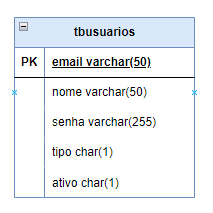
**SISTEMA DE LOGIN v2**

Nesse exemplo vamos criar um sistema de login em PHP com cadastro de usuário e área distintas para usuário administradores e usuários comuns.

Toda vez que um usuário se cadastrar estará inativo até que o administrador o ative. O administrador também terá a opção de “promover” o usuário para administrador.

**DIAGRAMA DO BANCO DE DADOS**

Abaixo temos a tabela tbusuarios que será utilizada nesse exemplo.



**SCRIPT PARA CRIAÇÃO DO BD E TABELA**

*create database bdloginv2;*

*use bdloginv2;*

*create table tbusuarios(*

*email varchar(50) not null primary key,*

*nome varchar(50),*

*senha varchar(255),*

*tipo char(1),*

*ativo char(1)*

*);*

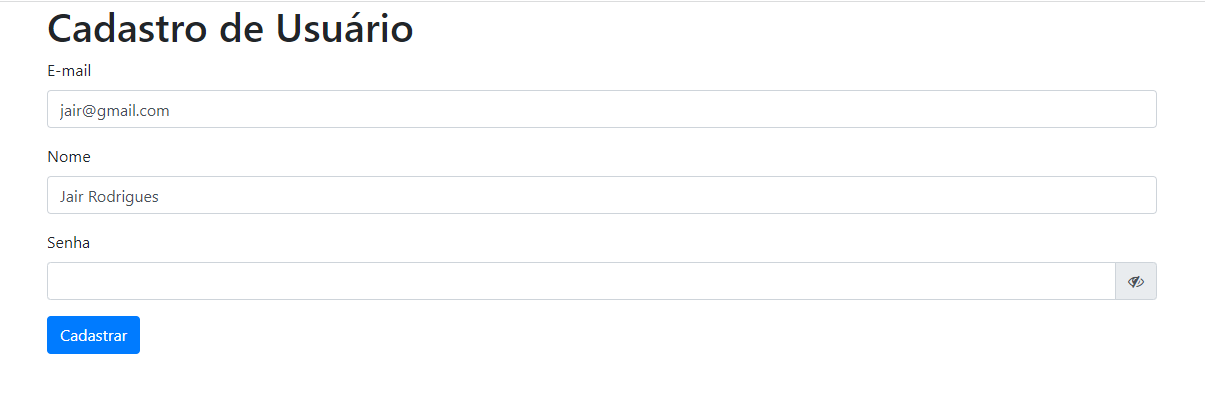
*insert into tbusuarios values ('admin@admin', 'Administrador', '$2y$10$STeq6OVcIHIk4xYLI5ZwrOlgcjx2NQCjfljKvw6lq3BDJzvKUUuN.', 'a', 's'),*

*('jair@gmail.com', 'Jair Rodrigues', '$2y$10$meI6hDfIJOJfUopdJKCUQ.62As3kxpl92CTn8OzvAxtFyhKZVIjze', 'u', 's');*

Deixei dois usuários pré cadastrados para podermos testar. Um administrador (admin@admin) cuja senha é admin e um usuário comum ([jair@gmail.com](mailto:jair@gmail.com)) cuja senha é 123456.

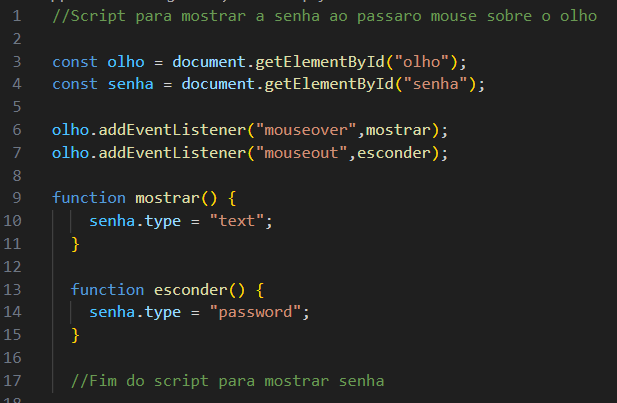
**TELA DE CADASTRO DE USUÁRIOS (telacadastrousuario.php)**

A tela de cadastro foi criada usando o site bootstrapformbuilder.com com algumas pequenas alterações estéticas.



