**FATEC ITAQUAQUECETUBA**

**Curso de Tecnologia em Gestão da Tecnologia da Informação**

**PROJETO INTERDISCIPLINAR**

**SISTEMA GERENCIADOR DE CONTATO (SGDC)**

**Lucas Medeiros Galindo**

**Responsável:**

**Profº Marcos Vasconcelos de Oliveira**

**2021**

**FATEC ITAQUAQUECETUBA**

**Curso de Tecnologia em Gestão da Tecnologia da Informação**

**PROJETO INTERDISCIPLINAR**

**Título do trabalho: SISTEMA GERENCIADOR DE CONTATO (SGDC)**

**Responsável:**

**Profº Marcos Vasconcelos de Oliveira**

**Integrante: Lucas Medeiros Galindo**

**RA:1550781721020**

**Emai**l: [medeiros.lucas1993@gmail.com](mailto:medeiros.lucas1993@gmail.com)

**2021**

**SUMÁRIO**

[Título do projeto 5](#_Toc74650100)

[Participantes 5](#_Toc74650101)

[Área ou linha de pesquisa 5](#_Toc74650102)

[Palavras-chave 5](#_Toc74650103)

[Duração do trabalho 5](#_Toc74650104)

[Contextualização 5](#_Toc74650105)

[Formulação do problema 6](#_Toc74650106)

[Objetivos 7](#_Toc74650107)

[Objetivo geral 7](#_Toc74650108)

[Objetivo específico 7](#_Toc74650109)

[Justificativa do trabalho 7](#_Toc74650110)

[Metodologia e plano de trabalho 7](#_Toc74650111)

[Especificações dos requisitos 7](#_Toc74650112)

[Requisitos funcionais 7](#_Toc74650113)

[Requisitos não funcionais 8](#_Toc74650114)

[Modelagem do sistema 8](#_Toc74650115)

[Diagramas 9](#_Toc74650116)

[Diagramas de classes 9](#_Toc74650117)

[Diagramas de caso de uso 10](#_Toc74650118)

[Diagrama de atividade 13](#_Toc74650119)

[Diagrama de sequência 14](#_Toc74650120)

[Protótipo de telas 15](#_Toc74650121)

[Código fonte 18](#_Toc74650122)

[HTML 18](#_Toc74650123)

[Alterar\_dados.aspx 18](#_Toc74650124)

[Cadastro.aspx 21](#_Toc74650125)

[Default.aspx 24](#_Toc74650126)

[Editar\_contato.aspx 25](#_Toc74650127)

[Menu.aspx 28](#_Toc74650128)

[Novo\_contato.aspx 30](#_Toc74650129)

[Pesquisar.aspx 32](#_Toc74650130)

[Estilos 35](#_Toc74650131)

[Base.css 35](#_Toc74650132)

[Cadastro.css 36](#_Toc74650133)

[Editar\_contato.css 37](#_Toc74650134)

[Editar\_dados.css 38](#_Toc74650135)

[Index.css 40](#_Toc74650136)

[Menu.css 41](#_Toc74650137)

[Novo\_contato.css 42](#_Toc74650138)

[Pesquisar.css 43](#_Toc74650139)

[Scripts 45](#_Toc74650140)

[Vanillatilt.js 45](#_Toc74650141)

[Controladores 67](#_Toc74650142)

[ContatoController.cs 67](#_Toc74650143)

[UsuarioController.cs 69](#_Toc74650144)

[Modelos 71](#_Toc74650145)

[Conexao.cs 71](#_Toc74650146)

[ContatoModel.cs 71](#_Toc74650147)

[UsuarioModel.cs 72](#_Toc74650148)

[Cronograma 72](#_Toc74650149)

[Bibliografia 73](#_Toc74650150)

# Título do projeto

Sistema gerenciador de contatos

# Participantes

Lucas Medeiros Galindo

# Área ou linha de pesquisa

Sistema gerenciador de contatos

# Palavras-chave

Gerenciamento, contatos, serviço, nuvem, web.

