

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET
CÂMPUS GUARAPUAVA

LUCAS ANDREY GROXKO JACOMEL

LIBSYSTEM : SISTEMA WEB PARA UMA BIBLIOTECA DE ESCOLA PÚBLICA ESTADUAL

TRABALHO DE CONCLUSÃO DE CURSO DO CURSO SUPERIOR EM TECNOLOGIA
EM SISTEMAS PARA INTERNET

GUARAPUAVA
2º Semestre de 2018

LUCAS ANDREY GROXKO JACOMEL

LibSystem: um sistema web para uma biblioteca de uma escola pública estadual

Monografia de Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Sistemas para Internet – TSI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Guarapuava, como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador : Prof. Dr. William Alberto Cruz Castañeda.

Co-orientador : Prof. Dr. Roni Fabio Banaszewski.

GUARAPUAVA
2º Semestre de 2018

ATA DE DEFESA DE MONOGRAFIA DE TRABALHO DE CONCLUSÃO DE CURSO DO CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

No dia **29 de novembro de 2018**, às 16:30 horas, em sessão pública nas dependências da Universidade Tecnológica Federal do Paraná Câmpus Guarapuava, ocorreu a banca de defesa da de Trabalho de Conclusão de Curso intitulada: **“Sistema Web de Bibliotecas Escolares”** do acadêmico **Lucas Andrey Groxko Jacomel** sob orientação do professor **Prof. Dr. William Alberto Cruz Castañeda** do curso de Tecnologia em Sistemas para Internet.

Banca Avaliadora	
Membro	Nome
Orientador	Prof. Dr. William Alberto Cruz Castañeda
Coorientador	Prof. Dr. Roni Fabio Banaszewski
Avaliador 1	Prof. Dr. Roni Fabio Banaszewski
Avaliador 2	Prof. Me. Paulo André Filipak

Situação do Trabalho	
Situação	<input checked="" type="checkbox"/> Aprovado <input type="checkbox"/> Aprovado com ressalvas <input type="checkbox"/> Reprovado <input type="checkbox"/> Não compareceu
Encaminhamento do trabalho para biblioteca	<input checked="" type="checkbox"/> Autoriza o encaminhado para biblioteca <input type="checkbox"/> Manter sigilo para publicação ou geração de patente

Guarapuava, 29 de novembro de 2018.

A folha de aprovação assinada encontra-se na coordenação do curso (ou programa).

RESUMO

Jacomel, Lucas Andrey Groxko. LibSystem: um sistema web para uma biblioteca de uma escola pública estadual. 62f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava 2018.

O município de Guarapuava possui 57 escolas da rede pública, localizadas dentro do perímetro urbano, sendo que 22 delas são estaduais e as 35 restantes municipais. Em relação a infraestrutura bibliotecária destas escolas, existem locais que não possuem sistema informatizado devido aos altos custos de aquisição e manutenção. Desta forma, o conceito de biblioteca eletrônica permite colocar à disposição dos usuários um sistema no qual os processos básicos são de natureza eletrônica e automatizados. O objetivo desse trabalho é desenvolver um sistema web de biblioteca escolar que atenda os requisitos do sistema bibliotecário de uma escola pública do município de Guarapuava. Para o desenvolvimento deste trabalho utiliza-se a metodologia Rational Unified Process (RUP) dividido em quatro etapas sendo elas a concepção, elaboração, construção e testes. Este trabalho contribuirá para que as escolas da rede pública de Guarapuava modernizem e organizem melhor seus processos bibliotecários.

Palavras-chave : Sistemas de computação. Bibliotecas. Escolas.

ABSTRACT

Jacomel, Lucas Andrey Groxko. LibSystem a web system for a library of a state public school. 62p. Completion of Course Work - Technology Course in Internet Systems, Federal Technological University of Paraná. Guarapuava 2018.

The municipality of Guarapuava has 57 public schools within the urban perimeter, in which 22 are state and 35 are municipal. Regarding the librarian infrastructure of these schools, there are places that do not have computerized system due to high acquisition and maintenance costs. Thus, the concept of electronic library allows to make available to users with a system in which basic processes are electronic and automated nature. The objective of this work is to develop a web system of school library that meets the requirements of a librarian system of a public school in the municipality of Guarapuava. For the development of this work, the Rational Unified Process (RUP) methodology is used and divided into four stages: design, elaboration, construction and testing. This work will help to modernize and organize Guarapuava's public schools in their librarian processes.

Key words: Computer systems. Libraries. Schools.

LISTA DE FIGURAS

Figura 1 – Funcionalidade de catalogação do OpenBiblio.....	14
Figura 2 – Interface gráfica de cadastro de leitores no BibLivre.....	15
Figura 3 – Interface gráfica de pendências do Meu Pergamum por usuários.....	16
Figura 4 – Representação do funcionamento da arquitetura cliente-servidor.....	18
Figura 5 – Equivalência entre tabelas e classes entities.....	22
Figura 6 – Fases e disciplinas da metodologia RUP.....	24
Figura 7 – Ciclo da metodologia RUP.....	25
Figura 8 – Protótipo da interface gráfica de login do sistema.....	29
Figura 9 – Protótipo da interface gráfica após logar com sucesso.....	30
Figura 10 – Protótipo da interface gráfica que dá acesso aos formulários de cadastro	30
Figura 11 – Protótipo da interface gráfica de cadastro de turmas.....	31
Figura 12 – Protótipo da interface gráfica de cadastro de alunos.....	32
Figura 13 – Protótipo da interface gráfica de cadastro de gêneros literários.....	32
Figura 14 – Protótipo da interface gráfica de cadastro de livros literários.....	33
Figura 15 – Protótipo da interface gráfica de cadastro de empréstimos.....	34
Figura 16 – Protótipo da interface gráfica de devolução.....	34
Figura 17 – Diagrama relacional da estrutura de banco de dados.....	39
Figura 18 – Interface gráfica de login.....	40
Figura 19 – Interface gráfica principal.....	41
Figura 20 – Interface gráfica de cadastro de turmas.....	43
Figura 21 – Interface gráfica de cadastro de alunos.....	44
Figura 22 – Interface gráfica de cadastro de empréstimos.....	46
Figura 22a – Interface gráfica de exibição dos livros do empréstimo ao clicar no botão VER.....	46
Figura 23 – Interface gráfica de adição de livros ao empréstimo.....	47
Figura 24 – Interface gráfica de devolução.....	48
Figura 25 – Relatório de livros disponíveis.....	48
Figura 26 – Relatório de livros emprestados.....	49
Figura 27 – Relatório de alunos que possuem livros emprestados.....	50

LISTA DE FIGURAS

Figura 28 – Gráficos das respostas da pergunta: Qual o nível de satisfação em relação ao sistema?.....	57
Figura 29 – Gráfico das respostas da pergunta: Avalie o processo de empréstimo.....	57
Figura 30 – Gráfico das respostas da pergunta: Avalie o controle de circulação dos livros literários.....	58
Figura 31 – Gráfico das respostas da pergunta: Avalie o controle de circulação dos livros didáticos.....	58
Figura 32 – Gráfico das respostas da pergunta: Avalie o design da interface.....	59
Figura 33 – Gráfico das respostas da pergunta: Avalie a usabilidade do sistema.....	59
Figura 34 – Gráfico das respostas da pergunta: Avalie o processo de gerenciamento de informação da biblioteca com um todo.....	60
Figura 35 – Gráfico das respostas da pergunta: Avalie a funcionalidade de cadastros....	60
.....	
Figura 36 – Gráfico das respostas da pergunta: Avalie a funcionalidade de empréstimos	
.....	61
Figura 37 – Gráfico das respostas da pergunta: Avalie e funcionalidade de devoluções..	
.....	61
Figura 38 – Respostas da pergunta: Avalie os relatórios de pendência de livros.....	62

LISTA DE QUADROS

Quadro 1 Requisitos e critérios de aceitação.....	26
Quadro 2 Tabelas que compõem o banco de dados do sistema evidenciando campo, utilidade e relacionamentos.....	36

LISTA DE SIGLAS

SIGLA	SIGNIFICADO
API	Application Programming Interface
CRUD	Create Read Update Delete
HQL	Hibernate Query Language
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
JEE	Java Enterprise Edition
JPA	Java Persistence API
JPQL	Java Persistence Query Language
JSF	Java Server Faces
JSP	Java Server Pages
MVC	Model View Controller
PHP	Hypertext Preprocessor
PUC-PR	Pontifícia Universidade Católica do Paraná
REGEX	Regular Expression
RUP	Rational Unified Process
SAB	Sistema de automação Bibliotecário
SE	Standart Edition
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structured Query Language
TCP	Transmission Control Protocol
TIC	Tecnologia da Informação e Comunicação
URL	Uniform Resource Locator
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language

SUMÁRIO

1. INTRODUÇÃO	10
1.1 OBJETIVOS	11
1.1.1 Objetivo Geral	11
1.1.2 Objetivos Específicos	11
2 RESENHA LITERÁRIA	12
2.1 ESTADO DA ARTE	12
2.1.1 Openbiblio	14
2.1.2 Biblivre	15
2.1.3 Pergamum	15
2.1.3.1 Pergamum na rede de biblioteca da prefeitura de Vitória-ES	17
2.2 TECNOLOGIAS CLIENTE-SERVIDOR EMPREGADAS	17
2.2.1 Java Enterprise Edition – JEE	18
2.2.2 Java Server Faces – JSF	19
2.2.3 Java Persistence API – JPA	19
2.2.4 Servidor Web Apache Tomcat	20
2.2.5 Servidor de banco de dados MySQL MariaDB	21
2.2.6 Framework – Hibernate	21
2.2.7 Bean Validator	22
2.3 METODOLOGIA RUP	23
3 ANÁLISE E PROJETO DO SISTEMA	26
3.1 ANÁLISE DO SISTEMA	26
3.2 PROJETO DO SISTEMA	29
3.2.1 Projeto do protótipo das interfaces gráficas do sistema	29
3.2.2 Projeto da base de dados do sistema	35
4 DESENVOLVIMENTO	40
4.1 FUNCIONALIDADE DE CADASTROS	42
4.1.1 Cadastro de turmas	42
4.1.2 Cadastro de classificação livros literários	42
4.1.3 Cadastro de alunos	43
4.1.4 Cadastro de livros literários	44

SUMÁRIO

4.2 FUNCIONALIDADE DE EMPRÉSTIMOS.....	45
4.2.1 Cadastro de empréstimos.....	45
4.2.1.1 Cadastro de livros do empréstimo.....	46
4.3 FUNCIONALIDADE DE DEVOLUÇÕES.....	47
4.3.1 Devolução.....	47
4.4 RELATÓRIOS.....	48
4.4.1 Livros Disponíveis.....	48
4.4.2 Livros Emprestados.....	49
5 RESULTADOS.....	50
6 CONCLUSÃO.....	52
6.1 TRABALHOS FUTUROS.....	53
7 REFERÊNCIAS.....	54
APÊNDICE A – QUESTIONÁRIO SOBRE TESTE DE SOFTWARE COM RESPOSTAS	57

1. Introdução

No mundo globalizado e na sociedade da informação, cujas tendências se sustentam pela acelerada inclusão de tecnologias da informação e comunicação (TICs), podemos perceber uma nova realidade em todos os setores e organizações da sociedade.

Especialmente no setor educacional público, as mudanças por essa inclusão das TICs criaram alterações nos hábitos de uso da informação de setores como o bibliotecário. Impulsionando a organização para a busca de um processo de modernização de suas estruturas e maior agilidade na prestação de serviços à comunidade escolar.

No contexto local do setor educacional público de Guarapuava, a cidade possui 57 escolas da rede pública, sendo 22 delas estaduais e 35 municipais. Porém existem algumas escolas desta rede que não possuem sua sede própria e dividem a infraestrutura com uma outra. Além disso, cada uma tem a sua biblioteca separada e não possuem um sistema de gerenciamento dos serviços biblioteconômicos.

O desenvolvimento deste trabalho está sendo dividido e executado em quatro etapas baseadas na metodologia *Rational Unified Process* (RUP), contribuindo para que as escolas da rede pública de Guarapuava modernizem e organizem a sua biblioteca.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver um sistema de gerenciamento web de bibliotecas para escolas públicas na cidade de Guarapuava.

1.1.2 Objetivos Específicos

- Obter e analisar requisitos para o sistema;
- Estruturar as funcionalidades do sistema segundo a análise de requisitos;
- Modelar e implementar o banco de dados do Sistema;
- Desenvolver todas as funcionalidades do sistema (autenticação, cadastros, empréstimos, relatórios, devoluções);
- Testar todas as funcionalidades do sistema;
- Validar o sistema;

2. Resenha Literária

2.1 Estado da Arte

A necessidade de estar atualizado e envolvido com avanços tecnológicos é de extrema importância para profissionais das escolas da rede pública, como, por exemplo, os bibliotecários. Se antes ficavam restritas aos limites físicos de uma biblioteca e de uma coleção, agora o uso difundido da tecnologia a serviço da informação transpõe barreiras físicas e institucionais (Da Silva; Dias, 2010).

Além da biblioteconomia, áreas ligadas à informação como a arquivologia e a ciência da computação trabalham em conjunto, já que estão intimamente relacionadas à construção, fluxo, organização e armazenagem das informações. (Da Silva; Dias, 2010).

Os sistemas desenvolvidos baseados neste perfil passam a ser chamados de Sistemas de Automação Bibliotecários (SABs). Os SABs são sistemas de bases de dados com uma finalidade específica, projetados para controlar atividades de uma biblioteca (Da Silva; Dias, 2010).

Os SABs baseados na filosofia *open-source* apresentam como um dos principais motivos para o seu constante crescimento de uso e desenvolvimento, a tendência da utilização do Sistema Operacional Linux.