**BOTÃO PARA EXIBIR A SENHA (script.js)**

Para esse sistema resolvi fazer algo diferente na questão da confirmação da senha, ao invés de dois campos com um js para verificar se são iguais criei um botão que ao passar o mouse em cima exibe a senha digitada. Esse arquivo script.js deverá ser salvo dentro da pasta js.



Abaixo outra forma de fazer onde temos que clicar no botão para que seja exibida a senha mudando também o ícone do olho.



**CADASTRO DE USUÁRIOS (cadastrousuario.php)**

Esse arquivo será responsável por pegar os dados dos usuários e armazenar na tabela de usuários.



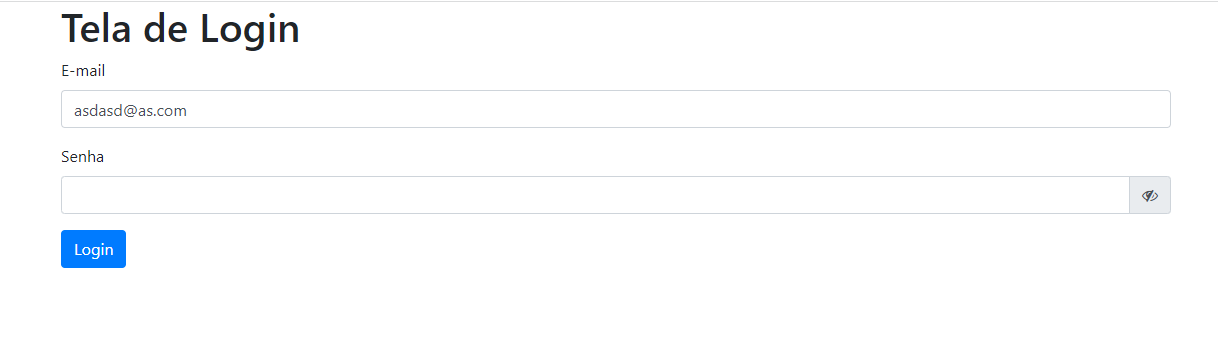
**Alguns pontos a observar:**

Temos um comando novo que é o **password\_hash** que irá criptografar a senha antes de armazenar na tabela.

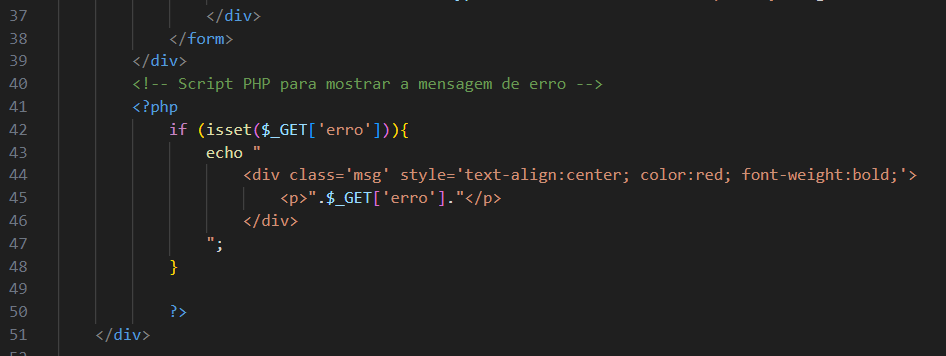
Outro detalhe é que no insert temos **‘u’** para o campo tipo e **‘n’** para ativo. Optei nesse caso pelas opções u para usuário normal e a para administrador. E no caso do ativo teremos **‘s’** para sim e **‘n’** para não.

**TELA DE LOGIN (telalogin.php)**

A tela de login segue a mesma estética da tela de cadastro sendo bastante simples. Apenas os campos de email e senha.



