento serão listadas a seguir.

Etapa 1 - Definição das características do sistema:

Inicialmente foram definidos alguns requisitos e características do sistema, dessa forma, os módulos que o compõem foram projetados e a sua estrutura pode ser melhor compreendida no diagrama da Figura 6.

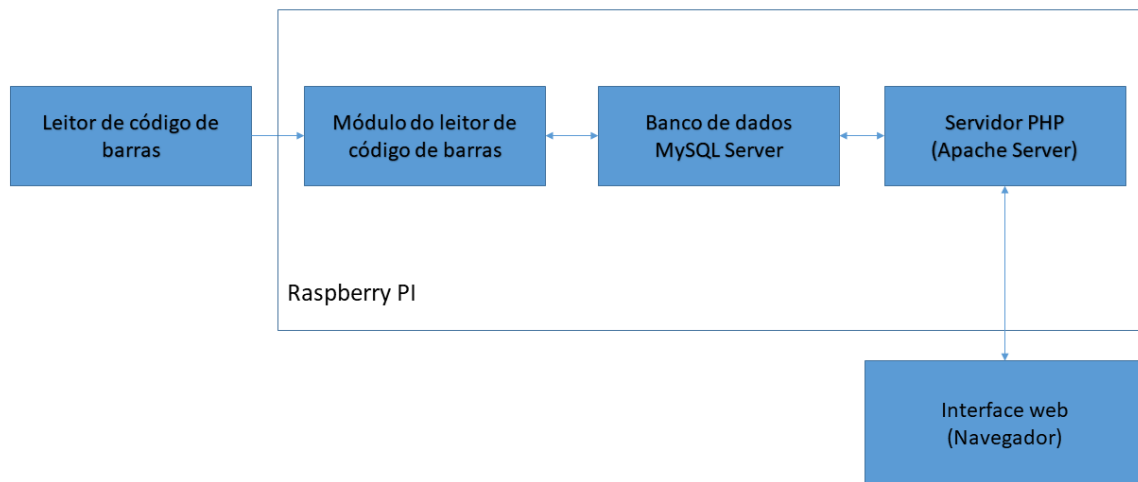


Figura 6 - Diagrama de blocos do sistema

Fonte: Autoria própria.

Etapa 2 – Instalação do sistema operacional

Antes de iniciar o desenvolvimento do sistema, foi instalado na *Raspberry PI* o sistema operacional *Raspbian Jessie*. Esse sistema é uma versão do *Linux* que foi desenvolvida com base no *Debian* e é exclusiva para a plataforma da *Raspberry PI*.

Etapa 3 – Primeira fase do desenvolvimento do módulo leitor de código de barras

Com o hardware da *Raspberry Pi* configurado, foi iniciado o desenvolvimento do primeiro módulo do sistema. Esse módulo foi o de leitura de código de barras, que foi desenvolvido na linguagem C e foi utilizado, inicialmente, apenas para fazer a leitura do código de barras.

Etapa 4 – Modelamento do banco de dados do sistema

Em um segundo momento foi esquematizado o banco de dados do sistema, projetado para armazenar todos os dados referentes aos alunos, equipamentos e também aos empréstimos realizados.

Etapa 5 – Desenvolvimento da interface web

Com o banco de dados modelado, iniciou-se o desenvolvimento da interface *web* do sistema. Para esse desenvolvimento, primeiramente foi feito um estudo da linguagem PHP e de AJAX, que é uma combinação de diferentes linguagens e scripts de desenvolvimento *web*. A interface *web* foi desenvolvida de maneira simplificada apenas para que fossem realizados testes preliminares. Posteriormente essa interface foi melhorada para que o sistema ficasse mais apresentável para o usuário final.

Etapa 6 – Instalação e configuração dos servidores web e banco de dados

A interface *web* precisa de um servidor *web* para ser acessada, e o banco de dados também precisa de um servidor para armazenamento dos dados. Estas funções foram configuradas na própria *Raspberry Pi*. Para isso, foram instalados os programas necessários para exercer essas funcionalidades: *Apache* e *MySQL Server*.

Etapa 7 – Segunda fase do desenvolvimento do módulo leitor de código de barras

Após a configuração dos servidores *web* e de banco de dados, iniciou-se o desenvolvimento de uma integração entre o módulo que faz as leituras dos códigos de barra e a interface *web*. O módulo das leituras foi modificado nessa etapa, onde foram adicionados códigos para que ele se comunique com a interface *web*, fazendo com que o sistema funcione corretamente.

Etapa 8 – Testes e validações

Com o sistema em funcionamento, foram realizadas as verificações e os testes necessários para comprovar o seu funcionamento de maneira estável. Com isso, o sistema foi validado e dado como pronto para ser utilizado no controle dos empréstimos dos equipamentos.

Etapa 9 – Criação do Shell Script

Após a finalização do sistema, foi criado um *shell script*, o qual servirá para instalações futuras do sistema. Uma vez baixado e executado, esse arquivo faz com que a *Raspberry Pi* fique pronta para ser utilizada apenas com o sistema de controle de empréstimos.

Etapa 10 – Desenvolvimento do manual de instruções

A última etapa do desenvolvimento foi uma etapa de extrema importância. Nessa etapa foi desenvolvido um manual de instruções para o usuário, o qual explica todas as funcionalidades do sistema e também ensina como fazer o uso delas. O manual pode ser visualizado no anexo 1 deste trabalho.

Materiais utilizados

Para o desenvolvimento do sistema foram utilizadas algumas ferramentas de *hardware* e *software*, as quais estão listadas nos tópicos a seguir:

- *Raspberry PI*;
- Leitor de código de barras *Bematech*;
- *Apache server*;
- *MySQL server*.

3.2. Instalação do sistema operacional (SO) *Jessie Lite*

Para que o sistema operacional funcione na *Raspberry PI* é necessário que ele esteja instalado em um cartão SD. Essa instalação foi feita com o auxílio do software *Win32 Disk Imager*, que é um software que faz a instalação da imagem do sistema no cartão SD. A Figura 7 mostra a interface do programa utilizado.

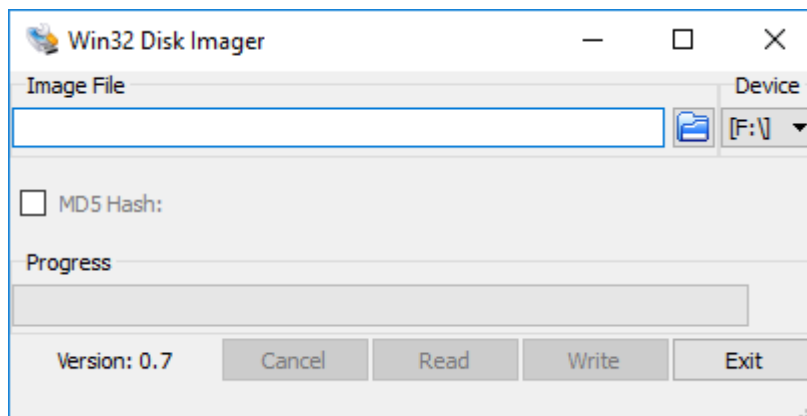


Figura 7 – Interface do programa *Win32 Disk Imager*.
Fonte: Autoria própria.

3.3. Desenvolvimento do módulo leitor de código de barras

O módulo leitor de código de barras foi desenvolvido em linguagem C. Inicialmente esse módulo apenas fazia a leitura dos códigos de barra, pois ele foi desenvolvido com o intuito de testar os leitores de código de barras disponíveis.

Posteriormente, esse módulo foi modificado para se comunicar com a interface *web* e com o banco de dados. O fluxograma da Figura 8 explica de forma simplificada o funcionamento desse módulo. O código fonte está no anexo 3 deste trabalho.

O *software* desenvolvido para esse módulo opera fazendo a verificação de uma tabela do banco de dados, a tabela interface. Essa tabela é alimentada pela interface *web* e nela está definida em qual página o sistema se encontra no momento.

Quando o software identifica a página em que a interface está, ele aguarda que o código de barras seja lido por meio do leitor e o guarda na tabela auxDados. Esse ciclo é repetido infinitamente e é essencial para o funcionamento do sistema.

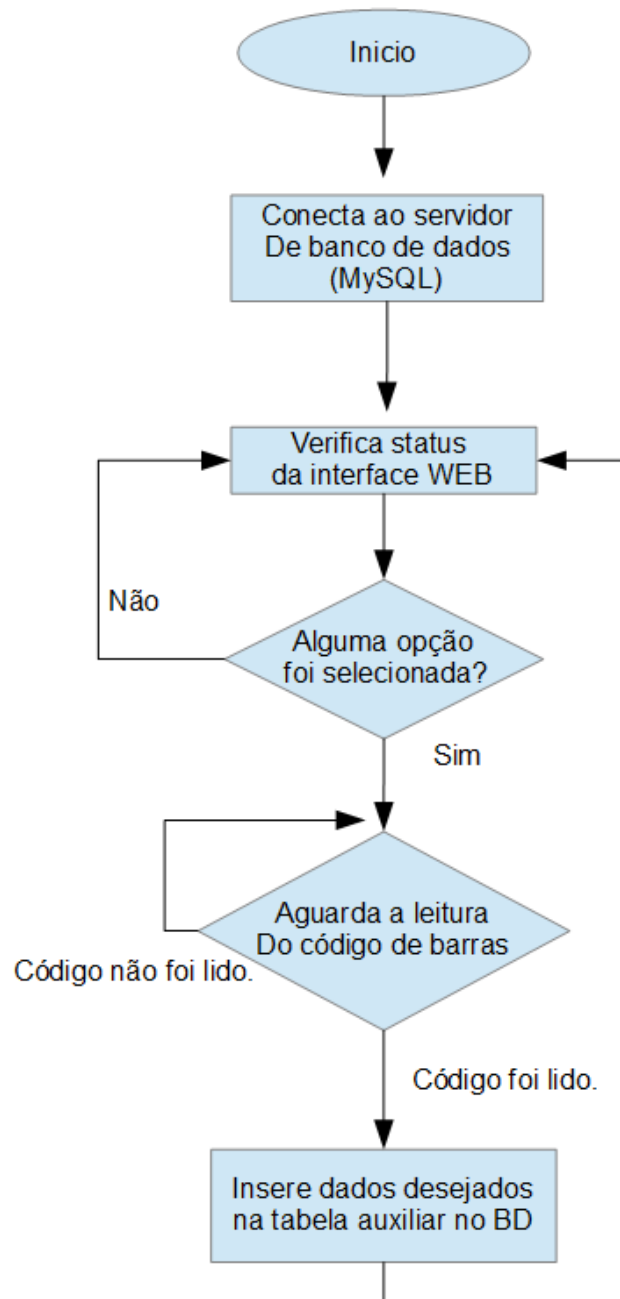


Figura 8 - Fluxograma do modulo leitor de código de barras.
Fonte: Autoria própria.

Para a leitura do código de barras é utilizado um *hardware* específico para essa função como os da Figura 9. Testes foram executados nesse *hardware* e constatou-se que o seu método de leitura é semelhante ao de um teclado, ou seja, o *hardware* faz a leitura do código de barras

e retorna em um formato que o sistema entende como se ele tivesse sido digitado através de um teclado tradicional.



Figura 9 – Leitores de código de barra.
Fonte: Autorial própria.

A leitura de caracteres oriundos de um teclado tradicional é feita através de uma função da linguagem C que se encontra na biblioteca *stdio.h*. Essa função, chamada *scanf*, faz a leitura de dados enviados por uma fonte externa, como um teclado por exemplo. Essa função pode efetuar a leitura de vários tipos de dados, no caso desse módulo o tipo de dado é o *double*, esse tipo foi escolhido devido ao tamanho do código de barras. Essa função possui algumas particularidades, ela recebe parâmetros do tipo de dados e um endereço onde deve guardar os dados lidos da entrada de texto.

Para fazer a manipulação e as consultas no banco de dados são utilizadas funções da biblioteca *mysql.h*. As funções dessa biblioteca permitem consultar determinada base de dados e incluir dados nela. Essa biblioteca é extremamente importante nessa aplicação, pois esse módulo é baseado em consultar uma tabela e incluir valores em outra. Essa comunicação precisa estar funcionando corretamente pois ela insere na interface *web* os dados que alimentam o sistema, que vão desde os cadastros de alunos e equipamentos, até a geração dos relatórios específicos.

Para verificar o funcionamento do módulo de leitura, foram feitos alguns testes com dois leitores de código de barras diferentes, um BEMATECH e um *Datamax*, utilizando alguns RA's e alguns códigos de patrimônio. O funcionamento do módulo foi verificado por um teste

na página de empréstimos. Esse teste está descrito detalhadamente na seção de resultados e discussões.

3.4. Projeto do banco de dados do sistema

O banco de dados do sistema foi projetado sob a plataforma do SGBD MySQL e foi pensado para armazenar dados dos alunos, equipamentos e empréstimos destes. Esse banco de dados também possui algumas tabelas auxiliares que são utilizadas para fazer a integração entre o módulo leitor de código de barras, que foi desenvolvido em linguagem C, com a interface *web*. As tabelas do banco de dados podem ser visualizadas na Figura 10.

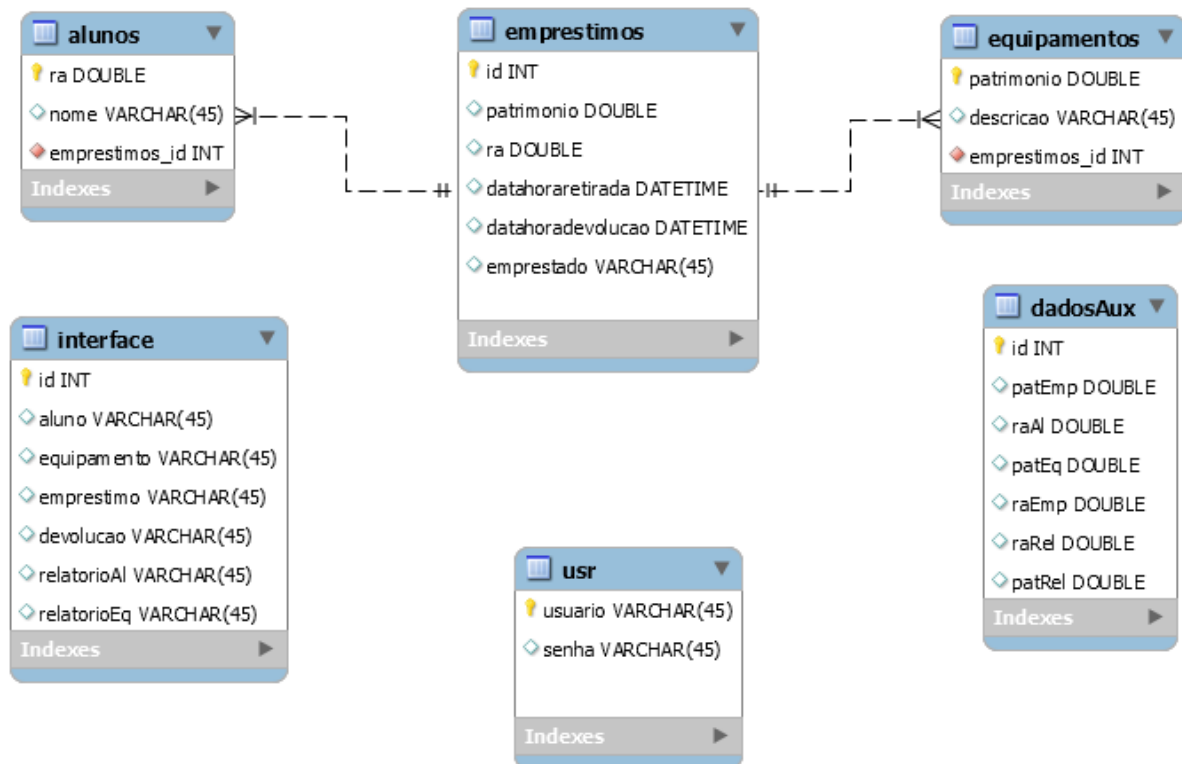


Figura 10 – Diagrama das tabelas do banco de dados.
Fonte: Autoria própria.