Com tais fatos, os bibliotecários, que trabalham com a informação, como sua matéria-prima, é necessário que se adaptem com o processo de automação para melhoria dos serviços prestados pelas bibliotecas.

A automação do ambiente bibliotecário tem como principal objetivo oferecer ao usuário uma base de dados com informações de materiais adquiridos pelas escolas, de forma a facilitar seu acesso (Rodrigues; Prudêncio, 2009).

Se as bibliotecas escolares querem oferecer melhores serviços aos usuários e cumprir sua missão educacional, será necessário adaptar as tecnologias de acordo com às necessidades e quantidades de informação dispostas por meio da utilização de um SAB que atenda os recursos disponíveis sem se tornar obsoleto a longo prazo (Da Silva; Dias, 2010).

Diante disso, a biblioteca passa a assumir que deixará de ser considerada tradicional e passará a ser chamada de eletrônica. A biblioteca eletrônica é o termo utilizado que se refere ao sistema nos quais processos básicos da biblioteca são de natureza eletrônica (Da Silva; Dias, 2010).

Isto implica a ampla utilização de computadores e de suas facilidades na construção de índices on-line, busca de textos completos e na recuperação e armazenamento de registros (Nascimento, 2010).

Definir um SAB não é uma tarefa simples por que com o decorrer dos anos sempre haverá avanços na área de desenvolvimento de sistemas web, mas especialmente em softwares para automação de bibliotecas (Côrte et. al, 1999)

A escolha de um software exige, fundamentalmente, a análise das ferramentas disponíveis, funções, recursos, potencialidade e a parceria com a empresa fornecedora, ou seja, existem várias funções para cada tipo de biblioteca, já que as formas de trabalho e prestação de serviços variam de uma para outras (Côrte, et. al, 1999).

Malheiros (2013), analisa que o processo de informatização das bibliotecas no Brasil passou a viver uma nova fase, caracterizada pela disponibilidade de recursos avançados – máquinas e softwares de nova geração que chegaram ao país nos últimos anos - ,apontando como causa, tanto de defasagem como do novo surto de desenvolvimento.

Malheiros (2013) alerta para o fato de que não existe um sistema ideal e, mesmo que a escolha seja a mais acertada, poderá não atender completamente os requisitos funcionais e de *performance*, além de executar o backup desejado e as opções de proteção, a custo compatível com o orçamento disponível.

Por esse motivo é importante que a biblioteca determine seus próprios requisitos obrigatórios e solicite as operações desejáveis somente após certificar-se de que as funções básicas e necessárias estejam plenamente atendidas. Qualquer iniciativa de informatização de uma biblioteca ou centro de documentação deve identificar a cultura, missão, objetivos, programas de trabalho, características com relação a sua abrangência temática, serviços e produtos oferecidos (Côrte et. al, 1999).

2.1.1 Openbiblio

O Openbiblio é o software livre para administração de bibliotecas mais utilizado no Brasil. O Sistema é capaz de gerenciar com facilidade os processos de uma biblioteca que incluem administração, relatórios, cadastro e circulação.

Uma das suas principais características é seguir os padrões internacionais US MARC - padrão internacional para transferência de dados entre sistemas de bibliotecas (SLB, 2010).

Por ser criado em PHP (*Hypertext PreProcessor*) e ter sua base de dados em My SQL (*Structured Language Query*), o sistema permite ser acessado através de um navegador web. Com isso pode ser implantado de forma centralizada diretamente de um servidor web, podendo ter acesso de qualquer lugar do mundo, ou então instalado localmente em um computador rodando Linux ou Windows (SLB, 2010).

A Figura 1 apresenta a interface gráfica do OpenBiblio, com as opções de empréstimo, catalogação, administração, relatórios. Destaca-se o módulo de catalogação que permite adicionar e classificar o acervo

Los campos marcados con * son requeridos.

Añadir nuevo Bibliografía:

* Tipo de material:	audio tapes ▾
* Colección:	audiolibro infantil ▾
* Número de entrada:	<input type="text"/>
Mostrar en OPAC:	<input checked="" type="checkbox"/>
Campos USMARC:	
* Title:	<input type="text"/>
Remainder of title:	<input type="text"/>
Statement of responsibility, etc.:	<input type="text"/>
* Personal name:	<input type="text"/>

Figura 1 - Funcionalidade de catalogação do OpenBiblio.

2.1.2 Biblivre

Em 2005, a empresa Sociedade dos Amigos da Biblioteca Nacional sob a presidência de Paulo Marcondes Ferraz, propôs o projeto de desenvolvimento da versão de um conjunto de programas de computador conhecido como Biblivre ilustrado pela Figura 2, seu propósito era tentar informatizar bibliotecas dos mais variados portes e tão somente propiciar a comunicação entre elas (Alauzo et. al, 2014).

Nesse sentido, desde o início o desenvolvimento do sistema que é ilustrado pela Figura 2 foi previsto que este software fosse oferecido livremente às bibliotecas que desejam utilizá-lo, estudá-lo, modificá-lo de acordo com os termos de licença que o caracterizassem como software livre. Devido a esta característica, o projeto passou a se chamar Biblioteca Livre (Alauzo et. al, 2014).

Figura 2 – Interface gráfica de cadastro de leitores no BibLivre.

2.1.3 Pergamum

O sistema integrado de bibliotecas Pergamum é um software informatizado de gerenciamento de bibliotecas que foi desenvolvido por profissionais da área de processamento de dados em parceria com bibliotecários, da Pontifícia Universidade Católica do Paraná (PUC-PR). Ele contempla as principais funções de uma biblioteca,

funcionando de forma integrada, com o objetivo de facilitar a gestão dos centros de informação, bem como melhorar a rotina diária dos seus usuários (Pergamum, 2013).

Sua estrutura está dividida em nove módulos: Módulos de parâmetros, estatísticas, aquisição, catalogação, circulação, relatórios, estatísticas, consulta ao catálogo e internet (Pergamum, 2013).

O uso do Pergamum trouxe um grande avanço na tecnologia da informação para o trabalho em rede, pois facilitou o acesso à informação, visto que permitiu ao usuário e operador consultar o acervo das bibliotecas da rede e, em uma única busca (Pergamum, 2013).

Entre as suas principais vantagens destaca-se o módulo Web, por disponibilizar o catálogo em formato eletrônico, apresentando recursos facilitadores no qual o usuário obtém prontamente as informações sobre o livro e/ou autor desejado. Além disso, a pesquisa é de fácil acesso sendo possível recuperar a informação por qualquer identificador da obra: autor, título, número de registro entre outros (Pergamum, 2013).

A Figura 3 mostra o módulo de pendências o qual verifica os livros que os usuários possuem e não foram devolvidos. Além disso mostra quantas renovações é permitido o usuário efetuar.

The screenshot shows the 'Meu Pergamum' user interface. The top navigation bar includes the UFRPR logo, the user's name, and a 'Logout' button. A sidebar on the left lists services such as 'Empréstimo', 'Perfil de interesse', 'Solicitações', 'Comentários', 'Dados pessoais', and 'Alteração de senha'. The main content area is titled 'Títulos pendentes' and contains a table with the following data:

	Data Devolução	Nº Renovações / Limite	Multas parciais (R\$)
• Desenvolvendo aplicações web com JSP, Servlets, Javasever faces, Hibernate, EJB 3 persistence e AJAX / 2007 - Livros	09/06/2017	0 / 3	0
• Java Web Services : implementando / 2010 - Livros	09/06/2017	0 / 3	0
• PostoreSQL : guia do programador / 2008 - Livros	05/06/2017	0 / 3	0
• SQL : um guia para iniciantes - 3. ed. / 2009 - Livros	05/06/2017	0 / 3	0

Below the table, there are sections for 'Títulos reservados' and 'Cesta Permanente'. The footer of the page indicates '© 2000 - 2014. Pergamum. Todos os direitos reservados.'

Figura 3 – Interface gráfica de pendências do meu pergamum por usuários.

2.1.3.1 Pergamum na rede de bibliotecas da prefeitura de Vitória-ES

Segundo Pereira et. al (2013) a rede Pergamum é considerada uma das melhores do Brasil, sendo referência para todo país no que se refere à prestação de serviços no âmbito bibliotecário.

As bibliotecas que fazem parte dessa rede ainda não possuem um sistema gerenciador de serviços biblioteconômicos, fato que estimulou os profissionais bibliotecários a solicitarem junto a secretaria de educação, a automação das bibliotecas por meio de um sistema seguro e efetivo (Pereira et. al, 2013).

O processo de implantação do sistema de automação e administração de bibliotecas escolares se inicia com o processo de articulação na Secretaria de Educação em função do diagnóstico realizado juntos às bibliotecas das Escolas Municipais de Ensino Fundamental. Tal análise vem sendo realizado desde o processo inicial de revitalização dos espaços escolares no ano de 1999. Desde então, os bibliotecários atuantes na rede vêm apontando a necessidade eminente de aquisição de um sistema de automação e administração de bibliotecas, pois várias tentativas de aquisição e desenvolvimento de um sistema já foram feitas.

2.2 Tecnologias cliente-servidor empregadas

Nesse capítulo são abordadas as diversas tecnologias que permitiram o desenvolvimento desse projeto como o Java EE, JSF, JPA entre outras.

Em uma arquitetura cliente-servidor há um servidor sempre em funcionamento, que atende as requisições dos clientes.

Um exemplo clássico é a aplicação web na qual um servidor web que atende a requisições de *browsers* de clientes. Na Figura 4 é ilustrado que quando recebe-se uma requisição de um cliente, o servidor web responde enviando o objeto requisitado a ele. Observa-se que os clientes não se comunicam diretamente uns com os outros.

O HTTP É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na *World Wide Web* (WWW). Para que a transferência de dados na Internet seja realizada, o protocolo HTTP necessita estar agregado a outros dois protocolos de rede: TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*).

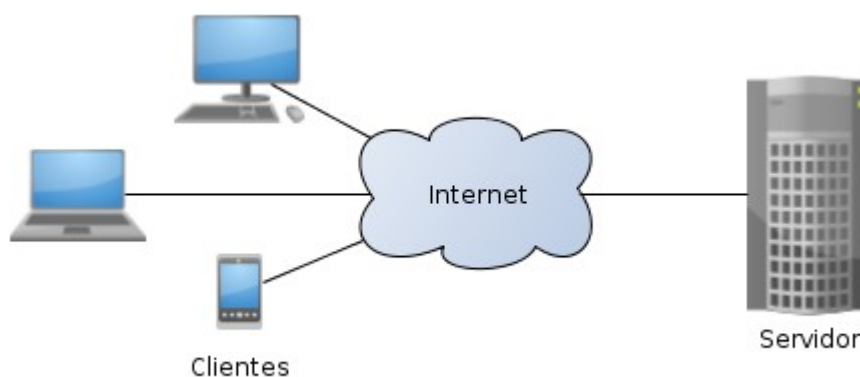


Figura 4 – Representação do funcionamento da arquitetura cliente-servidor.

2.2.1 Java Enterprise Edition – JEE

É uma plataforma *open-source* multicamadas, centrada no servidor, que utiliza um conjunto de tecnologias (Java Server Faces, Servlets) que reduzem a complexidade do desenvolvimento, implantação e gerenciamento de aplicações, auxiliando na utilização de regra de negócio e também na persistência no banco de dados (Enterprise Java Beans - EJB).

Alem disso, suporta conexões com o banco de dados, e possibilita flexibilidade da arquitetura do projeto de forma simples (Paz; Dias, 2016).

O JEE é construído sobre a plataforma Java SE (*Standart Edition*) e oferece um conjunto de APIs (*Application Programming Interface*) para desenvolvimento e execução de aplicações portáteis, robustas, escaláveis, confiáveis e seguras no lado do servidor.

As principais APIs disponibilizadas pelo JEE para desenvolvimento de aplicações cliente-servidor são: Java Server Pages (JSP), Java Servlets, Java Server Faces (JSF), Enterprise Java Beans (EJB), Java Persistence (API).

O JEE foi escolhido para ser utilizado no desenvolvimento desse projeto por ser uma plataforma do Java de fácil compreensão e possuir um conjunto de recursos que outras linguagens e plataformas já estudadas não possuem e um ambiente de desenvolvimento fácil de ser instalado e configurado em diferentes sistemas operacionais.

2.2.2 Java Server Faces – JSF

Após os Servlets, algumas tecnologias foram criadas com o intuito de simplificar o desenvolvimento de aplicações Web. Parte desse esforço procurava oferecer uma melhor arquitetura, pois os Servlets concentravam muitas responsabilidades em um único artefato.

JSF trouxe uma abordagem voltada para interfaces de usuário, disponibilizando para o programador uma rica coleção de componentes, bem como uma melhor separação entre as camadas da aplicação. Com o JSF, o programador pode escrever views em XHTML e através de *Data Binding* conectá-las a classes Java

Um dos termos mais comuns quando se trabalha com essa tecnologia é o Managed Bean, ou bean gerenciado que é definido como o controller do MVC. É ele o responsável por fazer a ligação entre a visão (as páginas XHTML) e o modelo do sistema (NetBeans, 2018).

Foi usado no projeto por que é possível fazer a integração entre páginas web e classes Java ajudando na organização do código separando-o em 3 camadas distintas por meio do padrão MVC. Nesse padrão as páginas de XHTML equivale as views, os Beans Gerenciados aos Controllers e o banco de dados ao Model.

2.2.3 Java Persistence API – JPA

A JPA é baseada no conceito de Plain Old Java Object (POJOs), que incorpora ideias de renomados *frameworks* de persistência para padronizar o mapeamento O/R em Java.