# Duração do trabalho

Início: MARÇO/2021 - Término: JUNHO/2021

# Contextualização

A história da lista telefônica do Brasil certamente faz parte da história da telefonia do país. Em janeiro de 1881, a Companhia Telefônica do Brasil foi fundada na Rua da Quitanda, no Rio de Janeiro, sendo a primeira empresa de telefonia do país. Na mesma ocasião, a empresa publicou em uma das páginas do Jornal do Commercio, publicado em 21 de agosto de 1881, a primeira lista telefônica do Brasil.

Para pesquisar números de telefone, as pessoas usam anúncios em jornais. No início de cada mês, as informações do novo proprietário de telefone eram publicadas no jornal, disponibilizando o número do telefone e o endereço. Foi dessa forma que a Companhia de Telégrafos Urbanos (Ferdinand Rodde & Co.) publicou na edição do dia 8 de janeiro de 1884, no jornal A Província de S. Paulo, hoje O Estado de S. Paulo, a primeira lista telefônica da cidade de São Paulo, com 22 assinantes.

Com o avanço do número de assinantes, as listas deixaram de ser publicadas nos jornais e passaram a ser impressas pelas próprias empresas telefônicas. As listas telefônicas permaneceram a ser editadas pelas próprias concessionárias de serviços telefônicos até 1946, principalmente pela Companhia Telefônica Brasileira, a maior empresa do Brasil na época.

Softwares e aplicativos de listas telefônicas são disponibilizados para instalação e utilização nas mais variadas e diversas plataformas como computadores, tablets ou celulares, necessitando-se ou não de Internet para realizar uma procura.

As versões on-line das tradicionais listas impressas são produzidas por editoras especializadas em todo o mundo. As funcionalidades das listas on-line são semelhantes às tradicionais, incluem listagens alfabéticas de empresas que podem ser pesquisas por nome da empresa, tipo de atividade, produto ou serviço. Diferentemente das listas impressas limitadas por espaço físico, as versões on-line permitem através de um único meio (Internet) obter informações sobre centenas de milhares de empresas ao redor do mundo, bem como obter informações adicionais sobre o perfil das empresas e seus produtos e serviços. Uma das vantagens das versões on-line é que elas são atualizadas diariamente, permitindo maior precisão das listagens já que informações podem ser alteradas a qualquer momento.

Assim como a guardar telefones de empresas, as pessoas também têm a necessidade de armazenar seus contatos pessoais, números de amigos, parentes e afins. A forma mais comum utilizada era a de utilizar um pequeno caderno e realizar as anotações dos dados manualmente com uma caneta.

Com o surgimento dos telefones celulares e telefones com memória interna, a possibilidade de registro desses dados se tornou o primeiro passo para um armazenamento totalmente digital, entretanto, esses dados ficam limitados a estes aparelhos, e caso fossem danificados, esses dados seriam perdidos.

Em meio a esta dificuldade, a utilização de serviços online de armazenamento tem se tornado cada vez mais populares devido a sua confiabilidade e portabilidade, estando sempre disponível para o usuário.

# Formulação do problema

Quantas vezes alguém precisou ter acesso a sua agenda de contatos independentemente de onde estava ou trocou de telefone e perdeu todos os seus contatos?

# Objetivos

## Objetivo geral

Desenvolver um sistema online que organize e gerencie os contatos pessoais.

## Objetivo específico

O usuário só poder ver contatos que ele cadastrou.

# Justificativa do trabalho

Para gerenciar melhor os contatos, eles não devem ser armazenados em dispositivos físicos, mas sim em nuvem. Portanto, um software de gerenciamento de contatos será criado para armazená-los, torná-los disponíveis e compatível com qualquer dispositivo a qualquer momento.

O sistema terá a capacidade de adicionar, editar e excluir contatos, separá-los por grupos e torná-los acessíveis aos usuários do sistema. Através deste site, será possível gerenciar os contatos adicionados de uma forma simplificada.