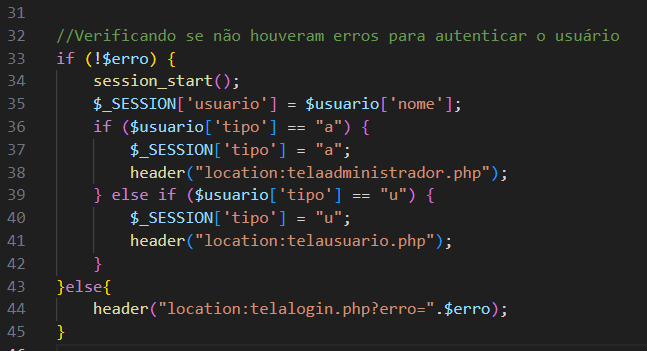
No final do arquivo temos um pequeno código PHP que tem como finalidade exibir a mensagem de erro no caso de ocorrer algum problema com o login:



**FAZENDO O LOGIN (login.php)**

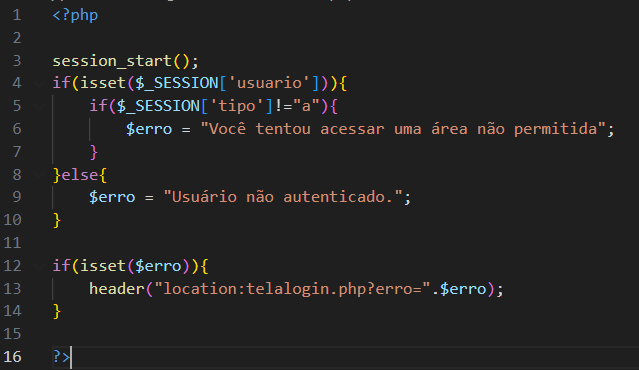
O login.php seraá o responsável por verificar e validar o usuário antes de dar acesso ao site. Nesse caso temos algumas diferenças em relação ao projeto anterior. Primeiro o PHP verifica se o nome de usuário foi encontrado, caso não retorna o erro de usuário não encontra. Após isso verifica se o usuário está ativo e por último se a senha está correta, sempre retornando o erro para a página telalogin.php. Outro ponto que vale observar com atenção é a verificação da senha, pois como usamos um hash (senha criptografada) temos que usar o comando password\_verify para comparar a senha digitada com a que está no BD.





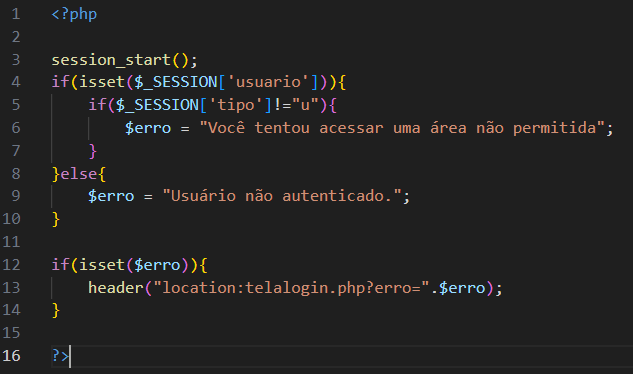
**VERIFICANDO SE O USUÁRIO É ADMINSTRADOR (validaradm.php)**

A ideia desse sistema é que o usuário comum tenha acesso a sua área específica e o administrador idem. Portanto criaremos dois arquivos de validação que irão verificar se o usuário se autenticou corretamente, caso não será enviado de volta para a tela de login com a mensagem de erro.



**VERIFICANDO SE É UM USUÁRIO COMUM (validarusuario.php)**

Nesse caso temos o mesmo processo acima só que para o usuário comum.



**FAZENDO O LOGOUT (logout.php)**

Após acessar a área restrita precisamos de uma opção para poder sair também, no caso fazer o logout. Vamos criar o arquivo logout.php que servirá para o administrador e para o usuário comum também.