Essas tabelas foram criadas através de comandos SQL no terminal do MySQL instalado no *Raspbian*. Os comandos de criação encontram-se no anexo 2 deste trabalho.

A tabela *alunos* armazena os dados dos alunos autorizados a fazer empréstimos de equipamentos, a tabela de *equipamentos* guarda os dados dos equipamentos disponíveis para

empréstimo. Essas tabelas alimentam a tabela *empréstimos*, sendo que um aluno pode emprestar vários equipamentos ao mesmo tempo, e um equipamento só pode ser emprestado para um aluno. A tabela *empréstimos* armazena um histórico detalhado de cada um dos empréstimos e isso é extremamente útil na hora da geração dos relatórios.

As tabelas *interface* e *dadosAux* são utilizadas para fazer a integração entre o módulo leitor de código de barras e a interface *web* do sistema.

Por último e talvez uma das mais importantes, a tabela *usr* é a tabela onde estão cadastrados os usuários do sistema. Essa tabela, apesar de simples, permitiu a implementação de um controle de acessos para o sistema.

Para demonstrar o funcionamento do banco de dados, foram feitas algumas consultas diretamente no terminal do MySQL. Essas consultas foram realizadas em três tabelas, a de alunos, de equipamentos e de empréstimos. Foram inseridos alguns dados reais e outros fictícios no banco de dados a fim de provar o seu funcionamento.

3.5. Desenvolvimento da interface WEB

A interface *web* é onde o usuário faz a inserção dos dados e opera o sistema. Essa interface foi desenvolvida com o auxílio de linguagens de descrição como HTML e linguagens de programação como PHP, além do AJAX, que é uma combinação de linguagens e técnicas de desenvolvimento *web*.

Dentro da interface *web* existem páginas que fazem o cadastro de alunos, equipamentos e empréstimos, além de páginas que geram relatórios. Em cada uma delas existe um código PHP atrelado a uma página HTML que faz a interface com o usuário. Para a inserção de dados oriundos de códigos de barra, foram implementadas duas formas para a sua inserção, através de um leitor de código de barras, ou através da digitação por meio de um teclado comum.

Todas as páginas são protegidas por um sistema de controle de usuários. Este controle é feito através de um script PHP que faz uma consulta em um servidor e identifica se o usuário está ou não autorizado a acessá-lo. No diagrama da Figura 11 é possível observar a estrutura da interface *WEB*.

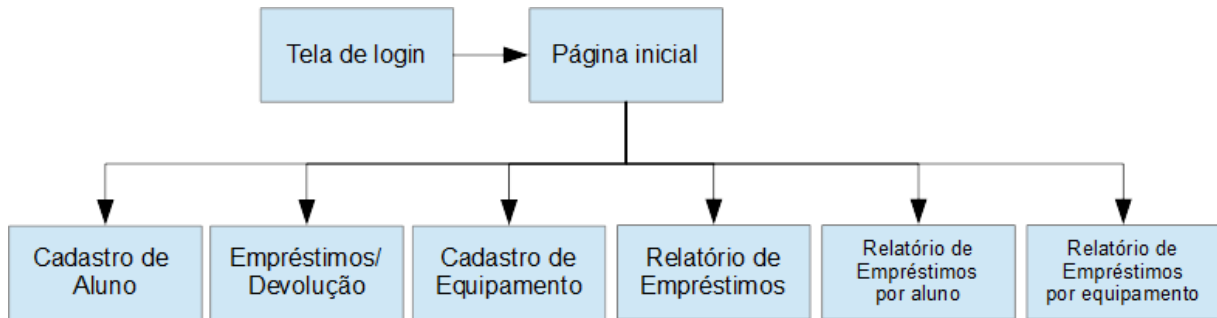


Figura 11 – Diagrama geral da interface web.
Fonte: Autoria própria.

3.5.1. Cadastro de Alunos

A página de cadastro de alunos foi desenvolvida para fazer a leitura dos dados do aluno (nome e RA) e armazená-los no banco de dados. Essa página é composta por um formulário HTML, que possui um *script* desenvolvido em AJAX que faz uma interface com o módulo de leitura dos códigos de barra. Os dados obtidos no formulário são enviados para o banco de dados através de um script PHP. O diagrama da Figura 12 mostra a estrutura da página e auxilia na compreensão do seu funcionamento.

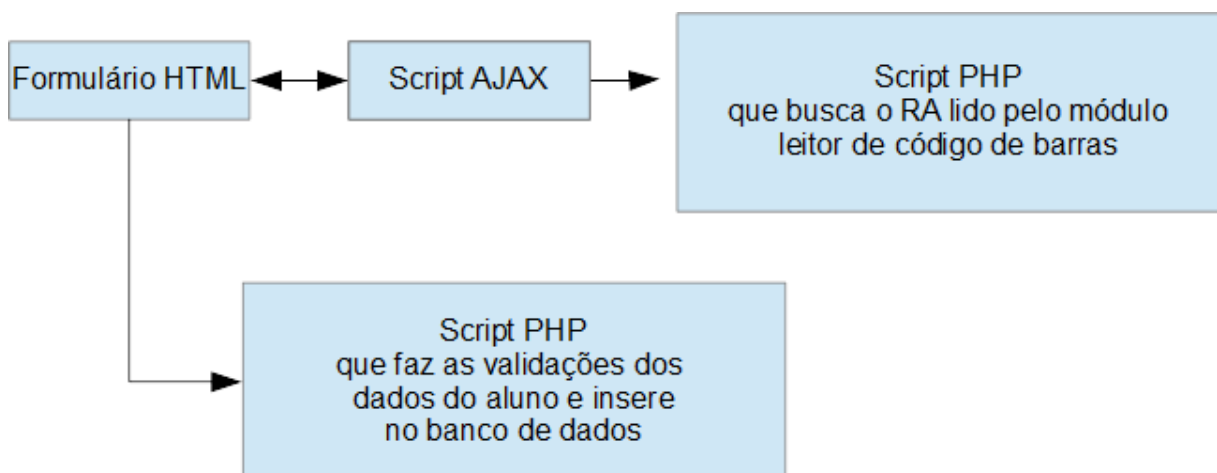


Figura 12 – Diagrama da página de cadastro de alunos.
Fonte: Autoria própria.

O formulário HTML do cadastro de alunos possui dois campos, sendo que um deles deve ser digitado e o outro é obtido através de um *script* AJAX e um PHP que fazem a busca na tabela *auxDados* retornando o RA que foi lido pelo leitor de código de barras.

Os dados são enviados para a tabela alunos do banco de dados também por meio de um *script* PHP. Antes de serem gravados, esses dados passam por algumas validações; caso algum deles esteja inconsistente, o sistema informa o usuário e ele deve reinserir os dados.

Para demonstrar as validações foram realizados alguns testes no sistema, nesses testes foram inseridos dados inválidos para que o sistema fizesse as validações e não permitisse a entrada dos dados errados no banco de dados.

3.5.2. Cadastro de equipamentos

O cadastro de equipamentos é uma página bem semelhante à página descrita na seção anterior, diferindo apenas pelo fato de que ela faz a leitura dos dados dos equipamentos de laboratório. Nessa página é obtida uma descrição do equipamento e o seu número de patrimônio que são armazenados no banco de dados. O seu diagrama estrutural está demonstrado na Figura 13.

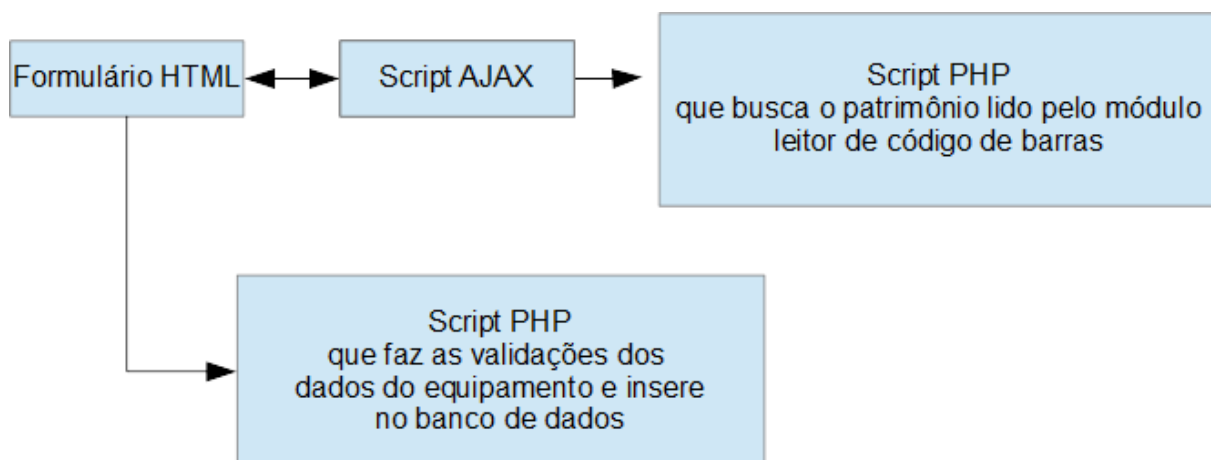


Figura 13 – Diagrama da página de cadastro de equipamentos.
Fonte: Autoria própria.

Existem dois campos no formulário de envio de dados do equipamento, assim como na página de aluno, um desses campos deve ser digitado e o outro é obtido pelos *scripts*. Essa página também possui um processo de validação que informa o usuário em caso de inconsistência dos dados.

Assim como no cadastro de alunos, foram realizados alguns testes para comprovar o funcionamento das validações da página de cadastro de equipamentos. Neles foram inseridos dados inconsistentes para que o sistema executasse as rotinas de validação.

3.5.3. Página de empréstimos

A página de empréstimos é onde o sistema exerce sua principal função, que é fazer o controle dos empréstimos dos equipamentos de maneira automatizada. A página foi projetada para fazer a captação dos dados do empréstimo com o auxílio de um *script* em AJAX e envia os dados através de um código em PHP. A devolução dos equipamentos também é feita nessa seção, apenas fazendo a seleção dessa opção.

As validações que garantem a integridade do sistema também são realizadas nessa página. Elas verificam diversas variáveis e só permitem o cadastro de um empréstimo se os dados passarem pelas validações, e isso evita problemas e falhas durante os processos de empréstimo e devolução. O diagrama da Figura 14 demonstra o funcionamento da página de empréstimos.

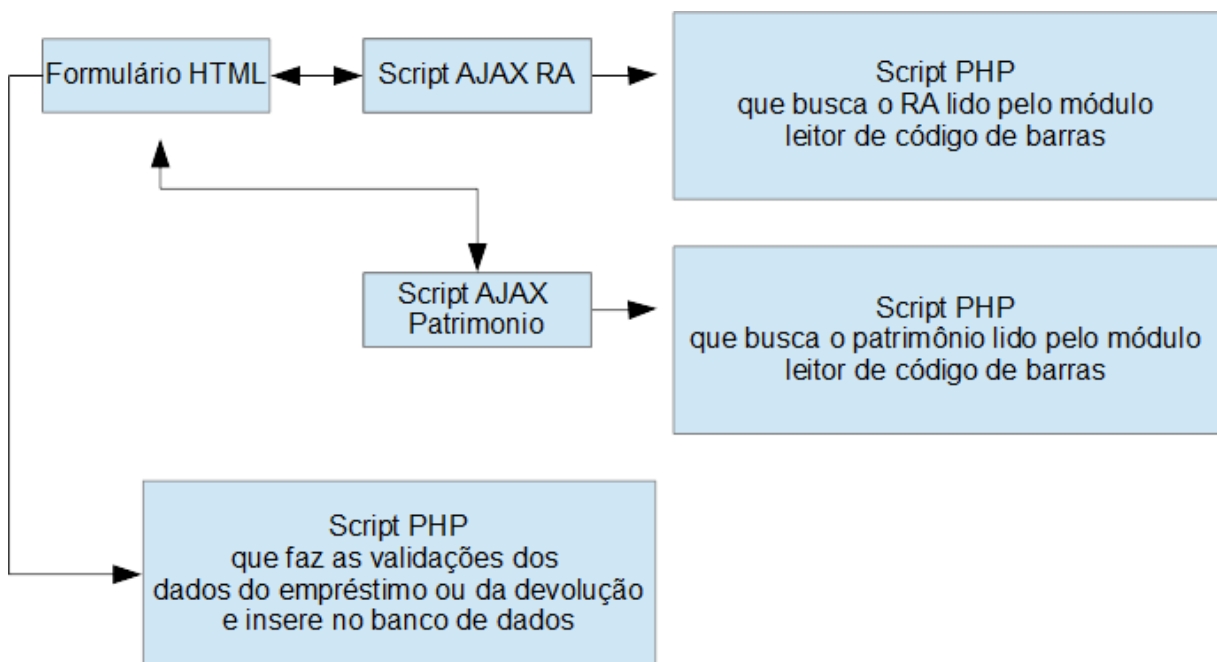


Figura 14 – Diagrama da página de empréstimos.
Fonte: Autoria própria.

O formulário HTML da página de empréstimos possui dois dados que são obtidos via *scripts* AJAX e PHP, e um dado onde é informado se é um empréstimo ou uma devolução.

Esses dados passam por validações no *script* PHP e são enviados para a tabela de empréstimos, em caso de empréstimo, ou apenas possuem seu campo atualizado (campo “emprestado” pode ser setado como “SIM” ou como “NÃO”) em caso de devolução. Essa página PHP também insere as datas de empréstimo ou devolução de maneira automática na tabela.

A validação da página de empréstimos foi feita através de alguns testes, eles foram realizados através da inserção de alguns dados. Esses dados foram validados ou não pelo sistema e inseridos no banco de dados. Os resultados obtidos estão descritos de maneira mais clara na seção de resultados.

3.5.4. Relatórios

Foram implementadas três páginas de relatório, uma delas que faz o relatório de todos os empréstimos já efetuados, outra que mostra os empréstimos realizados por determinado aluno e uma que mostra um histórico dos empréstimos de determinado equipamento.

A página do relatório geral foi desenvolvida com um *script* PHP que faz uma consulta geral no banco de dados e faz a exibição em uma página HTML.

A página de relatório de empréstimos por aluno possui um formulário para que o RA seja informado, sendo sua obtenção realizada através de um *script* AJAX e uma página PHP que fazem a interface com o módulo leitor de códigos de barra. Um *script* PHP faz a busca dos empréstimos realizados pelo aluno selecionado e informa todos os empréstimos realizados por ele. A Figura 15 mostra o diagrama de funcionamento dessa página.

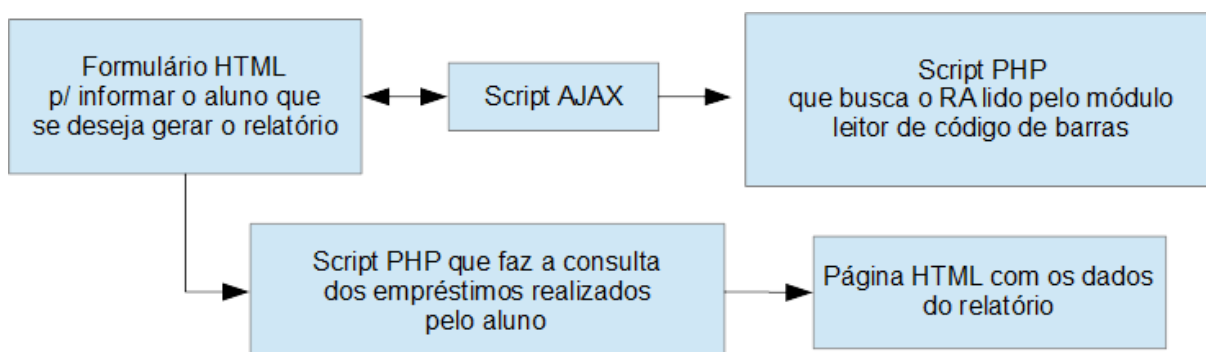


Figura 15 – Diagrama de funcionamento da página Relatório de empréstimos para o aluno.
Fonte: Autoria própria.