# Metodologia e plano de trabalho

Para o desenvolvimento do software de gerenciamento de contatos foi utilizado a linguagem de programação C#, conectado corretamente a um banco de dados SQL, construído através do SGBD MySQL, utilizando a metodologia orientadas a objetos, com base no diagrama de caso de uso, diagrama de classes, diagrama de atividades e diagrama de sequência.

## Especificações dos requisitos

### Requisitos funcionais

RF01- O site iniciará em uma tela de autenticação do usuário para acesso ao sistema;

RF02- Após acessar o sistema o usuário poderá alterar seus dados e de seus contatos;

RF03- A página inicial apresentará um menu com todas as opções necessárias para cada usuário;

RF04- O sistema possuirá funções de inclusão, exclusão e alteração de contatos;

RF05- O sistema deverá armazenar informações importantes dos contatos adicionados pelo usuário;

RF06- O usuário só terá acesso aos contatos que ele próprio cadastrou;

### Requisitos não funcionais

RNF01- O usuário deverá criar um cadastro para ingressar ao sistema;

RNF02- O sistema será constituído da linguagem de programação C# (ASP.Net Web Forms);

RNF03- O banco de dados desempenhará da linguagem SQL e terá como SGBD o MySql;

RNF04- O padrão de arquitetura de software irá consistir no MVC e o paradigma de programação em OO;

## Modelagem do sistema

Para este projeto será utilizado quatro modelos UML, diagrama de classes, diagrama de caso de uso, diagrama de atividades e diagrama de sequência.

Diagrama de classes: usado para representar graficamente todas as classes contidas no sistema.

Diagrama de caso de uso: Usado para representar graficamente O comportamento dos participantes externos vinculados ao sistema. Atores são aqueles que podem acessar o sistema.

Diagrama de atividades: Usado para demonstrar graficamente o algoritmo e parte do sistema que também pode representar ações das pessoas interagindo com o sistema.

Diagrama de sequência: é utilizado para demostrar de modo gráfico a interação dos atores com os métodos ou operações do sistema.

Modelo entidade e relacionamento: é utilizado para demostrar de modo gráfico o esquema relacional do banco de dados, mostrando os atributos das entidades e seus relacionamentos.

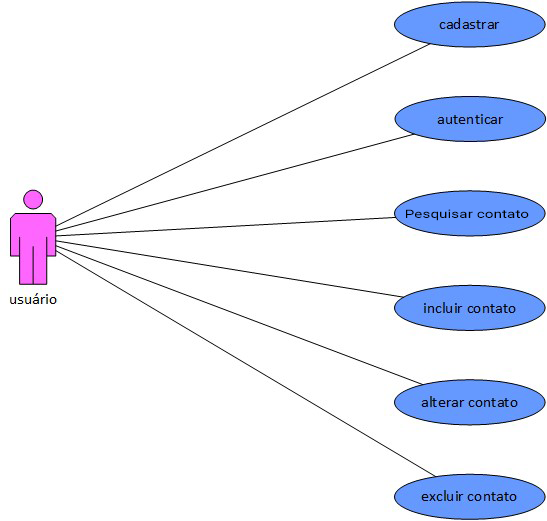
## Diagramas

### Diagramas de classes

Diagrama

Descrição gerada automaticamente

### Diagramas de caso de uso



#### UC1 – Cadastrar

|  |  |
| --- | --- |
| **Nome do caso de uso** | Cadastrar |
| **Ator** | usuário |
| **Pré-condições** | Sistema estar carregado |
| **Cenário principal** | * O usuário clica em cadastrar; * O site carrega o formulário de cadastro; * O usuário preenche o formulário de cadastro; * O usuário clica em “salvar”; * O sistema emite um alerta de sucesso; |
| **Cenário alternativo** | O usuário não realiza o cadastro |

#### UC2 – Autenticar

|  |  |
| --- | --- |
| **Nome do caso de uso** | Autenticar |
| **Ator** | usuário |
| **Pré-condições** | O usuário deve possuir cadastro ativo |
| **Cenário principal** | * O sistema carrega em ela inicial; * O usuário preenche o formulário de login com seus dados; * O usuário clica em login; * O sistema carrega em menu principal |
| **Cenário alternativo** | O usuário não efetua o login |

