



Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal Catarinense
Campus São Bento do Sul

Amanda Luiza Ilg

TRABALHO FINAL - SISTEMA DE BIBLIOTECA

São Bento do Sul

2021

Amanda luiza Ilg

TRABALHO FINAL - SISTEMA DE BIBLIOTECA

São Bento do Sul

2021

SUMÁRIO

INTRODUÇÃO	7
LEVANTAMENTO DE REQUISITOS	8
REQUISITOS FUNCIONAIS	8
REQUISITOS NÃO FUNCIONAIS	9
Regras de negócio	10
PROJETO	11
DIAGRAMA DE CASOS DE USO	11
DIAGRAMA DE CLASSES	12
REVISÃO BIBLIOGRÁFICA	15
ENGENHARIA DE SOFTWARE	15
UML	15
HTML	16
CSS	17
PHP	17
JAVASCRIPT	18
SQL - MYSQL	18
MODELO DE DESENVOLVIMENTO XP (EXTREME PROGRAMMING)	19
BANCO DE DADOS	22
MODELO DE ENTIDADE E RELACIONAMENTO	22
DIAGRAMA DE ENTIDADE E RELACIONAMENTO	23
MODELO FÍSICO - SQL	24
DESENVOLVIMENTO	26
CONEXÃO COM BANCO DE DADOS	26
TELA LOGIN	27
TELA CADASTRO DE USUÁRIO	28
TELA CONSULTA DO ACERVO	30
TELA PERFIL	32

TELA ÁREA DO FUNCIONÁRIO	35
Modal Delete	36
TELA CADASTRO DE LIVRO	37
TELA CADASTRO DE FUNCIONÁRIO	38
TELA MODIFICAR LIVRO	38
TELA ERRO	40
TESTES DE SOFTWARE	42
CONSIDERAÇÕES FINAIS	46
REFERÊNCIAS	47

1 INTRODUÇÃO

Esse documento consiste na documentação do desenvolvimento de um software para uma biblioteca. A proposta do software é oferecer login no sistema, consulta do acervo da biblioteca e cadastro de novos livros, conforme os recebimentos. O modelo de processo de software a ser adotado é o modelo da metodologia ágil XP (eXtreme Programming). Quanto às tecnologias e ferramentas a serem utilizadas, o software será desenvolvido na linguagem de marcação HTML, em conjunto com as linguagens de programação PHP e JavaScript. O Banco de Dados da biblioteca será em um servidor local e será construído no MySQL Workbench. Outras ferramentas que serão utilizadas para o desenvolvimento são o BrModelo, Astah Community, Visual Studio Code, Wamp Server e editores de texto gratuitos.

2 LEVANTAMENTO DE REQUISITOS

Sommerville (2018) e Visco (2019) dizem que o processo de levantamento de requisitos de um software consiste na compreensão das necessidades do cliente para o software e como ele será utilizado, antes do início do desenvolvimento. Nessa etapa são elencadas as atividades, os serviços, as características, o desempenho desejado e as imitações que o software terá. Para que, assim, os problemas a serem solucionados pelos desenvolvedores realmente existam e não sejam “achismos”. Desse modo, haverá maior eficiência e as necessidades do cliente serão atendidas.

2.1 REQUISITOS FUNCIONAIS

Os requisitos funcionais devem conter a descrição de uma função a ser executada pelo sistema, qual a origem dos requisitos, quais informações serão passadas para o usuário e quais são as restrições lógicas do sistema. São os comportamentos do sistema para entradas específicas. (SOMMERVILLE, 2018)

Identificador	Descrição	Prioridade	Req. Relacionados
RF001	O sistema deve permitir o controle de acesso de usuários por meio de login.	Alta	RN001, RF006
RF002	O sistema deve permitir a consulta dos livros do acervo da biblioteca por qualquer usuário cadastrado.	Alta	RN001, RF001
RF003	O sistema deve possibilitar ao usuário adicionar novos livros ao acervo.	Alta	RN002, RN003, RN004, RF007
RF004	O sistema deve permitir que haja a correção dos dados de determinado livro pelos usuários.	Alta	RN002, RF007

RF005	O sistema deve permitir a exclusão de livros pelos usuários.	Alta	RN002, RF007
RF006	O sistema deve possibilitar o cadastro de novos usuários no sistema.	Alta	RN001, RF001
RF007	O sistema deve efetuar a verificação dos usuários autorizados	Alta	RF003, RF004, RF005, RN003, RN004

2.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais estão sempre ligados aos requisitos funcionais e podem ser dois tipos: lógicos ou tecnológicos. As restrições lógicas são as regras de negócio e serão listadas no item abaixo. E as restrições tecnológicas dizem respeito à limitação de tecnologia existente, disponível para o uso ou que os desenvolvedores tenham domínio. O descumprimento de um requisito não funcional, ao contrário dos funcionais, pode comprometer todo o sistema. (WAZLAWICK, 2010; SOMMERVILLE, 2018)

ID	Descrição	Prioridade	Categoria	Req. Relacionados
RNF001	O sistema deve ser desenvolvido para Web, utilizando a linguagem HTML	Alta	Operacional	
RNF002	O sistema deve ter sua parte lógica desenvolvida em linguagem PHP e JavaScript	Alta	Operacional	
RNF003	O sistema deve ter seu layout desenvolvido responsivamente, com Bootstrap e CSS	Alta	Operacional	
RNF004	O sistema deve ter seu	Alta	Operacional	

	banco de dados hospedado localmente no servidor do Wamp Server			
--	---	--	--	--

2.2.1 Regras de negócio

As regras de negócio regulamentam as restrições do sistema quanto às restrições lógicas, ou seja, são as diretrizes a serem seguidas e que são definidas pelo cliente. (SCHEMES, 2020)

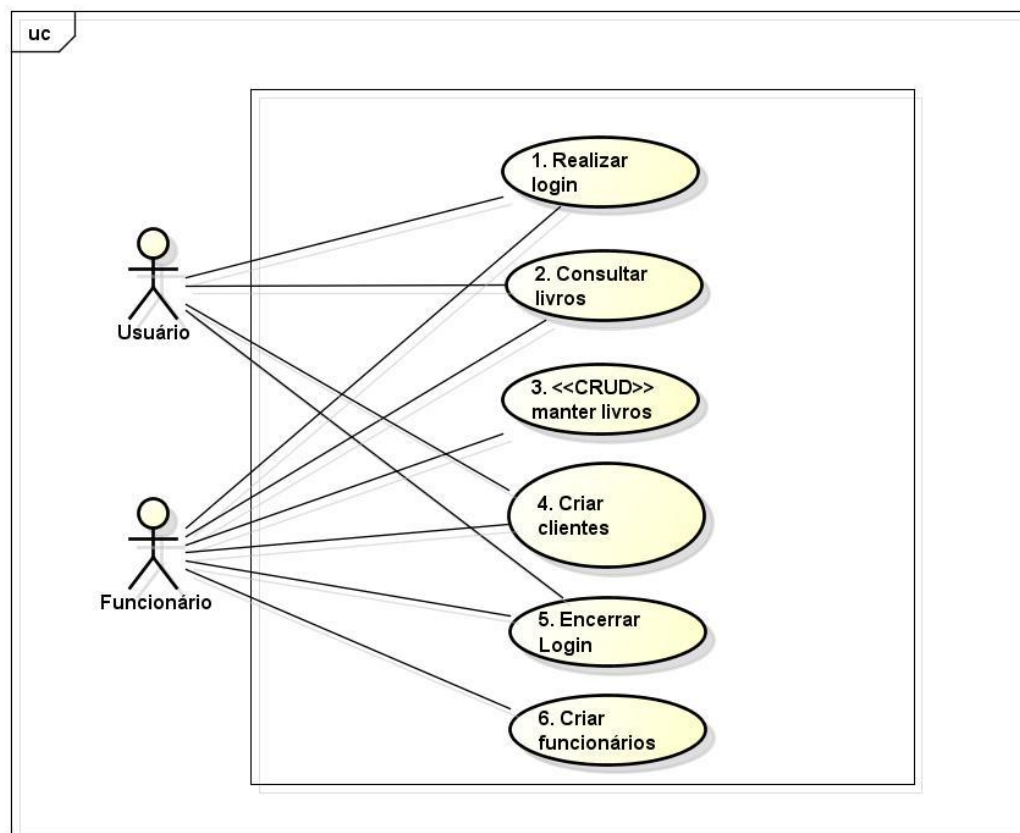
Identificador	Descrição	Req. Relacionados
RN001	O sistema não deve permitir que usuários não cadastrados entrem no sistema.	RF001, RF006, RF002
RN002	O sistema não deve permitir que usuários não autorizados realizem operações de cadastro, exclusão ou alteração de livros.	RF003, RF004, RF005, RF007
RN003	O sistema não deve permitir o cadastro de livros sem que os campos obrigatórios sejam preenchidos.	RF003
RN004	O sistema deve considerar os diferentes níveis hierárquicos de acesso: cliente e funcionário	RF007

3 PROJETO

De acordo com Coutinho(2020), na etapa de projeto será definido a parte técnica, sendo tudo que o software precisa para funcionar. Nela são considerados os aspectos como linguagem de programação, arquitetura do sistema, SGBD(Sistema Gerenciador de Banco da Dados), interface gráfica, hardware, etc como diz Pinto(2007). A importância dessa etapa é fornecer uma espécie de “esqueleto” do software, facilitando seu desenvolvimento.

3.1 DIAGRAMA DE CASOS DE USO

No diagrama de casos de uso é detalhado os usuários do sistema e suas interações. Sua importância se dá na representação das interações do usuário com o sistema, ajudando na definição e organização dos requisitos funcionais e especificando os requisitos e contexto do sistema. (LUCIDCHART, 2019)

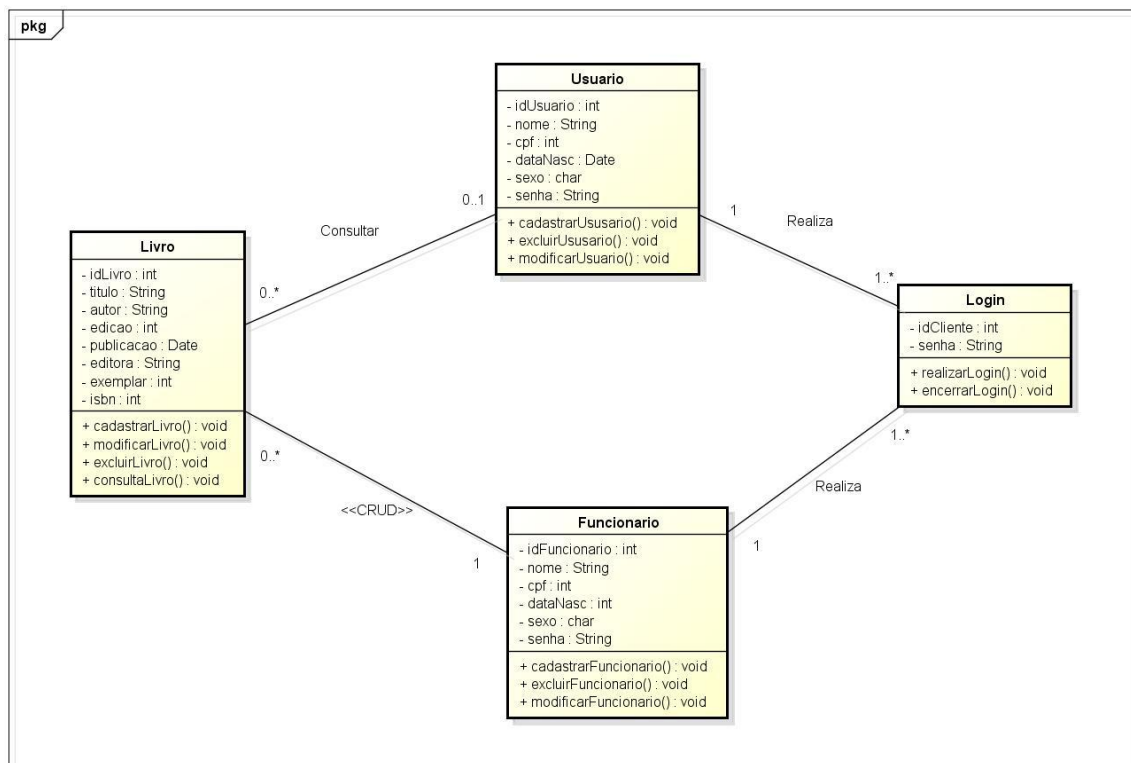


Os casos de uso podem ser explicados da seguinte forma:

- 1 - Realizar login: Todo usuário deve informar seu login e senha para ter acesso ao sistema, para controle de acessos.
- 2 - Consultar livros: Todo usuário deve poder visualizar o acervo de livros da biblioteca.
- 3 - <<CRUD>> Manter livros: O funcionário pode criar, consultar, alterar e remover livros do acervo.
- 4 - Criar clientes: Todo usuário pode cadastrar novos clientes.
- 5 - Encerrar login: É necessário encerrar o login para qualquer usuário do sistema.
- 6 - Criar funcionários: O funcionário pode cadastrar novos funcionários ao sistema

3.2 DIAGRAMA DE CLASSES

O Diagrama de Classes, assim como o Diagrama de Casos de Uso, faz parte da modelagem de requisitos, que têm sua importância na representação da descrição do que o cliente deseja, sendo a base para a criação de um projeto de software, bem como os requisitos necessários para que ele esteja dentro do esperado. No Diagrama de Classes, utiliza-se do cenário para representar as classes, que são a abstração de um objeto da vida real, suas relações com outras classes, seus atributos e operações que ela deve realizar. (PRESSMAN, 2011)



powered by Astah

No diagrama de classes existem quatro classes, sendo elas Usuario, Livro, Funcionario e Login. Cada classe tem seu id e seus atributos, que foram elencados de acordo com as requisições do cliente. Os relacionamentos acontecem entre Usuario e Login, pois um usuário pode realizar um ou vários logins no sistema, e da mesma forma um Funcionario pode realizar um ou vários logins. O relacionamento de Usuario e Livro se dá pois o Usuario irá consultar um livro do acervo, sendo que nenhum ou vários usuários podem consultar nenhum ou vários livros. E o relacionamento entre Funcionario e Livro consiste na criação, consulta, alteração, destruição e manutenção pelo Funcionario dos livros do acervo, onde um Funcionario pode realizar as operações de CRUD em nenhum ou vários livros.

No Diagrama de classes é possível visualizar os casos de uso da seguinte forma: Os atores se transformaram em classes, juntamente com os objetos que manipulam, sendo Livro e Login. Os casos de uso podem ser vistos mais explicitamente nos métodos de cada classe, por exemplo o Caso de Uso 1 está no método realizarLogin() da classe Login, o Caso de Uso 2 está no método consultaLivro() de Livro, o Caso de Uso 3 está nos métodos cadastrarLivro(),

modificarLivro(), consultaLivro() e excluirLivro() de Livro, já o Caso de Uso 4 está nos métodos cadastrarUsuario(), excluirUsuario() e modificarUsuario() de Usuario e cadastrarFuncionario(), excluirFuncionario() e modificarFuncionario() de Funcionario e o último Caso de Uso, sendo o 5, está no método encerrarLogin().

4 REVISÃO BIBLIOGRÁFICA

4.1 ENGENHARIA DE SOFTWARE

“O mundo moderno não funciona sem software”(SOMMERVILLE, 2019). Os softwares regem o mundo atual, pois a informatização é parte da rotina, sendo assim, tudo que fazemos está ligado a algum software.

O conceito “engenharia de software” surge em 1968, após a crise do software, período no qual o desenvolvimento de software era difícil, pois os defeitos não eram removidos e não se comportavam de maneira confiável, compreensível e eficiente, gerando diversos problemas (SILVA, 2015). Assim, a engenharia de software é uma disciplina que engloba todos os aspectos relacionados ao desenvolvimento de um software, desde seu início, onde são especificados os requisitos, até a manutenção desse sistema depois de pronto, e também os métodos e ferramentas necessários para a construção e documentação. (SOMMERVILLE, 2019)

A grande dificuldade da engenharia de software está em agradar o cliente. E para isso ela se utiliza de um processo sistemático, chamado processo de software, que possui quatro atividades principais, sendo elas: especificação, desenvolvimento, validação e evolução do software. Mas esse processo pode ser feito de várias formas, consistindo nas metodologias de desenvolvimento (SOMMERVILLE, 2019). Para que o desenvolvimento seja eficiente são necessários alguns aspectos, tais como comunicação, planejamento, boa execução e checagem dos resultados. Somente assim será possível garantir que o software seja seguro, confiável e de qualidade, que é o objetivo buscado pela engenharia de software, para que outros eventos como a crise do software não ocorram novamente. (PRESSMAN, 2011)

4.2 UML

A UML(*Unified Modeling Language* ou Linguagem Unificada de

Modelagem) é uma linguagem de notação gráfica para a modelagem e documentação de sistemas orientados a objetos. Para isso, são definidos elementos gráficos, como formas geométricas e outros elementos, a fim de representar uma abordagem do sistema através de diagramas, sendo uma abstração dele. O principal objetivo desses diagramas é fornecer uma visão clara e única do sistema. Não é preciso seguir à risca a notação de cada diagrama, podendo ser modificado conforme seu uso (NOLETO, 2020).

Existem três utilidades para os diagramas em um sistema, sendo: estímulo para o foco e discussão de um problema, documentação do sistema ou como descrição detalhada para gerar o código-fonte. (SOMMERVILLE, 2019).

Os diagramas da UML podem ser divididos em dois grupos: os estruturais, que especificam detalhes da arquitetura e estrutura do sistema, sendo a parte estática, e os comportamentais, que especificam a parte dinâmica do sistema, sendo o comportamento do sistema (VENTURA, 2021).

Existem treze tipos de diagramas, que permitem a visualização do sistema de diversas formas. Os cinco principais e mais usados são: diagrama de atividades, diagrama de casos de uso, diagrama de sequência, diagrama de classes e diagrama de máquinas de estado (SOMMERVILLE, 2019).

4.3 HTML

HTML (*HyperText Markup Language*) é uma linguagem de marcação criada por Tim Berners-Lee e o elemento mais essencial de uma aplicação Web. Ele permite a criação, estruturação, organização e formatação de documentos, seções, parágrafos, links, imagens e outros itens para páginas da internet. Ele permite realizar a estruturação básica de um website, pois a estilização deles é feita por meio de outras linguagens, como CSS e JavaScript. Seu objetivo não é criar um software, mas um conjunto de estruturas de uma página web e arrumar os componentes visuais. É uma ferramenta utilizada para fornecer informações para as pessoas que acessam os sites, navegadores e mecanismos de busca. (BERNARDO, 2020)

A extensão desses arquivos é .html ou .htm e podem ser visualizados em

qualquer navegador Web. Normalmente, um site consiste em várias páginas que são interligadas pelo código. Este é escrito por meio de tags ou elementos, que são os blocos de construção. Esses blocos são as marcações de conteúdo que o HTML faz, dividindo e criando hierarquia entre os blocos. (MARQUES, 2019; LONGEN, 2021)

4.4 CSS

CSS é a sigla para *Cascading Style Sheets*, ou seja, Folhas de Estilo em Cascatas. É utilizada para definir o estilo e a aparência de documentos desenvolvidos em linguagem de marcação (XML, HTML e XHTML). Ele surgiu, pois as linguagens de marcação não ofereciam os recursos necessários para o programador formatar as páginas de acordo com seu estilo, deixando-as mais atrativas. Utilizando o CSS é possível modificar cor, fonte, fundo, espaçamento, layout e muito mais. É uma ferramenta bastante diversa para a personalização de páginas para internet. Essa folha de estilos funciona por meio de camadas, que podem ser reutilizadas em mais elementos do código HTML. As principais vantagens do uso dessa linguagem associada ao HTML são a manutenção de um padrão de formatação, o controle da estilização em uma folha somente, facilitando a compreensão, a criação de formatações mais focadas no usuário e responsivas e a diminuição dos custos, pois evita a repetição e diminui as chances de erro. (GONÇALVES, 2021; OKUBO, 2021; PEREIRA, 2009)

4.5 PHP

PHP é uma linguagem de programação de alto nível voltada, principalmente, para o desenvolvimento Web, pois favorece a comunicação usuário-servidor. É uma linguagem de scripts open source, ou seja, é aberta e tem uma grande comunidade de desenvolvedores atuando na melhoria dela, além de poder ser comercializada sem exigir licenças de uso. Tem alta popularidade, pois tem sintaxe simples e fácil de aprender, tem alto desempenho, pois processa uma alta quantidade de dados e se integra com vários Bancos de Dados e é multiplataforma, sendo utilizada em grandes

empresas por essas características. Por ser escrita por meio de scripts, ela é processada no momento em que é executada, funcionando muito bem quando aliada às páginas HTML. Assim, seu principal uso é atrelado às aplicações Web, mas também é utilizada em scripts de linha de comando e aplicações para desktop. (FERREIRA, 2019; NOLETO, 2020)

4.6 JAVASCRIPT

JavaScript é uma linguagem de programação de alto nível, que tem suas aplicações voltadas para páginas Web, assim como o PHP. Foi criada pela Netscape Communications Corporation, com o objetivo de possibilitar maior dinamicidade e deixar mais agradável determinados processos Web. É uma linguagem leve, interpretada, orientada a objetos e é baseada em protótipos, mais conhecida como linguagem de script. É uma das mais importantes tecnologias para front-end, juntamente com HTML, CSS e PHP. Mas diferente do PHP, ela não realiza uma conexão com o servidor, sendo processada localmente e até mesmo sem internet. JavaScript é baseada em eventos e conta com diversas bibliotecas e frameworks, como o JQuery, o mais popular framework JavaScript. As principais vantagens desta linguagem de programação estão na sua versatilidade, rapidez, sintaxe fácil, leveza e alta compatibilidade com navegadores e plataformas. Porém, quando o assunto é segurança, ela deixa muito a desejar, não tendo muitos recursos nessa área. (SILVA, 2015; DORNELLES, 2016; ROVEDA, 2020)

4.7 SQL - MYSQL

O SQL, Structured Query Language ou Linguagem de Consulta Estruturada é uma linguagem padrão para lidar com bancos de dados relacionais. Ele não necessita de conhecimentos avançados em programação para ser utilizado para realizar consultas e pedidos de dados. Foi criado para que o acesso para modificação de dados pudesse ser simultâneo, descomplicado e unificado. Os principais comandos utilizados nessa linguagem são insert (inserir), search (pesquisar), update (atualizar) e delete (deletar), porém muitas outras consultas mais

avançadas podem ser realizadas. O SQL tem extrema importância no mundo, pois está sendo usado em todas as plataformas que armazenam algum dado, gerando uma demanda muito grande pelas empresas, sendo um dos pilares para quem quer trabalhar com tecnologia. (SILVEIRA, 2019; BACCA, 2019)