Entre as principais inclusões na JPA destacam-se:

- **POJOS Persistentes:** Talvez o aspecto mais importante da JPA seja o fato que os objetos são POJOs, significando que os objetos possuem design simples que não dependem da herança de interfaces ou classes de frameworks externos. Qualquer objeto com um construtor default pode ser feito a persistência sem nenhuma alteração numa linha de código. Mapeamento Objeto-Relacional com JPA é inteiramente dirigido a metadados. Isto pode ser feito através de anotações no código ou através de um XML definido externamente.

- **Consultas em Objetos:** As consultas podem ser realizadas através da Java Persistence Query Language (JPQL), uma linguagem de consulta que é derivada do EJB SQL e transformada depois para SQL. As consultas usam um esquema abstraído que é baseado no modelo de entidade como oposto às colunas na qual a entidade é armazenada.
- **Configurações simples:** Existe um grande número de características de persistência que a especificação oferece, todas são configuráveis através de anotações, XML ou uma combinação das duas. Anotações são simples de usar, convenientes para escrever e fácil de ler. Além disso, JPA oferece diversos valores defaults, portanto para já sair usando JPA é simples, bastando algumas anotações.

A partir do arquivo persistence.xml é possível editar as configurações padrões de acordo com as particularidades do usuário, como a URL da base dados onde serão os registros do projeto, a senha e o nome de usuário do MySQL. Esse arquivo apresenta uma linha de código para definir a criação automática de tabelas que pode ser none (não criando tabelas automaticamente, muito usado após o hibernate já ter criado as tabelas e na ocasião de não precisar alterar a estrutura do banco), create (muito usado na primeira execução do projeto, após deletar o banco devido a modificações na estrutura), drop-and-create (que é deletado e criado novamente após a execução, muito usado para fazer testes). Foi usado no trabalho para realizar e persistir os dados durante as operações de CRUD.

2.2.4 Servidor Web Apache Tomcat

Robusto e eficiente de aplicações Java para Web a ser utilizado num ambiente de produção, é capaz de atuar também como um servidor web ou integrado a um servidor web que provê um servidor web HTTP puramente em Java, mais especificamente um contêiner de servlet. Desenvolvido pela Apache Software Foundation, é distribuído como software livre como a implementação de referência para as tecnologias JavaServlet e JSP (Vukotic; Goodwill, 2011).

Esse servidor foi escolhido para ser utilizado nesse trabalho por ser um servidor adequado as tecnologias java, possuir uma ampla documentação e muito utilizado em aplicações JEE.

2.2.5 Servidor de banco de dados MySQL MariaDB

É um servidor de banco de dados relacional que se aproxima cada vez mais das facilidades em produtos proprietários, independente do sistema operacional que for utilizado (Gilmore, 2011).

O MySQL oferece diversos tipos de mecanismos para gerenciamento de dados, conhecidos como mecanismo de armazenamento. Cada mecanismo ostenta seus próprios pontos fortes e fracos, e devem ser aplicados de forma seletiva para acomodar o melhor uso que se espera para os seus dados (Gilmore, 2011).

Além disso permite a indexação e busca full-text como um recurso que melhora o desempenho da mineração de dados a partir das colunas baseadas em texto. Com esse recurso é possível produzir resultados na ordem de relevância, de acordo com a proximidade com que a *query* casa as colunas de texto indexadas na linha (Gilmore, 2011).

O MySQL foi escolhido para o desenvolvimento devido sua fácil integração com diversas linguagens de programação e de fato por ser um SGBD gratuito e popular, contendo uma ampla documentação.

2.2.6 Framework - Hibernate

É um projeto que oferece uma solução para o problema de gerenciamento de dados persistentes em Java. Se relaciona com o banco de dados, onde esse relacionamento é conhecido como mapeamento objeto/relacional para java, deixando o desenvolvedor livre para se concentrar em problemas da lógica do negócio.

Sua simplicidade em configuração, dá ao desenvolvedor algumas regras para que sejam seguidas como padrões de desenvolvimento ao escrever sua lógica de negócio e suas classes persistentes (Gonçalves, 2007).

O Hibernate disponibiliza para seus usuários um mecanismo de consulta de dados, que posteriormente diminui o tempo de desenvolvimento que seria gasto em

uma determinada aplicação. Por utilizar sua própria linguagem de consulta o Hibernate Query Language (HQL) disponibiliza a função de consulta e retorno de dados, fazendo a conversão entre os registros permitindo que a aplicação utilize qualquer banco de dados SQL (Magalhães; Silva, 2013).

O benefício de utilização dessa tecnologia no projeto é fato do desenvolvedor não precisar criar todas as tabelas que compõem a estrutura do banco manualmente. As tabelas são criadas a partir das classes que recebem a anotação “@Entity” e devem possuir um atributo com a anotação @Id porém seu nome não precisa ser necessariamente “id”. Como ilustrado na Figura 5 as classes com anotação “@Entity” são equivalentes a tabelas do banco de dados, os atributos dessas classes equivalentes a um campo dessa tabela e os registros do banco são objetos.

```
@Entity
@Table(name = "turmas")
public class Turma implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    //TODO mudar para enum ou int
    @NotEmpty(message = "DIA DA SEMANA - nao pode ser vaziao")
    private String dia;

    @NotEmpty(message = "HORARIO - nao pode ser vaziao")
    private String horario;
```

Figura 5 – Equivalência entre tabelas e classes entitis.

2.2.7 Bean Validator

É uma especificação Java para tratar validações de dados de forma centralizada, pois as validações são inseridas no próprio modelo através de anotações, assim possibilitando a consistência das informações em diferentes camadas além de possuir integrações com outras especificações como JSF, JPA, Hibernate (Neri, 2017).

Esse recurso foi utilizado no projeto para que vários campos de diversos formulários fossem validados fazendo com que o banco de dados não salve um novo objeto caso um ou mais campos contenham dados inválidos. É possível criar mensagens personalizadas ao usuário alertando e informando qual é o campo e

porque as informações estão inválidas. Permite emitir mensagens de alerta por cada campo quando o usuário estiver tentando salvar um objeto contendo informações inválidas. Se todas as informações do objeto estiver válidas esse mesmo é salvo direto no banco de dados. A partir desse recurso é possível validar um tamanho mínimo e máximo de dados que o campo pode suportar, no caso se o campo permitir que apenas números inteiros sejam dados válidos é possível definir um intervalo de valores válidos e também criar validações de campos que possuem um mesmo formato padrão por meio das REGEX.

2.3 Metodologia RUP

O Rational Unified Process (RUP), é um *framework* de processo da engenharia de *software* que fornece práticas testadas na indústria de software e gerência de projetos. Por ser um *framework* de processo, pode ser customizado conforme as necessidades organizacionais e do projeto, dessa forma, pode-se trabalhar um RUP mais “leve e ágil” ou “mais pesado”.

O RUP permite que a equipe do projeto escolha as atividades e os artefatos para serem produzidos, reduzindo assim, o excesso de documentação para torná-lo mais ágil.

A figura 6 ilustra de forma gráfica a metodologia RUP dividida nas seguintes fases:

- **Concepção/inicialização:** Identifica e descreve quais recursos e funções compõem o sistema. Foi necessário realizar uma entrevista com um profissional responsável por uma biblioteca para obter informações iniciais assim como ter conhecimento de quais módulos irão compor o sistema e a modelagem do banco de dados. Foram utilizados recursos como a UML para modelagem dos diagramas do sistema e Balsamiq Mockup para prototipação de interfaces (Pressman, 2011).
- **Elaboração :** Envolve as atividades de obtenção e análise de requisitos. Refina e expande as informações da fase anterior realizando o levantamento de requisitos funcionais e não funcionais assim como a construção dos modelos de análise. Além disso inclui a modelagem rápida do projeto que envolve a

estruturação do banco de dados, dos módulos do sistema que irão compor o sistema. Esta etapa demonstra a viabilidade da arquitetura mas não oferece os recursos e funções necessárias para usar o sistema. Foi utilizado a ferramenta MySQL workbench que facilitou a modelagem e as modificações da estrutura do banco de dados (Pressman, 2011).

- **Construção:** Compreende a codificação, testes e desenvolvimento de protótipos dos módulos do sistema com integração com o banco de dados. Modelos de requisitos são contemplados nesta fase. Implementa recursos e funções necessárias do sistema. Além disso, realizam-se atividades de integração (montagens de componentes e teste de integração) de todas as funcionalidades, módulos do sistema e do banco de dados (Pressman, 2011).
- **Validação/Transição** – Compreende os últimos estágios da atividade da construção do sistema e a primeira parte da atividade de emprego genérico. Valida-se o software com usuários finais para testes beta e o feedback dos usuários relataram defeitos e mudanças necessárias (Pressman, 2011).

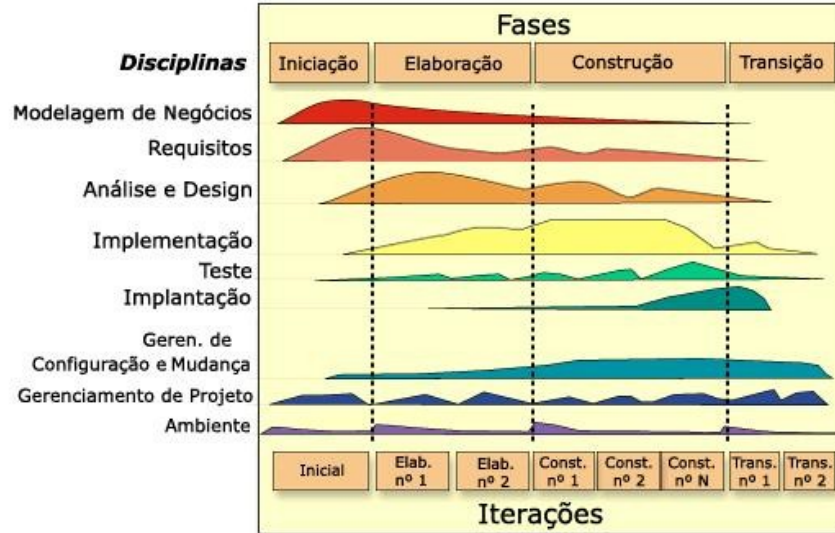


Figura 6 – Fases e disciplinas da metodologia RUP (Pressman, 2011).

Outra característica interessante dele é a aplicação do modelo de ciclo de vida iterativo e incremental ilustrado na Figura 7. Neste ciclo cada iteração, parte do software é desenvolvida, sendo os artefatos da nova iteração superior à iteração anterior.

O desenvolvimento iterativo e incremental permite aos desenvolvedores o aprendizado em relação ao software, possibilitando assim, a localização de futuros problemas em fases iniciais.

Em relação as disciplinas gerência de configuração e mudança, gerenciamento de projeto e ambiente não foram empregadas no desenvolvimento desse projeto.



Figura 7 – Ciclo da metodologia RUP (Pressman, 2011).

O RUP é descrito a partir de três perspectivas. Na perspectiva dinâmica, o RUP identifica o ciclo de desenvolvimento do projeto em quatro fases sequenciais sendo, cada fase, finalizada por um marco principal. As fases do RUP são iniciação, elaboração, construção e transição. A Figura 6 apresenta as fases do RUP (Bessa; Barros, 2009).

A escolha da metodologia RUP no projeto deu-se ao fato de ser uma metodologia de desenvolvimento ágil dividida em fases sequenciais, sendo cada uma dependente da fase anterior e possível de ser divididas em várias disciplinas.

3 Análise e projeto do sistema

Esse capítulo apresentará as funcionalidades necessárias para construção do sistema que foram coletadas durante uma entrevista que foi realizada com o profissional responsável pela biblioteca do estabelecimento visitado. A partir da coleta desses requisitos por meio dessa entrevista, o desenvolvimento desse projeto foi dividido em 4 etapas sequenciais através da metodologia RUP que está sendo utilizada para desenvolvê-lo.

3.1 Análise do sistema

Nessa seção irá ser apresentada uma documentação gerada a partir da entrevista realizada com o responsável pela biblioteca. As histórias foram coletadas por meio da entrevista realizada e poderão ser vistas no quadro abaixo:

Requisito	Critérios de aceitação
Como usuário gostaria de acessar o sistema	<ul style="list-style-type: none"> Os usuários são cadastrados diretamente no banco de dados. Se os dados digitados pelo usuário de login e senha estiverem de acordo com as informações contidas no banco de dados o usuário poderá acessar o sistema e será encaminhado a página principal, caso contrário o usuário permanecerá na página de login.
Como usuário gostaria de cadastrar todas as turmas do colégio	<ul style="list-style-type: none"> O campo nome não pode ser vazio, o primeiro dígito deve ser formado por qualquer número e o último por qualquer letra do alfabeto, o tamanho padrão deve ser 2, o nome deve ser único, caso for cadastrado uma turma com o nome existe, esse registro é ignorado.

