



UNIVERSIDADE FEDERAL DO OESTE DO PARÁ
INSTITUTO DE ENGENHARIA E GEOCIÊNCIAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Fernando Almeida do Carmo

Proposta de projeto - Sistema para gerenciamento de dados
cadastrais de usuários e pacientes da unidade de pronto
atendimento (UPA) Santarém-PA

Sistemas Distribuídos

18/12/2024

I- Documento de Visão de Projeto

1. Objetivo

O propósito deste documento é coletar, analisar e definir as necessidades de alto-nível e características do projeto de software, focando nas potencialidades requeridas pelos afetados e usuários-alvo, e como estes requisitos serão abordados no projeto de software.

A visão do projeto documenta o ambiente geral de processos a ser desenvolvido para o sistema durante o projeto, fornecendo a todos os envolvidos uma descrição compreensível deste e de suas macro-funcionalidades.

O Documento de Visão de Projeto documenta apenas as necessidades e funcionalidades do sistema que estão sendo atendidas no projeto de software.

2. Descrição do Projeto

2.1. Justificativa

A informatização está cada vez mais presente em nossa sociedade, exercendo grandes influências dentro das empresas, das universidades e até mesmo dentro de nossas casas, ela está presente em nosso cotidiano. Um exemplo de informatização o qual pode ser citado é o E-SUS Hospitalar do Hospital da Lagoa, Roberli Pinto, que é um sistema de informatização da gestão hospitalar que visa à automatização dos processos de cuidado com o paciente e também dos aspectos administrativos, como controle de almoxarifado, carga horária e frequência de profissionais e redução dos documentos em papel, a qual transformou de maneira significativa a economia, acelerando os processos de atendimentos, fazendo-se com que haja um grande controle de dados. Nesse contexto, espera-se que com a implementação de um sistema informatizado nas unidades de pronto atendimento e nos sistemas de cadastro, se possa acelerar o processo de preenchimento de formulários com eficiência, assim tendo resultados próximos ou iguais no qual foi aplicado nos sistemas da E-SUS Hospitalar do Hospital da Lagoa, Roberli Pinto. Na Unidade de Pronto Atendimento (UPA), localizada na cidade de Santarém no estado do Pará, caso não exista um sistema de informação para cadastramento de pacientes, as entradas devem ser registradas todas manualmente, preenchidas as fichas de primeiro atendimento e concomitantemente serão lançados em um livro os nomes para futuras consultas.

Nessa perspectiva, esse modelo torna o serviço do recepcionista mais trabalhoso e cansativo, e ainda, quando um paciente precisa da cópia da sua ficha do primeiro atendimento, os funcionários do arquivo médico sentem muitas dificuldades em achar a ficha do paciente. Diante deste cenário, surge a necessidade do desenvolvimento de um sistema de informação para cadastramento de usuários e pacientes. Dessa forma, com a construção do sistema de atendimento poderá ajudar a eliminar os grandes problemas na demora do atendimento e garantindo a segurança e disponibilidade dos dados do paciente.

2.2. Objetivo

Desenvolver de um sistema web para o cadastramento dos pacientes dentro de uma Unidade de Pronto Atendimento (UPA) no município de Santarém-Pará, a fim de realizar um gerenciamento informatizado das informações dos usuários e suas consultas médicas e garantindo a segurança da informação.

3. Envolvimento

3.1. Abrangência

O projeto estará disponível por meio de um sistema web, onde um super usuário fará o cadastro de outros usuários que serão responsáveis pelo cadastro de médicos e pacientes e fazer as ações que envolvem estes atores como Realizar Consulta entre Médico e Paciente. Estes cadastros serão processados por um API (Interface de Programação de Aplicações).

Público alvo: Setor público, Unidade de Pronto Atendimento.

3.2. Papel das Partes Interessadas

3.2.1 - Cliente

Descrição	<input type="checkbox"/> Responsável pelos requisitos funcionais e não funcionais do sistema. <input type="checkbox"/> Responsável pelos testes do sistema para homologação do projeto.
Papel no desenvolvimento	<input type="checkbox"/> Atuar como facilitador e especificação dos requisitos sistêmicos perante a equipe de desenvolvimento. <input type="checkbox"/> Garantir que as regras de negócio sejam suportadas pela base legal. <input type="checkbox"/> Validar e aprovar os requisitos.