Existem muitos SGBDs (Sistema de Gerenciamento de Banco de Dados), que utilizam a linguagem SQL de maneira muito parecida para a criação e manipulação de bancos de dados relacionais. O mais utilizado deles é o MySQL, pois é gratuito e de código aberto, o que torna-o muito popular. Foi desenvolvido pela MySQL AB, depois comprado pela Sun Microsystems e, atualmente, pertence à Oracle, que implantou políticas mais restritivas. Os desenvolvedores, em desacordo com essas políticas, desenvolveram o projeto MariaDB, uma cópia aberta e gratuita.. (PISA, 2012)

4.8 MODELO DE DESENVOLVIMENTO XP (EXTREME PROGRAMMING)

A metodologia de desenvolvimento ágil XP (eXtreme Programming) é amplamente utilizada. Seus primeiros conceitos foram desenvolvidos em 1980, mas atualmente têm sido aprimorados e chamados de IXP (Industrial XP), mais refinado para o trabalho em grandes empresas e organizações. (PRESSMAN, 2011)

A metodologia XP foi desenvolvida segundo os ideais do Manifesto Ágil e contém cinco valores, que são a base para essa metodologia. São eles comunicação, simplicidade, feedback (retroalimentação), coragem e respeito. É necessária uma boa comunicação para que o trabalho seja desenvolvido com eficácia e eficiência, garantindo que todos saibam o que fazer. A simplicidade diz respeito a projetar somente o necessário imediatamente, sem se preocupar com coisas futuras, assim, achar problemas se torna mais rápido. O que nos leva ao valor do feedback, onde, através de testes dos projetos simples, os erros são achados para que possam ser consertados rapidamente. A coragem é necessária, pois muitas mudanças ocorreram e é necessário coragem para arriscar em novas soluções. E o respeito é essencial dentro de uma equipe, pois sem ele o trabalho se torna impossível, virando um ambiente de brigas. (PRESSMAN, 2011)

O processo XP consiste em quatro etapas ou atividades, sendo a primeira o planejamento, onde é necessário ouvir o cliente e o que ele busca, levantando os requisitos do software. A segunda é a etapa de projeto, que segue o princípio da simplicidade, onde é preferível algo simples do que elaborado e complexo. É importante falar que essa etapa acontece constantemente no projeto, pois o sistema é elaborado durante seu desenvolvimento. A terceira etapa é a codificação, onde os desenvolvedores devem pôr a “mão na massa” e realizar a codificação de pequenos protótipos do projeto, para que possam ser juntados aos poucos, conforme o sistema for desenvolvido. E a quarta etapa são os testes, que já iniciam na etapa de codificação, mas que se intensificam com a especialização do código. (PRESSMAN, 2011)

O XP possui muitas práticas, algumas serão apresentadas na imagem abaixo:

Princípio ou prática	Descrição
Propriedade coletiva	Os pares de desenvolvedores trabalham em todas as áreas do sistema de modo que não se desenvolvem ‘ilhas de conhecimento’, e todos os desenvolvedores assumem a responsabilidade por todo o código. Qualquer um pode mudar qualquer coisa.
Integração contínua	Assim que o trabalho em uma tarefa é concluído, ele é integrado ao sistema completo. Após qualquer integração desse tipo, todos os testes de unidade no sistema devem passar.
Planejamento incremental	Os requisitos são registrados em ‘cartões de história’, e as histórias a serem incluídas em um lançamento são determinadas de acordo com o tempo disponível e com sua prioridade relativa. Os desenvolvedores decompõem essas histórias em ‘tarefas’ de desenvolvimento (Figuras 3.5 e 3.6).
Representante do cliente	Um representante do usuário final do sistema (o cliente) deve estar disponível em tempo integral para o time de programação. Em um processo como esse, o cliente é um membro do time de desenvolvimento, sendo responsável por levar os requisitos do sistema ao time, visando sua implementação.
Programação em pares	Os desenvolvedores trabalham em pares, conferindo o trabalho um do outro e oferecendo o apoio necessário para que o resultado final seja sempre satisfatório.
Refatoração	Todos os desenvolvedores devem refatorar o código continuamente logo que sejam encontradas possíveis melhorias para ele. Isso mantém o código simples e de fácil manutenção.
Projeto (<i>design</i>) simples	Deve ser feito o suficiente de projeto (<i>design</i>) para satisfazer os requisitos atuais, e nada mais.
Lançamentos pequenos	O mínimo conjunto útil de funcionalidade que agregue valor ao negócio é desenvolvido em primeiro lugar. Os lançamentos do sistema são frequentes e acrescentam funcionalidade à primeira versão de uma maneira incremental.
Ritmo sustentável	Grandes quantidades de horas extras não são consideradas aceitáveis, já que o efeito líquido muitas vezes é a diminuição da qualidade do código e da produtividade no médio prazo.
Desenvolvimento com testes <i>a priori</i> (<i>test-first</i>)	Um <i>framework</i> automatizado de teste de unidade é utilizado para escrever os testes de um novo pedaço de funcionalidade antes que ela própria seja implementada.

Fonte: SOMMERVILLE, 2019

Os debates sobre essa metodologia são muitos, pois há um grande grau de dificuldade na sua implantação e alguns autores, como Sommerville (2019) dizem que é um modelo falho e que é impossível de ser implementado em equipes, mas alguns, como Pressman (2011), dizem que para que a metodologia funcione, como aconteceu em muitas organizações, é necessário o conhecimento e adaptação às

necessidades específicas de quem está utilizando-a.

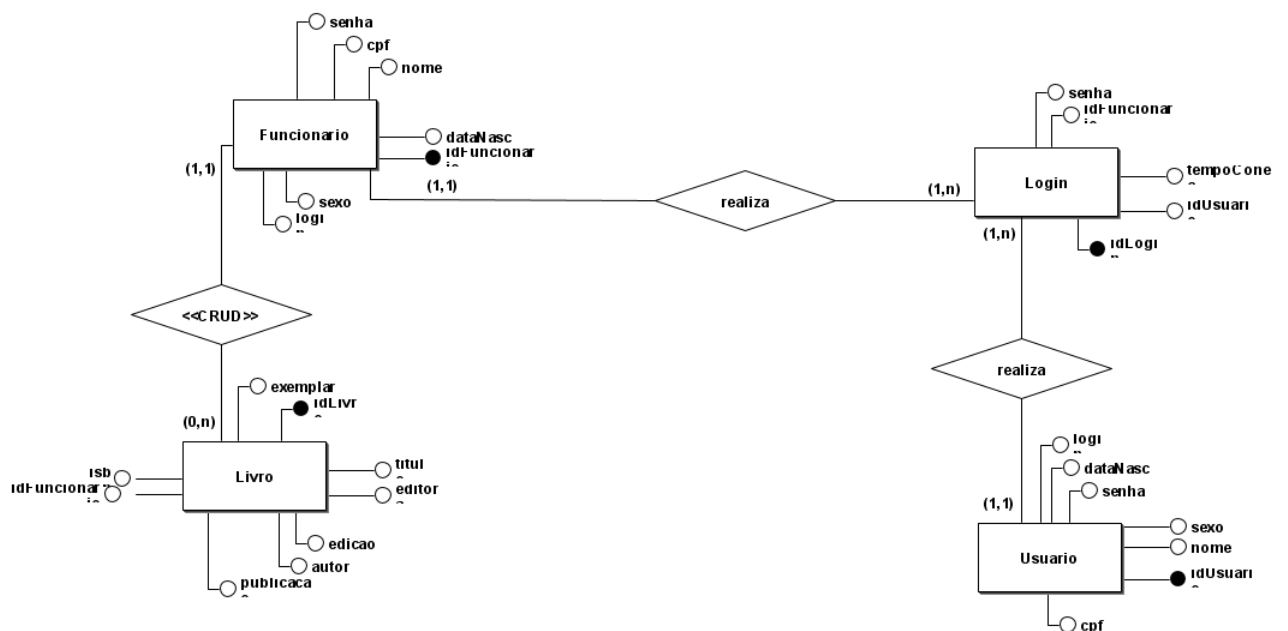
5 BANCO DE DADOS

Banco de Dados, como o próprio nome sugere, são conjuntos de dados, armazenados de maneira organizada, consistente, protegida e acessível facilmente. Para realizar todos esses procedimentos com os dados são utilizados os SGBDs (Sistemas de Gerenciamento de Banco de Dados), softwares que permitem que os dados sejam manipulados e protegidos. (ALECRIM, 2018)

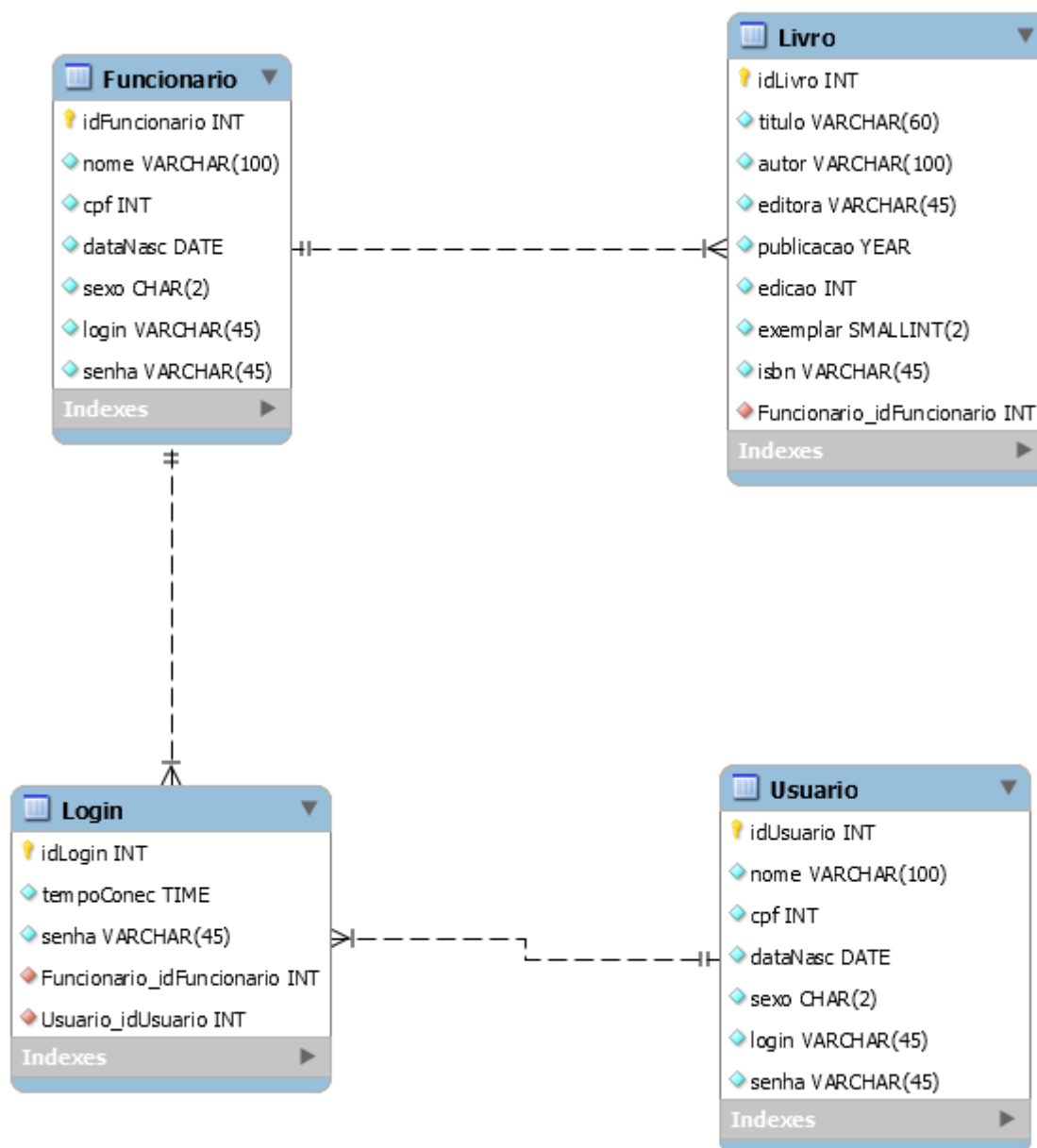
Os bancos de dados são representações da realidade, chamadas de minimundos, e alterações nos minimundos resultam em mudanças no banco de dados. Eles têm uma lógica coerente, sendo ordenados. Existem diferentes tipos de bancos de dados, sendo mais utilizado o banco de dados relacional. (MACHADO, 2014)

É impossível utilizarmos a internet sem falarmos de bancos de dados, pois eles estão em todos os lugares. Normalmente um usuário leigo não vê ele, pois são projetados para estarem escondidos, porém cada informação a seu respeito é armazenada em algum banco de dados do mundo. (ALECRIM, 2018)

5.1 MODELO DE ENTIDADE E RELACIONAMENTO



5.2 DIAGRAMA DE ENTIDADE E RELACIONAMENTO



No modelo do Banco de Dados, temos quatro entidades, sendo elas Usuario, Login, Funcionario e Livro. Elas possuem seus identificadores, bem como demais atributos. A entidade Usuario se relaciona com Login, pois um Usuario pode realizar um ou muitos Logins no sistema. Da mesma forma, um Funcionario pode realizar um ou vários Logins. O relacionamento entre Funcionario e Livro se dá pelo motivo de que um Funcionario pode cadastrar, alterar ou excluir um ou vários livros do acervo da Biblioteca.