Quadro 1 – Requisitos e critérios de aceitação

Requisito	Critérios de aceitação
	<ul style="list-style-type: none"> • O campo horário contem um <i>select</i> contendo opções fixas de (1, 5), esse campo deve ser preenchido obrigatoriamente • O campo dia da semana contem um <i>select</i> contendo opções fixas de (Segunda a Sexta), esse campo deve ser preenchido obrigatoriamente. • O campo turno contem um <i>select</i> contendo opções fixas (Manhã e Tarde), esse campo deve ser preenchido obrigatoriamente. • O campo nome do professor de português não pode ser vazio, só pode conter letras tanto maiúscula e minúscula e acentuadas e espaços.
<p>Como usuário gostaria de cadastrar todas as classificações de livros disponíveis na biblioteca</p>	<ul style="list-style-type: none"> • O campo nome não pode ser vazio • O campo cor não pode ser vazio
<p>Como usuário gostaria de cadastrar todos os alunos da escola</p>	<ul style="list-style-type: none"> • Campo nome deve conter somente letras e espaços • Se não tiver alguma turma cadastrada não é possível o cadastrar aluno, pois o campo não permite valor nulo. • Campo e-mail não pode ser nulo e deve seguir uma máscara padrão criada pela REGEX. • Campo matrícula não pode ser nulo e deve ser único • Nome do alunos devem ser exibidos em uma tabela em ordem alfabética

Continuação Quadro 1 - Requisitos e critérios de aceitação

Requisito	Critérios de aceitação
<ul style="list-style-type: none"> • Como usuário gostaria de cadastrar todos os livros literários 	<ul style="list-style-type: none"> • O campo código deve ser único • Nome dos livros devem ser mostrados em uma tabela em ordem alfabética.
<ul style="list-style-type: none"> • Como usuário gostaria de cadastrar empréstimos para os alunos 	<ul style="list-style-type: none"> • Não pode possuir nenhum campo vazio; • Só poderão ser adicionado no máximo 4 livros por empréstimo; • Só serão exibidos no select em ordem alfabética os livros cujo sua disponibilidade seja igual a T. • Após de ser realizado um empréstimo a disponibilidade dos livros é modificada para F, assim desaparecendo da lista e não sendo possível ser emprestado para um aluno que realizar um futuro empréstimo enquanto o aluno que emprestou não devolver.
<ul style="list-style-type: none"> • Como usuário desejo efetuar a devolução de um livro 	<ul style="list-style-type: none"> • A partir de um select é exibida uma lista de todos livros ordenados em ordem alfabética cujo a disponibilidade seja F, quando um livro é selecionado e usuário clica no botão de confirmação, esse livro volta a ficar disponível para um futuro empréstimo e desaparece dessa lista.
<ul style="list-style-type: none"> • Como usuário desejo visualizar um relatório de livros que estão emprestados 	<ul style="list-style-type: none"> • A partir desse um relatório esses livros poderão ser visualizados por título, nome do aluno que está com posse e a turma que esse aluno pertence.

Continuação Quadro 1 - Requisitos e critérios de aceitação

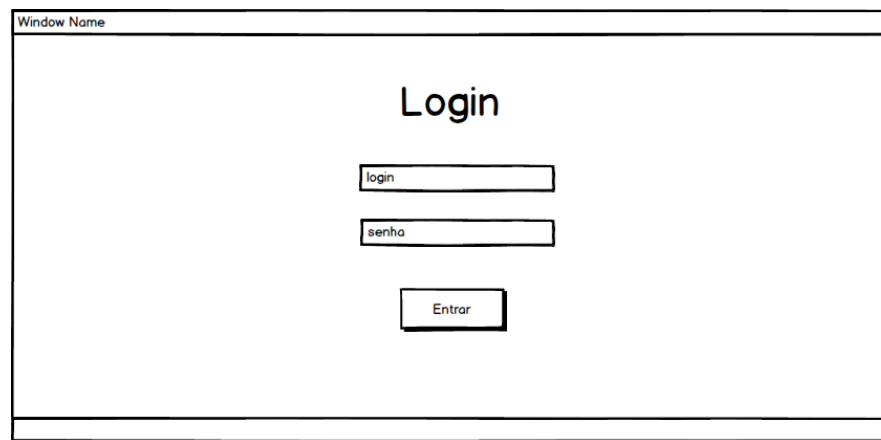
3.2 Projeto do sistema

Nessa seção é apresentado um protótipo das interfaces gráficas e a estrutura do banco que deram origem a esse sistema.

3.2.1 Projeto do protótipo das interfaces gráficas do sistema

Os protótipos das interfaces gráficas desse sistema foram construídos utilizando a ferramenta Balsamiq Mockup. Essa ferramenta fornece vários recursos e componentes que dão ao desenvolvedor uma noção de como será o produto final antes e durante seu desenvolvimento. Apesar do *design* das interfaces gráficas implementadas não saírem iguais ao desenho devido ao tempo disponível para o desenvolvimento desse sistema, suas funcionalidades continuaram sendo as mesmas, porém algumas tiveram a suas funcionalidades reduzidas conforme os desenhos e devido sua complexidade de desenvolvimento e exigir muito tempo de dedicação do que o tempo disponível.

Na figura 8 é possível visualizar a interface gráfica de login do sistema. Essa foi uma das interfaces gráficas mais simples a ser desenvolvida, será visto mais adiante que ficou muito próxima a realidade, ela possui apenas 2 campos um que o usuário digita seu nome ou e-mail e outro para digitar sua senha e um botão de entrar. Se os dados digitados após clicar nesse botão corresponderem a dados existentes no banco de dados o usuário é encaminhado para a interface gráfica da figura 9 tendo acesso a todas as funcionalidades, caso contrário o usuário permanece nessa mesma interface gráfica.



The image shows a wireframe of a login window. At the top left, there is a label 'Window Name'. The main content area is centered and contains the word 'Login' in a large font. Below it are two input fields: the first is labeled 'login' and the second is labeled 'senha'. Below these fields is a button labeled 'Entrar'.

Figura 8 – Protótipo da interface gráfica de login do sistema.

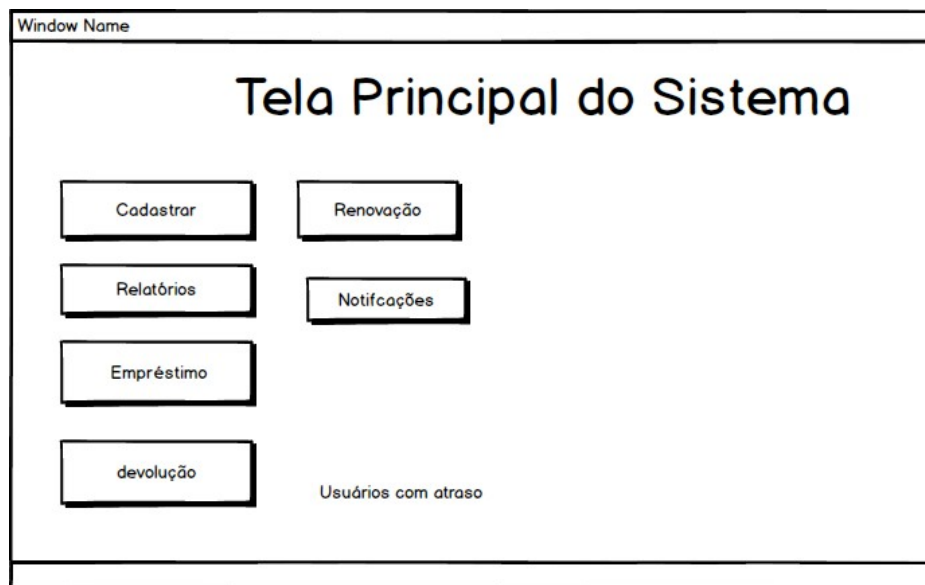


Figura 9 – Protótipo da interface gráfica após login com sucesso.

A figura 10 mostra a interface gráfica de cadastro, que quando o usuário clicar em algum desses botões ele é encaminhado a um formulário de cadastro relacionado ao que escrito no botão que serão mostrados nas próximas figuras.

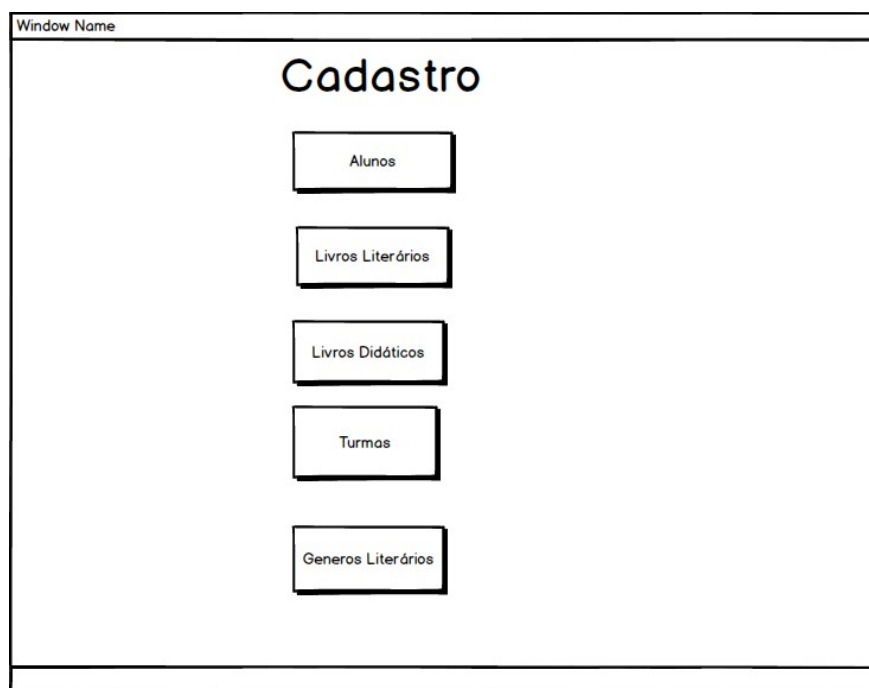


Figura 10 – Protótipo da interface gráfica que dá acesso aos formulários de cadastro.

A figura 11 mostra um protótipo da interface gráfica de cadastro de turmas. Nessa interface gráfica é possível cadastrar uma nova turma se todas as informações estiverem válidas as quais serão futuramente vinculadas aos alunos, se o usuário tentar salvar dados inválidos é emitido uma mensagem por campo alertando o por que o dado entrado está inválido. O cadastro de turmas é simples possuindo apenas 3 campos (nome o qual deve ser único, dia da semana para emprestar livros, nome do professor de português).

Window Name

Formulário de Cadastro de Turmas

Turma

Dia de emprestar livro

Professor de Português

Figura 11 – Protótipo da interface gráfica de cadastro de turmas.

A figura 12 mostra um protótipo da interface gráfica de cadastro de alunos. Nessa interface gráfica é possível cadastrar um novo aluno se todas as informações estiverem válidas, caso o usuário tente salvar dados inválidos é emitido uma mensagem por campo alertando o por que o dado entrado está inválido. No “comboBox” ao lado do texto turma é exibido todas as turmas que foram cadastradas e todo o aluno que for cadastrado deverá estar vinculado a uma turma.

Group Name

Formulário de Cadastro de alunos

Nome

Código

E-mail

Celular

Turma

Nome da mãe

Nome do pai

Figura 12 – Protótipo da interface gráfica de cadastro de alunos.

A figura 13 mostra um protótipo da interface gráfica de cadastro de gêneros literários. Nessa interface gráfica é possível cadastrar um novo gênero se todas as informações estiverem válidas as quais serão futuramente vinculadas aos livros literários, se o usuário tentar salvar dados inválidos é emitido uma mensagem por campo alertando o por que o dado entrado está inválido. O cadastro de gêneros é simples possuindo apenas 2 campos (nome, cor da etiqueta que é representado).

Window Name

Formulário de cadastro de gêneros literários

Gênero literário

Cor da etiqueta

Figura 13 – Protótipo da interface gráfica de cadastro de gêneros literários.

A figura 14 mostra um protótipo da interface gráfica de cadastro de livros literários. Nessa interface gráfica é possível cadastrar um novo livro literário se todas as informações estiverem validas, caso o usuário tente salvar dados inválido é emitido uma mensagem por campo alertando o por que o dado entrado está inválido. No “comboBox” ao lado do texto gênero é exibido todas os gêneros que foram cadastradas e todo o livro que for cadastrado deverá estar vinculado a um gênero.

Window Name

Formulário de Cadastro de Livros

Título

Autor

Gênero Literario

Número de páginas

Edição Ano


ISBN


Figura 14 – Protótipo da interface gráfica de cadastro de livros literários.

A figura 15 mostra um protótipo da interface gráfica de empréstimo de livros, onde é exibido 2 “inputs”, um que permite pesquisar por um aluno que está cadastrado e outro que permite pesquisar os livros cadastrados cujo o campo disponibilidade seja = 'T', quando um livro é adicionado ao empréstimo o valor desse campo é alterado para 'F' e volta para 'T' quando ele é devolvido, e dois campos *date* que são controlados manualmente, porém não impede que os empréstimos sejam realizados em datas pertencentes ao passado e ao futuro e a data de devolução ser passado a data do empréstimo.

Formulário de empréstimos

Window Name

Data de empréstimo 

Data prevista para devolução 

Aluno

Buscar Livros

Id	Titulo	Devolução

Figura 15 – Protótipo da interface gráfica de cadastro de empréstimos.

A figura 16 mostra um protótipo da interface gráfica de devolução onde é exibido um *input* que o usuário digita o código do livro, se o sistema encontrar um livro com o código que foi digitado esse livro voltará para o *select* da interface gráfica de empréstimo. A devolução não é registrada em uma tabela do banco de dados ela apenas muda a disponibilidade de um livro de 'F' para 'T', por que os alunos podem devolver livros de empréstimo diferente num mesmo momento ou devolver de um empréstimo em diferentes momentos assim ficando difícil serem controladas.

Window Name

Devolução

Digite o Código do Livro

Figura 16 – Protótipo da interface gráfica de devoluções.

3.2.2 Projeto da Base de Dados do Sistema

A base de dados é uma das principais partes de um sistema web, pois é nela que serão armazenados todos os dados dos cadastros que já foram realizados no sistema e também serve para fazer buscas desses dados que estão armazenados no servidor MySQL. Essa base precisa ser bem projetada antes do desenvolvimento, pois muitas vezes ocorre a necessidade de modificar a estrutura da base de dados durante o desenvolvimento do sistema, como por exemplo ter a necessidade de acrescentar novos campos às tabelas ou novas tabelas ao banco, ou eliminar campos ou tabelas existentes que estão sendo redundantes ou desnecessários após o desenvolvimento da aplicação.