O relatório de empréstimos por equipamento foi desenvolvido em uma estrutura bem parecida com a da página de relatório por aluno. A diferença está no *script* de comunicação com o módulo de leitura de código de barras. Nesse caso o *script* obtém o patrimônio do equipamento desejado ao invés do RA. O código PHP que faz as consultas necessárias para se obter o histórico de empréstimos do equipamento desejado e ainda retorna a quantia de vezes que esse equipamento foi emprestado, o que pode ajudar a evitar o desgaste prematuro desse equipamento. O diagrama da Figura 16 mostra a estrutura da página.

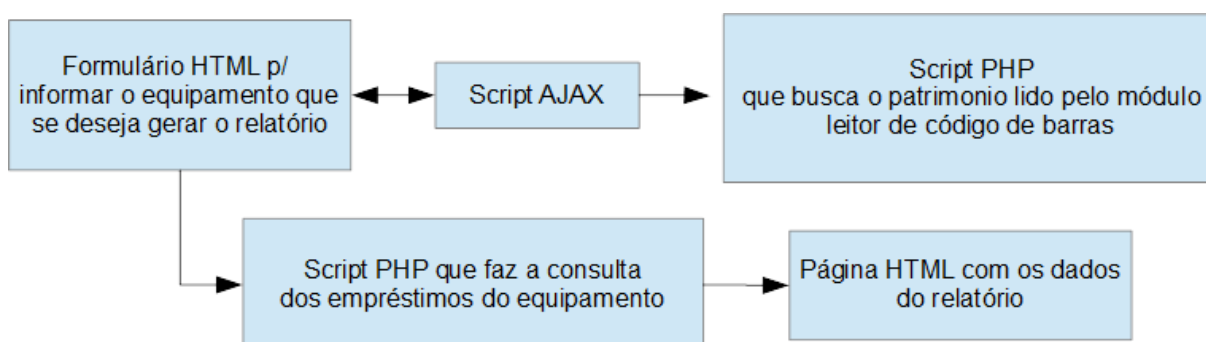


Figura 16 – Diagrama da página Relatório de empréstimos por equipamento.

Fonte: Autoria própria.

A geração de relatórios é um dos pontos mais importantes do projeto, dessa forma garantir o funcionamento dessa função é essencial para o sistema. Foram inseridos alguns empréstimos no banco de dados e posteriormente solicitou-se através das páginas de relatório informações sobre esses empréstimos, as quais passaram pelas validações do sistema e estão demonstradas na seção de resultados.

4. RESULTADOS E DISCUSSÕES

Este capítulo apresenta os resultados obtidos durante o desenvolvimento deste trabalho. Eles foram obtidos utilizando exemplos de RA's de alunos e patrimônio de equipamentos e contemplam todas as funções do sistema. Durante a obtenção dos resultados foram testadas diversas situações, as quais serão descritas detalhadamente na apresentação individual de cada um.

4.1. Módulo leitor de código de barras

Enquanto o *software* rodava na *Raspberry PI*, observou-se o número de patrimônio na interface desenvolvida, o qual pode ser visualizado na Figura 17. Para confirmar o funcionamento do *software* do módulo, na interface *web* foi utilizado o botão que faz a solicitação do código. Conforme mostra a Figura 18.

Para completar a função de empréstimo, realizou-se a leitura de um RA, conforme mostra a Figura 19. O funcionamento do módulo foi comprovado novamente na interface *web* fazendo a solicitação do valor lido. O resultado pode ser visualizado na Figura 20.



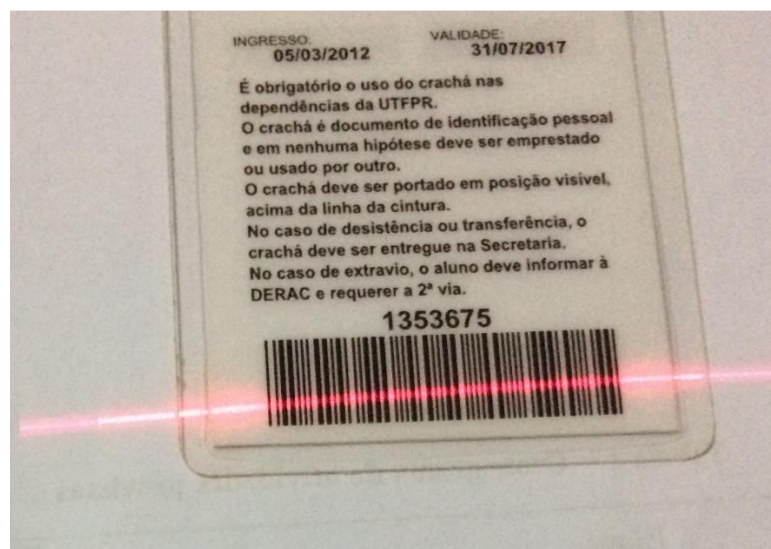
**Figura 17 – Número de patrimônio utilizado para os testes.
Fonte: Autoria própria.**

The screenshot shows a web browser window with the URL `192.168.100.16/pemprestimo.php`. The page is titled "Empréstimo de Equipamento" and features a sidebar menu on the left with options like "Cadastrar Aluno", "Cadastrar Equipamento", and "Relatório de empréstimos". The main content area includes a form with the following elements:

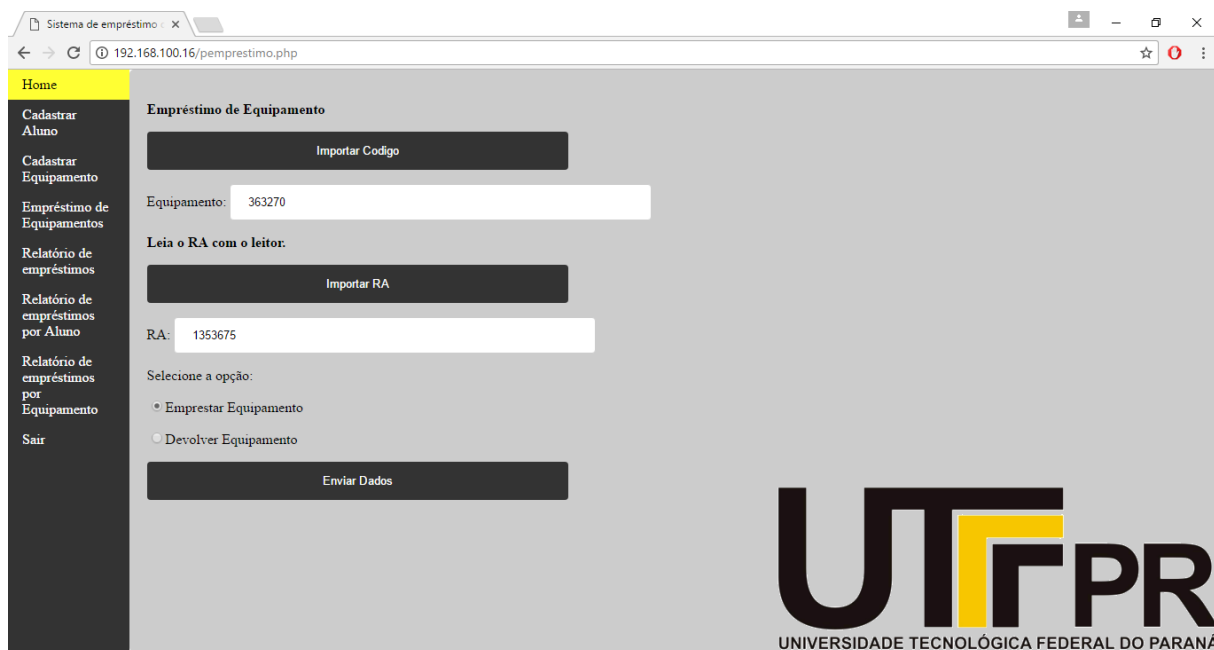
- A button labeled "Importar Código".
- An input field for "Equipamento:" containing the value "363270".
- A section titled "Leia o RA com o leitor." with a button "Importar RA".
- An input field for "RA:".
- A section "Selecione a opção:" with two radio buttons: "Emprestar Equipamento" (selected) and "Devolver Equipamento".
- A button labeled "Enviar Dados".

The UTFPR logo (Universidade Tecnológica Federal do Paraná) is visible in the bottom right corner of the page.

**Figura 18 – Patrimônio do equipamento conforme lido pelo módulo leitor de código de barras.
Fonte: Autoria própria.**



**Figura 19 - Número de RA utilizado nos testes.
Fonte: Autoria própria.**



Sistema de empréstimo - x

192.168.100.16/pemprestimo.php

Home

Cadastrar Aluno

Cadastrar Equipamento

Empréstimo de Equipamentos

Relatório de empréstimos

Relatório de empréstimos por Aluno

Relatório de empréstimos por Equipamento

Sair

Empréstimo de Equipamento

Importar Código

Equipamento: 363270

Leia o RA com o leitor:

Importar RA

RA: 1353675

Selecione a opção:

Emprestar Equipamento

Devolver Equipamento

Enviar Dados

UTPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Figura 20 – RA lido pelo módulo leitor de código de barras.

Fonte: Autoria própria.

Com esses testes foi possível comprovar o funcionamento do *software* que faz a leitura dos códigos de barra. Ele executa funções semelhantes nas páginas de cadastro de aluno, equipamentos e páginas de relatório.

Através do resultado dos testes, ficou comprovada a eficácia das funções escolhidas para o desenvolvimento do módulo e também da lógica usada para a sua programação.

4.2. Banco de dados

Conforme descrito na metodologia, nessa seção está demonstrado o funcionamento do banco de dados do sistema.

A primeira consulta foi executada na tabela de alunos, o resultado pode ser observado na Figura 21.


```
mysql> select * from alunos;
+----+-----+-----+
| id | RA      | nome                               |
+----+-----+-----+
| 36 | 1353675 | Andressa dos Santos Leão         |
| 37 | 1253921 | Aline Rocha Leão                 |
| 38 | 834637  | Renan Felipe T. B. Ikeda         |
| 39 | 363271  | João                              |
| 40 | 363264  | Maria                             |
+----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Figura 21 - Tabela de alunos no banco de dados do sistema.
Fonte: Autoria própria.

Posteriormente a tabela de equipamentos foi verificada, os dados exibidos na consulta podem ser visualizados na Figura 22.

```
mysql> select * from equipamentos;
+----+-----+-----+
| id | patrimonio | descricao                          |
+----+-----+-----+
| 7  | 363951    | Fonte                              |
| 8  | 363274    | Multimetro                         |
| 9  | 363270    | Projetor                           |
| 10 | 363262    | FPGA                               |
| 11 | 363278    | Datapool                           |
| 12 | 363275    | Ponte LCR                          |
| 13 | 363260    | Osciloscópio                       |
+----+-----+-----+
7 rows in set (0.00 sec)

mysql>
```

Figura 22 – Tabela de equipamentos no banco de dados.
Fonte: Autoria própria.

Por último foi executada uma consulta relacionada a tabela de empréstimos, essa consulta pode ser observada na Figura 23.