5.3 MODELO FÍSICO - SQL

Script:

```
create database Biblioteca;
```

```
use Biblioteca;
```

```
create table Funcionario(
```

```
    idFuncionario int unique not null auto_increment,
```

```
    nome varchar(100) not null,
```

```
    cpf int not null unique,
```

```
    dataNasc date not null,
```

```
    sexo char(2) not null,
```

```
    login varchar(45) not null,
```

```
    senha varchar(45) not null,
```

```
    primary key(idFuncionario)
```

```
);
```

```
create table Usuario(
```

```
    idUsuario int unique not null auto_increment,
```

```
    nome varchar(100) not null,
```

```
    cpf int not null unique,
```

```
    dataNasc date not null,
```

```
    sexo char(2) not null,
```

```
    login varchar(45) not null,
```

```
    senha varchar(45) not null,
```

```
    primary key(idUsuario)
```

```
);
```

```
create table Livro(
```

```
    idLivro int unique not null auto_increment,
```

```
    titulo varchar(60) not null,
```

```

    autor varchar(100) not null,
    editora varchar(45) not null,
    publicacao year not null,
    edicao int not null,
    exemplar smallint not null,
    isbn varchar(45) not null,
    idFuncionario int not null,
    primary key(idLivro),
    foreign key(idFuncionario) references Funcionario(idFuncionario)
);

```

```

create table Login(
    idLogin int unique not null auto_increment,
    senha varchar(45) not null,
    tempoConec time not null,
    idUsuario int not null,
    idFuncionario int not null,
    primary key(idLogin),
    foreign key(idUsuario) references Usuario(idUsuario),
    foreign key(idFuncionario) references Funcionario(idFuncionario)
);

```

A estrutura física do Banco de Dados descrita acima é resultado da conversão do modelo lógico para a estrutura física. Cada tabela é a representação de uma entidade, juntamente com seus atributos. Os relacionamentos se dão da mesma forma e são representados pelas chaves estrangeiras presentes nas tabelas. Todos os atributos tem a característica de não serem nulos, pois são obrigatórios no sistema.

6 DESENVOLVIMENTO

6.1 CONEXÃO COM BANCO DE DADOS

Script da conexão com o banco de dados:

```
<?php
    $servidor = "localhost:3307";
    $usuario = "root";
    $senha = "";
    $dbname = "biblioteca";

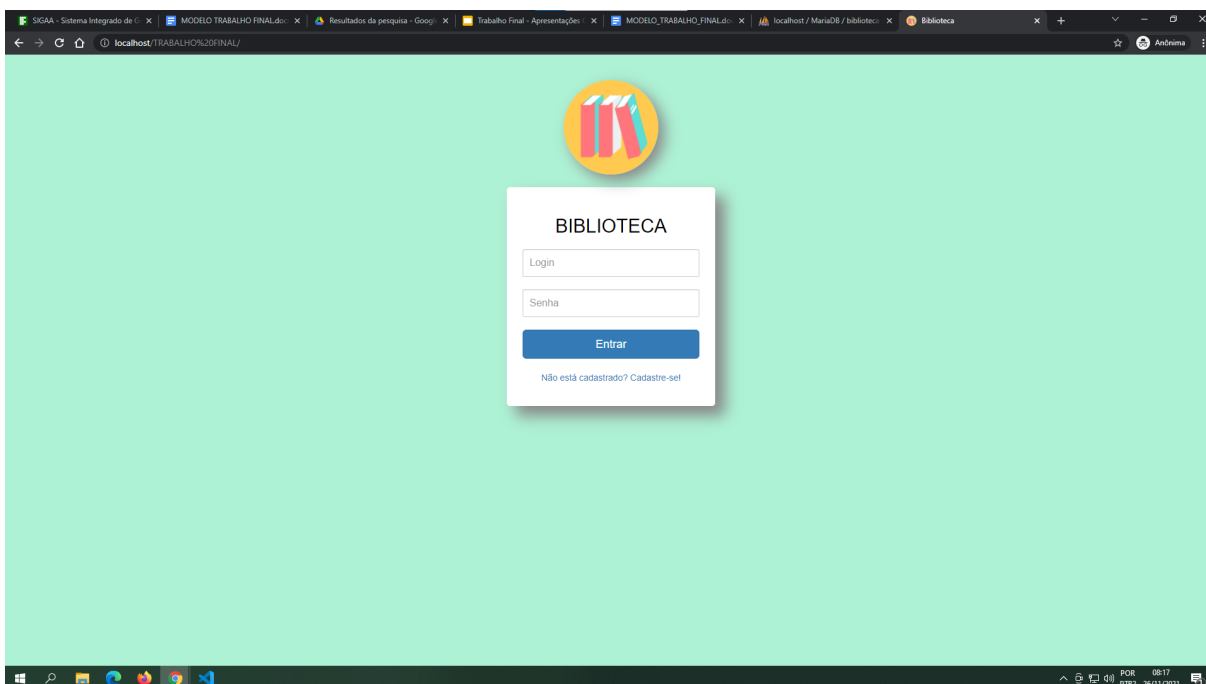
    //Criar a conexao
    $conn = mysqli_connect($servidor, $usuario, $senha, $dbname);

    if(!$conn){
        $_SESSION['ERRO'] = "Falha na conexão: " . mysqli_connect_error();
        header("Location:../erro/erro.php");
    }
?>
```

Primeiramente, para que seja realizado um comando em PHP, é necessário abrir a tag do código PHP, sendo “<?php” e ao final do código fechá-la, “?>”. Depois disso, foram criadas quatro variáveis, onde são armazenados, respectivamente, a localização do servidor e sua porta, o nome do usuário do banco, a senha desse usuário e, por último, o nome do banco de dados que deseja acessar. Essas variáveis são necessárias para a próxima instrução, que consiste em realizar a conexão com o banco de dados. Para isso, cria-se uma variável para armazenar o resultado da conexão e executa-se o comando *mysqli_connect()*, que realiza a conexão com um banco de dados especificado por meio da seguinte ordem de parâmetros para realização da função: *mysqli_connect*(nome do servidor e porta; nome do usuário; senha do usuário; nome do banco de dados). Dessa forma, a função realizará a conexão. Após isso, é necessário ver o resultado da conexão, pois podem haver erros de digitação nos dados para acesso ao banco. Essa

verificação se dará por meio de uma estrutura de seleção if, que verificará se o resultado de conexão é “false”, se sim, ocorreu um erro ao conectar com o banco de dados e o sistema criará uma variável de sessão que apresentará um texto de falha na conexão, juntamente com o resultado da função *mysqli_connect_error()*, que mostra qual foi o erro ocorrido ao tentar executar a conexão. Em seguida, o sistema redirecionará para a página de erros do sistema, onde irá recuperar essa sessão e apresentá-la na tela para visualização do usuário. Se o resultado da variável de conexão for verdadeiro, o sistema só continuará normalmente com a execução das próximas ações e eventos requeridos a ele.

6.2 TELA LOGIN



Esta é a tela para realizar login de todos os usuários, para que possam acessar ao sistema. O usuário deve estar cadastrado no sistema, caso contrário, há um link embaixo do botão entrar que o direciona para a tela de cadastro de usuário. Se ele estiver cadastrado, ao clicar no botão “ENTRAR”, o sistema realizará uma verificação por meio de uma consulta ao banco de dados verificando se há algum usuário com os dados digitados, como mostra o script abaixo:

```
$result_usuario = "SELECT * FROM view_func_user WHERE login =
'$usuario' && senha = '$senha' LIMIT 1";

$resultado_usuario = mysqli_query($conn, $result_usuario);
```

```
$resultado = mysqli_fetch_assoc($resultado_usuario);
```

Se a consulta achar um usuário com os dados correspondentes, ele será redirecionado para a tela de consulta ao acervo e caso não ache, ele criará uma sessão de erro no login, que exibirá a mensagem “Usuário ou senha Inválido”.

6.3 TELA CADASTRO DE USUÁRIO

Nesta tela será cadastrado um novo usuário ao sistema. Após preencher todos os dados, ele deve clicar no botão “CADASTRAR”, onde será realizada uma operação de inserção no banco de dados, de acordo com o script abaixo:

```
<?php

session_start();

include_once("../include/conexao.php");

$nome = $_POST["nome"];

$cpf = $_POST["cpf"];

$login = $_POST["login"];

$senha = $_POST["senha"];
```

```

$sexo = $_POST["sexo"];

$dataNasc = $_POST["dataNasc"];

$sql = "INSERT INTO USUARIO (nome, ".
"cpf, dataNasc, sexo, login, senha) ".
" VALUES (".
"''.$nome.'', ".
"''.$cpf.'', ".
"''.$dataNasc.'', ".
"''.$sexo.'', ".
"''.$login.'', ".
"''.$senha.'')";

if(mysqli_query($conn, $sql)){
    mysqli_close($conn);
    header("Location: ../index.php");
} else {
    mysqli_close($conn);
    $_SESSION["ERRO"] = "Ocorreu um erro ao Cadastrar o Usuário!";
    echo $sql;
    header("Location: ../erro/erro.php");
}

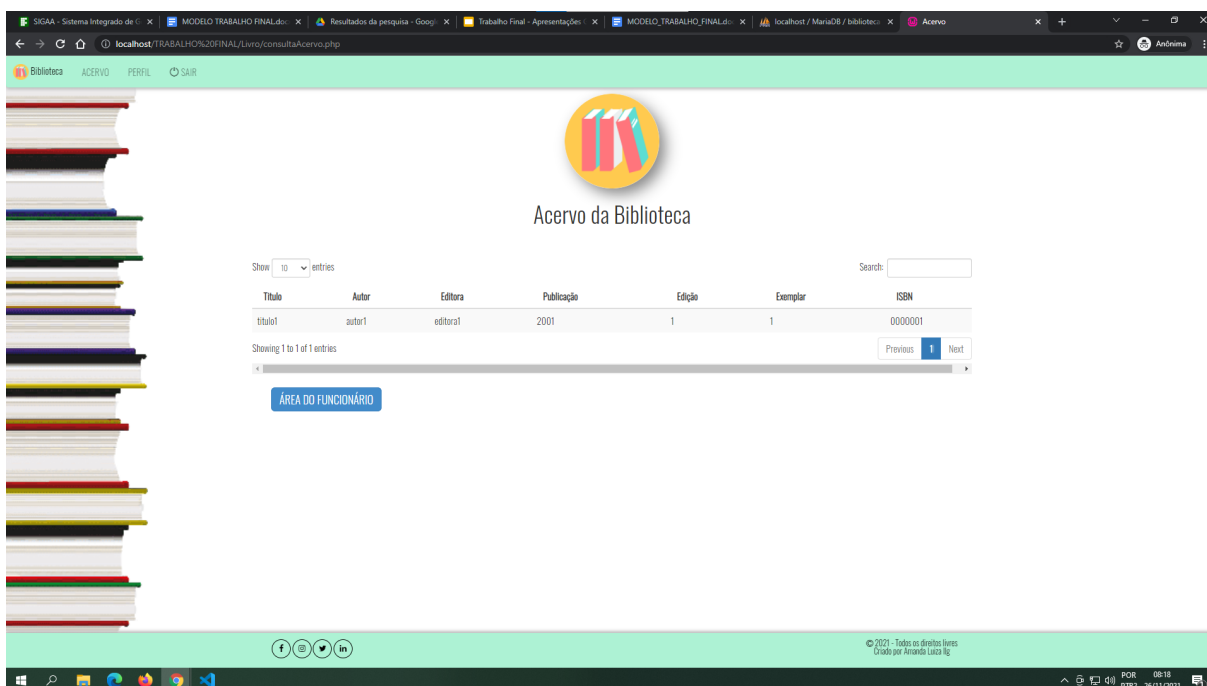
?>