	<input type="checkbox"/> Fornecer os parâmetros operacionais do sistema. <input type="checkbox"/> Acompanhar o desenvolvimento do sistema. <input type="checkbox"/> Decidir sobre a realização de reuniões com os colaboradores. <input type="checkbox"/> Participar da homologação das decisões relacionadas aos sistemas. <input type="checkbox"/> Participar da Homologação do produto final.
Insumos ao projeto de software	<input type="checkbox"/> Requisitos Funcionais. <input type="checkbox"/> Requisitos Não-Funcionais. <input type="checkbox"/> Casos de testes para homologação do sistema. <input type="checkbox"/> Consultas diversas para validação do sistema. <input type="checkbox"/> Homologação das aplicações.
Representante	Não definido ainda

3.2.2 - Equipe de Desenvolvimento

Descrição	<input type="checkbox"/> Responsável pela identificação dos requisitos do software e pelo desenvolvimento dos modelos estáticos e dinâmicos do projeto
Papel no desenvolvimento	<input type="checkbox"/> Identificar e descrever as necessidades do usuário, especificando as funcionalidades do software que irão atendê-las. <input type="checkbox"/> Levantar os requisitos funcionais e não funcionais do projeto. <input type="checkbox"/> Definir o que irá interagir com o sistema. <input type="checkbox"/> Identificar dentro da visão lógica do sistema, a melhor forma de acomodar as necessidades do usuário, e o impacto da solução adotada sobre os requisitos do sistema. <input type="checkbox"/> Em suma, responsável pela geração de um produto que atenda aos requisitos que foram identificados junto ao usuário.
Insumos ao projeto de software	<input type="checkbox"/> Documento de visão e demais documentos de requisitos de software. <input type="checkbox"/> Aplicação desenvolvida de acordo com o especificado (artefatos do projeto).
Representante	Fernando Almeida do Carmo

3.3. Papel dos Atores

Descrição	Administrador
Papel	Monitorar dúvidas, reclamações e possíveis falhas do sistema e alertar a equipe

	responsável. Monitorar também, os acessos dos usuários para verificar aqueles que precisam ser retirados do sistema, em caso de demissão e os que precisam ser cadastrados no sistema (novos servidores da UPA).
Insumos ao sistema	Gerenciar o número de ocorrências e selecionar por ordem de urgência.
Representante	A definir.

Descrição	Usuário/Cliente
Papel	Entrar em contato com o sistema para fins de monitoramento.
Insumos ao sistema	Contribuir com as possíveis falhas encontradas.
Representante	A definir.

4. Necessidades e Funcionalidades

4.1. Necessidade 1: Realizar cadastro e remoção de usuários

Id Func.	Descrição das Funcionalidades/atores envolvidos	Benefício
F1.1	Consultar se os dados do usuário ainda não foram cadastrados anteriormente, utilizando a API para realizar a validação	<Crítico>
	Super Administrador	
F1.2	Excluir os dados dos usuários do Banco de Dados, acionando a API para executar a operação	<Crítico>
	Super Administrador	

4.2. Necessidade 2: Realizar cadastro de Médico

Id Func.	Descrição das Funcionalidades/atores envolvidos	Benefício
F2.1	Consultar pelo CRM do médico se ainda não foi cadastrado no sistema utilizando a API tal consulta	<Crítico>
	Administrador	
F2.2	Inserir os dados do médico no Banco de Dados, utilizando a API para realizar o cadastro	<Crítico>
	Administrador	

4.3. Necessidade 3: Realizar cadastro de Paciente

Id Func.	Descrição das Funcionalidades/atores envolvidos	Benefício
F3.1	Consultar pelo CNS do paciente, utilizando a API para verificar se o registro já existe no sistema	<Crítico>
	Administrador	
F3.2	Inserir os dados do paciente no Banco de Dados, utilizando a API para realizar o cadastro	<Crítico>
	Administrador	

4.4. Necessidade 4: Agendamento e Cadastro de Consultas

Id Func.	Descrição das Funcionalidades/atores envolvidos	Benefício
F4.1	Validar se o médico e o paciente já estão cadastrados no sistema, utilizando a API para realizar a validação	<Crítico>
	Administrador	
F4.2	Cadastrar a consulta com informações de data, horário e local no Banco de Dados, utilizando a API	<Crítico>
	Administrador	
F4.3	Enviar uma notificação por e-mail ao médico com os detalhes da consulta, acionando a API para realizar o envio	<Crítico>
	Administrador	

5. Restrições Tecnológicas

- Deve ser acessível via internet.
- Deve ser restrito para um dispositivo por vez.
- Deve ser restrito um único acesso na conta por vez.