```
mysql> select * from emprestimos;
+----+-----+-----+-----+-----+-----+
| id | patrimonio | RA      | DataHoraRetirada | DataHoraDevolucao | emprestado |
+----+-----+-----+-----+-----+-----+
| 42 | 363270 | 1353675 | 2017-05-24 20:50:21 | 0000-00-00 00:00:00 | 1 |
| 43 | 363262 | 834637 | 2017-05-24 21:34:10 | 0000-00-00 00:00:00 | 1 |
| 44 | 363278 | 1253921 | 2017-05-24 21:34:39 | 0000-00-00 00:00:00 | 1 |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql>
```

Figura 23 – Tabela de empréstimos do banco de dados.

Fonte: Autoria própria.

4.3. Interface *WEB*

A interface *web* é a ferramenta de operação do sistema. É nela que o usuário faz a inserção de dados, controla os empréstimos e gera os relatórios desejados. Como a interface possui algumas particularidades, foram efetuados testes de validação que serão descritos detalhadamente nesta seção.

4.3.1. Página de *login*

A página de *login* é uma página relativamente simples, ela é apresentada ao usuário conforme a Figura 24. Nessa página o usuário deve informar seu nome de usuário e sua senha.

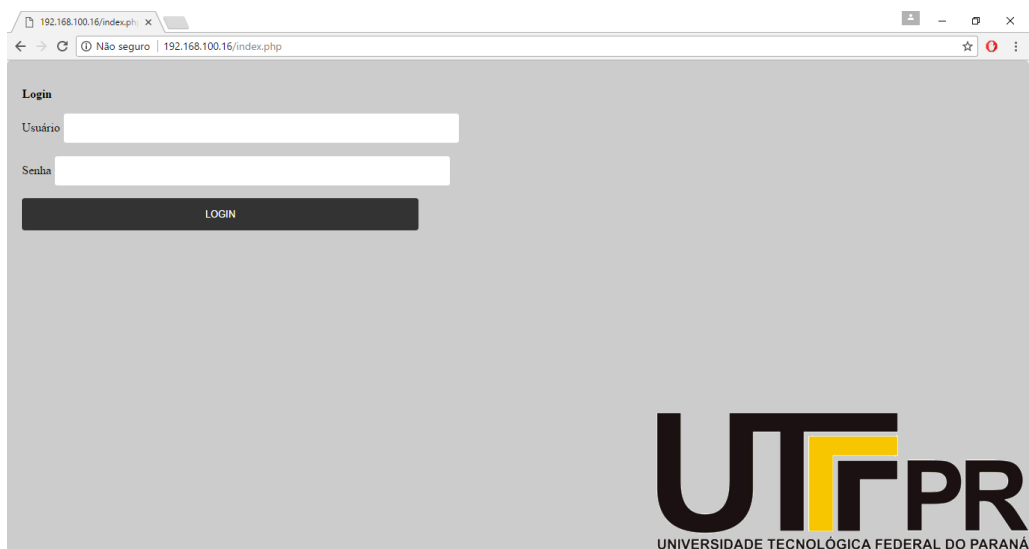


Figura 24 - Página de *login*.
Fonte: A autoria própria.

Caso o usuário esteja correto, o sistema é aberto conforme ilustra a Figura 25. Essa é a página inicial do sistema e através dela é possível acessar todas as funções, como empréstimo, relatórios e cadastros de aluno e equipamentos.

Em caso de um usuário não válido ser digitado, ou em caso de um erro na senha, o sistema exibe uma página de erro por alguns segundos e depois retoma a página de *login*. Essa tela de erro pode ser visualizada na Figura 26.



Figura 25 – Página inicial do sistema.
Fonte: A autoria própria.

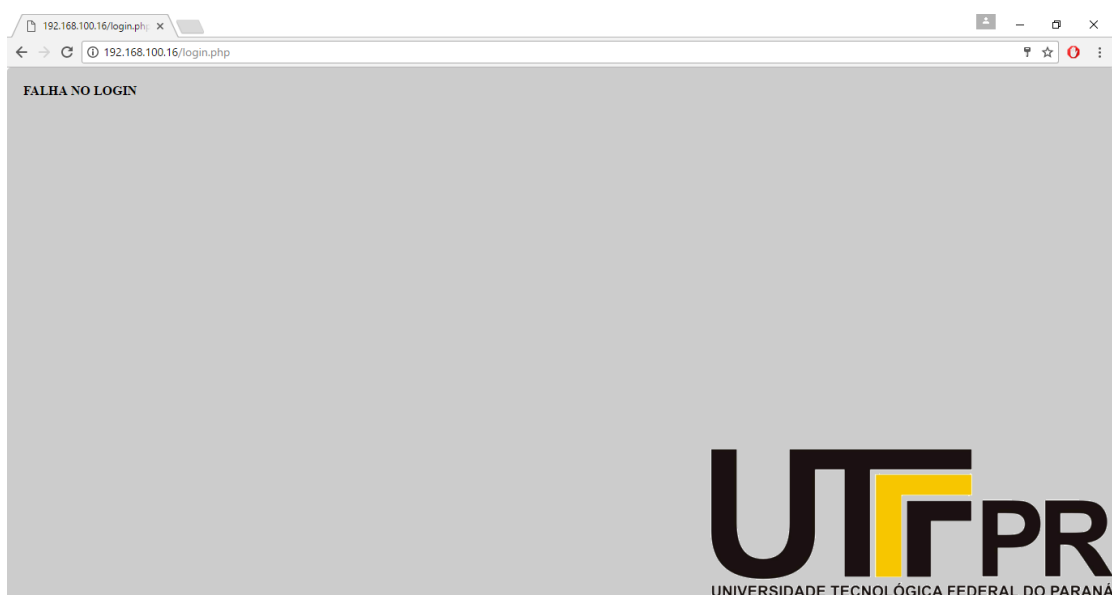


Figura 26 – Falha no login.
Fonte: Autoria Própria.

4.3.2. Cadastro de alunos

O cadastro de alunos é a primeira opção do menu do sistema. Nessa página é efetuado o cadastro dos alunos, conforme mostra a Figura 27. Deve-se adicionar nome e o RA do aluno, sendo que o RA é obtido através do botão “Importar RA”, que executa um *script* AJAX juntamente com um *script* PHP e busca o dado que foi lido pelo módulo leitor de código de barras.

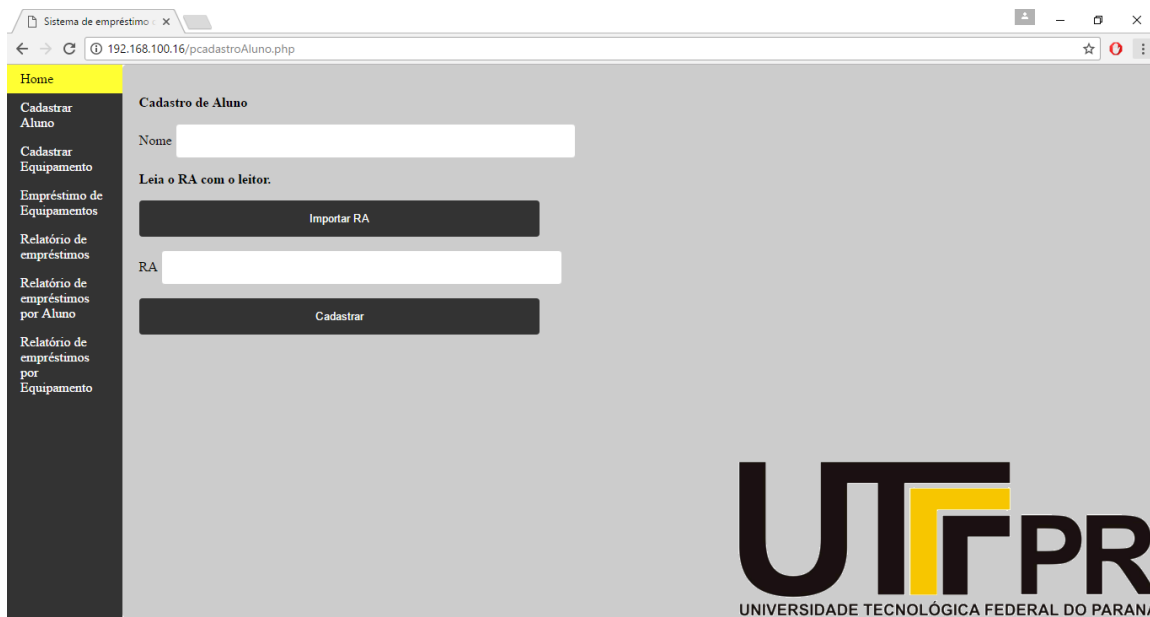


Figura 27 – Página de cadastro de alunos.
Fonte: Autoria própria.

Com os dados inseridos, eles são enviados através do botão “Cadastrar”. Quando esse botão é apertado, os dados passam por um *script* PHP que faz a validação dos dados. Essa validação pode indicar duas situações, caso o aluno já esteja cadastrado no sistema, é apresentada a tela da Figura 28 por alguns segundos, após isso o sistema volta para a página de cadastro. No caso de o cadastro não possuir nenhum erro, o sistema volta para a sua tela inicial e o aluno é cadastrado.

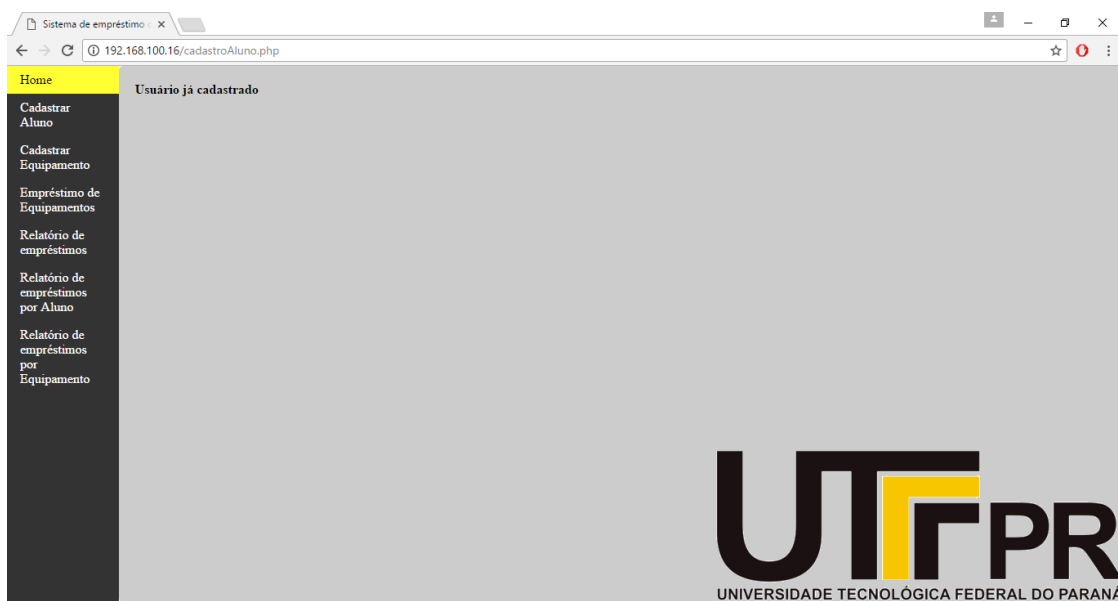


Figura 28 – Erro no cadastro de alunos.
Fonte: Autoria própria.

4.3.3. Cadastro de equipamentos

O cadastro de equipamentos é bem semelhante ao cadastro de alunos, nessa seção são cadastrados os equipamentos disponíveis para empréstimo. A página de cadastro de equipamentos é exibida como na Figura 29. A forma de comunicação entre a interface *web* e o módulo leitor de código de barras é exatamente a mesma do cadastro de alunos.

Assim como no cadastro de alunos, o cadastro de equipamentos passa por algumas validações antes de ser executado. Essas validações evitam duplicidade de dados e caso isso venha a acontecer, a tela da Figura 30 é apresentada. No caso de o cadastro passar pelas validações, o sistema volta para a tela inicial e o equipamento é cadastrado.

The image shows a web browser window with the URL '192.168.100.16/pcadastroEquipamento.php'. The page title is 'Cadastro de Equipamentos'. On the left, there is a dark sidebar with a 'Home' button and several menu items: 'Cadastrar Aluno', 'Cadastrar Equipamento', 'Empréstimo de Equipamentos', 'Relatório de empréstimos', 'Relatório de empréstimos por Aluno', and 'Relatório de empréstimos por Equipamento'. The main content area has a light gray background. At the top, it says 'Cadastro de Equipamentos'. Below that is a text input field labeled 'Equipamento:'. Underneath is the instruction 'Leia o código do equipamento com o leitor.' followed by a dark button labeled 'ImportarCodigo'. Below that is another text input field labeled 'Código:'. At the bottom of the form is a dark button labeled 'Cadastrar'. In the bottom right corner of the page, there is a large logo for 'UTFPR' with the text 'UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ' underneath it.

Figura 29 – Página de cadastro de equipamentos.
Fonte: Autoria própria.

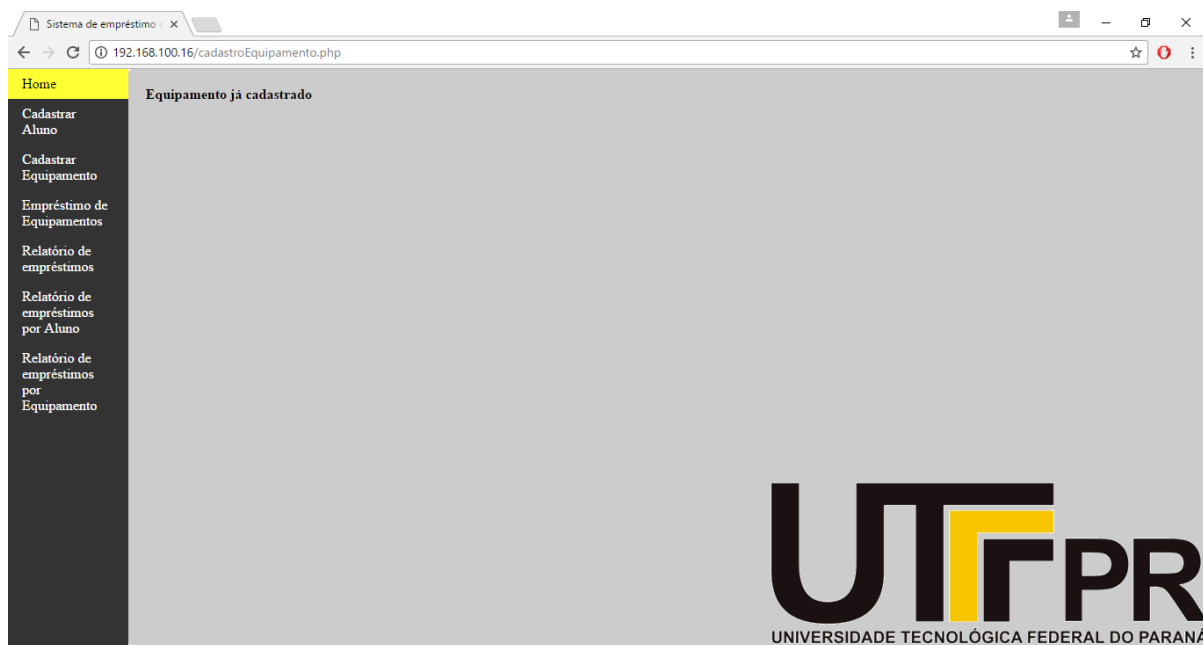


Figura 30 – Erro no cadastro de equipamentos.
Fonte: Autoria própria.

4.3.4. Empréstimo e devolução

A página de empréstimo e devolução é onde a principal função do sistema é executada. Nessa página são inseridos os dados do equipamento emprestado e do aluno que realizou o empréstimo, que são inseridos no sistema por meio de um formulário HTML. Ao acessar a página, ela é exibida conforme a Figura 31.

Figura 31 - Página de empréstimo e devolução.
Fonte: Autoria própria.

O patrimônio do equipamento é obtido por meio do botão “Importar Código” que executa um *script* em AJAX e faz a solicitação para o módulo leitor de código de barras. O mesmo acontece com o botão “Importar RA”, mas nesse caso ele faz a solicitação do RA.

O empréstimo possui algumas validações que geram mensagens de erro. As validações, juntamente com as mensagens de erro serão descritas a seguir.

- **Validação do campo equipamento**

Nessa validação, o sistema verifica se o equipamento está cadastrado para evitar que um dado inconsistente seja adicionado ao sistema. Caso um dado não seja validado, a mensagem de erro da Figura 32 é apresentada.

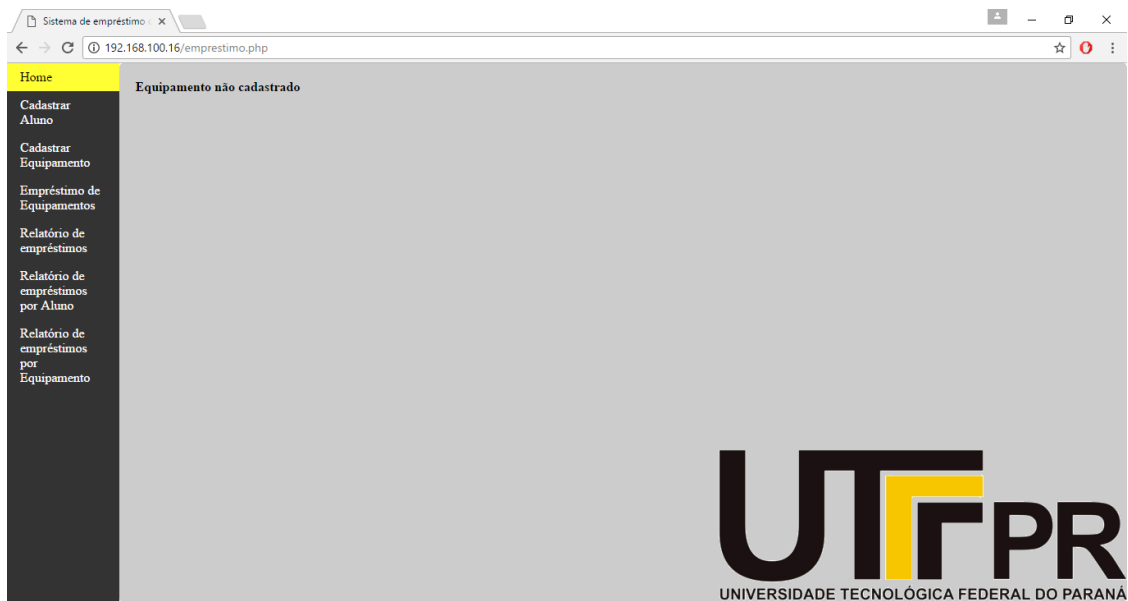


Figura 32 – Erro de falta de cadastro de equipamento.
Fonte: Autoria própria.

- **Validação do campo RA**

Essa validação verifica através de um *script* PHP o status do dado inserido no campo RA (aluno), caso ele não esteja cadastrado no sistema, é informada uma mensagem de erro. Essa mensagem está demonstrada na Figura 33.

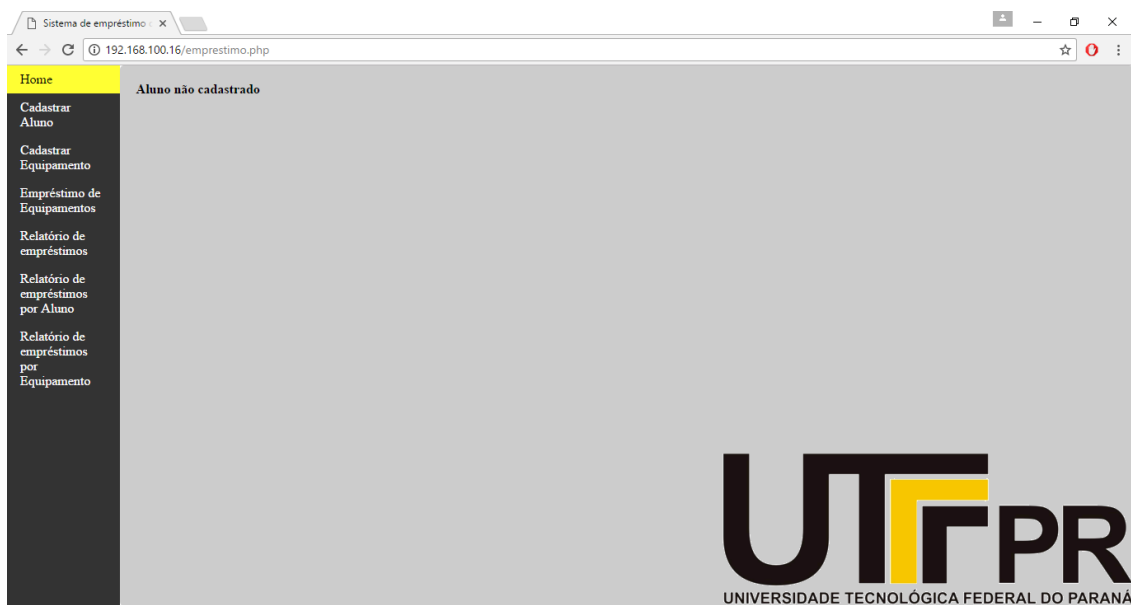


Figura 33 - Erro de aluno não cadastrado.
Fonte: Autoria própria.

- **Verificação de duplicidade no empréstimo**

Função extremamente importante no sistema, ela faz uma varredura no banco de dados através de um *script* PHP que verifica se já existe um empréstimo do equipamento que foi inserido. Ela evita que existam duplicidade nos empréstimos e isso é extremamente importante para não atrapalhar o fluxo de controle. Em uma possível tentativa de inserir um equipamento que já está emprestado, o sistema apresenta uma mensagem de erro por alguns instantes e retorna para a página de empréstimos. A mensagem apresentada pode ser visualizada na Figura 34.

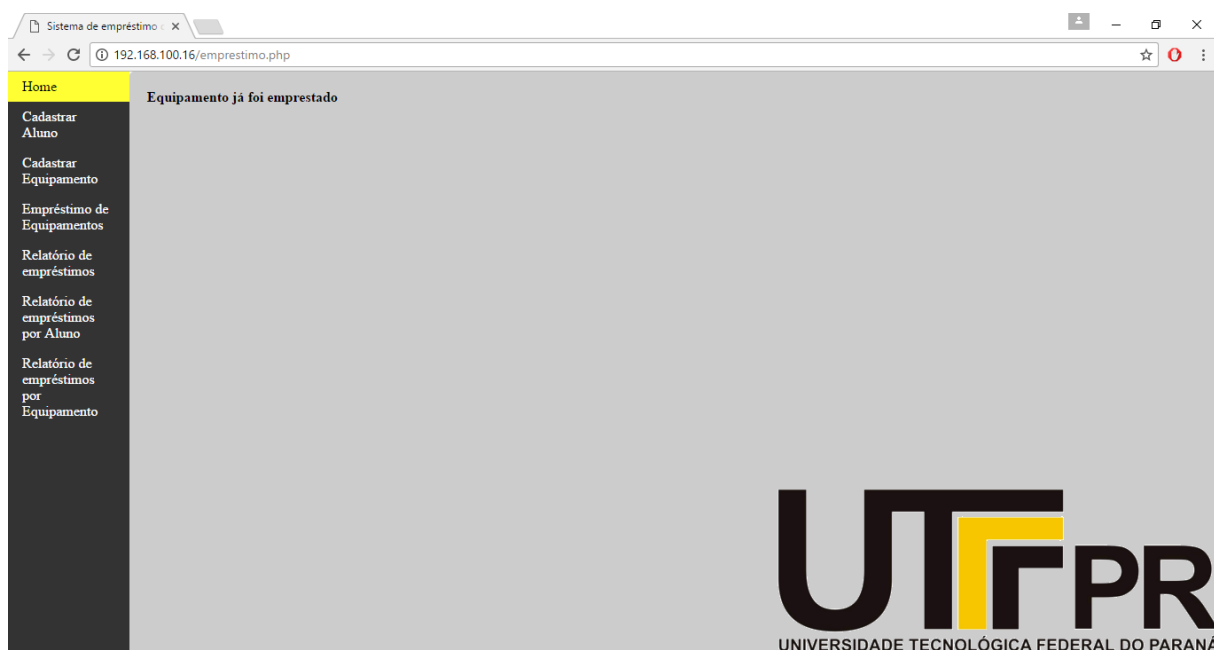


Figura 34 – Erro de duplicidade de empréstimos.

Fonte: Autoria própria.

Caso os dados inseridos passem sem problemas pelas validações, o empréstimo é cadastrado no banco de dados e o sistema retorna para a sua página inicial.

Para efetuar uma devolução, é preciso selecionar o campo “devolução” no formulário, sendo a inserção dos dados exatamente a mesma do caso do empréstimo. Para a devolução ser efetuada, é feita uma validação dos campos. Ela é também realizada por meio de um *script* PHP que varre o banco de dados e verifica se os dados inseridos são consistentes, ou seja, se existe um empréstimo “em aberto” com esses dados (RA e patrimônio). Caso não exista

nenhum empréstimo, o sistema apresenta uma mensagem de erro como a da Figura 35 por alguns instantes e depois retorna para a página de empréstimos.

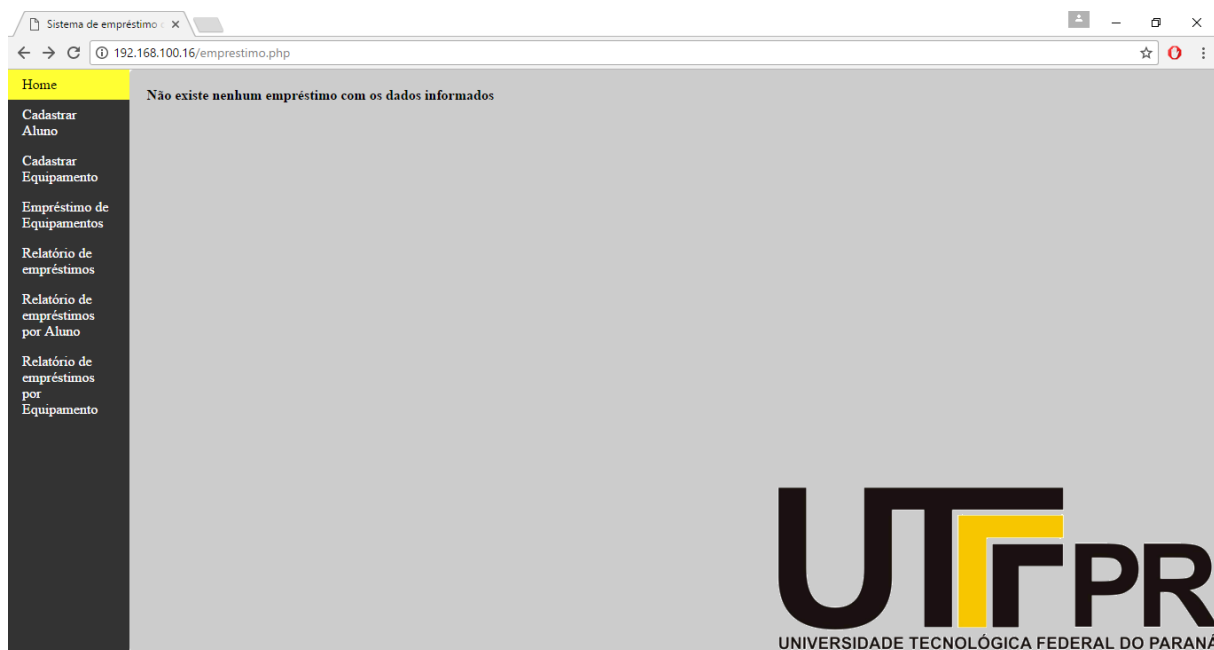


Figura 35 – Erro na devolução.
Fonte: Autoria própria.