#### UC3 – Pesquisar contato

|  |  |
| --- | --- |
| **Nome do caso de uso** | Pesquisar contato |
| **Ator** | usuário |
| **Pré-condições** | O usuário deve estar autenticado em sistema |
| **Cenário principal** | * O sistema carrega em menu principal; * O usuário clica em “novo contato”; * O sistema carrega o formulário de novos contatos; * O usuário preenche o formulário; * O usuário clica em “salvar”; |
| **Cenário alternativo** | O usuário clica no botão “cancelar” antes de concluir a operação de cadastro |

#### UC4 – Incluir contato

|  |  |
| --- | --- |
| **Nome do caso de uso** | Incluir contato |
| **Ator** | usuário |
| **Pré-condições** | O usuário deve estar autenticado em sistema |
| **Cenário principal** | * O sistema carrega em menu principal; * O usuário clica em “novo contato”; * O sistema carrega o formulário de novos contatos; * O usuário preenche o formulário; * O usuário clica em “salvar”; |
| **Cenário alternativo** | O usuário clica no botão “cancelar” antes de concluir a operação de cadastro |

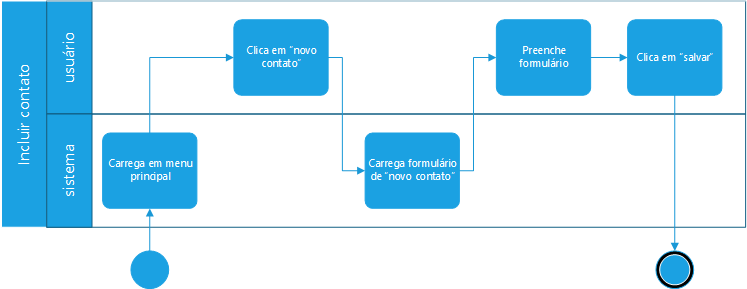
#### UC5 – Alterar contato

|  |  |
| --- | --- |
| **Nome do caso de uso** | Alterar contato |
| **Ator** | usuário |
| **Pré-condições** | O usuário deve estar autenticado em sistema |
| **Cenário principal** | * O usuário pesquisa o contato que deseja editar; * O formulário de edição carrega com os dados do usuário selecionado; * O usuário edita os campos desejados; * O usuário clica em salvar |
| **Cenário alternativo** | O usuário clica no botão “cancelar” antes de concluir a operação de alteração |

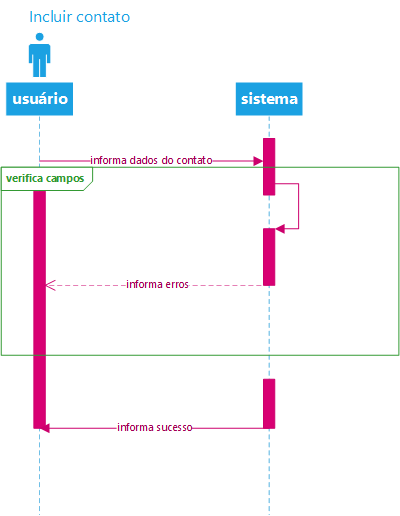
#### UC6 – Excluir contato

|  |  |
| --- | --- |
| **Nome do caso de uso** | Excluir contato |
| **Ator** | usuário |
| **Pré-condições** | O usuário deve estar autenticado em sistema |
| **Cenário principal** | * O usuário pesquisa o contato cadastrados; * O usuário clica em “excluir”; * O sistema emite uma mensagem de sucesso |
| **Cenário alternativo** | O usuário não clica em excluir |