6. Critérios de aceitação do Sistema

O sistema somente será implantado com o treinamento dos interessados e aprovação dos usuários.

7. Escopo Não Incluído no Sistema

O sistema não permite o agendamento de consulta para um médico e paciente se o médico já possuir um atendimento marcado para o mesmo horário.

8. Premissas e Restrições

Premissas	Restrições
------------------	-------------------

<ul style="list-style-type: none">• Não identificadas	<ul style="list-style-type: none">• O sistema não deve imprimir prontuários médicos sobre os pacientes.
---	---

II- Modelo de Dados

1. Modelo Conceitual do Banco de Dados

Para este estudo, delimitou-se quatro Tabelas (Entidades) para compor o sistema proposto. Desta forma, o primeiro passo foi a construção do Modelo Conceitual do Banco de Dados, também conhecido como Modelo Entidade Relacionamento (ER). A figura 1 mostra o modelo criado com a ferramenta BrModelo¹.

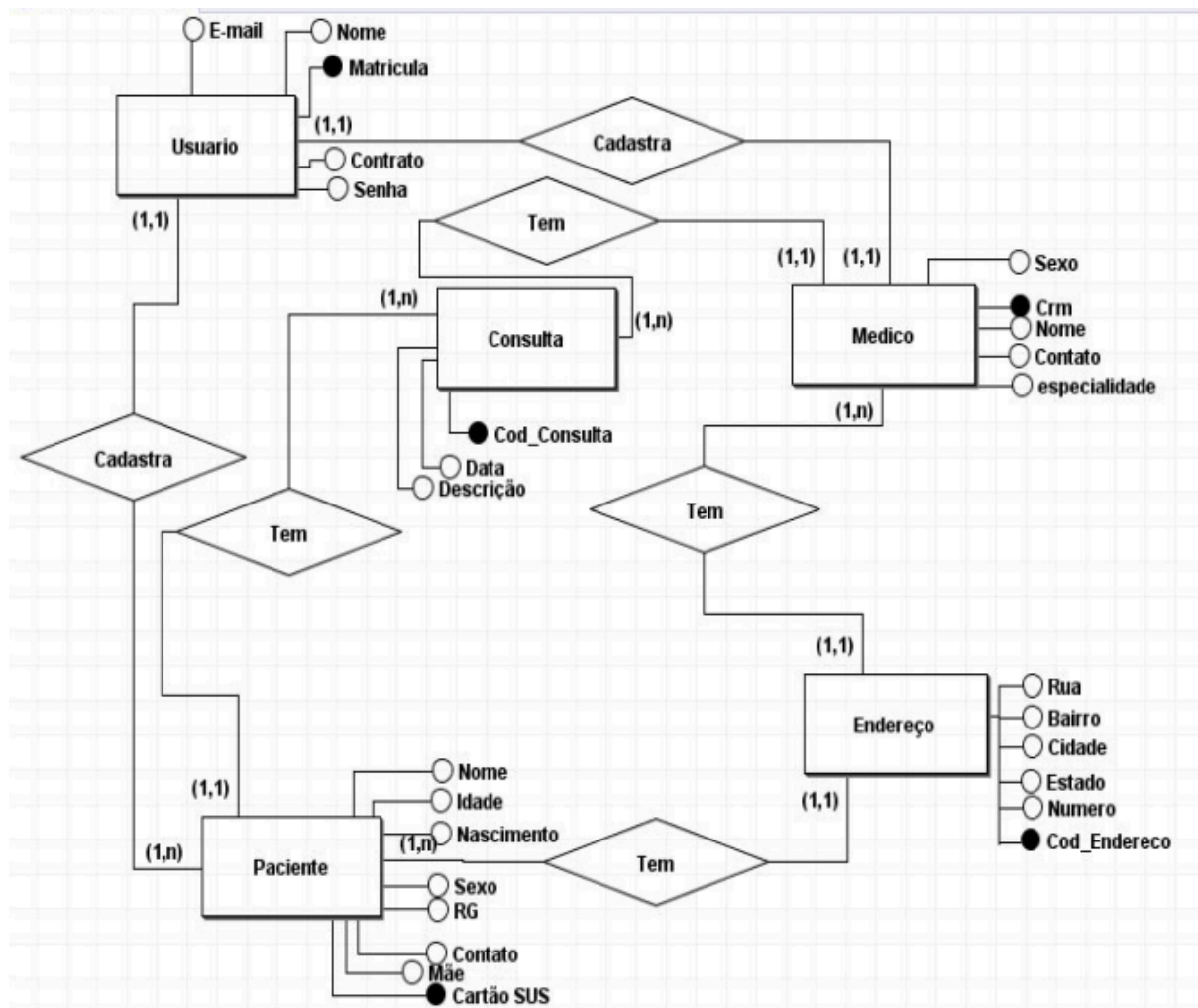


Figura 1: Modelo Conceitual do Sistema.

¹ <http://www.sis4.com/brModelo/>

III- Construção de Front-End do Sistema