Caso os dados sejam consistentes, o sistema efetua a devolução do equipamento, mas não exclui o empréstimo do banco de dados. Todo o histórico de empréstimos fica registrado para consultas futuras.

4.3.5. Relatórios

A seção de relatórios é importante pois apresenta todo o histórico dos empréstimos. Foram gerados três tipos de relatório, os quais serão descritos detalhadamente nos tópicos a seguir.

- **Relatório de empréstimos**

Esse relatório contempla todos os empréstimos já realizados, ele é gerado através de um *script* PHP e pode ser acessado pelo menu do sistema na opção “Relatório de empréstimos”. O relatório possui algumas informações importantes, como a data de empréstimo e devolução, e também exibe se o equipamento está ou não emprestado. Esse relatório mostra todos os

empréstimos, portanto é um histórico “geral”. O formato do relatório pode ser visualizado na Figura 36.

Equipamento	Aluno	Data retirada	Data devolução	Emprestado
7897517937164	834637	2017-05-13 20:38:07	2017-05-13 21:16:21	NÃO
7897517937164	834637	2017-05-13 21:13:07	2017-05-13 21:16:21	NÃO
7898929261298	834637	2017-05-13 21:13:45	0000-00-00 00:00:00	NÃO
7897517937164	834637	2017-05-13 21:17:36	2017-05-13 21:17:49	NÃO
363274	834637	2017-05-21 23:31:37	0000-00-00 00:00:00	SIM

Relatório gerado em: 2017-05-21 23:39:53

UTPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Figura 36 – Tela de relatório geral.
Fonte: Autoria própria.

- **Relatório de empréstimos por aluno**

O relatório de empréstimos por aluno gera um relatório de todos os empréstimos de um determinado aluno, que pode ser acessado através do menu do sistema na opção “Relatório de empréstimos por aluno”. Para sua geração, é necessário informar um RA no formulário, esse RA é lido pelo leitor de código de barras e importado através de um *script* AJAX combinado com um PHP. Esse método é o mesmo utilizado nos cadastros que foram descritos anteriormente. O RA também pode ser digitado no teclado, caso não se tenha o RA em mãos. A tela que aparece na seleção da opção de relatório pode ser visualizada na Figura 37.

Com o RA selecionado, o relatório pode ser gerado, caso não seja encontrado nenhum empréstimo cadastrado para aquele usuário, o sistema apresenta uma mensagem como a da Figura 38.

Sistema de empréstimo - x

192.168.100.16/prelatorioRA.php

Home

Cadastrar Aluno

Cadastrar Equipamento

Empréstimo de Equipamentos

Relatório de empréstimos

Relatório de empréstimos por Aluno

Relatório de empréstimos por Equipamento

Relatório de empréstimos por Aluno

Leia o RA com o leitor:

Importar RA

RA:

Gerar Relatório

UTFPB
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Figura 37 – Formulário para geração do relatório de empréstimos por aluno.
Fonte: Autoria própria.

Relatório de empréstimo - x

192.168.100.16/relatorioRA.php

Home

Cadastrar Aluno

Cadastrar Equipamento

Empréstimo de Equipamentos

Relatório de empréstimos

Relatório de empréstimos por Aluno

Relatório de empréstimos por Equipamento

Não foi encontrado nenhum registro de empréstimo para o RA 846218

UTFPB
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

23:49
21/05/2017

Figura 38 – Erro na geração do relatório de empréstimos por aluno.
Fonte: Autoria própria.

Se houverem registros, o relatório é apresentado conforme a Figura 39. Esse relatório é importante tanto para os alunos, quanto para o departamento, pois é possível manter o controle do histórico, evitando assim vários problemas.

Equipamento	Data retirada	Data devolução	Emprestado
7897517937164	2017-05-13 20:38:07	2017-05-13 21:16:21	NÃO
7897517937164	2017-05-13 21:13:07	2017-05-13 21:16:21	NÃO
7898929261298	2017-05-13 21:13:45	0000-00-00 00:00:00	NÃO
7897517937164	2017-05-13 21:17:36	2017-05-13 21:17:49	NÃO
363274	2017-05-21 23:31:37	0000-00-00 00:00:00	SIM

Relatório gerado em: 2017-05-21 23:48:30

UTPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Figura 39 – Relatório de empréstimos por aluno.

Fonte: Autoria própria.

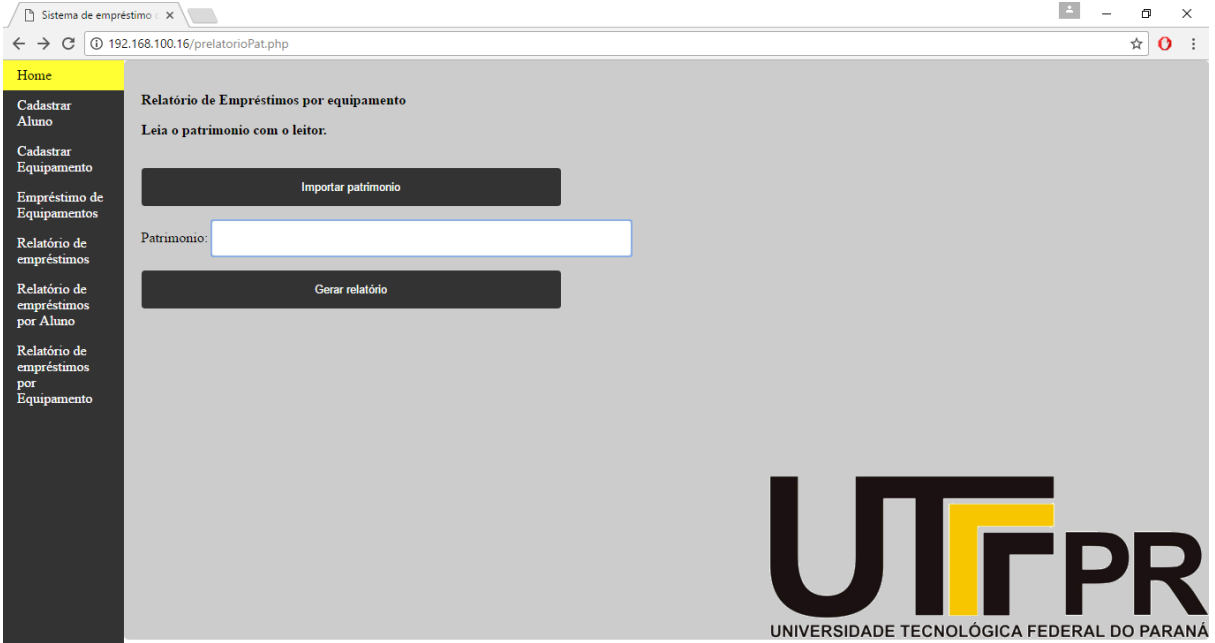
- **Relatório de empréstimos por equipamento**

A lógica desse relatório é semelhante ao relatório descrito na seção anterior, mas nesse caso é levantado um histórico do equipamento solicitado. Esse relatório é acessado pela opção “Relatório de empréstimos por equipamento”. Assim como no relatório por alunos, a inserção dos dados do equipamento que se deseja obter o relatório é feita através de um formulário combinado com AJAX e PHP. Esse formulário pode ser visualizado na Figura 40.

No caso de o equipamento informado não possuir nenhum empréstimo cadastrado, o sistema apresenta uma mensagem como a da Figura 41.

Para o caso de existirem dados de empréstimos, o sistema apresenta um relatório semelhante ao da Figura 42. Além da listagem de todos os empréstimos efetuados para o equipamento que foi solicitado, o sistema também informa quantas vezes esse equipamento foi emprestado. Esses dados combinados com o histórico são de extrema importância para o fluxo

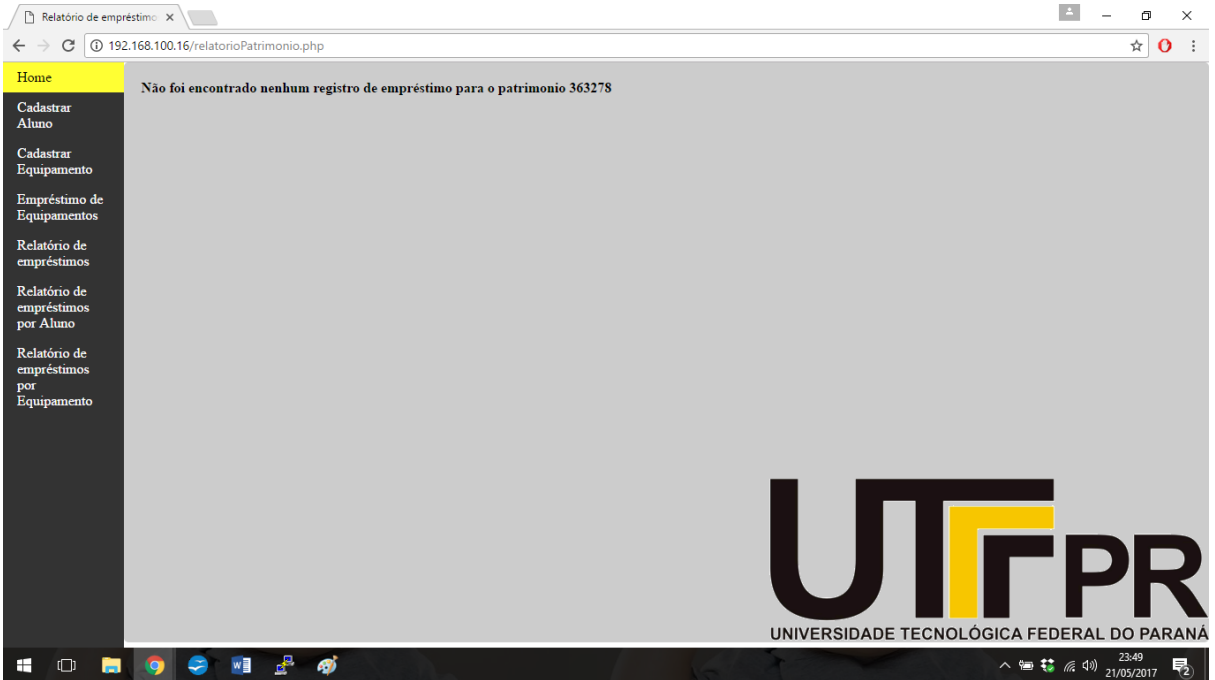
tanto de controle, quanto de gestão dos equipamentos, pois através disso é possível evitar o desgaste de equipamentos que já foram utilizados muitas vezes.



The screenshot shows a web browser window with the URL `192.168.100.16/prelatorioPat.php`. The page title is "Sistema de empréstimo". On the left, there is a dark sidebar menu with the following items: "Home" (highlighted in yellow), "Cadastrar Aluno", "Cadastrar Equipamento", "Empréstimo de Equipamentos", "Relatório de empréstimos", "Relatório de empréstimos por Aluno", and "Relatório de empréstimos por Equipamento". The main content area has a title "Relatório de Empréstimos por equipamento" and a subtitle "Leia o patrimonio com o leitor:". Below this, there is a button labeled "Importar patrimonio". Underneath, there is a text input field labeled "Patrimonio:" which is currently empty. Below the input field is a button labeled "Gerar relatório". In the bottom right corner of the page, there is a logo for "UTFPR" (UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ).

Figura 40 – Formulário para geração do relatório de empréstimos por equipamento.

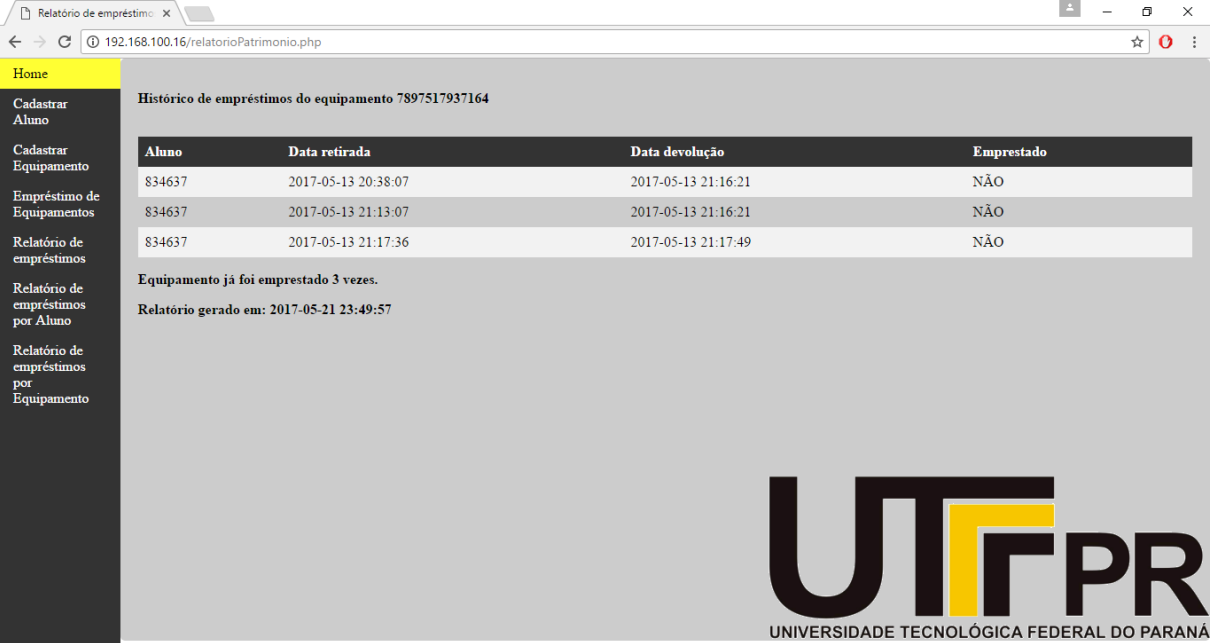
Fonte: Autoria própria.



The screenshot shows the same web browser window, but the URL is now `192.168.100.16/relatorioPatrimonio.php`. The sidebar menu is identical to the previous screenshot. The main content area displays a message: "Não foi encontrado nenhum registro de empréstimo para o patrimonio 363278". The "UTFPR" logo is still present in the bottom right corner. The Windows taskbar is visible at the bottom of the screen, showing the time as 23:49 on 21/05/2017.

Figura 41 – Erro no relatório por equipamentos.

Fonte: Autoria própria.



Relatório de empréstimos: x

192.168.100.16/relatorioPatrimonio.php

Home

Cadastrar Aluno

Cadastrar Equipamento

Empréstimo de Equipamentos

Relatório de empréstimos

Relatório de empréstimos por Aluno

Relatório de empréstimos por Equipamento

Histórico de empréstimos do equipamento 7897517937164

Aluno	Data retirada	Data devolução	Emprestado
834637	2017-05-13 20:38:07	2017-05-13 21:16:21	NÃO
834637	2017-05-13 21:13:07	2017-05-13 21:16:21	NÃO
834637	2017-05-13 21:17:36	2017-05-13 21:17:49	NÃO

Equipamento já foi emprestado 3 vezes.

Relatório gerado em: 2017-05-21 23:49:57

UTPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Figura 42 – Relatório de empréstimos por equipamento.
Fonte: Autoria própria.

5. CONCLUSÃO

O empréstimo de equipamentos é uma necessidade no meio acadêmico, seja para aulas práticas ou para o desenvolvimento de projetos fora do horário de aula, como PI's, TCC's e os projetos de disciplina que são propostos pelos professores. Existem diversos equipamentos, como osciloscópio, fonte regulável, gerador de função, FPGA's, entre outros, que possuem um custo elevado e praticamente nenhum aluno tem acesso a eles fora da faculdade.