### Diagrama de atividade



### Diagrama de sequência



# Protótipo de telas

Interface gráfica do usuário

Descrição gerada automaticamente

Figura : Tela de Login

Interface gráfica do usuário

Descrição gerada automaticamente

Figura : Tela de cadastro de usuário

Interface gráfica do usuário

Descrição gerada automaticamente

Figura : Menu principal

Interface gráfica do usuário

Descrição gerada automaticamente

Figura : Editar contato

Interface gráfica do usuário

Descrição gerada automaticamente

Figura : Editar dados do usuário

Interface gráfica do usuário

Descrição gerada automaticamente

Figura : cadastro de novos contatos

Interface gráfica do usuário

Descrição gerada automaticamente

Figura : Pesquisar contato

# Código fonte

## HTML

### Alterar\_dados.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="alterar\_dados.aspx.cs" Inherits="WebApplication1.Views.WebForm3" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<link href="../Styles/base.css" rel="stylesheet" />

<link href="../Styles/editar\_dados.css" rel="stylesheet" />

<title></title>

</head>

<body>

<form id="form1" runat="server" class="container">

<asp:Button ID="btnLogout" runat="server" Text="Sair" OnClick="btnLogout\_Click"/>

<div class="form">

<div class="form-control">

<label for="dadosNome">Nome:</label>

<asp:TextBox ID="dadosNome" runat="server"></asp:TextBox>

</div>

<div class="form-control">

<label for="dadosEmail">E-mail:</label>

<asp:TextBox ID="dadosEmail" runat="server"></asp:TextBox>

</div>

<div class="form-control">

<label for="dadosSenha">Senha:</label>

<asp:TextBox ID="dadosSenha" runat="server" TextMode="Password"></asp:TextBox>

</div>

<div class="form-control">

<label for="dadosConfSenha">Confirme a senha:</label>

<asp:TextBox ID="dadosConfSenha" runat="server" TextMode="Password"></asp:TextBox>

</div>

<div class="btn-control">

<asp:Button ID="dadosSalvar" runat="server" Text="Salvar" OnClick="dadosSalvar\_Click" />

<asp:Button ID="dadosCancelar" runat="server" Text="Cancelar" OnClick="dadosCancelar\_Click"/>

</div>

</div>

</form>

<script>

const dadosEmail = document.getElementById("dadosEmail");

const dadosSenha = document.getElementById("dadosSenha");

const dadosConfSenha = document.getElementById("dadosConfSenha");

dadosEmail.addEventListener("change", (event) => {

event.preventDefault();

let pattern = /^[a-z0-9.]+@[a-z0-9]+\.[a-z]+(\.[a-z]+)?$/gi;

let emailEstaOk = pattern.test(dadosEmail.value);

if (!emailEstaOk) {

alert("Formato de e-mail inválido! verifique os dados e insira novamente")

dadosEmail.value = ""

}

});

dadosConfSenha.addEventListener("keyup", (event) => {

event.preventDefault()

if (dadosConfSenha.value !== dadosSenha.value) {

dadosSenha.style.cssText = `box-shadow: 0px 0px 20px rgba(255, 0, 0, 1) !important;`;

dadosConfSenha.style.cssText = `box-shadow: 0px 0px 20px rgba(255, 0, 0, 1) !important;`;

} else {

dadosSenha.style.cssText = `box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5) !important;`;

dadosConfSenha.style.cssText = `box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5) !important;`;

}

});

</script>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using WebApplication1.Controllers;

using WebApplication1.Models;

namespace WebApplication1.Views

{

public partial class WebForm3 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["login"] == null)

{

Response.Redirect("default.aspx");

}

else

{

List<string> resposta = new List<string>();

UsuarioController controller = new UsuarioController();

resposta = controller.Pesquisar(Session["email"].ToString());

dadosNome.Text = resposta[1].ToString();

dadosEmail.Text = resposta[2].ToString();

}

}

protected void dadosCancelar\_Click(object sender, EventArgs e)

{

Response.Redirect("menu.aspx");

}

protected void btnLogout\_Click(object sender, EventArgs e)

{

Session["login"] = null;

Response.Redirect("default.aspx");

}

protected void dadosSalvar\_Click(object sender, EventArgs e)

{

if (dadosNome.Text == "")

{

Response.Write("<script>alert('Verifique o nome inserido')</script>");

}

else if (dadosEmail.Text == "")

{

Response.Write("<script>alert('Verifique o e-mail inserido')</script>");