Para que todos os cadastros realizados possam ser armazenados localmente na máquina que está sendo utilizada, foi instalado o servidor MySQL, o qual foi discutido na seção 2.3.6.

A base de dados do projeto é composta por 11 tabelas (users, roles, turmas, alunos, empréstimos_literarios, livros_literarios, classificacoes, empréstimos_literarios_livros_literarios, empréstimos_didaticos_livros_didaticos, livros_didaticos, empréstimos_didaticos), sendo que 2 delas (roles, users) não possuem relacionamento com outras tabelas, e são utilizadas na autenticação de usuários. Nesse projeto existem 2 relacionamentos (N x N), entre as tabelas, que serão descritos a seguir:

- (livros_literarios x empréstimos_literarios) : Existe a tabela intermediária (empréstimos_literarios_livros_literarios) que armazena as chaves estrangeiras dessas 2 tabelas durante a adição de livros literários ao empréstimo.
- (livros_literarios x empréstimos_didaticos) : Existe a tabela intermediária (empréstimos_didaticos_livros_didaticos) que armazena as chaves estrangeiras dessas 2 tabelas durante a adição de livros didáticos ao empréstimo.

Nome da tabela	Campos	Utilidade	Relacionamentos
roles	- email - name	autenticação	-
users	- email - senha	autenticação	-
turmas	- id - nome - horario_emprestimo - turno - nome_professor_portugues	armazenar todas as turmas que a escola possui	- alunos (1 x n)
alunos	- id - nome - email - celular - nome_mae - nome_pai - turma_id	armazenar todos os alunos que estudam na escola	- empréstimos_literarios (1 x n) - empréstimos didaticos (1 x n) - turmas (1 x 1)
livros_literarios	- id - codigo - titulo - autor - numero_paginas - isbn - ano - genero - classificacao_id - edicao	armazenar todos os livros que pertencem a biblioteca	- empréstimos_literarios (n x n) - classificacoes (1 x 1)
classificacao	- id - nome - cor_etiqueta	armazenar todas as classificações de livros existentes na biblioteca	-livros_literarios (1 x n)

Quadro 2 – Tabelas que compõem o banco de dados do sistema evidenciando campo, utilidade, relacionamentos

Nome da tabela	Campos	Utilidade	Relacionamentos
livros_didaticos	- id - codigo - titulo - disciplinas - pnld - série - editora - disponibilidade	Armazenar todos os livros didaticos que pertencem ao colégio	emprestimos_didaticos (n x n)
emprestimos_didaticos	- id - aluno_id - data_emprestimo - data_devolucao - status	Armazenar os empréstimos de livros didáticos realizado pelos alunos	- alunos (1 x 1) - livros_didaticos (n x n)
emprestimos_literarios	- id - aluno_id - data_emprestimo - data_devolucao - status	Armazenar os empréstimos de livros literários realizados pelos alunos	- alunos (1 x 1) - livros_literarios (n x n)
emprestimos_literarios_livros_literarios	- emprestimo_literario_id - livros_literario_id	Armazenar chaves primárias das tabelas livros_literarios e emprestimos_literarios, por se tratar de um relacionamento (n x n)	Armazenar chaves primárias das tabelas livros_literarios e emprestimos_literarios, por se tratar de um relacionamento (n x n)

Continuação Quadro 2 – Tabelas que compõem o banco de dados do sistema evidenciando campo, utilidade, relacionamentos

Nome da tabela	Campos	Utilidade	Relacionamentos
emprestimos_didaticos_livros_didaticos	- emprestimo_didatico_id - livros_didatico_id	Armazenar chaves primárias das tabelas livros_didaticos e emprestimos_didaticos, por se tratar de um relacionamento (n x n).	Como num relacionamento (n x n) não é possível armazenar várias chaves num mesmo campo e transformar o relacionamento em (1 x n) por questões de redundância, existe a necessidade de ter uma tabela intermediária como ponte para as 2 tabelas envolvidas nesse relacionamento. Essa tabela não possui chave primária.

Continuação Quadro 2 – Tabelas que compõem o banco de dados do sistema evidenciando campo, utilidade, relacionamentos

Para diminuir o trabalho de criação manual das tabelas foi usado o *framework* de persistência Hibernate, o qual as gera automaticamente, pois toda hora é necessário deletar o banco de dados por causa das modificações na estrutura no decorrer do desenvolvimento. A figura número 17 ilustra o diagrama da estrutura de banco de dados do sistema.

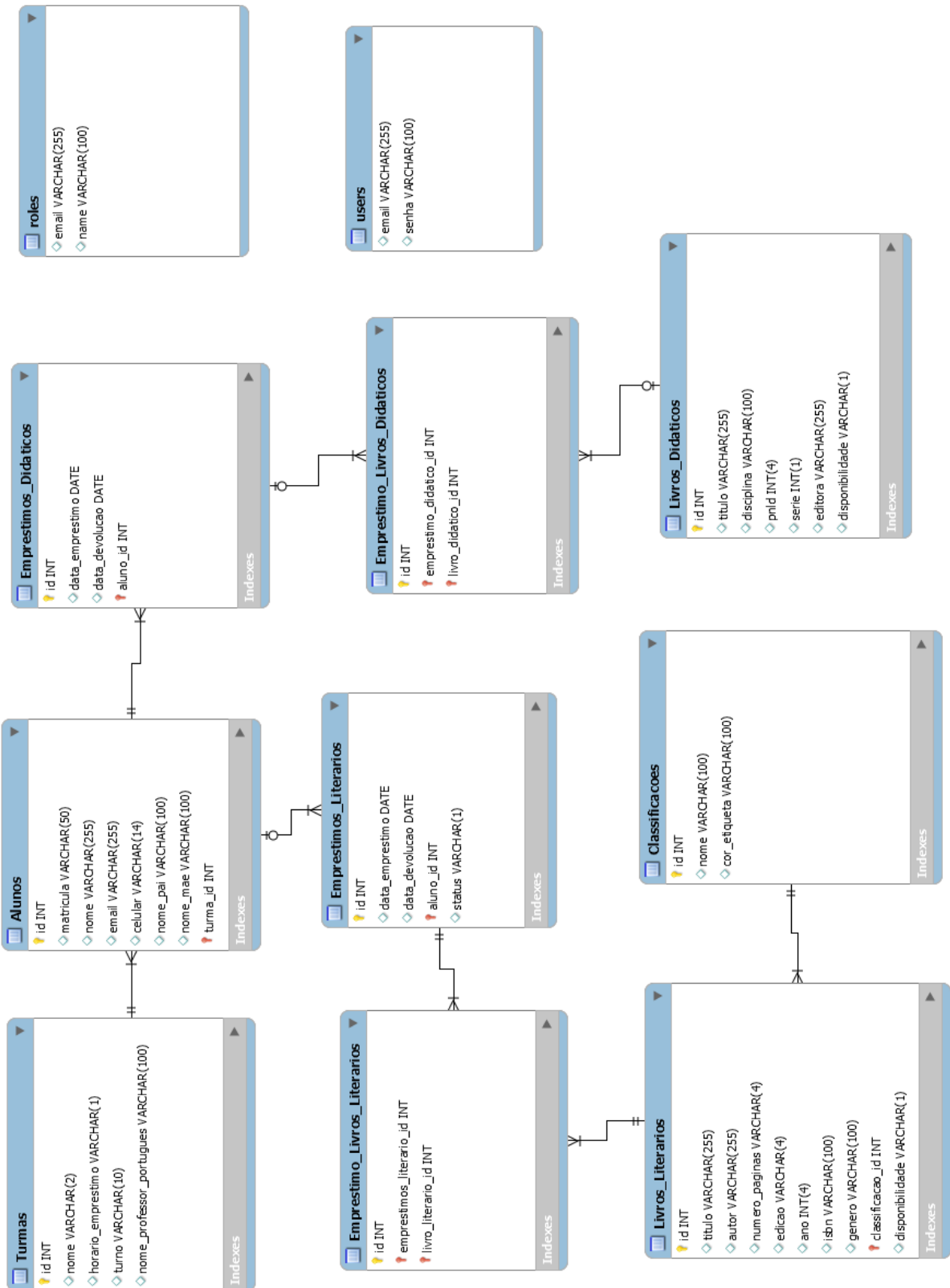


Figura 17 - Diagrama relacional da estrutura de banco de dados.

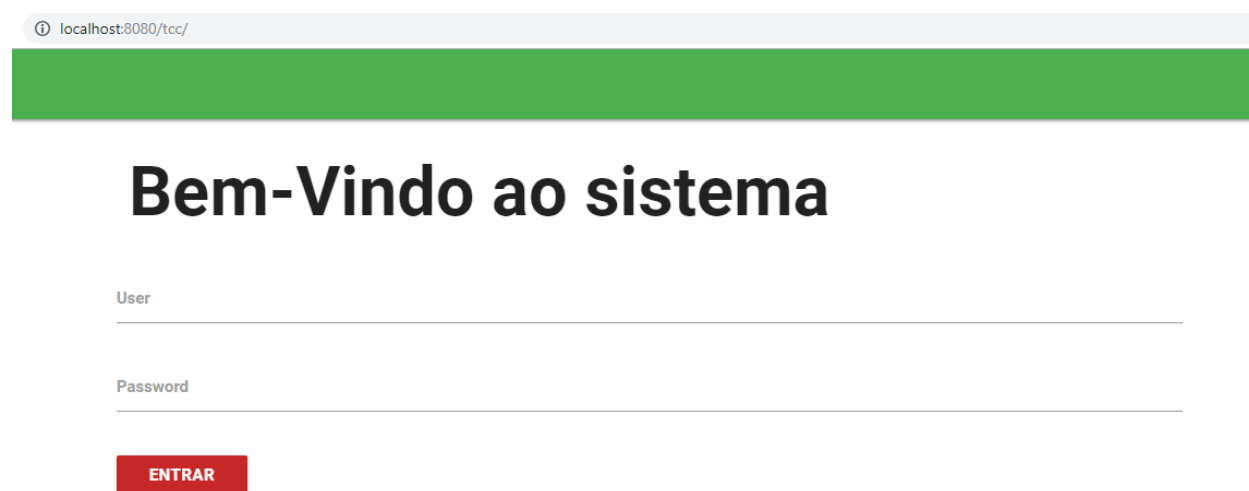
4. Desenvolvimento

O sistema desenvolvido tem como objetivo controlar os livros pertencentes ao acervo da biblioteca. Através dele é possível fazer as 4 operações de CRUD em alunos, turmas, classificações, livros literários, empréstimo e também efetuar devoluções de livros literários que não é associada a um CRUD, visualizar relatórios de livros disponíveis e emprestados, no caso dos emprestados é possível saber qual aluno esse livro está em posse e qual turma esse aluno pertence.

Conforme a entrevista realizada o sistema poderia ter mais funcionalidades do que as que foram desenvolvidas no momento, porém como a demanda de tempo para desenvolvimento foi maior que o tempo disponível para desenvolver esse trabalho isso acabou não sendo possível.

A primeira interface gráfica conforme ilustrada pela figura 18 que será possível visualizar no sistema é a de login, se os dados existentes no banco de dados corresponderem ao que usuário digitou ele será encaminhado a interface gráfica principal e criará uma sessão, caso contrário permanecerá nessa mesma.

Quando a sessão é destruída ou o usuário tenta acessar uma página que exige autenticação sem estar logado, o sistema redirecionar a essa interface gráfica.



localhost:8080/tcc/

Bem-Vindo ao sistema

User

Password

ENTRAR

Figura 18 – Interface gráfica de login.

Após efetuar o login com sucesso, o usuário tem acesso a todas as funcionalidades do sistema como ilustrado na Figura 19. Clicando na opção logout, a sessão é destruída e o mesmo redirecionado para a interface de login. A figura 19 mostra todos as funcionalidades que compõem o sistema:

Funcionalidade de Cadastro – ao clicar em cadastrar o usuário tem acesso a formulários de cadastro de turmas, alunos, livros literários e classificações.

Funcionalidade de Relatórios – ao clicar em relatórios, o usuário tem acesso a relatórios de livros disponíveis e emprestados.

Empréstimo – ao clicar em empréstimos, o usuário tem acesso a interface gráfica de empréstimo de livros onde na primeira parte é cadastrado o empréstimo e o aluno e na segunda é adicionado os livros ao empréstimo.

Devolução – ao clicar em devolução, o usuário tem acesso a interface gráfica para efetuar devoluções.

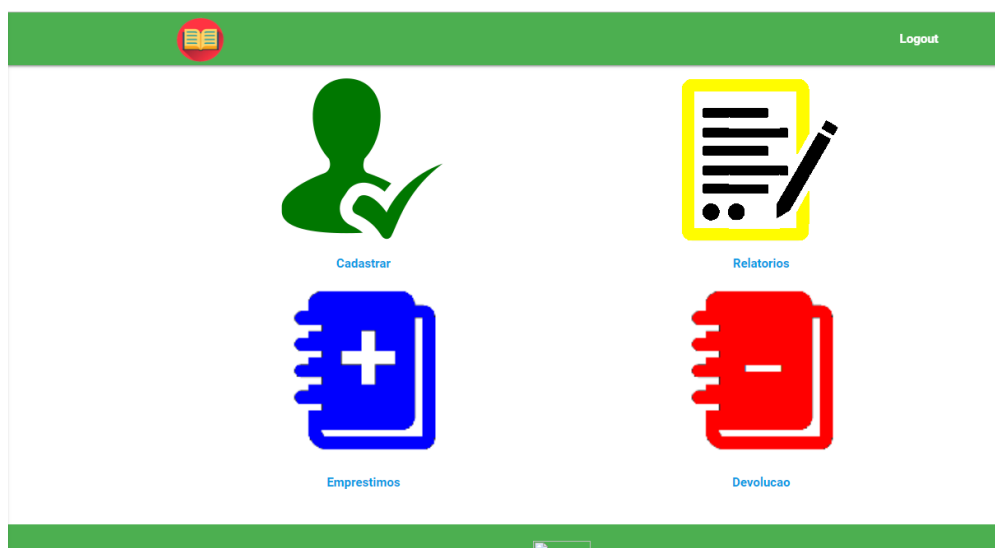


Figura 19 – Interface gráfica principal.