O departamento faz o empréstimo desses equipamentos nos horários em que o técnico de laboratório fica na instituição e faz o controle dos empréstimos de forma manual. Esse controle, conforme já foi descrito anteriormente, por vezes apresenta algumas falhas durante o processo, além de não manter um histórico dos empréstimos.

Este trabalho teve como proposta o desenvolvimento de um sistema automatizado para fazer o controle dos empréstimos dos equipamentos do DAELN-CM. Esse controle, além de automatizar o processo de registro do empréstimo, ainda mantém um histórico de fácil acesso por meio de relatórios.

Para o desenvolvimento do sistema foram feitas pesquisas bibliográficas para definição da estrutura tanto de *hardware*, quanto do *software*. Durante as pesquisas foram definidos os elementos que seriam utilizados para atender de forma mais prática os objetivos do projeto. A *Raspberry PI* foi utilizada para obter um sistema portátil, ou seja, com a possibilidade de mudar ele de lugar com facilidade. A escolha pela interface *web* também foi pela portabilidade, pois com ela o sistema fica acessível por meio de um navegador *web*, o que faz com que a sua operação seja extremamente simples.

Durante o desenvolvimento do trabalho algumas dificuldades foram encontradas, a principal delas foi a interface entre o sistema *web* e o programa desenvolvido na *Raspberry* para a leitura de código de barras. Para o funcionamento correto dessa interface entre as duas plataformas foram necessários diversos testes e validações para que o sistema não se comporte de maneira inesperada.

A interface *web* do sistema foi desenvolvida para ficar extremamente intuitiva e com operação facilitada. Para a questão de segurança, o sistema possui todas as validações para garantir a integridade dos dados inseridos e conseqüentemente a sua confiabilidade. Também possui um módulo de controle de *login*, o que torna o seu acesso permitido apenas a usuários

previamente cadastrados. Esse cadastro é feito diretamente no banco de dados do sistema, portanto somente pessoas autorizadas e instruídas para isso podem fazer esse cadastro.

Como os objetivos iniciais foram alcançados, pode-se dizer que o projeto foi concluído de maneira satisfatória. O controle dos empréstimos ficou automatizado e é possível consultar todo o histórico de empréstimos realizados.

Para trabalhos futuros ficam as seguintes sugestões:

- Desenvolvimento de aplicativos para dispositivos móveis que conversam com o sistema, o que aumentaria ainda mais a sua portabilidade;
- Desenvolvimento de um controle por *tag's* RFID ao invés dos códigos de barra;
- Desenvolvimento de um sistema de reserva de equipamentos;

REFERÊNCIAS

- BARR, M.; MASSA, A. *Programming Embedded Systems*. Sebastopol: O'Reilly, 1999.
- CAPIOTTI, Paulo. **Benefícios da Automação de Processos**. Blog IPROCESS – 25 de junho de 2012. Disponível em: <<http://blog.iprocess.com.br/2012/06/beneficios-da-automacao-de-processos/>>. Acesso em: maio de 2017.
- DAWIS, E. P., J. F. Dawis, Wei-Pin Koo (2001). *Architecture of Computer-based Systems using Dualistic Petri Nets*. Systems, Man, and Cybernetics, 2001 IEEE International Conference on Volume 3, 2001 Page(s):1554 - 1558 vol.3
- ELINUX. **RPI Hub**. Setembro 2013. Acessado em 04 de maio de 2017. Disponível em: http://elinux.org/RPi_Hub
- ESCOLA W3. **Tutoriais**. Acessado em 04 de maio de 2017. Disponível em: <http://escola.w3.pt/>
- GNU. **GNU GENERAL PUBLIC LICENSE**, 2017. Disponível em: <<https://www.gnu.org/licenses/gpl-3.0.en.html>> Acesso em: 05 de maio de 2017
- KORTH, H.F. e SILBERSCHATZ, A. **Sistemas de Bancos de Dados**, Makron Books, 2a. edição revisada, 1994.
- MOZILA DEVELOPER NETWORK. **HTML**, 2017. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em 06 de maio de 2017
- NATIONS, DANIEL. *Improve Your Understanding of Web Applications*, 2016. Disponível em: <<https://www.lifewire.com/what-is-a-web-application-3486637>>. Acesso em 04 de maio de 2017
- ORACLE. **MySQL**, 2017. Disponível em: <<https://www.oracle.com/mysql/index.html>>. Acesso em 06 de maio de 2017
- PALMEIRA, Thiago V. V. **Como funcionam as aplicações web**. Disponível em: <<http://www.devmedia.com.br/como-funcionam-as-aplicacoes-web/25888>>. Acesso em: 05 Maio de 2017
- PHP.NET. **Documentations**, 2017. Disponível em: <<http://php.net/>>. Acesso em 06 de maio de 2017
- PRUDENCIO, R. B. C.; RODRIGUES, A. M. M. **Automação: a inserção da biblioteca na tecnologia da informação**. 2009.
- RASPBERRY PI FOUNDATION, UK. **Raspberry Pi**. Disponível em: <<http://www.raspberrypi.org/faqs>>. Acesso em: 04 de maio de 2017.

REZENDE, Ricardo **Conceitos fundamentais de banco de dados**. 2006. Disponível em: <<http://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. Acesso em: 05 Maio de 2017

SGANDERLA, Kelly. **Estudo de Caso: Automatizar o processo (ou não)? Eis a questão!** Blog IPROCESS – 04 de fevereiro de 2013. Disponível em: <<http://blog.iprocess.com.br/tag/beneficios-da-automacao-de-processos/>>. Acesso em: Maio de 2017.

STADZISZ, P. C.; RENAUX, D. P. B. **Software Embarcado**. 2008. Disponível em: <http://www.pessoal.utfpr.edu.br/douglasrenaux/Stad_Renaux_Software_Embarcado.pdf>.

VARELA, Bernardina Cardoso. **Desenvolvimento de um sistema de Gestão de Biblioteca**. Cabo Verde: 2005

TEIXEIRA, José R. **Introdução ao MySQL**. Disponível em: <<http://www.devmedia.com.br/introducao-ao-mysql/27799>>. Acesso em: 08 de maio de 2017.

WELLING, Luke; THOMSON, Laura; FURMANKIEWICZ, Edson. **PHP e MySQL: desenvolvimento Web**. Rio de Janeiro: Campus, 2005

W3 Schools. **Tutorials**. Acessado em 04 de maio de 2017. Disponível em: <http://www.w3schools.com>

ANEXO 1**MANUAL DE INSTRUÇÕES**
SISTEMA DE CONTROLE DE
EMPRÉSTIMOS DE EQUIPAMENTOS

OBSERVAÇÃO: Todos os dados podem ser inseridos através do leitor de código de barras e também pelo teclado. Caso se deseje inserir pelo teclado, não há necessidade de apertar nos botões *Importar*, basta digitar na caixa de texto.

1. CRIAÇÃO DE UM USUÁRIO

Para acessar o sistema é preciso criar um usuário, esse usuário é criado diretamente no gerenciador de banco de dados do sistema. O acesso ao gerenciador é feito através de uma página web, ela pode ser acessada através de um navegador web (*Google Chrome, Mozilla Firefox, Internet Explorer*) pelo endereço de ip em que se encontra o servidor (Raspberry PI) combinado com o caminho */phpmyadmin*. (Exemplo: <http://192.168.100.16/phpmyadmin>)

O acesso ao gerenciador precisa de um usuário e uma senha, nesse caso ambos estão configurados como *root*. Com a tela do gerenciador aberta, deve-se acessar a base de dados *almoxarifado*, nela deve-se abrir a tabela *usr*, conforme mostra a figura 1.

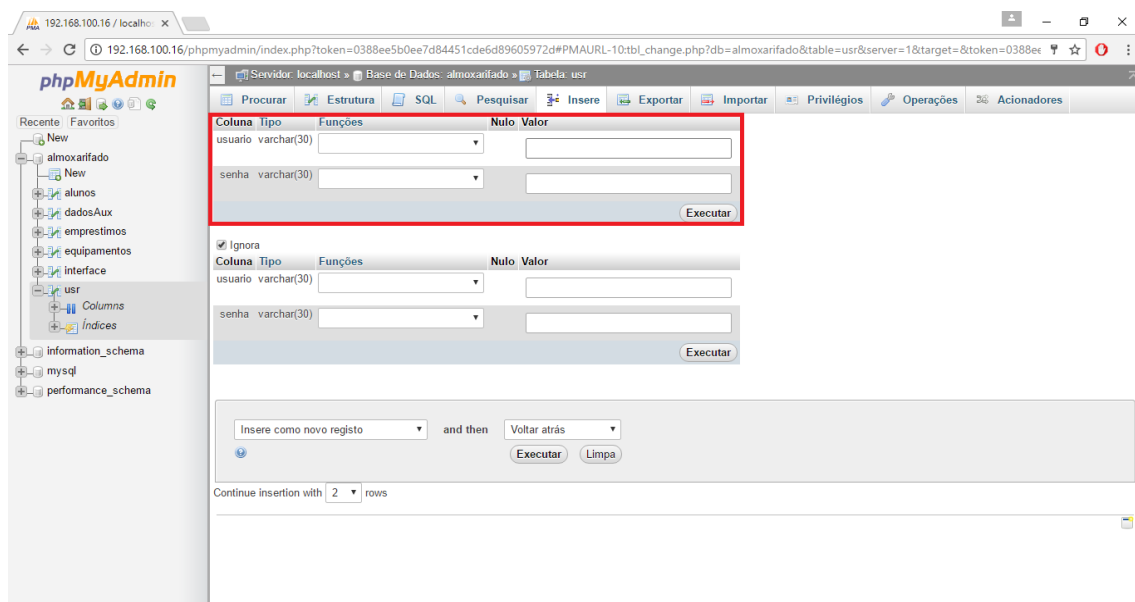


Figura 43 - Página de controle do banco de dados

O usuário, juntamente com a senha devem ser adicionados e deve-se apertar o botão executar. Fazendo esses passos, o usuário está criado e pronto para acessar o sistema.

Observação: Não são permitidos usuários com o mesmo nome.

2. CADASTRO DE ALUNOS

O cadastro de alunos é feito através da página *Cadastro de alunos*. Nesse cadastro é necessário colocar um nome e fazer a leitura do RA através do leitor de código de barras. Para obter o RA lido, basta clicar no botão *Importar RA*, com os dois dados (nome e RA), deve-se clicar no botão *Cadastrar*. A tela de cadastro de alunos pode ser visualizada na figura 2, juntamente com os botões *Importar RA* e *Cadastrar*.