Login e Cadastro de usuário

Assim como definido nas etapas iniciais do projeto e projetado na ferramenta Figma², a primeira tela do sistema é de Login, a Figura 1 mostra essa tela construída utilizando o *Framework Bootstrap*³ em consonância a linguagem de programação PHP, as quais fazem parte de um conjunto de ferramentas mais utilizadas no que tange a construção de sistemas web (SUPRIATMAJA *et. al*, 2022).

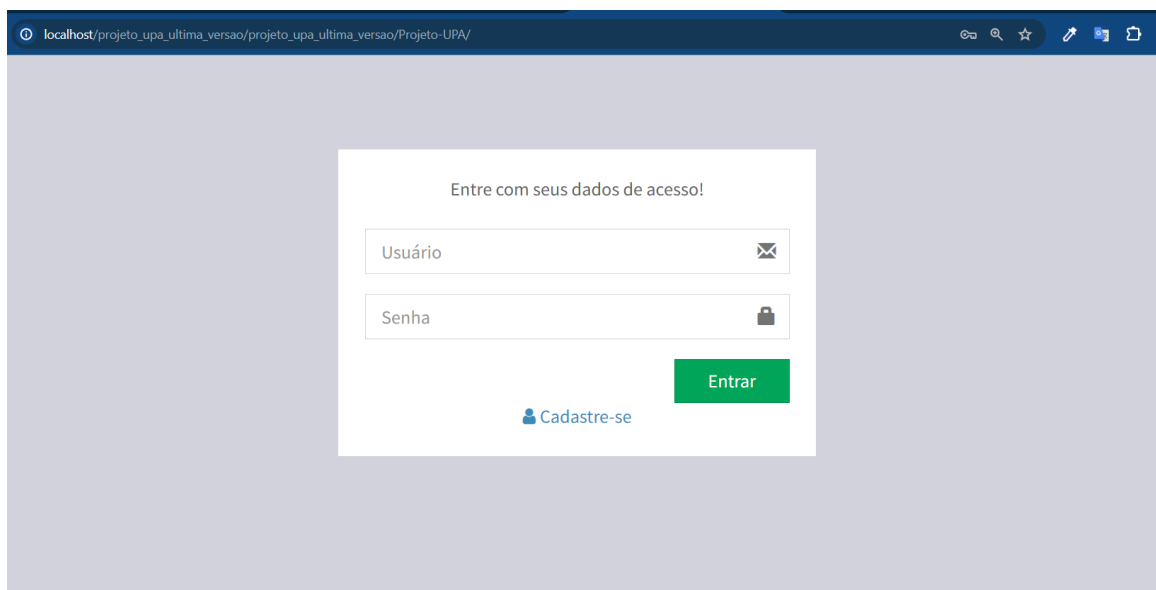


Figura 1: Tela de Login do sistema

Na página inicial construída há dois campos necessários para o usuário logar no sistema, o campo usuário é referente ao número de matrícula do usuário sendo este um importante parâmetro para a confidencialidade e autenticidade deste ator, tal número de matrícula deve ser informado pelo cadastro de cada usuário na unidade. Além disso, o usuário deve informar sua senha cadastrada. Caso este usuário não possua cadastro no sistema ou seja seu primeiro acesso ele deve seguir para a página de cadastro de usuário e informar os dados necessários, bem como mostra a Figura 2.