```

Após realizar a inserção, o usuário será direcionado para a página de login, ou, se houver falha na execução dessa instrução, para a página de erro.

O botão “VOLTAR” redireciona o usuário de volta para a tela de login.

6.4 TELA CONSULTA DO ACERVO



Na tela de consulta do acervo, os livros cadastrados no banco de dados são consultados e impressos na tela no formato de uma tabela. Conforme os scripts abaixo:

Script para consulta no banco de dados:

```
<?php

include_once("../include/conexao.php");

$sql = "select * from livro";

$resultado = mysqli_query($conn, $sql);

while($row = $resultado -> fetch_array()){

    $rows[] = $row;

}

mysqli_close($conn);

?>
```

Script para impressão no formato de tabela:

```

<div class="table-responsive">

    <table id="clienteTable" class="table table-striped
table-bordered" style="width:100%">

        <thead class="thead-dark">

            <tr>

                <th>Titulo</th>

                <th>Autor</th>

                <th>Editora</th>

                <th>Publicação</th>

                <th>Edição</th>

                <th>Exemplar</th>

                <th>ISBN</th>

            </tr>

        </thead>

        <tbody>

            <?php

                foreach($rows as $row){

                    ?>

                    <tr>

                        <td><?php echo $row['titulo'] ?></td>

                        <td><?php echo $row['autor'] ?></td>

                        <td><?php echo $row['editora'] ?></td>

                        <td><?php echo $row['publicacao'] ?></td>

                        <td><?php echo $row['edicao'] ?></td>

                        <td><?php echo $row['exemplar'] ?></td>

                        <td><?php echo $row['isbn'] ?></td>

```

```

        </tr>

        <?php
        }

        ?>

    </tbody>

</table>

</div>

```

A tabela de livros pode realizar a listagem deles limitada a uma quantidade por subpáginas, que pode ser definida e também realiza a busca de itens por meio da barra de pesquisa acima da tabela.

Há ainda o botão “ÁREA DO FUNCIONÁRIO”, que redireciona o usuário, quando autorizado, ou seja, quando ele estiver cadastrado como funcionário, para a página que possui as ações que somente o funcionário pode realizar.

6.5 TELA PERFIL

The screenshot shows a web browser window with the URL `localhost/Trabalho%20Final/Usuario/formEditUsuario.php`. The page has a green header with navigation links: Biblioteca, ACESSO, PERFIL, and SAIR. On the left, there is a decorative image of a stack of books. The main content area is titled 'Perfil' and contains a form with the following fields:

- Nome:** Administrador
- CPF:** 0
- Data de Nascimento:** dd/mm/aaaa
- Sexo:**
 - ☐ Masculino
 - ☐ Feminino
 - ☒ Outro
- Login:** admin
- Senha:** (masked with dots)

At the bottom right of the form are two buttons: 'Salvar' and 'Voltar'. The footer of the page includes social media icons and a copyright notice: '© 2021 - Todos os direitos reservados. Criado por Alexandre Lima Ig'.

Na tela de perfil, o sistema recupera as informações do usuário logado, por meio de sessões e consulta ao banco de dados, por meio do script abaixo:

```

<?php

include_once("../include/conexao.php");

if(isset($_SESSION['idUsuario'])){

    $id = $_SESSION['idUsuario'];

    $cpf = $_SESSION["cpfUsuario"];

    $sql = "SELECT * FROM view_func_user".

    " WHERE idUsuario = '". $id.'" and cpf = '". $cpf.'"';

    $sql_query = mysqli_query($conn, $sql);

    if(!$sql_query){

        $_SESSION["ERRO"] = "Ocorreu um erro ao recuperar o
Usuario!";

        mysqli_close($conn);

        header("Location: ../erro/erro.php");

    }

    $dadosUsuario = mysqli_fetch_row($sql_query);

    $id = $dadosUsuario[0];

    $nome = $dadosUsuario[1];

    $dataNasc = $dadosUsuario[3];

    $sexo = $dadosUsuario[4];

    $login = $dadosUsuario[5];

    $senha = $dadosUsuario[6];

}

?>

```

Essas informações são impressas na tela e podem ser modificadas pelo usuário, porém, para salvá-las no banco de dados o usuário deve clicar no botão

“SALVAR”, que fará a atualização dos dados utilizando o script abaixo, que primeiramente verifica em qual tabela do banco de dados está o usuário (se ele é um usuário comum ou um funcionário).

```
<?php

session_start();

include_once("../include/conexao.php");

$id = $_POST['idUserario'];

$nome = $_POST["nome"];

$cpf = $_POST["cpf"];

$login = $_POST["login"];

$senha = $_POST["senha"];

$sexo = $_POST["sexo"];

$dataNasc = $_POST["dataNasc"];

unset($_SESSION['cpfUsuario']);

if(isset($_POST['funcionario'])) {

    $sql = "UPDATE funcionario SET nome = '". $nome.

    "', cpf = '". $cpf.

    "', login = '". $login.

    "', senha = '". $senha.

    "', sexo = '". $sexo.

    "', dataNasc = '". $dataNasc.

    "' where idFuncionario = '". $_POST['idUserario']. "'";

    $_SESSION["idUserario"] = $_POST['idUserario'];

} else {

    $sql = "UPDATE usuario SET nome = '". $nome.
```



```

        "', cpf = '". $cpf.

        "', login = '". $login.

        "', senha = '". $senha.

        "', sexo = '". $sexo.

        "', dataNasc = '". $dataNasc.

        "' where idUsuario = '". $id. "'";

        $_SESSION["idUsuario"] = $id;

    }

    $_SESSION["cpfUsuario"] = $_POST["cpf"];

    if(mysqli_query($conn, $sql)){

        mysqli_close($conn);

        header("Location:formEditUsuario.php");

        $_SESSION['AtualSucesso'] = "Atualizado com sucesso!";

    } else {

        mysqli_close($conn);

        $_SESSION["ERRO"] = "Ocorreu um erro ao Alterar o Usuário!";

        header("Location: ../erro/erro.php");

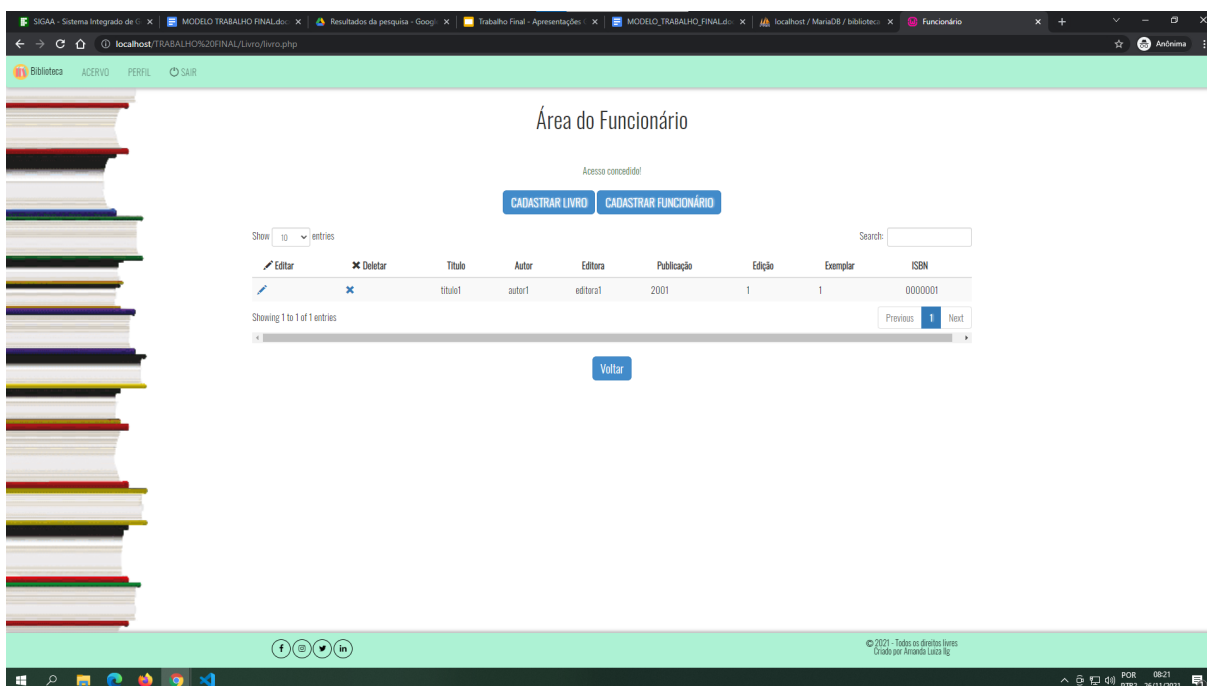
    }

?>

```

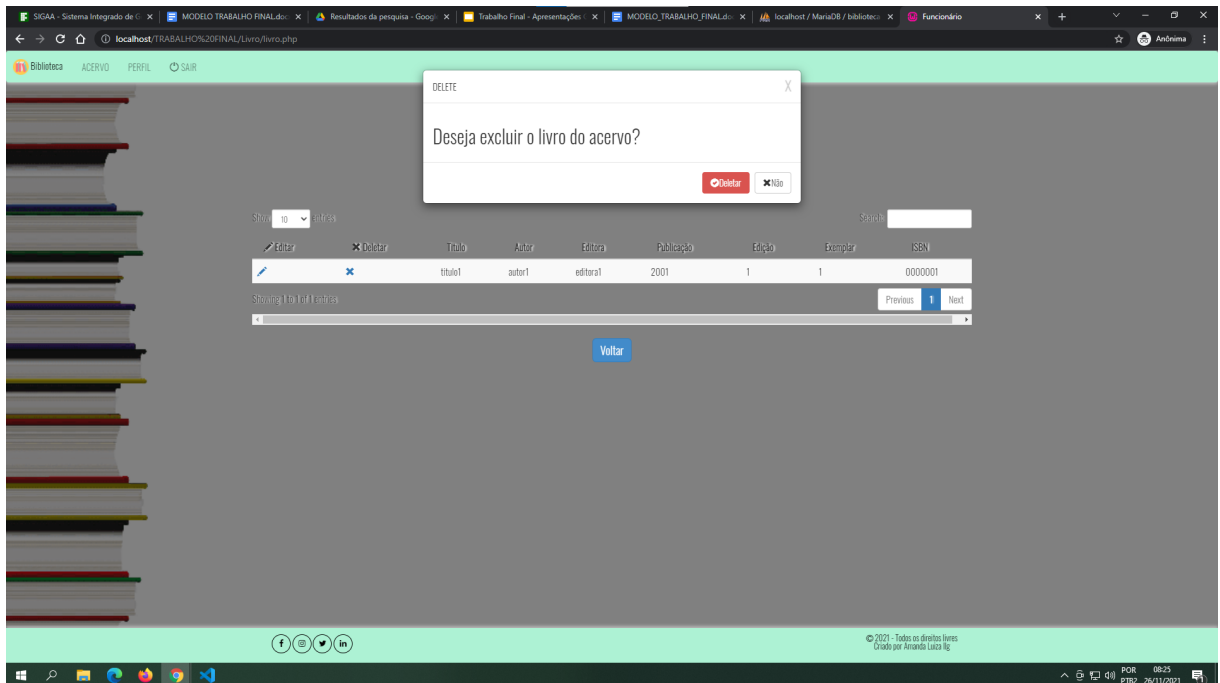
O botão “VOLTAR” redireciona o usuário de volta para a tela de consulta ao acervo.