The image shows a web browser window displaying the 'Cadastro de Aluno' page. The browser's address bar shows the URL '192.168.100.16/pcadastroAluno.php'. The page layout includes a dark sidebar on the left with a 'Home' button highlighted in yellow. The sidebar contains the following menu items: 'Cadastrar Aluno', 'Cadastrar Equipamento', 'Empréstimo de Equipamentos', 'Relatório de empréstimos', 'Relatório de empréstimos por Aluno', and 'Relatório de empréstimos por Equipamento'. The main content area is titled 'Cadastro de Aluno' and features a form with the following elements: a text input field for 'Nome', a section labeled 'Leia o RA com o leitor.' containing a dark 'Importar RA' button, and another text input field for 'RA' with a dark 'Cadastrar' button below it. In the bottom right corner of the page, the logo for 'UTFRPR' (Universidade Tecnológica Federal do Paraná) is displayed, consisting of the letters 'UTFRPR' in a bold, stylized font with a yellow square in the 'F', and the full name 'UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ' underneath.

Figura 44 - Cadastro de alunos

3. CADASTRO DE EQUIPAMENTOS

O cadastro de equipamentos é feito através da página *Cadastro de equipamentos*. Nesse cadastro é necessário colocar uma descrição para o equipamento e fazer a leitura do patrimônio através do leitor de código de barras. Para obter o patrimônio lido, basta clicar no botão *Importar Codigo*, com os dois dados (patrimônio e descrição), deve-se clicar no botão *Cadastrar*. A tela de cadastro de equipamentos pode ser visualizada na figura 2, juntamente com os botões *Importar Codigo* e *Cadastrar*.

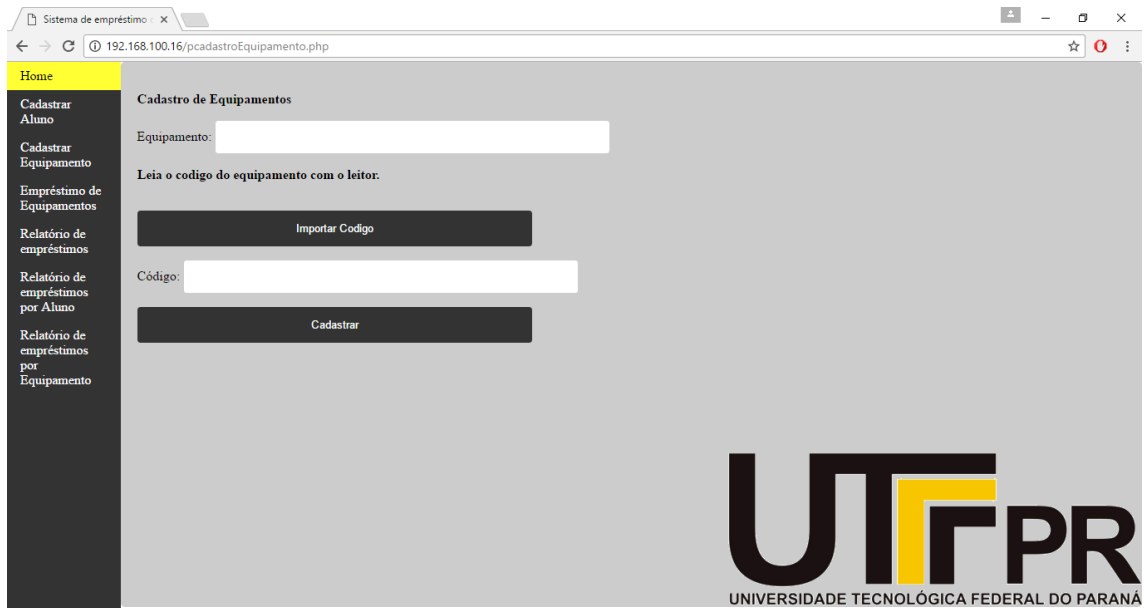
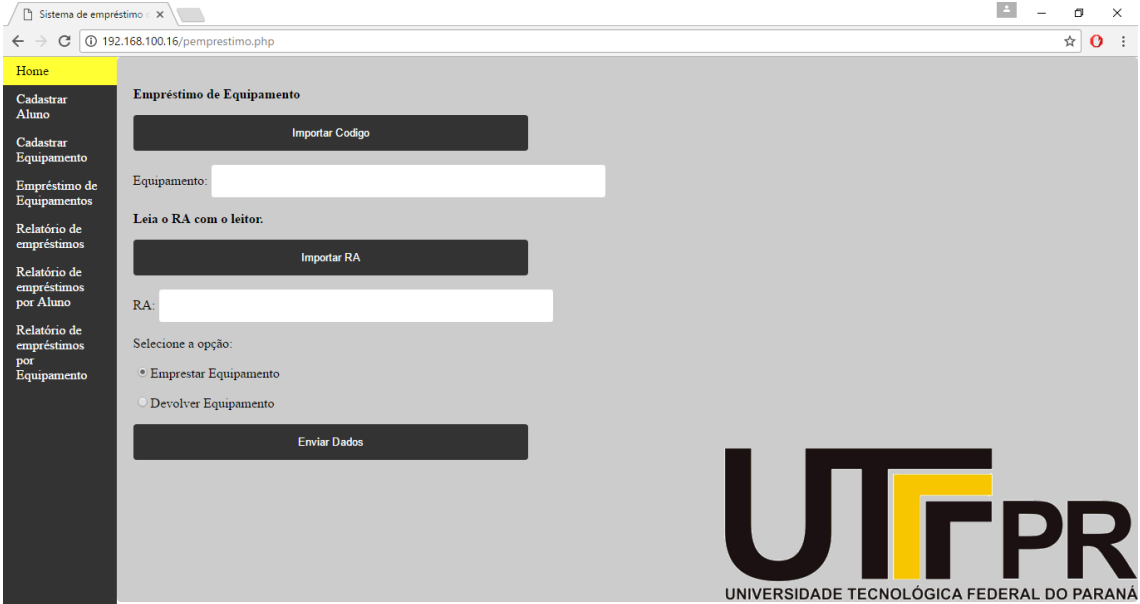


Figura 45 - Cadastro de Equipamentos

4. EMPRÉSTIMO DE EQUIPAMENTOS

Para efetuar o empréstimo de um equipamento é preciso acessar a página *Empréstimos*. Nessa página é necessário que se faça a inserção dos dados do equipamento emprestado e do aluno que está emprestando.

Para inserir o dado de patrimônio do equipamento, deve-se fazer a leitura com o leitor do código de barras e clicar no botão *Importar Código*. Depois do patrimônio aparecer no campo *Equipamento*, deve-se ler o RA do aluno com o leitor e clicar no botão *Importar RA*.



A imagem mostra a interface de usuário de um sistema web para empréstimos. O navegador indica o endereço 192.168.100.16/pemprestimo.php. O menu lateral à esquerda contém links para: Home, Cadastrar Aluno, Cadastrar Equipamento, Empréstimo de Equipamentos, Relatório de empréstimos, Relatório de empréstimos por Aluno e Relatório de empréstimos por Equipamento. O formulário principal, intitulado "Empréstimo de Equipamento", possui os seguintes elementos: um botão "Importar Código" para importar o número de patrimônio; um campo de texto rotulado "Equipamento:"; um botão "Importar RA" para importar o número de matrícula; um campo de texto rotulado "RA:"; duas opções de seleção por rádio: "Emprestar Equipamento" e "Devolver Equipamento"; e um botão "Enviar Dados" para finalizar o registro. No canto inferior direito, há o logotipo da UTEPR (Universidade Tecnológica Federal do Paraná).

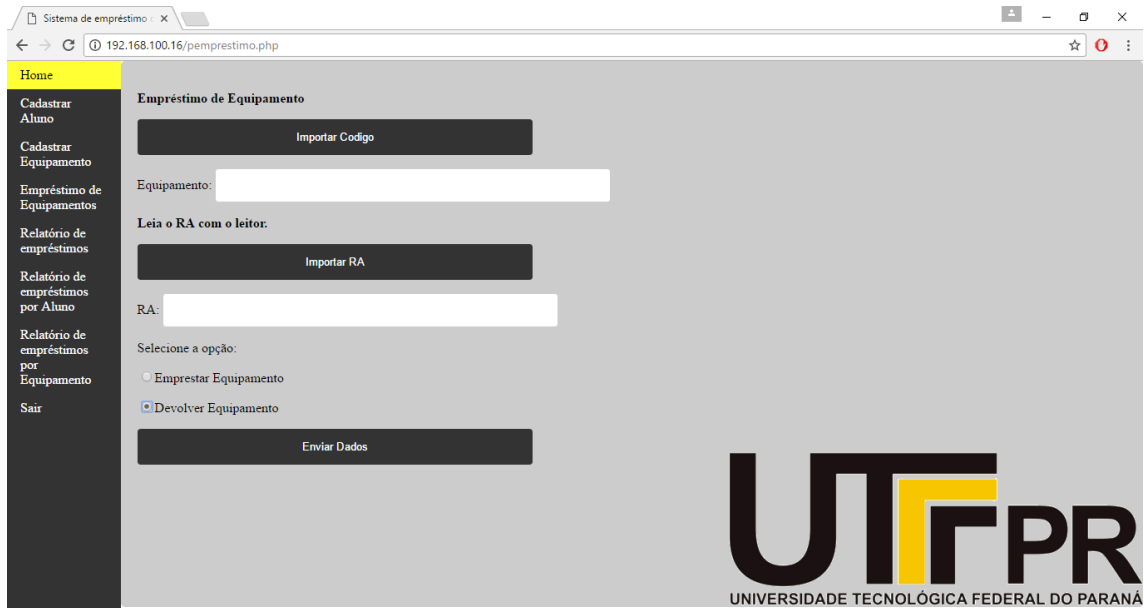
Figura 46 - Página de empréstimos

Com os dados de equipamento e aluno inseridos, deve-se clicar no botão *Enviar Dados*. Caso os dados estejam corretos, o empréstimo estará registrado, caso ocorra algum erro, o sistema informará em uma mensagem e retornará para a página de empréstimos.

5. DEVOLUÇÃO DE EQUIPAMENTOS

Para efetuar a devolução de um equipamento também deve-se acessar a página *Empréstimos*. Os dados devem ser inseridos da mesma forma que no empréstimo.

A diferença está na seleção da opção *Devolver Equipamento*. Essa opção deve ser selecionada conforme a figura 5.



The image shows a web browser window displaying the 'Sistema de empréstimo' application. The URL is 192.168.100.16/pemprestimo.php. The page has a dark sidebar on the left with navigation links: Home, Cadastrar Aluno, Cadastrar Equipamento, Empréstimo de Equipamentos, Relatório de empréstimos, Relatório de empréstimos por Aluno, Relatório de empréstimos por Equipamento, and Sair. The main content area is titled 'Empréstimo de Equipamento' and contains the following form elements: a 'Importar Código' button, an 'Equipamento:' text input field, a 'Leia o RA com o leitor:' section with an 'Importar RA' button, an 'RA:' text input field, a 'Selecione a opção:' section with two radio buttons: 'Emprestar Equipamento' (unselected) and 'Devolver Equipamento' (selected), and an 'Enviar Dados' button. The bottom right corner features the logo of 'UTPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ'.

Figura 5 - Página de empréstimos com a opção Devolver Equipamento selecionada

Com os dados de equipamento e aluno inseridos, deve-se clicar no botão *Enviar Dados*. Caso os dados estejam corretos, a devolução estará registrada, caso ocorra algum erro, o sistema informará em uma mensagem e retornará para a página de empréstimos.

6. RELATÓRIOS

O sistema permite a geração de três tipos de relatório, um que mostra todos os empréstimos já realizados, um que mostra os empréstimos realizados por determinado aluno e outro que mostra o histórico dos empréstimos de um equipamento.

O relatório de todos os empréstimos pode ser acessado através da opção *Relatório de Empréstimos*. Ao acessar a opção, o sistema mostra uma tabela com todos os empréstimos realizados. Um exemplo pode ser visualizado na figura 6.

Equipamento	Aluno	Data retirada	Data devolução	Empréstado
363270	1353675	2017-05-24 20:50:21	0000-00-00 00:00:00	SIM
363262	834637	2017-05-24 21:34:10	0000-00-00 00:00:00	SIM
363278	1253921	2017-05-24 21:34:39	0000-00-00 00:00:00	SIM

Relatório gerado em: 2017-05-28 19:51:17

UTPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Figura 6 - Exemplo de relatório

6.1. Relatório por aluno

O relatório de um determinado aluno pode ser acessado através da página *Relatório de empréstimos por Aluno*, nessa página deve-se informar o RA. Conforme dito anteriormente, esse RA pode ser obtido através do leitor do código de barras ou então digitado via teclado. Com o RA inserido, deve-se clicar no botão *Gerar Relatório*. A figura 7 mostra a página com o formulário.

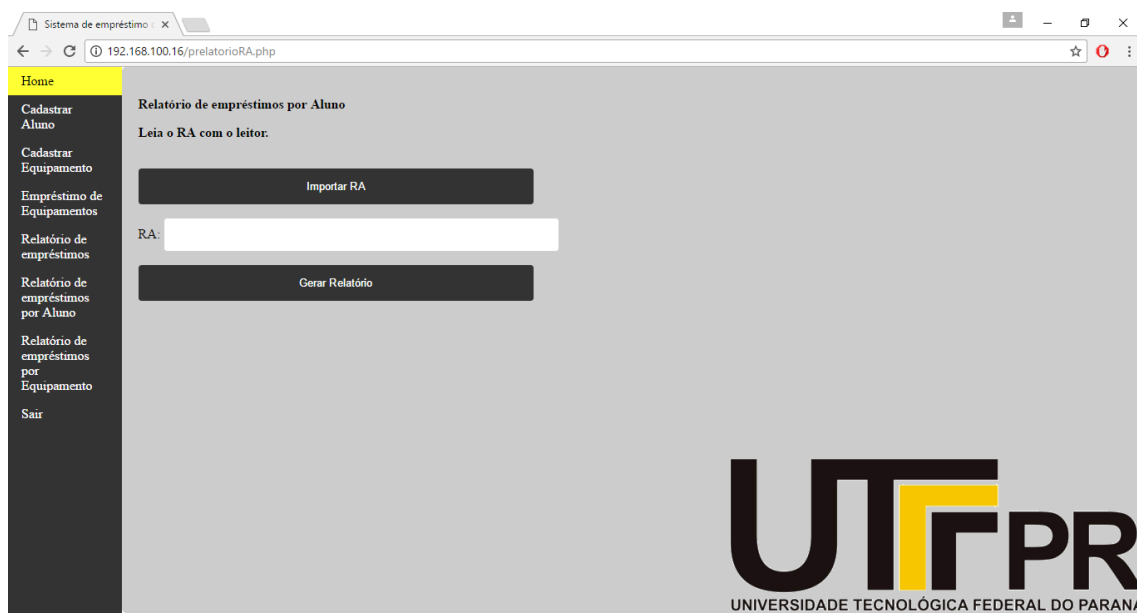


Figura 7 - Formulário para geração do relatório por aluno

Caso existam empréstimos para o aluno que foi inserido, uma tabela com todos os empréstimos é mostrada. Caso contrário, uma mensagem de erro é apresentada e o sistema retorna para o formulário.

6.2. Relatório por equipamento

Para gerar um relatório por equipamentos, deve-se acessar a opção *Relatório de empréstimos por equipamento* e inserir o patrimônio da mesma forma que o RA é inserido no relatório por aluno. O formulário para geração do relatório pode ser visualizado na figura 8.

The screenshot shows a web browser window with the URL `192.168.100.16/prelatorioPat.php`. The page title is "Relatório de Empréstimos por equipamento". Below the title, there is a text input field labeled "Patrimonio:" and two buttons: "Importar patrimônio" and "Gerar relatório". The sidebar menu on the left includes options like "Cadastrar Aluno", "Cadastrar Equipamento", "Empréstimo de Equipamentos", "Relatório de empréstimos", "Relatório de empréstimos por Aluno", "Relatório de empréstimos por Equipamento", and "Sair". The UTPR logo is displayed at the bottom right of the page.

Figura 8 - Formulário para geração de relatório por equipamento

Caso existam dados de empréstimos para o equipamento inserido, um relatório como o da figura 9 é mostrado. Detalhe que esse relatório mostra a quantidade de vezes que esse equipamento foi emprestado.

The screenshot shows a web browser window with the URL `192.168.100.16/relatorioPatrimonio.php`. The page title is "Relatório de empréstimos". Below the title, there is a table with the following data:

Aluno	Data retirada	Data devolução	Emprestado
1353675	2017-05-24 20:50:21	0000-00-00 00:00:00	SIM

Below the table, there is a text box stating "Equipamento já foi emprestado 1 vezes." and "Relatório gerado em: 2017-05-28 20:05:41". The sidebar menu on the left is the same as in Figure 8. The UTPR logo is displayed at the bottom right of the page.

Figura 9 - Formato do relatório por equipamentos

ANEXO 2