4.1 Funcionalidade de Cadastros

4.1.1 Cadastro de turmas

Essa interface gráfica pode ser acessada quando o usuário clicar em cadastrar conforme visto na Figura 20 e pode ser acessada diretamente pelas outras interfaces gráficas de cadastro que compõem o módulo, se clicar no símbolo do livro o usuário é redirecionado a interface de cadastro principal e ao clicar em logout a sessão é destruída sendo o usuário redirecionado para interface gráfica de login.

O formulário de cadastro de turmas mostrado na ilustração tem 5 campos sendo todos de preenchimento obrigatório e contendo validações do tipos de dados, sendo 2 inputs e 3 *selects* contendo opções fixas.

O campo turma só será válido se o tamanho for 2 e o primeiro dígito for qualquer número e o segundo qualquer letra.

O campo nome do professor de português só será válido se seus caracteres forem compostos somente por letras.

Abaixo do formulário é exibido uma tabela que mostra todos os registros cadastrados sendo possível atualizar e deletar. A turma só poderá ser deletada caso não possua nenhum aluno cadastrado, não é muito correto fazer o delete em cascata, por que se a turma deixar de existir os alunos não necessariamente deixarão de estudar no colégio, assim tendo que cadastrá-los novamente.

Por serem poucos registros não há necessidade de ser usado um campo como critério para ordenação de dados.

4.1.2 Cadastro de classificações de livros literários

Essa interface gráfica é similar a imagem ilustrada na Figura 20, usada para cadastro de classificações de livros, sendo obrigatório o preenchimento de todos os campos e também não possuindo um critério de ordenação.





Id	Nome	Dia	Horario	Professor	Turno	
1	6A	Terça-Feira	Primeiro	Telma	Tarde	 
2	6B	Terça-Feira	Terceiro	Telma	Tarde	 

Figura 20 – interface gráfica de cadastro de turmas.

4.1.3 Cadastro de Alunos

Na Figura 21 é visto que essa interface gráfica possui 7 campos, sendo o campo nome, e-mail, turma e matrícula de preenchimento obrigatório. O campo turma é diferente dos demais pois é um *select* com informações contidas no banco de dados, diferente do *select* mostrado na figura 20 que as opções são fixas, conforme as turmas são cadastradas ou excluídas esse *select* vai mudando suas opções.

Por ter uma grande quantidade de registros e haver necessidade de pesquisas a todo momento e não possuir um mecanismo de consultas a tabela abaixo do formulário mostra os alunos em ordem alfabética usando o nome critério de ordenação porém não definindo um segundo critério de desempate caso mais de um aluno ter o mesmo nome.

4.1.4 Cadastro de Livros Literários

Essa interface gráfica é similar a ilustrada pela Figura 21 e a que mais possui campos entre todas do sistema, possuindo 10 campos no total, sendo o campo título, autor, classificação, número de páginas, código de preenchimento obrigatório. Os livros possuem um campo chamado disponibilidade e são cadastrados com a disponibilidade = 'T' por padrão e que não pode ser visto no formulário e também ser atualizado manualmente, esse campo só pode ser atualizado quando esse livro é emprestado ou devolvido. Possui um *select* com o mesmo comportamento da interface gráfica de cadastro de alunos. Por serem em muitos registros há necessidade de serem ordenados por ordem alfabética usando o título com o critério e o código que é único como critério de desempate caso haja 2 livros iguais.

Id	Matricula	Nome	Email	Celular	Turma	Nome da mãe	Nome do pai	
6	6	Amanda	mandinha@gmail.		2A			
11	11	Ana	aninha2000@gma		7B			
27	26	Andres	andres@gmail.com		3A			

Figura 21 – Interface gráfica de cadastro de alunos.

4.2 Funcionalidade de empréstimos

4.2.1 Cadastro de empréstimos

Essa funcionalidade está sendo ilustrada na Figura 22. É cadastrado um empréstimo para um aluno, contendo um *select* que mostra o nome de todos os alunos cadastrados em ordem alfabética, e 2 campos de data contendo a data que ele emprestou e a data que ele deve devolver os livros emprestados sendo controlados de forma manual, pois o trabalho aumentaria se esses campos fossem controlados automaticamente, e também teria problemas da data que o aluno deve devolver coincidir com feriados, pois existem alguns feriados que não tem data fixa assim não podendo serem controlados e o código precisar de uma manutenção manual anualmente, também devido ao tempo investido em funcionalidades com prioridades maiores acabou não sendo possível evitar fazer com que um empréstimo não seja realizado numa data que está no passado ou no futuro também existe outro problema que não foi possível solucionar com relação a data a data que o aluno deve devolver poder estar no passado em relação a data do empréstimo.

Os empréstimos são mostrados em ordem decrescente, usando o id como critério de ordenação, pois quanto maior for o id do empréstimo mais recente será.

Ao clicar no botão (VER) que está a esquerda do nome do aluno, será mostrado todos os livros que o aluno emprestou nesse empréstimo mostrado na Figura 22a.

Aluno

Raissa

Data de emprestimo

Data de devolucao

SALVAR

Id	Aluno	Data-Emprestimo	Data-Devolucao	
VER	Raissa	2018-11-09 00:00:00.0	2018-11-19 00:00:00.0	[Edit/Delete]
VER	Valentina	2018-11-07 00:00:00.0	2018-11-14 00:00:00.0	[Edit/Delete]

Figura 22 – Interface gráfica de cadastro de empréstimos.

Livro Literário Livro Didático Logout

Titulo

Tucano da Amazonia

Quem mexeu no meu queijo

Figura 22a – Interface gráfica de exibição dos livros de um empréstimo ao clicar no botão VER.

4.2.1.1 Cadastro de livros do empréstimo

Essa interface está ilustrada na Figura 23. Por serem muitos registros o título dos livros no *select* são exibidos ordem alfabética e caso dois livros tenham o mesmo nome o critério de desempate é o código, só serão exibidos livros literários se o campo disponibilidade for = 'T', ao selecionar o título do livro e clicar no botão adicionar o status desse livro é mudado para F, assim não sendo possível ser inserido 2 vezes no mesmo empréstimo, ou emprestado para um próximo aluno enquanto não for devolvido, na tabela abaixo do botão se clicar no ícone ao lado do seu nome, o título do

livro volta ser apresentado no *select* pois o livro voltara para a disponibilidade = 'T', após clicar no campo fechar empréstimo o usuário é encaminhado para interface gráfica de empréstimo.

O empréstimo é cadastrado com o campo status = 'O' por padrão, quando o aluno devolve todos os livros pertencentes àquele empréstimo o sistema automaticamente muda o campo status = 'C'.



Figura 23 – Interface gráfica de adição de livros ao empréstimo.

4.3 Funcionalidade devoluções

4.3.1 Devolução

Essa interface gráfica ilustrada pela Figura 24 não realiza cadastros. Apenas exibe registros de uma instrução no seu *select* cujo a query é "SELECT I FROM LivroLiterario I WHERE I.disponibilidade = 'F' ORDER BY I.titulo" que filtra somente os livros com disponibilidade 'F' em ordem alfabética, após o livro ser selecionado e clicado no botão salvar sua disponibilidade é alterada para 'T', voltando a ficar disponível para futuros empréstimos.

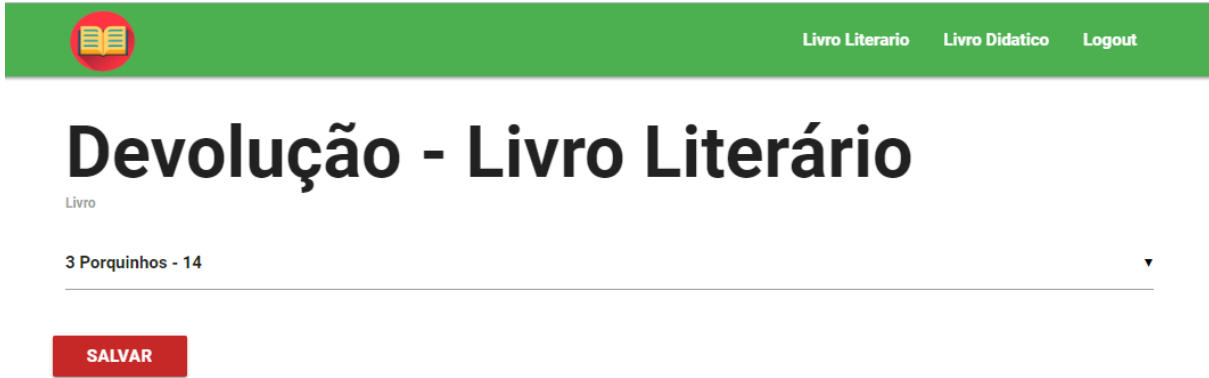


Figura 24 – Interface gráfica de devolução.

4.4 Relatórios

4.4.1 Livros Disponíveis

No relatório ilustrado pela Figura 25 são mostrados todos os livros que estão na biblioteca em ordem alfabética, cujo a *query* é "SELECT I FROM LivroLiterario I WHERE I.disponibilidade = 'T' ORDER BY I.titulo", sendo mostrados 10 em cada página, podendo esses serem emprestados para qualquer aluno no presente momento.

Codigo	Titulo
1	Gato de Botas
138	História do Campeonato Brasileiro
137	História do Futebol
116	Laços de família
2	Lusiadas
15	Menina e o Porquinho
132	Metamorfose
9	Mickey
6	Piratas da Malasia
5	Princesas da Disney

Figura 25 – Relatório de livros disponíveis.

4.4.2 Livros Emprestados

Nesse relatório são mostrados todos os livros que não estão na biblioteca, para qual aluno está emprestado no momento, qual turma esse aluno pertence e a data que foi emprestado e deve ser devolvido. Para escrever a *query* do relatório ilustrado pela Figura 26 foram necessários 4 passos:

1. Verificar todos os livros literários com disponibilidade = 'F'
2. Verificar o maior id de empréstimo desses livros pela tabela intermediária
3. Descobrir a qual aluno pertence esses empréstimos
4. Descobrir a turma que esses alunos estão cadastrados



Título	Aluno	Turma	Data do Empréstimo	Data da Devolucao
Menina e o Porquinho	Valentina	6A	2018-11-07 00:00:00.0	2018-11-14 00:00:00.0
Turma da Monica	Valentina	6A	2018-11-07 00:00:00.0	2018-11-14 00:00:00.0
Tucano da Amazonia	Enzo	6B		
Quem mexeu no meu queijo	Enzo	6B		
3 Porquinhos	Pablo	6A		
Bolsa Amarela	Pablo	6A		
Donald e Margarida	Enzo	6B		
Lusiadas	Enzo	6B		

Figura 26 – Relatório de livros emprestados.

5 Resultados

Neste capítulo são apresentados os resultados obtidos com os testes realizados em parceria com o profissional responsável pela biblioteca do colégio Estadual Newton Felipe Albach da cidade de Guarapuava. Após realizar os testes envolvendo várias situações de empréstimos foi possível fazer melhorias em algumas funcionalidades já existentes do sistema e descobrir novas.

Nos dias 18 e 19 de setembro junto com o funcionário responsável pela biblioteca foram realizados testes e essa mesma pessoa notou satisfeita com os vários benefícios que o sistema trará ao seu trabalho como um deles ilustrado na Figura 27, foi notado durante esses 2 dias que os empréstimos ainda estão sendo controlado em folhas de papeis que são impressas semanalmente para cada turma do colégio, assim essas folhas podendo ser perdidas ou danificadas e também causando problemas ao meio ambiente, além de ser difícil obter um controle de circulação dos livros e alunos que estão com livros atrasados. A funcionalidade de empréstimo é muito importante pois a partir dela que as folhas de papel serão eliminadas do cotidiano do trabalho do responsável pela biblioteca. O fato dos alunos e livros estarem ordenados por nome no *select* de empréstimos é uma grande economia de tempo, pois seria mais difícil encontrar determinado aluno ou livro se não estivesse ordenado e aumentaria mais o trabalho do funcionário.

Titulo	Aluno	Turma	Data do Empréstimo	Data da Devolucao
Bolsa Amarela	Amanda Mendes Moreira	3A	09/11/2018	19/11/2018
Rato Roeu a Roupa do Rei de Roma	Amanda Mendes Moreira	3A	09/11/2018	19/11/2018
3 Porquinhos	João Pedro Batista	8A	09/11/2018	19/11/2018
Piratas da Malasia	João Pedro Batista	8A	09/11/2018	19/11/2018

Figura 27 – Relatório de alunos que possuem livros emprestados.

No mês dezembro foi realizado um teste de validação do sistema aplicando um questionário que avalia vários aspectos do sistema, após esses testes serem realizados todos os entrevistados responderam esse questionário de acordo com suas opiniões pessoais. O questionário presente no APÊNDICE A abrange onze questões que avaliam design, funcionalidades e usabilidade do sistema. As Figuras de 28 a 38 ilustram as perguntas referentes a esse questionário e os gráficos associados as respostas dadas pelos entrevistados, numa escala de 1 a 5 (péssimo, ruim, regular, bom, ótimo).