² <https://www.figma.com/>

³ <https://getbootstrap.com/>

Informe seus dados para acesso!

Fernando Almeida do Carmo

fernandocarmo@gmail.com

93992405793

2019008428

Cadastrar

Fazer Login

Figura 2: Tela de Cadastro no sistema

Após fazer o cadastro no sistema, o usuário pode retornar para a página de Login para acessar o sistema. Vale destacar que as informações devem ser validadas pelo sistema back-end, caso haja alguma inconsistência nos dados informados, o sistema mostrará uma mensagem com o respectivo erro, assim como mostra a Figura 3.

Entre com seus dados de acesso!

Usuário ou Senha não correspondem!

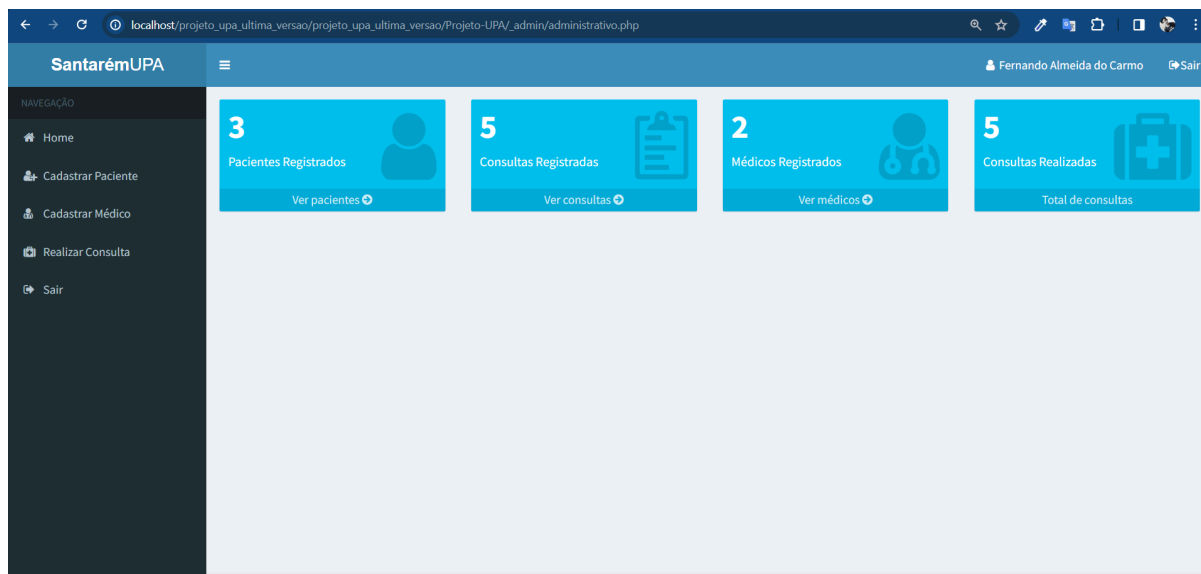
eutinho@gmail.com

Entrar

Cadastre-se

Figura 2: Mensagem de erro caso ocorra inconsistências

Se tais etapas apresentadas ocorrem com sucesso, o usuário poderá então visualizar a página inicial da Dashboard do sistema, bem como mostra a Figura 3.



Apesar de incompleto, a Figura 3 já mostra um avanço substancial do projeto, apresentando os principais componentes do sistema, como a barra lateral, onde pode-se identificar o menu para as páginas de acesso ao cadastro de Pacientes, Médicos e Consultas. Além disso, o usuário poderá visualizar os respectivos quantitativos de Pacientes, Consultas e Médicos cadastrados. Como objetivos futuros, pretende-se inserir gráficos para mostrar a evolução temporal de tais análises.

Em cada menu mostrado na imagem direciona o usuário para um formulário específico de cada ator mencionado anteriormente, cada formulário de cadastro estão em desenvolvimento, bem a estrutura as respectivas validações a qual fazem parte da próxima entrega da disciplina que será a integração de tais páginas com o back-end, para tanto utilizará as ferramentas MySQL, PHP e Bootstrap para a construção destes mecanismos do sistema, bem como a conexão com banco de dados.