6.6 TELA ÁREA DO FUNCIONÁRIO



Esta é uma tela de atalho para todas as funcionalidades que um funcionário pode realizar no sistema. Só é possível acessá-la se o usuário logado for um usuário. Nesta tela, assim como na de consulta, são listados os livros cadastrados em uma tabela e é possível pesquisá-los e estabelecer uma quantidade de itens por subpágina. O botão “CADASTRO DE LIVRO” e “CADASTRO DE FUNCIONÁRIO” direcionam para as páginas para cadastro de cada um destes objetos. Há também as opções de editar e deletar os livros da tabela, por meio dos ícones presentes na própria tabela. O ícone do lápis direciona o usuário para a página de edição dos dados do livro. O ícone do “X” dispara a ação de uma *modal*, que está descrita abaixo.

6.6.1 *Modal Delete*



Ao clicar no “x” da área do funcionário a ação de uma modal de confirmação de exclusão é disparada, fazendo com que o resultado acima seja apresentado. Se o usuário realmente deseja excluir, ele deve clicar no botão “Deletar”, que executará a ação de deletar o livro do banco de dados, mas se a ação não é desejada, é possível clicar no botão “Não” ou no “X” no canto superior direito.

6.7 TELA CADASTRO DE LIVRO

A tela de cadastro de livro consiste em um formulário que deve ser

preenchido pelo funcionário e, após isso, ele deve clicar no botão “CADASTRAR”, para que seja feita uma inserção, semelhante a do cadastro do usuário, mudando os atributos e a tabela de inserção. E o botão “Voltar” redireciona de volta à página de consulta ao acervo.

6.8 TELA CADASTRO DE FUNCIONÁRIO

The screenshot shows a web browser window displaying the 'Cadastro de Funcionário' form. The browser's address bar shows the URL 'localhost:TRABALHO%20FINAL/Usuario/formFuncionario.php'. The page has a green header with navigation links: 'Biblioteca', 'ACERVO', 'PERFIL', and 'SAIR'. The form itself is titled 'Cadastro de Funcionário' and contains the following fields:

- Nome:** A text input field with the placeholder 'Nome Completo'.
- CPF:** A text input field with the placeholder 'CPF'.
- Data de Nascimento:** A date input field with the placeholder 'dd/mm/aaaa' and a calendar icon.
- Sexo:** Three radio button options: 'Masculino', 'Feminino', and 'Outro'.
- Login:** A text input field with the placeholder 'Login'.
- Senha:** A text input field with the placeholder 'Senha'.

At the bottom of the form are two blue buttons: 'CADASTRAR' and 'VOLTAR'. The footer of the page includes social media icons (Facebook, Twitter, LinkedIn) and a copyright notice: '© 2021 - Todos os direitos reservados. Criado por Alexandre Lusa Jr.'.

A tela de cadastro de funcionário, assim como as outras de cadastro, é um formulário que, depois de preenchido pelo funcionário, pode ser inserido no banco de dados do sistema por meio do clique no botão “CADASTRAR”. A inserção se dá por meio de um script muito semelhante ao de inserção do usuário, mudando somente a tabela de inserção. O botão “VOLTAR” redireciona de volta para a página de consulta ao acervo.

6.9 TELA MODIFICAR LIVRO

Alterar Livro

Titulo:

Autor:

Editora:

Publicação:

Edição:

Exemplar:

ISBN:

© 2021 - Todos os direitos reservados. Criado por Amanda Lúcia Rg.

Na tela de alteração dos dados de um livro, ocorre a recuperação dos dados de livro por meio de scripts semelhantes aos de recuperação dos dados do usuário na tela de perfil. Esses dados recuperados podem ser modificados e salvos no banco de dados por meio do botão “Salvar”, que realizará uma ação de atualização dos dados por meio do seguinte script:

```
<?php

session_start();

include_once("../include/conexao.php");

$id = $_POST["idLivro"];

$titulo = $_POST["titulo"];

$autor = $_POST["autor"];

$editora = $_POST["editora"];

$publicacao = $_POST["publicacao"];

$edicao = $_POST["edicao"];

$exemplar = $_POST["exemplar"];

$isbn = $_POST["isbn"];
```

```

$idFuncionario = $_SESSION['idUserario'];

$sql = "UPDATE livro SET titulo = '". $titulo.

"', autor = '". $autor.

"', editora = '". $editora.

"', publicacao = '". $publicacao.

"', edicao = '". $edicao.

"', exemplar = '". $exemplar.

"', isbn = '". $isbn.

"', idFuncionario = '". $idFuncionario.

"' where idLivro = '". $id. "'";

if(mysqli_query($conn, $sql)){

    mysqli_close();

    header("Location:consultaAcervo.php");

} else {

    mysqli_close($conn);

    $_SESSION["ERRO"] = "Ocorreu um erro ao Alterar o Livro!";

    header("Location: ../erro/erro.php");

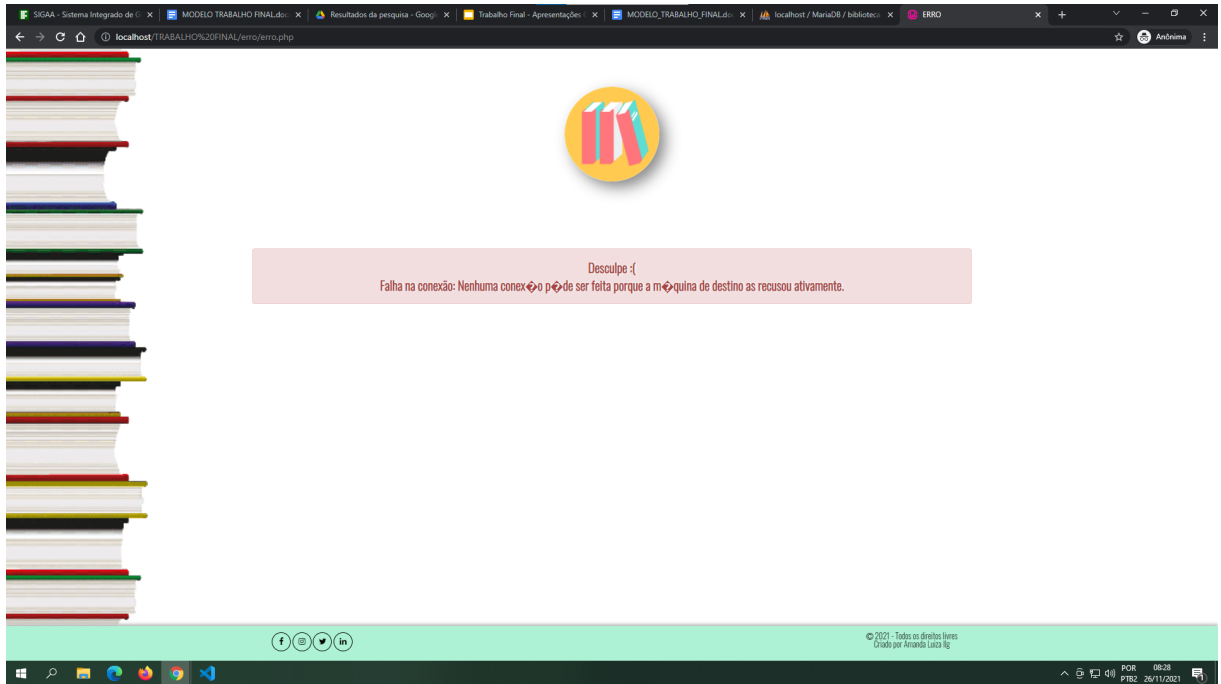
}

?>

```

O botão “Voltar” redireciona para a página de consulta ao acervo.

6.10 TELA ERRO

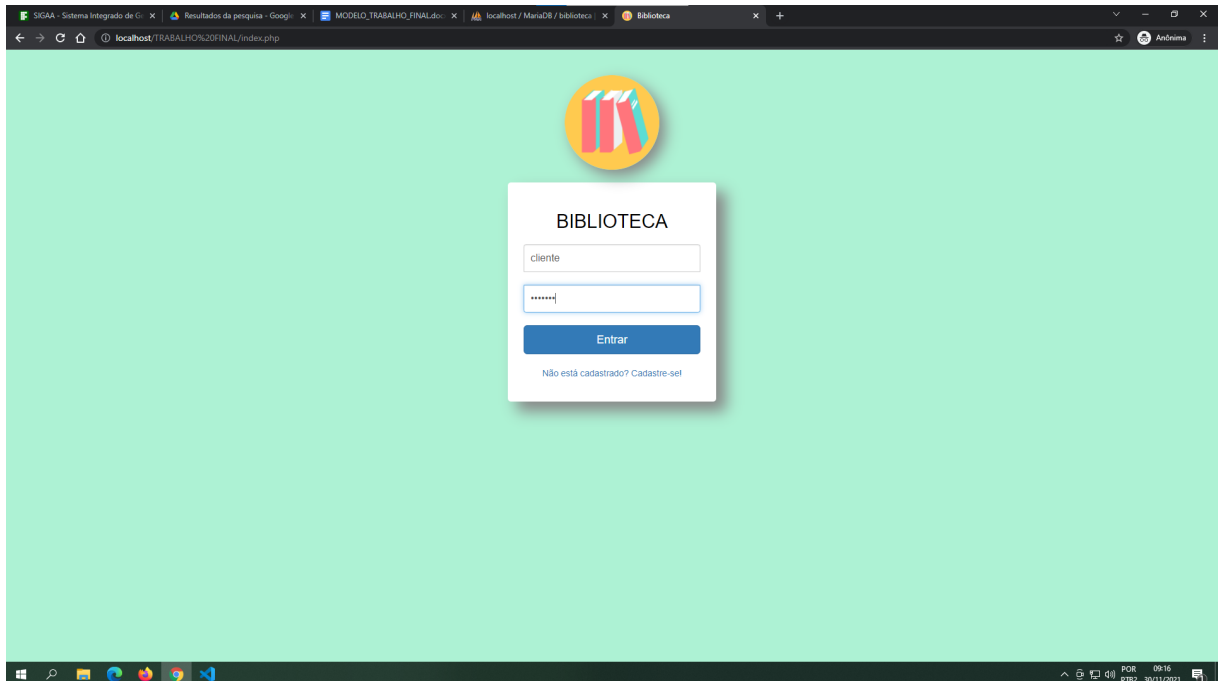


Esta tela é o redirecionamento para qualquer erro que ocorrer na execução de uma ação do sistema. Ao ocorrer o erro, uma variável de sessão é criada com um valor de mensagem do erro e direciona à esta página. A variável de sessão é recuperada nesta tela e sua mensagem é apresentada ao usuário no quadro vermelho de alerta.