}

else if (dadosSenha.Text.Length < 6)

{

Response.Write("<script>alert('Verifique a senha inserida! Ela deve conter no mínimo 6 caracteres</script>");

}

else if (dadosSenha.Text != dadosConfSenha.Text)

{

Response.Write("<script>alert('Verifique a senha inserida! os campos de senha devem ser iguais</script>");

}

else

{

try

{

UsuarioModel usuario = new UsuarioModel(dadosNome.Text, dadosEmail.Text, dadosSenha.Text);

usuario.Id = int.Parse(Session["login"].ToString());

UsuarioController controller = new UsuarioController();

bool resposta = controller.Alterar(usuario);

if (resposta)

{

Response.Write("<script>alert('Usuário alterado com sucesso!')</script>");

dadosNome.Text = "";

dadosEmail.Text = "";

dadosSenha.Text = "";

dadosConfSenha.Text = "";

}

else

{

Response.Write("<script>alert('Erro ao alterar o cadastro do usuário!')</script>");

dadosNome.Text = "";

dadosEmail.Text = "";

dadosSenha.Text = "";

dadosConfSenha.Text = "";

}

}

catch (Exception erro)

{

Console.WriteLine(erro);

//throw erro;

}

}

}

}

}

### Cadastro.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="cadastro.aspx.cs" Inherits="WebApplication1.Views.WebForm2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<link href="../Styles/base.css" rel="stylesheet" />

<link href="../Styles/cadastro.css" rel="stylesheet" />

<title></title>

</head>

<body>

<form id="form1" runat="server" class="container">

<div class="form">

<div class="form-control">

<label for="cadastroNome">Nome:</label>

<asp:TextBox ID="cadastroNome" runat="server"></asp:TextBox>

</div>

<div class="form-control">

<label for="cadastroEmail">E-mail:</label>

<asp:TextBox ID="cadastroEmail" runat="server"></asp:TextBox>

</div>

<div class="form-control">

<label for="cadastroSenha">Senha:</label>

<asp:TextBox ID="cadastroSenha" runat="server" TextMode="Password"></asp:TextBox>

</div>

<div class="form-control">

<label for="cadastroConfSenha">Confirme a senha:</label>

<asp:TextBox ID="cadastroConfSenha" runat="server" TextMode="Password"></asp:TextBox>

</div>

<div class="btn-control">

<asp:Button ID="cadastroSalvar" runat="server" Text="Salvar" OnClick="cadastroSalvar\_Click" />

<asp:Button ID="cadastroCancelar" runat="server" Text="Cancelar" OnClick="cadastroConfSenha\_Click"/>

</div>

</div>

</form>

<script>

const cadastroEmail = document.getElementById("cadastroEmail");

const cadastroSenha = document.getElementById("cadastroSenha");

const cadastroConfSenha = document.getElementById("cadastroConfSenha");

cadastroEmail.addEventListener("change", (event) => {

event.preventDefault();

let pattern = /^[a-z0-9.]+@[a-z0-9]+\.[a-z]+(\.[a-z]+)?$/gi;

let emailEstaOk = pattern.test(cadastroEmail.value);

if (!emailEstaOk) {

alert("Formato de e-mail inválido! verifique os dados e insira novamente")

cadastroEmail.value = ""

}

});

cadastroConfSenha.addEventListener("keyup", (event) => {

event.preventDefault()

if (cadastroConfSenha.value !== cadastroSenha.value) {

cadastroSenha.style.cssText = `box-shadow: 0px 0px 20px rgba(255, 0, 0, 1) !important;`;

cadastroConfSenha.style.cssText = `box-shadow: 0px 0px 20px rgba(255, 0, 0, 1) !important;`;

} else {

cadastroSenha.style.cssText = `box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5) !important;`;

cadastroConfSenha.style.cssText = `box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5) !important;`;

}

});

</script>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using WebApplication1.Controllers;

using WebApplication1.Models;

namespace WebApplication1.Views

{

public partial class WebForm2 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void cadastroConfSenha\_Click(object sender, EventArgs e)