Estes passos são importantes para utilização dos principais conceitos da disciplina, permitindo o ganho de experiência no que tange a utilização de ferramentas tecnológicas para a construção de sistemas web, bem como arquitetura de um sistema computacional do tipo *dashboard* e identificação de problemas do mundo real, além da capacidade analítica.

IV- Construção de Back-End do Sistema

Login e Cadastro de usuário

Após a criação da Tela de Login utilizando o *Framework Bootstrap*⁴, que utiliza estilização baseado em HTML, CSS e JavaScript, foi utilizada a linguagem PHP para a implementação da Conexão desta tela com a base de dados de usuários do sistema, tal base foi construída utilizando o Banco de Dados MySQL. A Figura 1 mostra a tela (Front-end) e a Figura 2 mostra o trecho de código que faz tal conexão entre esta tela e o banco de dados a fim de verificar se o usuário solicitante está cadastrado no sistema.

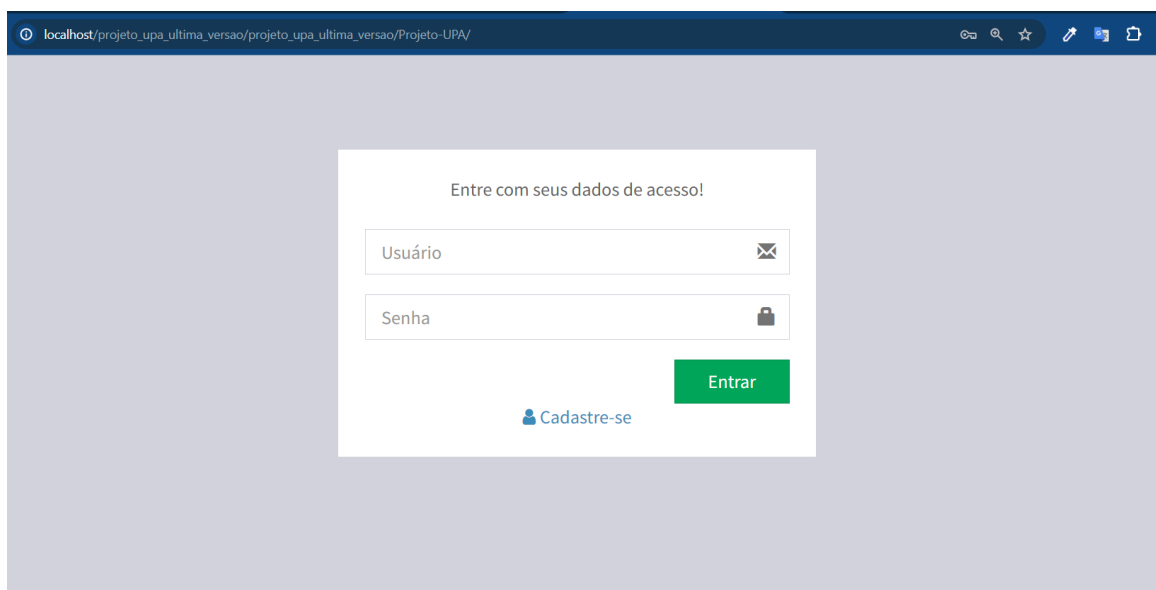


Figura 1: Tela de Login do sistema

⁴ <https://getbootstrap.com/>

```

<?php
    session_start();
    include_once "conexaoUPA.php";

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

    if($login == "" || $senha == ""){
        $_SESSION['msg'] = "Campos em branco!";
        header("Location: index.php");
    }
    else{
        //checa se existe parametro passado no campo matricula
        if (isset($login)) {
            $mat = ''; //retornar matricula do BD
            $pass = ''; //retornar senha do BD
            $sql = 'SELECT matricula FROM usuario WHERE matricula='.$login.'';
            $resul_user_bd = $dbcon->query($sql);

            //tratamento do resultado da busca
            if ($resul_user_bd->num_rows > 0) {
                while($row = $resul_user_bd->fetch_assoc()) {
                    $mat.= $row["matricula"];
                }
            }else{
                $_SESSION['msg'] = "Usuário não encontrado!";
                header("Location: index.php");
            }
        }
    }

```

Figura 2: Conexão com Banco de Dados e Validação de Usuários

Como mostra a Figura 2, os dados do banco de dados estão inseridos no arquivo conexaoUPA.php. Após incluir tal arquivo no projeto são passados os dados digitados no formulário da tela de login pelo usuário. Logo após essa etapa são feitas series de validações nos dados, bem como se os campos estão em brancos, se a matricula e/ou senha estão inclusos no banco de dados.