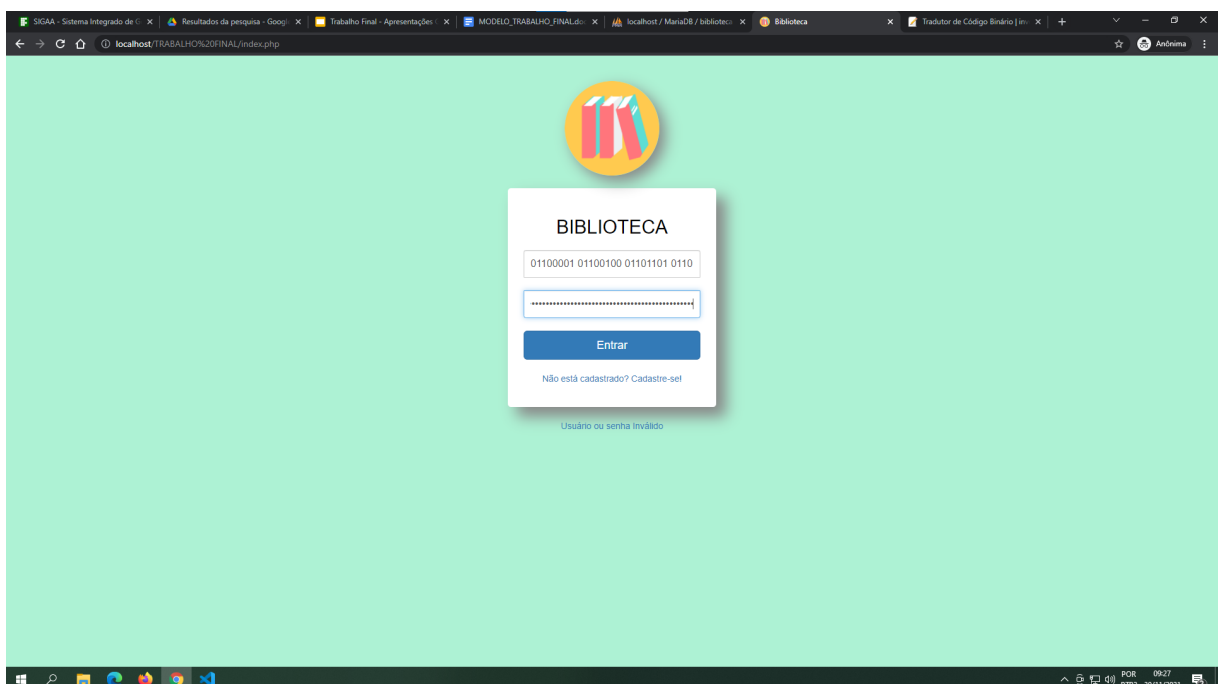
7 TESTES DE SOFTWARE

Será realizado o teste da Tela de Login, tentando inserir valores em formato de texto, números, binário e comandos SQL nos campos de login e senha e entrar no sistema. O resultado esperado é o erro dessas tentativas, informando que usuário e senha estão inválidos. A entrada no sistema será autorizada somente quando o usuário e senha forem iguais aos dados do banco.

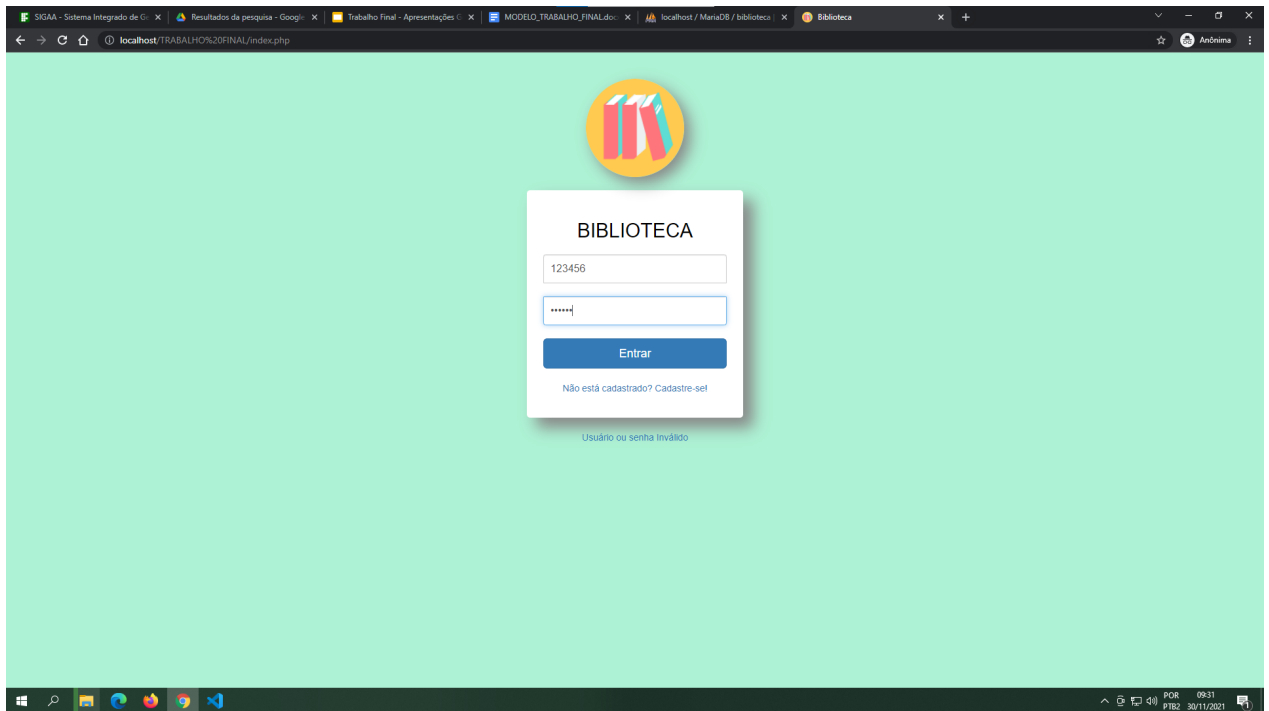
Teste com texto aleatório em ambos os campos:



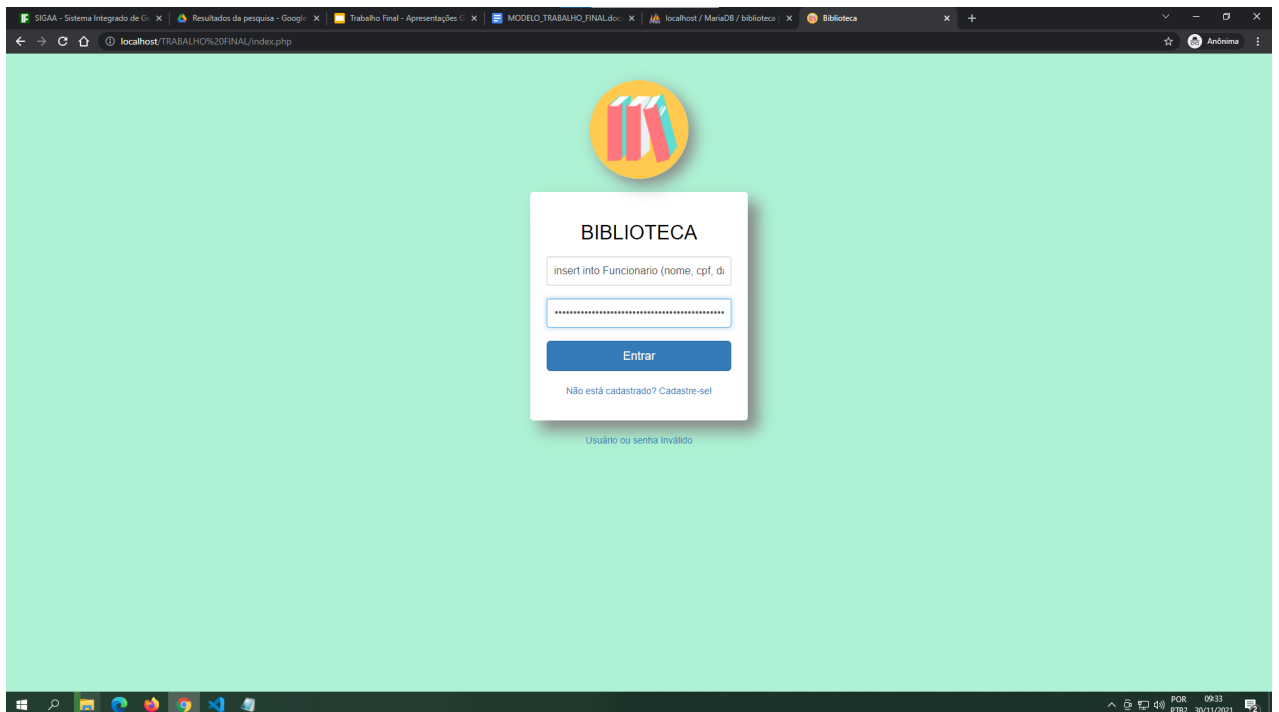
Teste com caracteres binários de um usuário cadastrado:



Teste com números aleatórios em ambos os campos:

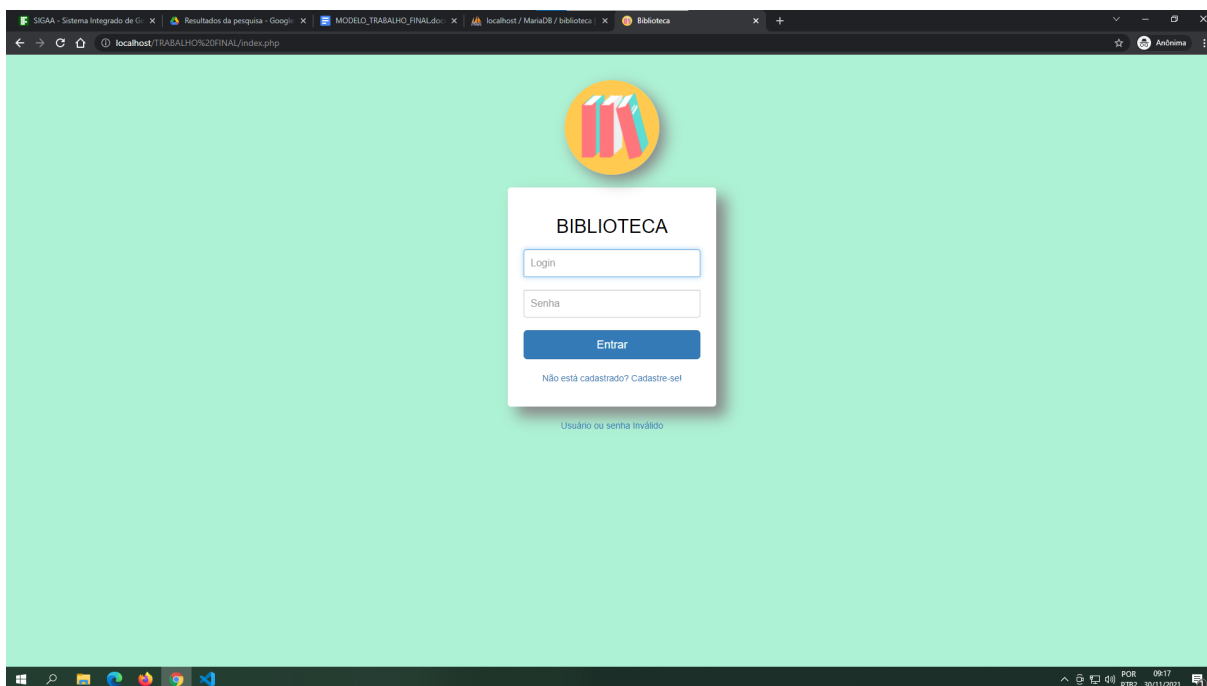


Teste com comando SQL de insert:

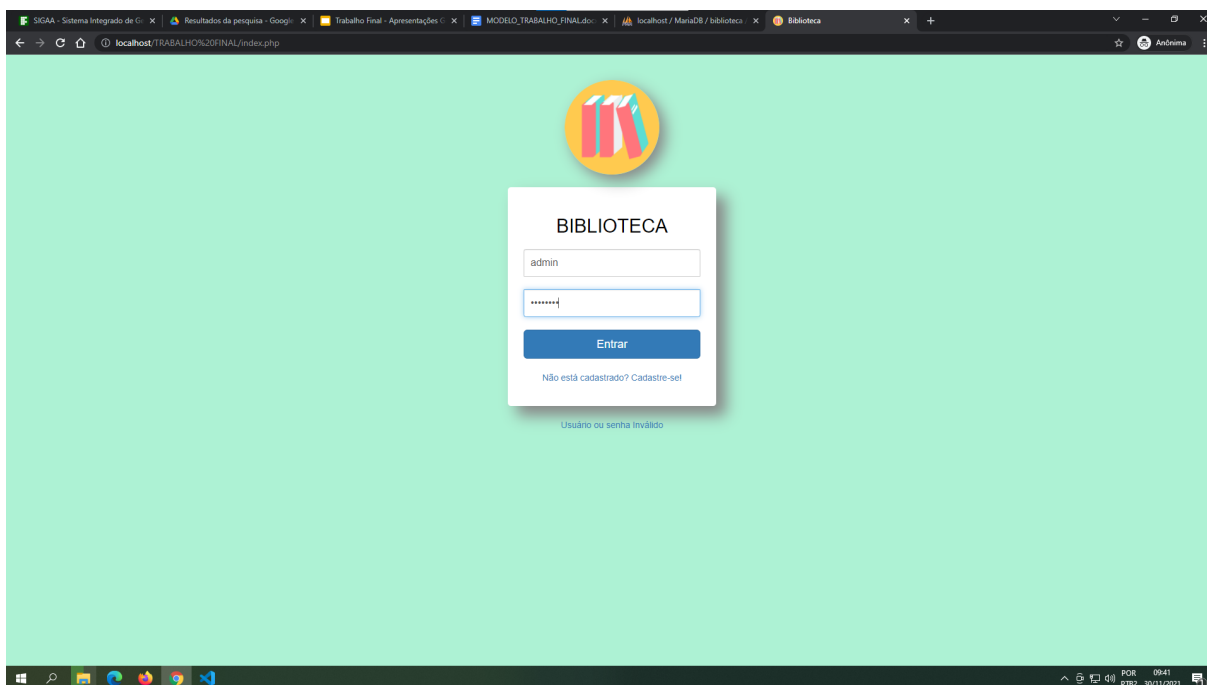


O método `mysqli_real_escape_string()` impede que comandos SQL sejam executados, pois ele escapa com caracteres especiais, como aspas.

Em nenhum dos testes o login foi realizado, resultando no erro de usuário ou senha inválidos, como mostrado abaixo:



Somente ao realizar o login com um usuário cadastrado foi possível entrar no sistema, como mostrado abaixo:



Acervo da Biblioteca

Show 10 entries

Search:

Título	Autor	Editora	Publicação	Edição	Exemplar	ISBN
titulo1	autor1	editora1	2001	1	1	0000001

Showing 1 to 1 of 1 entries

Previous 1 Next

ÁREA DO FUNCIONÁRIO

© 2021 - Todos os direitos reservados. Criado por Amanda Lima Rg

30/11/2021 09:42

8 CONSIDERAÇÕES FINAIS

Considerando todo o desenvolvimento do sistema, desde seu levantamento de requisitos necessários, tanto funcionais, não funcionais e regras de negócio, construção dos diagramas de casos de uso e de classes, para melhor visualização das necessidades do sistema, desenvolvimento dos modelos conceitual, lógico e físico do banco de dados, até a programação do sistema, pode-se considerar que o sistema atende os requisitos definidos, sem a apresentação de falhas graves. Podem haver limitações, pois o produto final não sofreu um processo de análise rigoroso, mas considerando seu uso comum, ele cumpre os objetivos definidos. É um software simples, mas que pode ser aprimorado para usos mais amplos e com mais funcionalidades. Considerando o aprofundamento bibliográfico, a abordagem dos temas é sucinta, mas apresenta as bases de cada tópico, dando uma noção básica ao leitor sobre os temas.

9 REFERÊNCIAS

VINCO. **5 técnicas para o levantamento de requisitos de software**. 2019.

Disponível em: <https://blog.vinco.com.br/levantamento-de-requisitos-de-software/>. Acesso em: 15 nov. 2021.

WAZLAWICK, Raul Sidnei. **Requisitos**. 2010. Disponível em:

https://edisciplinas.usp.br/pluginfile.php/4178861/mod_resource/content/1/slides%2003%20Modelagem%20de%20Requisitos.pdf#:~:text=que%20restri%C3%A7%C3%B5es%20%C3%B3gicas%20. Acesso em: 15 nov. 2021.

SCHEMES, Taynara. **O que são regras de negócio em TI?** 2020. Disponível em:

<https://conteudo.movidesk.com/regras-de-negocio-ti/#o-que-sao-regras-de-negocio-em-ti>. Acesso em: 15 nov. 2021.

COUTINHO, Thiago. **Desenvolvimento de Software**. 2021. Disponível em:

https://www.google.com/url?q=https://www.voitto.com.br/blog/artigo/o-que-e-desenvolvimento-de-software&sa=D&source=docs&ust=1637953088404000&usg=AOvVaw3_vlltQ68L1v7MDrPoYfGG. Acesso em: 15 nov. 2021.

PINTO, Hudson Lamounier. **Desenvolvimento de Software**. 2007. Disponível em:

<https://www.devmedia.com.br/atividades-basicas-ao-processo-de-desenvolvimento-de-software/5413>. Acesso em: 15 nov. 2021.

LUCIDCHART. **Diagrama de caso de uso UML**. 2021. Disponível em:

<https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>. Acesso em: 15 nov. 2021.

SILVA, Débora. **Engenharia de software**. 2015. Disponível em:

<https://www.estudopratico.com.br/engenharia-de-software-aspectos-praticos-da-producao-de-um-sistema-de-software/>. Acesso em: 15 nov. 2021.

CARDILLI, Danilo. **Conceitos de Software e Engenharia de Software**. 2010.

Disponível em:

<https://www.devmedia.com.br/conceitos-de-software-e-engenharia-de-software/15730>. Acesso em: 15 nov. 2021.

NOLETO, Cairo. **UML: o que é, para que serve e quando usar essa linguagem de notação?**. o que é, para que serve e quando usar essa linguagem de notação?.

2021. Disponível em: <https://blog.betrybe.com/tecnologia/uml/>. Acesso em: 15 nov. 2021.

VENTURA, Plínio. **O que é UML**. 2019. Disponível em:

<https://www.ateomomento.com.br/diagramas-uml/>. Acesso em: 15 nov. 2021.

MARQUES, Rafael. **O que é HTML? Entenda de forma descomplicada**. 2019.

Disponível em: homehost.com.br/blog/tutoriais/o-que-e-html/. Acesso em: 15 nov.

2021.

LONGEN, Andrei. **O Que é HTML? Guia Básico Para Iniciantes**. 2021. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos>. Acesso em: 15 nov. 2021.

BERNARDO, Fernanda. **HTML**: tudo que você precisa saber sobre o assunto!. tudo que você precisa saber sobre o assunto!. 2020. Disponível em: <https://blog.betrybe.com/html/>. Acesso em: 15 nov. 2021.

GONÇALVES, Ariane. **O que é CSS? Guia Básico para Iniciantes**. 2021. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>. Acesso em: 15 nov. 2021.

OKUBO, Beatriz. **Você Sabe o que é CSS? Entenda Como Funciona e Para que Serve**. 2021. Disponível em: <https://br.godaddy.com/blog/voce-sabe-o-que-e-css-entenda-como-funciona-e-para-que-serve/>. Acesso em: 15 nov. 2021.

PEREIRA, Ana Paula. **O que é CSS?** 2009. Disponível em: <https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm>. Acesso em: 15 nov. 2021.

FERREIRA, Kellison. **O que é PHP e por que você precisa conhecer essa linguagem de programação web**. 2019. Disponível em: <https://rockcontent.com/br/blog/o-que-e-php/>. Acesso em: 15 nov. 2021.

NOLETO, Cairo. **O que é PHP**: o guia básico desta linguagem de programação!. o guia básico desta linguagem de programação!. 2020. Disponível em: <https://blog.betrybe.com/desenvolvimento-web/php/>. Acesso em: 15 nov. 2021.

SILVA, Giancarlo. **O que é e como funciona a linguagem JavaScript?** 2015. Disponível em: <https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>. Acesso em: 15 nov. 2021.

ROVEDA, Ugo. **JavaScript**: o que é, para que serve e como funciona o js?. o que é, para que serve e como funciona o JS?. 2020. Disponível em: <https://kenzie.com.br/blog/javascript/>. Acesso em: 15 nov. 2021.

DORNELLES, Nemora. **JavaScript para iniciantes**: o que é, origens, para que serve e "hello world". o que é, origens, para que serve e "Hello World". 2016. Disponível em: <https://becode.com.br/javascript-para-iniciantes-origens-o-que-e-para-que-serve/>. Acesso em: 15 nov. 2021.

PISA, Pedro. **O que é e como usar o MySQL?** 2012. Disponível em: <https://www.techtudo.com.br/noticias/2012/04/o-que-e-e-como-usar-o-mysql.ghtml>.

Acesso em: 15 nov. 2021.

SILVEIRA, Paulo. **O que é SQL?** 2019. Disponível em:
<https://www.alura.com.br/artigos/o-que-e-sql>. Acesso em: 15 nov. 2021.

BACCA, Carolina Cozer. **O que é SQL e para que ele serve?** 2019. Disponível em:
<https://www.tecmundo.com.br/software/146482-sql-que-ele-serve.htm>. Acesso em:
15 nov. 2021.

ALECRIM, Emerson. **Bancos de dados são mais importantes nas nossas vidas do que a gente imagina.** 2018. Disponível em:
<https://tecnoblog.net/245120/banco-de-dados-importancia/>. Acesso em: 15 nov. 2021.

SOMMERVILLE, Ian. **Engenharia de software.** 10. ed. São Paulo: Pearson Prentice Hall, 2019. xi, 756 p. ISBN 9788579361081 (broch.).

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional.** 7. ed. Porto Alegre: Bookman, 2011. xxviii, 780 p. ISBN 9788563308337.

FELIPE NERY RODRIGUES MACHADO. **Banco de Dados.** 3. São Paulo 2014 0
ISBN 9788536509846.