```

-- phpMyAdmin SQL Dump
-- version 4.2.12deb2+deb8u2
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: 18-Maio-2017 Ã s 20:06
-- VersÃ£o do servidor: 5.5.54-0+deb8u1
-- PHP Version: 5.6.29-0+deb8u1
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8 */;
--
-- Database: `almoxarifado`
--
-----
--
-- Estrutura da tabela `alunos`
--
CREATE TABLE IF NOT EXISTS `alunos` (
`id` int(11) NOT NULL,
`RA` double NOT NULL,
`nome` varchar(200) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=41 DEFAULT CHARSET=latin1;
--
-- Estrutura da tabela `dadosAux`
--
CREATE TABLE IF NOT EXISTS `dadosAux` (
`id` int(11) NOT NULL,
`patEmp` double NOT NULL,
`raAl` double NOT NULL,
`patEq` double NOT NULL,
`raEmp` double NOT NULL,
`raRel` double NOT NULL,
`patRel` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Estrutura da tabela `emprestimos`
--
CREATE TABLE IF NOT EXISTS `emprestimos` (
`id` int(11) NOT NULL,
`patrimonio` double NOT NULL,
`RA` double NOT NULL,

```

```

`DataHoraRetirada` datetime NOT NULL,
`DataHoraDevolucao` datetime NOT NULL,
`emprestado` tinyint(1) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=45 DEFAULT CHARSET=latin1;
--
-- Estrutura da tabela `equipamentos`
--
CREATE TABLE IF NOT EXISTS `equipamentos` (
  `id` int(11) NOT NULL,
  `patrimonio` double NOT NULL,
  `descricao` varchar(255) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1;
--
-- Estrutura da tabela `interface`
--
CREATE TABLE IF NOT EXISTS `interface` (
  `aluno` tinyint(1) NOT NULL,
  `equipamento` tinyint(1) NOT NULL,
  `emprestimo` tinyint(1) NOT NULL,
  `devolucao` tinyint(1) NOT NULL,
  `id` int(11) NOT NULL,
  `relatorioAl` int(11) NOT NULL,
  `relatorioEq` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Estrutura da tabela `usr`
--
CREATE TABLE IF NOT EXISTS `usr` (
  `usuario` varchar(30) NOT NULL,
  `senha` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Extraindo dados da tabela `usr`
--
INSERT INTO `usr` (`usuario`, `senha`) VALUES
('admin', 'admin123');
--
-- Indexes for dumped tables
--
--
-- Indexes for table `alunos`
--
ALTER TABLE `alunos`
ADD PRIMARY KEY (`id`,`RA`);
--
-- Indexes for table `dadosAux`
--
ALTER TABLE `dadosAux`
ADD PRIMARY KEY (`id`);
--
-- Indexes for table `emprestimos`
--

```

```

ALTER TABLE `emprestimos`
ADD PRIMARY KEY (`id`);
--
-- Indexes for table `equipamentos`
--
ALTER TABLE `equipamentos`
ADD PRIMARY KEY (`id`,`patrimonio`);
--
-- Indexes for table `interface`
--
ALTER TABLE `interface`
ADD PRIMARY KEY (`id`);
--
-- Indexes for table `usr`
--
ALTER TABLE `usr`
ADD PRIMARY KEY (`usuario`);
--
-- AUTO_INCREMENT for dumped tables
--
--
-- AUTO_INCREMENT for table `alunos`
--
ALTER TABLE `alunos`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=41;
--
-- AUTO_INCREMENT for table `emprestimos`
--
ALTER TABLE `emprestimos`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=45;
--
-- AUTO_INCREMENT for table `equipamentos`
--
ALTER TABLE `equipamentos`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=14;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT
*/;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS
*/;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
*/;

```


ANEXO 3

```

1 #include <wiringPi.h>
2 #include <wiringSerial.h>
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <string.h>
6 #include <mysql/mysql.h>
7
8 #define LOCALHOST "localhost"
9 #define USR "root"
10 #define SENHA "root"
11 #define DATABASE "almoxarifado"
12
13 #define QUERYAI "SELECT * FROM interface WHERE aluno = '1';"
14 #define QUERYEq "SELECT * FROM interface WHERE equipamento = '1';"
15 #define QUERYEmEq "SELECT * FROM interface WHERE equipamento = '2';"
16 #define QUERYEm "SELECT * FROM interface WHERE emprestimo = '1';"
17 #define QUERYDv "SELECT * FROM interface WHERE devolucao = '1';"
18 #define QUERYRelA "SELECT * FROM interface WHERE relatorioAI = '1';"
19 #define QUERYRelEq "SELECT * FROM interface WHERE relatorioEq = '1';"
20
21 MYSQL conexao;
22 unsigned char op;
23 unsigned int consultaOp();
24 void adAluno();
25 void adEquipamento();
26 void emprestimo();
27 void devolucao();
28 void relatorioRA();
29 void relatorioEq();
30
31
32 int main(){
33
34 mysql_init(&conexao);
35
36 if (mysql_real_connect(&conexao, LOCALHOST, USR, SENHA, DATABASE, 0, NULL, 0)){
37 printf("CONEXAO OK\n");
38 }
39 else{
40 printf("ERRO %d: %s", mysql_errno(&conexao), mysql_error(&conexao));
41 }
42
43 op = 0;
44 while (1){
45 op = consultaOp();
46 switch (op){
47 case 1:{
48 adAluno();
49 break;
50 }
51 case 2:{
52 adEquipamento();
53 break;

```

```

54 }
55 case 3:{
56 emprestimo();
57 break;
58 }
59 case 4:{
60 emprestimo();
61 break;
62 }
63 case 6:{
64 relatorioRA();
65 break;
66 }
67 case 7:{
68 relatorioEq();
69 break;
70 }
71 }
72 }
73 }
74 unsigned int consultaOp(){
75 MYSQL_RES *respAl, *respEq, *respEmp, *respEmpEq, *respDv, *respRelA, *respRelEq;
76 MYSQL_ROW *linhas;
77 MYSQL_FIELD *colunas;
78 int nAl, nEq, nEmp, nEmpEq, nDv, nRelA, nRelEq;
79 nAl = nEq = nEmp = nEmpEq = nDv = nRelA = nRelEq = 0;
80 int numFields = 0, numRows = 0, i = 0;
81 char *aux;
82
83
84 mysql_query(&conexao, QUERYAl);
85 respAl = mysql_store_result(&conexao);
86 nAl = mysql_num_rows(respAl);
87
88 mysql_query(&conexao, QUERYEq);
89 respEq = mysql_store_result(&conexao);
90 nEq = mysql_num_rows(respEq);
91
92 mysql_query(&conexao, QUERYEm);
93 respEmp = mysql_store_result(&conexao);
94 nEmp = mysql_num_rows(respEmp);
95
96 mysql_query(&conexao, QUERYEmEq);
97 respEmpEq = mysql_store_result(&conexao);
98 nEmpEq = mysql_num_rows(respEmpEq);
99
100 mysql_query(&conexao, QUERYDv);
101 respDv = mysql_store_result(&conexao);
102 nDv = mysql_num_rows(respDv);
103
104 mysql_query(&conexao, QUERYRelA);
105 respRelA = mysql_store_result(&conexao);
106 nRelA = mysql_num_rows(respRelA);
107
108 mysql_query(&conexao, QUERYRelEq);
109 respRelEq = mysql_store_result(&conexao);
110 nRelEq = mysql_num_rows(respRelEq);
111

```

```

112 if (nAl) return 1;
113 if (nEq) return 2;
114 if (nEmp) return 3;//{
115 if (nDv) return 4;
116 if (nEmpEq) return 5;
117 if (nRelA) return 6;
118 if (nRelEq) return 7;
119 }
120 void adAluno(){
121 char query[255], queryUp[255];
122 double ra;
123 int validaRA = 0;
124 ra = 0;
125 while (!validaRA){
126 scanf("%lf", &ra);
127
128 if (ra) validaRA = 1;
129 printf("%d", consultaOp());
130 }
131 sprintf(query, "UPDATE dadosAux SET raAl = '%.0lf' where dadosAux.id = '1';", ra);
132 if (mysql_query(&conexao, query)) printf("%s\n", mysql_error(&conexao));
133 sprintf(queryUp, "UPDATE interface SET equipamento = '0', emprestimo = '0',
devolucao = '0', aluno = '0' WHERE interface.id = '1';");
134 mysql_query(&conexao, queryUp);
135 op = 0;
136 }
137 void adEquipamento(){
138 char query[255], queryUp[255];
139 double patrimonio;
140 patrimonio = 0;
141 printf("Patrimonio equipamento:\n");
142 while(!patrimonio) {
143 scanf("%lf", &patrimonio);
144 }
145 sprintf(query, "UPDATE dadosAux SET patEq = '%.0lf' where dadosAux.id = '1';",
patrimonio);
146 if (mysql_query(&conexao, query)) printf("%s\n", mysql_error(&conexao));
147 sprintf(queryUp, "UPDATE interface SET equipamento = '0', emprestimo = '0',
devolucao = '0', aluno = '0' WHERE interface.id = '1';");
148 mysql_query(&conexao, queryUp);
149 op = 0;
150 }
151 void emprestimo(){
152 char query[255];
153 double patrimonio, ra;
154 patrimonio = ra = 0;
155 int op = 0;
156 printf("Patrimonio \n");
157 while(!patrimonio){
158 scanf("%lf", &patrimonio);
159 }
160 sprintf(query, "UPDATE dadosAux SET patEmp = '%.0lf' where dadosAux.id = '1';",
patrimonio);
161 if (mysql_query(&conexao, query)) printf("%s\n", mysql_error(&conexao));
162 while(op!=5){
163 op = consultaOp();
164 }
165 printf("RA do equipamento emprestado \n");

```

```

166 while(!ra){
167 scanf("%lf", &ra);
168 }
169 sprintf(query, "UPDATE dadosAux SET raEmp = '%.0lf' where dadosAux.id = '1';",
ra);
170 if (mysql_query(&conexao, query)) printf("%s\n", mysql_error(&conexao));
171 op = 0;
172 }
173 void relatorioRA(){
174 char query[255], queryUp[255];
175 double ra;
176 ra = 0;
177 while(!ra && op == 6) {
178 printf("RA: \n");
179 op = consultaOp();
180 scanf("%lf", &ra);
181 if (ra) op = 0;
182 }
183 sprintf(query, "UPDATE dadosAux SET raRel = '%.0lf' where dadosAux.id = '1';",
ra);
184 if (mysql_query(&conexao, query)) printf("%s\n", mysql_error(&conexao));
185 sprintf(queryUp, "UPDATE interface SET equipamento = '0', emprestimo = '0',
devolucao = '0', aluno = '0' WHERE interface.id = '1;");
186 mysql_query(&conexao, queryUp);
187 op = 0;
188 }
189 void relatorioEq(){
190 char query[255], queryUp[255];
191 double eq;
192 eq = 0;
193 while(!eq && op == 7) {
194 printf("RA: \n");
195 op = consultaOp();
196 scanf("%lf", &eq);
197 if (eq) op = 0;
198 }
199 sprintf(query, "UPDATE dadosAux SET eqRel = '%.0lf' where dadosAux.id = '1';",
eq);
200 if (mysql_query(&conexao, query)) printf("%s\n", mysql_error(&conexao));
201 sprintf(queryUp, "UPDATE interface SET equipamento = '0', emprestimo = '0',
devolucao = '0', aluno = '0' WHERE interface.id = '1;");
202 mysql_query(&conexao, queryUp);
203 op = 0;

```

204 }