Com base nas respostas desse questionário foi possível ver o nível de aprovação e satisfação do sistema em relação aos entrevistados. A partir desse questionário foi possível validar o sistema, notar os efeitos, mudanças e melhorias que irá causar nos processos diários da biblioteca após sua implantação.

Com esse questionário foi possível verificar, validar e avaliar o sistema, notou-se pelos entrevistados que a gestão dos processos e controle da biblioteca irão melhorar o trabalho do profissional encarregado. Pelas respostas dos entrevistados observou-se que o controle do processo de gerenciamento e relatórios de pendências estão entre os principais benefícios da implantação desse sistema na biblioteca.

6. Conclusão

Este trabalho teve como objetivo desenvolver um sistema de gerenciamento web de bibliotecas para escolas estaduais da rede pública da cidade de Guarapuava, pois auxilia o profissional responsável pela biblioteca a gerenciar os livros que a pertencem.

O sistema atende a demanda da biblioteca da escola onde foram realizados os testes. Foi percebido durante os testes que para o sistema possa ser utilizado em todas as escolas públicas estaduais é necessário a padronização da manipulação dos acervos em cada uma delas, pois cada biblioteca possui diferentes formas de classificar e gerenciar seu acervo.

Visto que foi preciso modernizar a forma de controlar os empréstimos, esse sistema foi planejado a partir de uma entrevista com um bibliotecário de uma escola pública estadual.

Por se tratar de uma aplicação web permite acesso remoto em diferentes máquinas a partir de uma arquitetura cliente-servidor, eliminando os altos custos de manutenção e aquisição desse software, sendo algo inovador dentro desse contexto.

Entre as vantagens desse sistema estão:

- Digitalização das informações;
- Automatização dos processos de geração de relatórios, empréstimo e devolução;
- Consultas de empréstimos e livros;
- Acesso remoto a várias máquinas;
- Autenticação por meio de banco de dados;

Após a aplicação de um questionário por meio das respostas dadas pelos entrevistados notou-se que alguns aspectos do sistema precisam ser melhorados como a interface e a usabilidade.

Foi observado que o sistema não pode ser implementado a curto prazo, pois existem exemplares nessa biblioteca que não possuem um número de identificação, assim ficando difícil de ser identificado durante uma devolução. Pois isto pode gerar uma inconsistência no sistema, às vezes ocorre de mais de um exemplar com o mesmo título e sem um número estar emprestado durante um mesmo momento. Essa inconsistência no sistema ao devolver um livro sem número pode eliminar uma pendência existente e criar uma inexistente no relatório de livros emprestados.

6.1 Trabalhos Futuros

Entre as futuras funcionalidades do sistema são descritas as seguintes:

1. Cálculo multas por livro e dia atrasado, no contexto de bibliotecas privadas e municipais.
2. Bloquear empréstimo para leitores que possuem multas sem pagar ou livros atrasados até a multa ser paga e os livros devolvidos.
3. Área de acesso ao leitor, nessa área o leitor poderá visualizar todo o acervo da biblioteca, suas pendências e dívidas com a biblioteca e reservar livros para um futuro empréstimo
4. Criar módulo de cadastro de bibliotecários, nessa área o *admin* poderá cadastrar novos bibliotecários para utilizar o sistema.
5. Consultar alunos com livros atrasados por turma.
6. Permitir com que o leitor renove 3 vezes no máximo um livro de um empréstimo.
7. Não permitir empréstimos de livros inadequados a idade do leitor.

7. Referências

ALAUZO, Jorge Luiz Cativo; SILVA, Danielle Lima; FERNANDES, Tatiana Brandão. FUNCIONALIDADES DE UM SOFTWARE LIVRE DE AUTOMAÇÃO DE BIBLIOTECAS: uma avaliação do Biblivre. **RacIn**, João Pessoa, v.2, n.2, p. 23-43, jul.-dez 2014.

BESSA, Alan Teles; BARROS, Márcio de Oliveira. Integração entre o PMBOK e RUP. Disponível em:
<https://www.researchgate.net/publication/257026796_Integracao_entre_o_PMBOK_e_RUP>. Acesso em 14 novembro de 2018.

CORTE, Adelaide Ramos; ALMEIDA, Ieda de Muniz; PELLEGRINI, Ana Emília; LOPES, Ildeu Ordini; SAENGER, José Carlos; ESMERALDO, Maria Bernadete P.; PEREIRA, Maria Cristina Moraes; FERREIRA, Rosana Rika M. C.; do LAGO. Automação de bibliotecas e centros de documentação: o processo de avaliação e seleção de softwares. *Ci. Inf.*, Brasília, p.241-256, set-dez 1999.

DA SILVA, Marcelo Bezerra; DIAS, Guilherme Ataíde. O sistema de automação em bibliotecas OpenBiblio aplicado à disciplina automação em bibliotecas. *Biblionline*, João Pessoa, v. 6, n. 1, p. 53-71, 2010.

GILMORE, W. Jason. Dominando PHP e MySQL. 2ª edição. Rio de Janeiro : Alta Books, 2011.

GONÇALVES, Edson. Desenvolvendo aplicações web com jsp, servlets, jsf, hibernate, ejb 3 persistence e ajax. Rio de Janeiro : Ciência Moderna, 2007.

MAGALHÃES. Carlos Filipe; SILVA, Alexandre Paulino Sierra. Utilização do Framework Hibernate em Aplicação Java Web, UNIPAR, Paranavaí, 2013.

MALHEIROS, Tânia Milca de Carvalho. NECESSIDADE DE INFORMAÇÃO DO USUÁRIO COM DEFICIÊNCIA VISUAL: UM ESTUDO DE CASO DA BIBLIOTECA DIGITAL E SONORA DA UNIVERSIDADE DE BRASÍLIA. 305 f. Dissertação de Mestrado – Universidade de Brasília, Brasília, 2013.

NASCIMENTO, Flávia Cristina Brito. A IMPORTÂNCIA DA EDUCAÇÃO A DISTÂNCIA NA FORMAÇÃO DE BIBLIOTECÁRIOS: A INTERAÇÃO ENFINANTES E ASCENDENTES. 54 f. Monografia (Bacharelado em Biblioteconomia) – Universidade Federal da Paraíba, João Pessoa, 2010.

NERI, Emmanuel. Criando validações de Bean Validation customizadas.

Disponível em <<https://emmanuelneri.com.br/2017/05/30/criando-validacoes-de-bean-validation-customizadas/>>. Acesso em: 6 de novembro de 2018.

Netbeans. Guia de Referência JSF - JavaServer Faces.

Disponível em <https://netbeans.org/kb/docs/web/jsf20-intro_pt_BR.html>. Acesso em: 14 de novembro de 2018.

PAZ, Marcelo Augusto Lobo; DIAS, Jayme Willian. DESENVOLVIMENTO WEB UTILIZANDO JAVA EE, UNIPAR, Paranavaí, 2016.

Pergamum - Rede Informatizada de Bibliotecas Públicas do Paraná no Sistema Pergamum: da concepção à ação.

Disponível em:

<https://www.pergamum.pucpr.br/redepergamum/trabs/artigo_cbdd_completo_2013.pdf>. Acesso em 06 de dezembro de 2018.

PEREIRA, Gleice; DOS SANTOS, Emanuela Brambilla; CALMON, Maria Aparecida. AUTOMAÇÃO DA REDE DE BIBLIOTECAS ESCOLARES DA PREFEITURA DE VITÓRIA. 3º Seminário de Informação em Arte, Rio de Janeiro, 2013.

PMG – Prefeitura Municipal de Guarapuava. Endereço das escolas municipais. Disponível em <<http://www.guarapuava.pr.gov.br/cidadao/unidades-municipais-de-ensino/>>. Acesso em: 02 de maio de 2017.

PRESSMAN, Roger S. Engenharia de Software Uma Abordagem Profissional. 7º edição. Porto Alegre : AMGH , 2011.

RODRIGUES, Anielma Maria Marques; PRUDÊNCIO, Ricardo Bastos Cavalcante. AUTOMAÇÃO: a inserção da biblioteca na tecnologia da informação. Biblionline, João Pessoa, v. 5, n. 1/2, 2009

SLB – Software Livre do Brasil. A nova versão do Openbiblio 0.6 pt.BR já está disponível. Disponível em <<http://softwarelivre.org/portal/noticias/a-nova-versao-do-openbiblio-0.6-pt.br-ja-esta-disponivel1>>. Acesso em 30 de abril de 2017.

VUKOTIC, Aleksa; GOODWILL, James. Apache Tomcat 7. 1ª edição. New York : Apress, 2011.

APÊNDICE A – QUESTIONÁRIO SOBRE TESTE DE SOFTWARE COM RESPOSTAS.

1 - Qual o nível de satisfação em relação ao sistema?

6 respostas

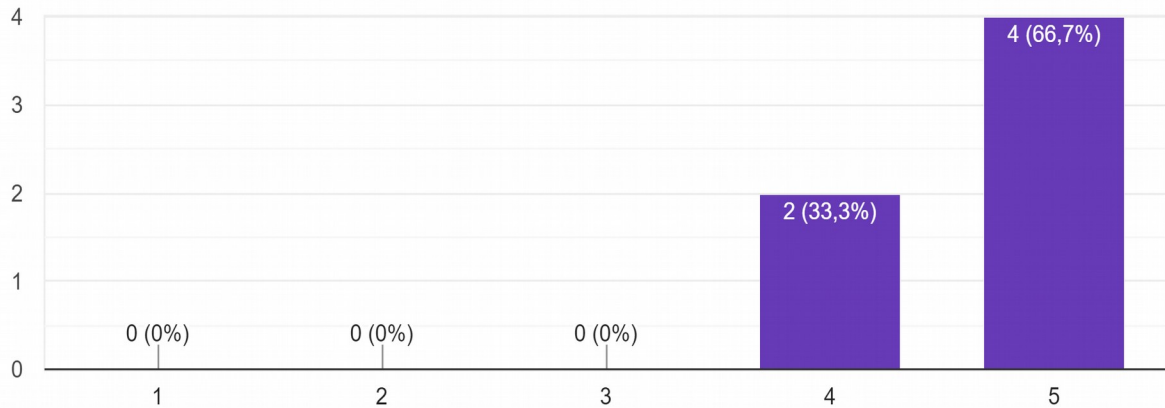


Figura 28 – Gráfico das respostas da pergunta: Qual o nível de satisfação em relação ao sistema?.

2 - Avalie o processo de empréstimo

6 respostas

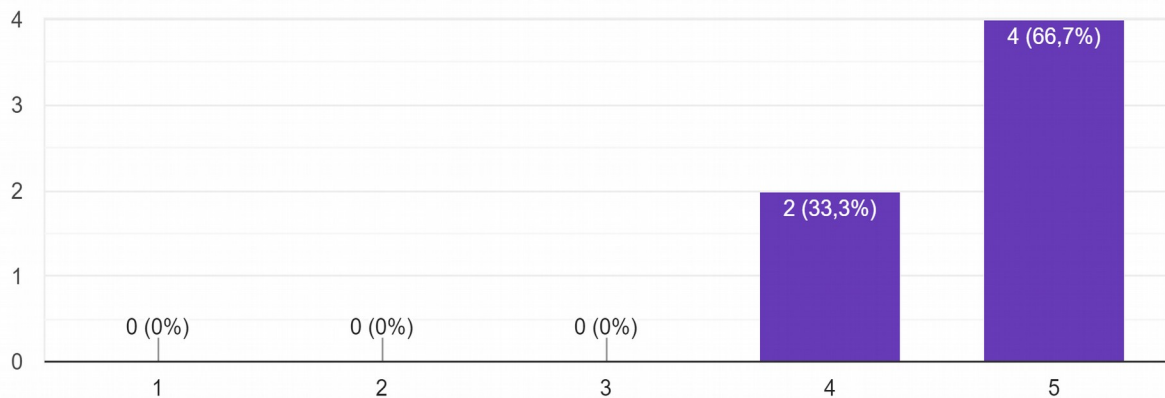


Figura 29 – Gráfico das respostas da pergunta: Avalie o processo de empréstimo.

3 - Avalie o controle de circulação dos livros literários

6 respostas

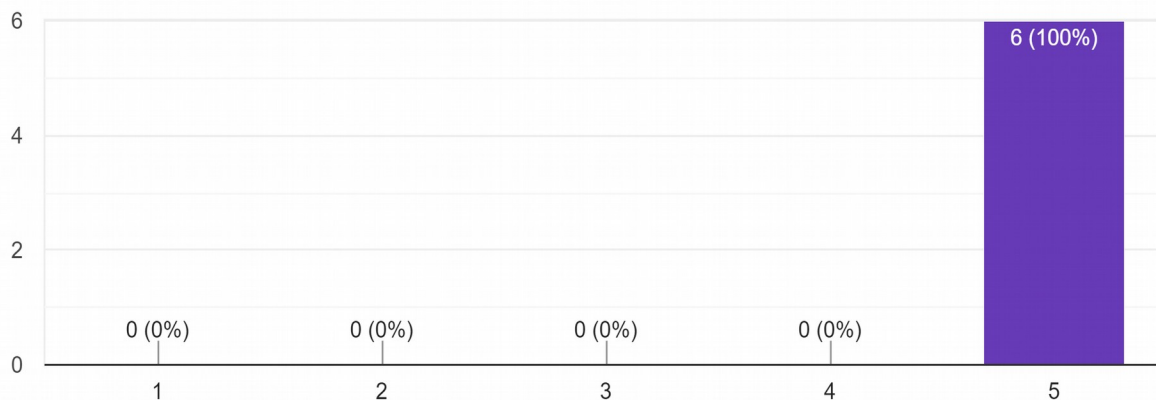


Figura 30 – Gráfico das respostas da pergunta: Avalie o controle de circulação dos livros literários.