{

Response.Redirect("default.aspx");

}

protected void cadastroSalvar\_Click(object sender, EventArgs e)

{

if (cadastroNome.Text == "")

{

Response.Write("<script>alert('Verifique o nome inserido')</script>");

}

else if (cadastroEmail.Text == "")

{

Response.Write("<script>alert('Verifique o e-mail inserido')</script>");

}

else if (cadastroSenha.Text.Length < 6)

{

Response.Write("<script>alert('Verifique a senha inserida! Ela deve conter no mínimo 6 caracteres</script>");

}

else if (cadastroSenha.Text != cadastroConfSenha.Text)

{

Response.Write("<script>alert('Verifique a senha inserida! os campos de senha devem ser iguais</script>");

}

else

{

try

{

UsuarioModel usuario = new UsuarioModel(cadastroNome.Text, cadastroEmail.Text, cadastroSenha.Text);

UsuarioController controller = new UsuarioController();

bool resposta = controller.Inserir(usuario);

if (resposta)

{

Response.Write("<script>alert('Usuário cadastrado com sucesso!')</script>");

cadastroNome.Text = "";

cadastroEmail.Text = "";

cadastroSenha.Text = "";

cadastroConfSenha.Text = "";

}

else

{

Response.Write("<script>alert('Erro ao cadastrar usuário!')</script>");

cadastroNome.Text = "";

cadastroEmail.Text = "";

cadastroSenha.Text = "";

cadastroConfSenha.Text = "";

}

}

catch (Exception erro)

{

Response.Write("<script>alert('Erro ao cadastrar usuário!')</script>");

Console.WriteLine(erro);

}

}

}

}

}

### Default.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="default.aspx.cs" Inherits="WebApplication1.Views.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<link href="../Styles/base.css" rel="stylesheet" />

<link href="../Styles/index.css" rel="stylesheet" />

<title></title>

</head>

<body>

<form id="form1" runat="server" class="container">

<div class="form">

<div class="form-control">

<label for="txbLogin">Login:</label>

<asp:TextBox ID="txbLogin" runat="server" required></asp:TextBox>

</div>

<div class="form-control">

<label for="txbPassword">Senha:</label>

<asp:TextBox ID="txbPassword" runat="server" TextMode="Password" required></asp:TextBox>

</div>

<div class="form-control">

<asp:Button ID="btnLogin" runat="server" Text="Entrar" OnClick="btnLogin\_Click" />

</div>

<p>Não possui login? <span><a href="cadastro.aspx">Cadastre-se</a></span></p>

</div>

</form>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using WebApplication1.Controllers;

namespace WebApplication1.Views

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnLogin\_Click(object sender, EventArgs e)

{

List<string> resposta = new List<string>();

Session["login"] = null;

try

{

UsuarioController usuario = new UsuarioController();

resposta = usuario.Pesquisar(txbLogin.Text);

if (resposta[0] == "Usuário não encontrado")

{

Response.Write("<script>alert('Usuário não localizado!')</script>");

}

else if (resposta[3] != txbPassword.Text)

{

Response.Write("<script>alert('Usuário ou senha incorreta!')</script>");

}

else

{

Session["login"] = resposta[0].ToString();

Session["usuario"] = resposta[1].ToString();

Session["email"] = resposta[2].ToString();

Response.Redirect("menu.aspx");

}

}

catch (Exception erro)

{

Console.WriteLine(erro);

//throw erro;

}

}

}

}

### Editar\_contato.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="editar\_contato.aspx.cs" Inherits="WebApplication1.Views.editarcontato" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<link href="../Styles/base.css" rel="stylesheet" />

<link href="../Styles/editar\_contato.css" rel="stylesheet" />

<title></title>

</head>

<body>

<form id="form1" runat="server" class="container">

<asp:Button ID="btnLogout" runat="server" Text="Sair" OnClick="btnLogout\_Click" />

<div class="form">

<div class="form-control btn-control">

<asp:TextBox ID="editarBuscar" runat="server"></asp:TextBox>

<asp:Button ID="editarBtnBuscar" runat="server" Text="Pesquisar" OnClick="editarBtnBuscar\_Click" />