Assim como na Tela de Login, a tela de cadastro de usuários possui um formulário que envia os dados por meio de um código em PHP para validações e após essa etapa os salva no banco de dados. A Figura 3 mostra esse formulário de cadastro bem como a Figura 4 mostra o trecho de código para validação e cadastro no banco de dados.

Figura 3: Tela de Cadastro no sistema

```

if($nome == "" || $email == "" || $senha == "" || $matricula == "" || $contato == "" || $conf == ""){
    $_SESSION['msg'] = "Necessário preencher todos os campos!";
    header("Location: cadastro.php");
}elseif($senha != $conf){
    $_SESSION['msg'] = "As senhas não correspondem!";
    header("Location: cadastro.php");
}elseif((strlen($senha)) < 6){
    $_SESSION['msg'] = "A senha deve ter no mínimo 6 caracteres!";
    header("Location: cadastro.php");
}elseif((strlen($senha)) > 12){
    $_SESSION['msg'] = "A senha deve ter no máximo 12 caracteres!";
    header("Location: cadastro.php");
}
else{
    $senha = base64_encode($senha);
    $sql=$dbcon->query("INSERT INTO usuario(matricula, nome, email, contato, senha) VALUES('$matricula', '$nome', '$email', '$contato', '$senha')");

    if($sql){
        $_SESSION['msgcad'] = "Cadastro efetuado com sucesso!";
        header("Location: index.php");
    }else{
        $_SESSION['msg'] = "Erro ao cadastrar usuário! Verifique seus dados e tente novamente! ";
        header("Location: cadastro.php");
    }
}
}

```

Figura 4: Cadastro de Usuário no Banco de Dados

O trecho de código da Figura 4 mostra todos os campos digitados pelo usuário no formulário de cadastro, bem como valida um por um para então salvar no banco de dados. Vale destacar na etapa de cadastro da senha do usuário, que tal senha, digitada em texto bruto pelo usuário, está sendo **encriptada** utilizando o método de codificação de dados base64 e salva desta forma no banco de dados, bem como mostra a Figura 5. Após o cadastro, o usuário pode realizar o login no sistema e caso tenha alguma inconsistência em qualquer etapa anterior, o sistema informará ao usuário mensagens de erros, bem como mostra a Figura 6

			matricula	nome	email	contato	senha
<input type="checkbox"/>	 Editar	 Copiar	 Remover	2019008428	Maurício de Sousa	tinhofernando44@gmail.com	93992405793 ZmFjZmFjMDA=
<input type="checkbox"/>	 Editar	 Copiar	 Remover	2019008422	Fernando Almeida	inadiaalmeida10@gmail.com	93992405793 dGluaG8wMA==
<input type="checkbox"/>	 Editar	 Copiar	 Remover	2019008421	Maurício dos Santos	mna@gmail.com	93992405793 MTIzNDU2
<input type="checkbox"/>	 Editar	 Copiar	 Remover	12345678	Fernando Almeida do Carmo	teste@gmail.com	93992405793 MTIzNDU2Nzg=
<input type="checkbox"/>	 Editar	 Copiar	 Remover	200100	Fernando Almeida	tinhoalmeida10@gmail.com	93992405793 MjAwMXRpbmhw
<input type="checkbox"/>	 Editar	 Copiar	 Remover	2001001	Admim	admin@gmail.com	93992405793 YWRtaW4xMjM=
<input type="checkbox"/>	 Editar	 Copiar	 Remover	2011008422	Fernando Almeida do Carmo	fernandocarmo@gmail.com	93992405793 MzkwLnBpY28=
<input type="checkbox"/>	 Editar	 Copiar	 Remover	20100000	Carla Mariana	carla@gmail.com	93992405793 Y2FybGEzMjM=