4 - Avalie o controle de circulação dos livros didáticos

6 respostas

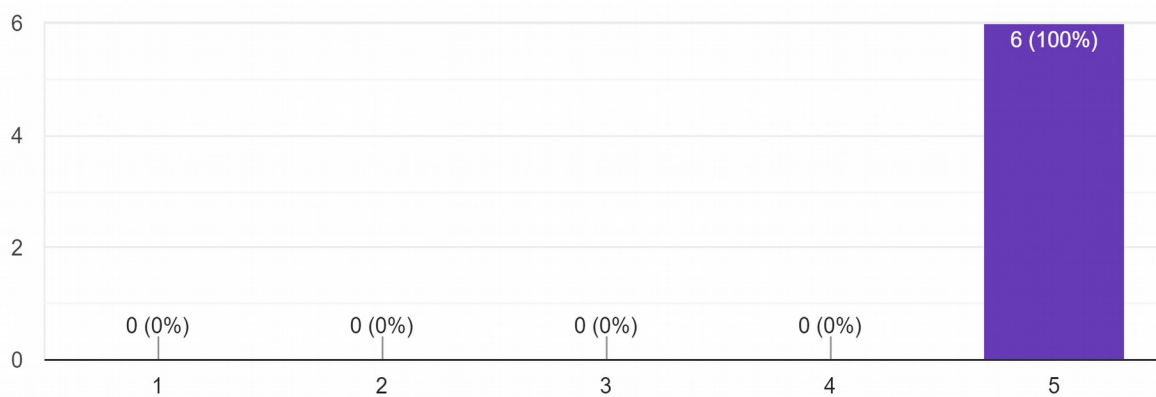


Figura 31 – Gráfico das respostas da pergunta: Avalie o controle de circulação dos livros didáticos.

5 - Avalie o design da interface

6 respostas

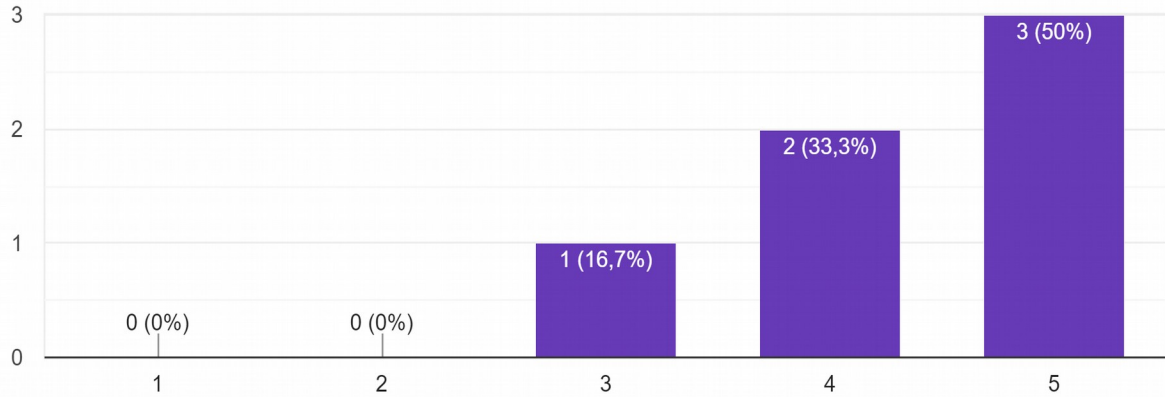


Figura 32 – Gráfico das respostas da pergunta: Avalie o design da interface.

6 - Avalie a usabilidade do sistema

6 respostas

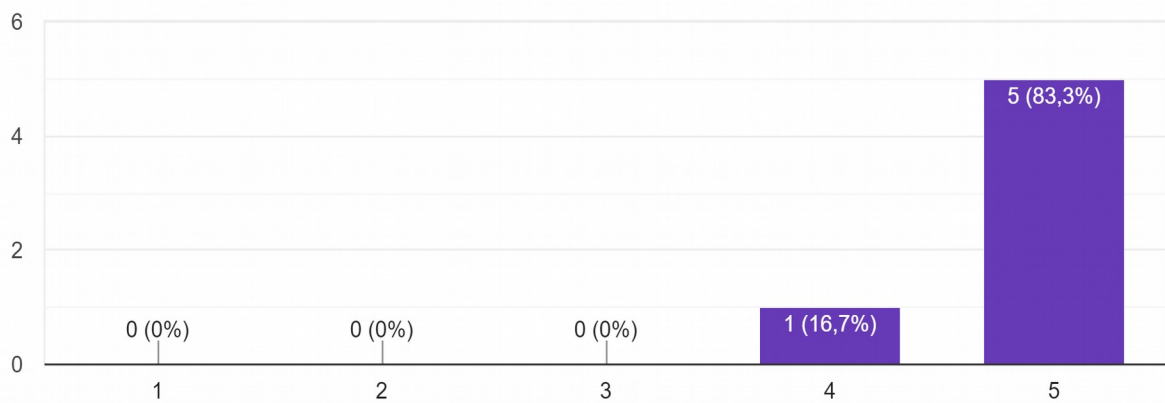


Figura 33 – Gráfico das respostas da pergunta: Avalie a usabilidade do sistema.

7 - Avalie o processo de gerenciamento de informação da biblioteca como um todo

6 respostas

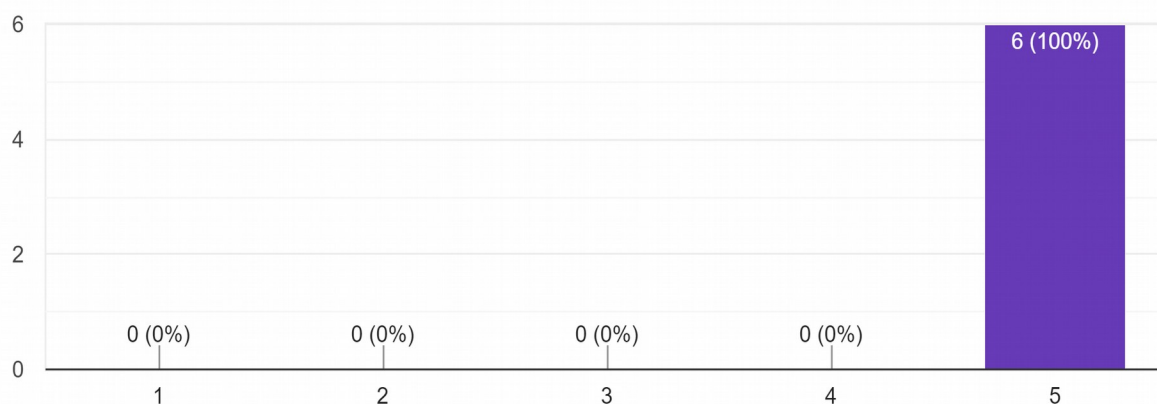


Figura 34 – Gráfico das respostas da pergunta: Avalie o processo de gerenciamento de informação da biblioteca com um todo.

8 - Avalie a funcionalidade de cadastros

6 respostas

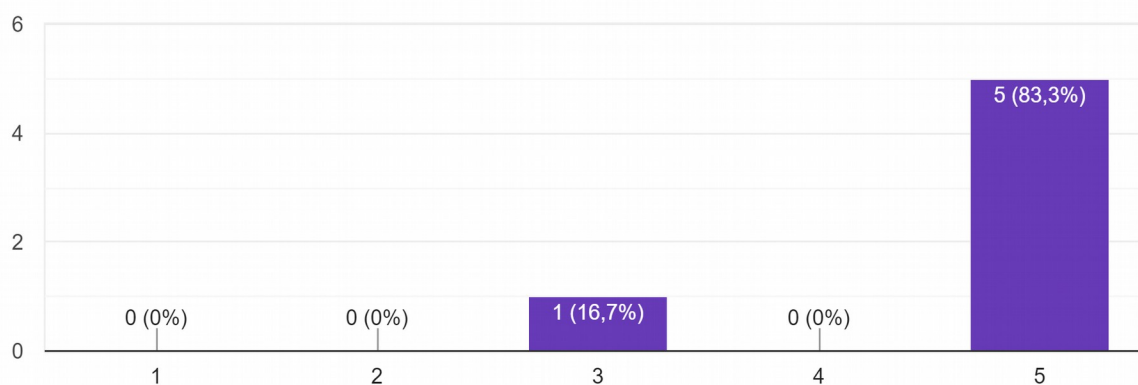


Figura 35 – Gráfico das respostas da pergunta: Avalie a funcionalidade de cadastros.

9 - Avalie a funcionalidade de empréstimos

6 respostas

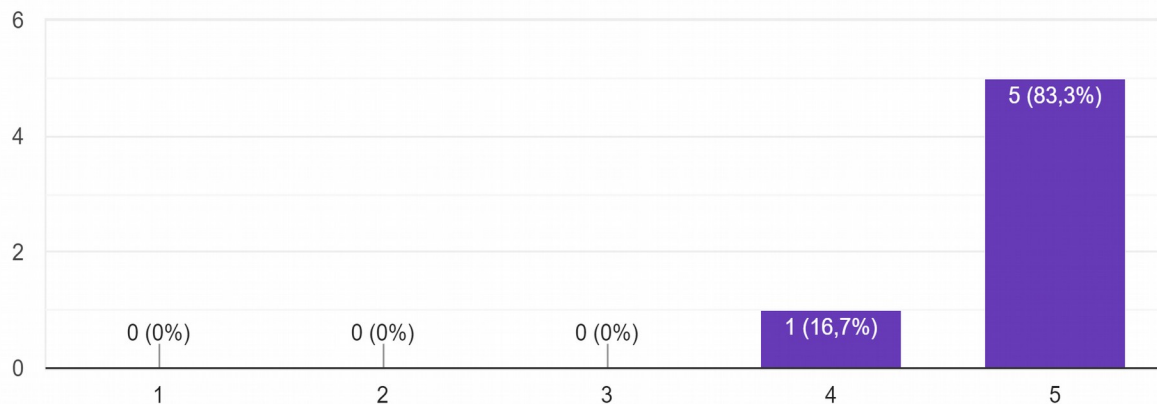


Figura 36 – Gráfico das respostas da pergunta: Avalie a funcionalidade de empréstimos.

10 - Avalie a funcionalidade de devoluções

6 respostas

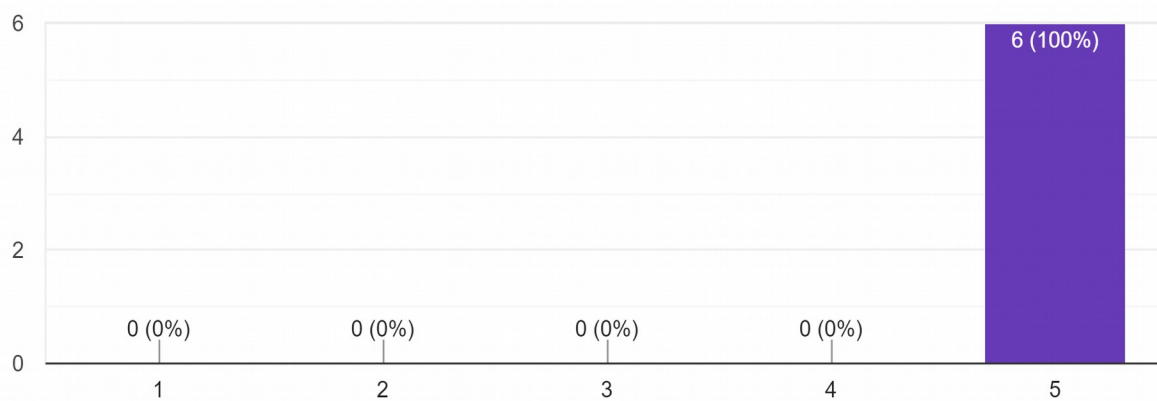


Figura 37 – Gráfico das respostas da pergunta: Avalie a funcionalidade de devoluções.

11 - Avalie os relatórios de pendência de livros

6 respostas

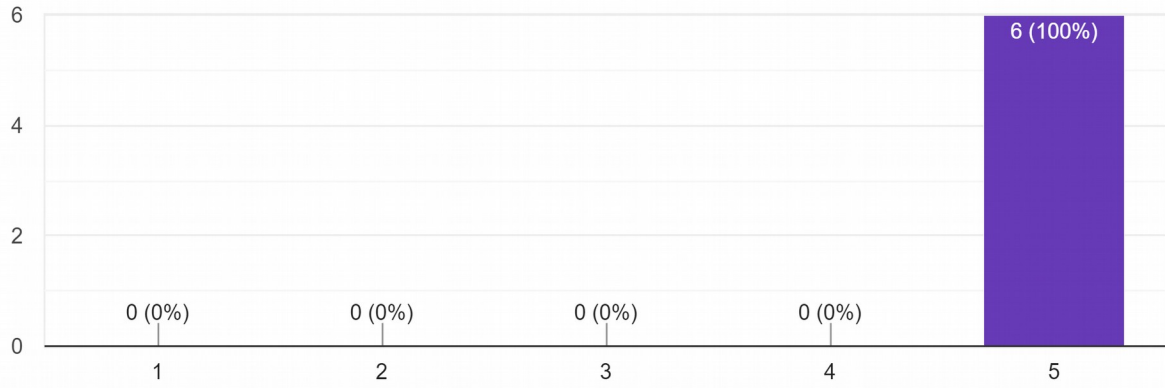


Figura 38 – Gráfico das respostas da pergunta: Avalie os relatórios de pendência de livros.