Figura 5: Exemplo de senhas encriptadas na banco de Dados

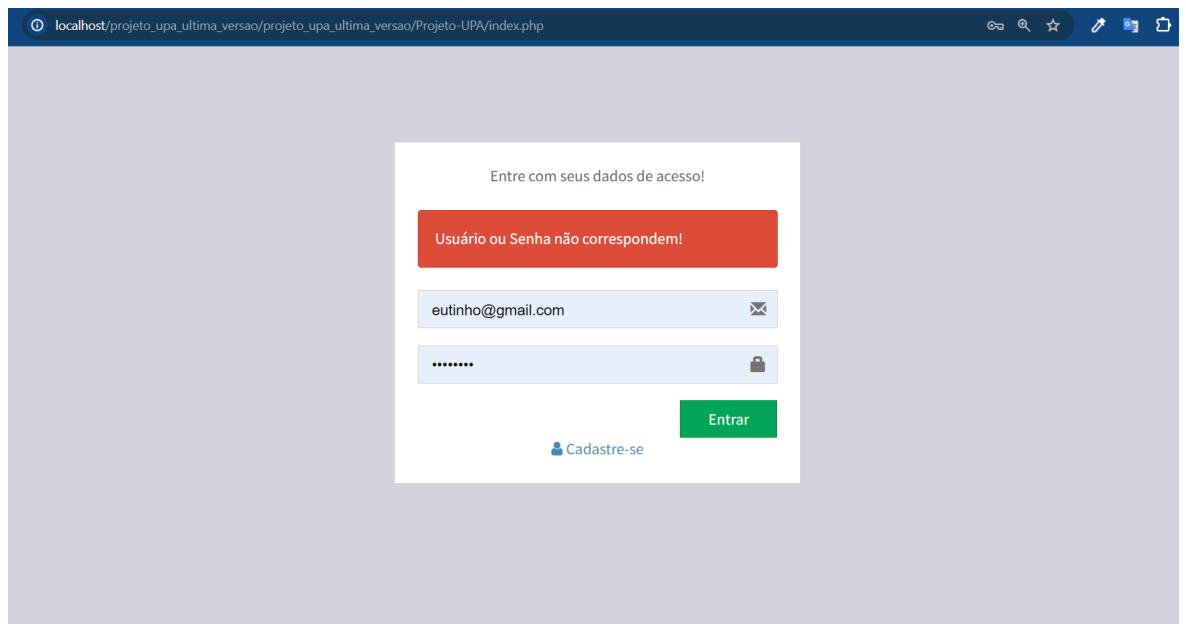


Figura 6: Mensagem de erro caso ocorra inconsistências

Estes são os principais passos de Back-end do sistema desenvolvido até o momento, o mecanismo de Dashboard, bem como cadastro de médicos e pacientes ainda estão sendo implementados e serão apresentados etapas futuras.

Estes passos são importantes para utilização dos principais conceitos da disciplina, permitindo o ganho de experiência no que tange a utilização de ferramentas tecnológicas para a construção de sistemas web, bem como arquitetura de um sistema computacional do tipo *dashboard* e identificação de problemas do mundo real, além da capacidade analítica.

Proposta de Implementação da API no Sistema de Gerenciamento de Consultas

A fim de aprimorar a arquitetura e escalabilidade do sistema de gerenciamento de dados de usuários e pacientes da UPA, proponho a implementação de uma API RESTful. Essa funcionalidade permitirá que o sistema seja acessado de forma mais flexível e integrada com outros serviços, facilitando a comunicação entre o frontend e o backend. A API será responsável por gerenciar as operações essenciais, como cadastro de pacientes, médicos e agendamento de consultas.

Com a implementação dessa API, será possível garantir maior modularidade ao sistema, além de possibilitar o acesso remoto e a comunicação entre diferentes componentes e plataformas, como aplicativos móveis e sistemas externos. A comunicação será realizada por meio de requisições HTTP, utilizando métodos como GET, POST, PUT e DELETE, atendendo às necessidades de escalabilidade e integração com outras soluções tecnológicas.

Essa mudança também contribui para o alinhamento do sistema com boas práticas de desenvolvimento de software, proporcionando uma base sólida para futuras expansões.

Referências

SUPRIATMAJA, Gede Agus et al. Sistem Informasi Perpustakaan Menggunakan Framework Bootstrap Dengan PHP Native dan Database MySQL Berbasis Web Pada SMP Negeri 2 Dawan. **Jurnal Teknologi Ilmu Komputer**, v. 1, n. 1, p. 7-15, 2022.