</div>

<div class="form-control">

<asp:Label ID="contatoID" runat="server" Text="" Visible="False"></asp:Label>

<label for="editarNome">Nome:</label>

<asp:TextBox ID="editarNome" runat="server"></asp:TextBox>

</div>

<div class="form-control">

<label for="editarTelefone">Telefone:</label>

<asp:TextBox ID="editarTelefone" runat="server" MaxLength="11"></asp:TextBox>

</div>

<div class="form-control">

<label for="editarEmail">E-mail:</label>

<asp:TextBox ID="editarEmail" runat="server"></asp:TextBox>

</div>

<div class="form-control">

<label for="editarGrupo">Grupo:</label>

<asp:TextBox ID="editarGrupo" runat="server"></asp:TextBox>

</div>

<div class="form-control btn-control">

<asp:Button ID="btnSalvar" runat="server" Text="Salvar" OnClick="btnSalvar\_Click" />

<asp:Button ID="btnCancelar" runat="server" Text="Cancelar" OnClick="btnCancelar\_Click" />

</div>

</div>

</form>

<script>

const editarTelefone = document.getElementById("editarTelefone");

const editarEmail = document.getElementById("editarEmail");

editarTelefone.addEventListener("keyup", (event) => {

event.preventDefault();

let valor;

valor = editarTelefone.value.replace(/\D/g, "");

editarTelefone.value = valor;

});

editarEmail.addEventListener("change", (event) => {

event.preventDefault();

let pattern = /^[a-z0-9.]+@[a-z0-9]+\.[a-z]+(\.[a-z]+)?$/gi;

let emailEstaOk = pattern.test(editarEmail.value);

if (!emailEstaOk) {

alert("Formato de e-mail inválido! verifique os dados e insira novamente")

editarEmail.value = ""

}

});

</script>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using WebApplication1.Controllers;

using WebApplication1.Models;

namespace WebApplication1.Views

{

public partial class editarcontato : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["login"] == null)

{

Response.Redirect("default.aspx");

}

}

protected void btnLogout\_Click(object sender, EventArgs e)

{

Session["login"] = null;

Response.Redirect("default.aspx");

}

protected void btnCancelar\_Click(object sender, EventArgs e)

{

editarBuscar.Text = "";

editarNome.Text = "";

editarTelefone.Text = "";

editarEmail.Text = "";

editarGrupo.Text = "";

Response.Redirect("menu.aspx");

}

protected void editarBtnBuscar\_Click(object sender, EventArgs e)

{

List<string> resposta = new List<string>();

try

{

ContatoController controller = new ContatoController();

resposta = controller.Pesquisar(editarBuscar.Text, Session["login"].ToString());

if (resposta[0] == "Contato não encontrado")

{

editarNome.Text = "";

editarTelefone.Text = "";

editarEmail.Text = "";

editarGrupo.Text = "";

Response.Write("<script>alert('Contato não localizado!')</script>");

}

else

{

contatoID.Text = resposta[0].ToString();

editarNome.Text = resposta[1].ToString();

editarTelefone.Text = resposta[2].ToString();

editarEmail.Text = resposta[3].ToString();

editarGrupo.Text = resposta[4].ToString();

}

}

catch (Exception erro)

{

Console.WriteLine(erro);

//throw erro;

}

}

protected void btnSalvar\_Click(object sender, EventArgs e)

{

try

{

ContatoModel contato = new ContatoModel(editarNome.Text, editarTelefone.Text, editarEmail.Text, editarGrupo.Text, Session["login"].ToString());

contato.Id = int.Parse(contatoID.Text);

ContatoController controller = new ContatoController();

bool resposta = controller.Atualizar(contato);

if (resposta)

{

Response.Write("<script>alert('Contato atualizado com sucesso!')</script>");

}

else

{

Response.Write("<script>alert('erro ao atualizar contato!')</script>");

}

}

catch (Exception erro)

{

Console.WriteLine(erro);

//throw erro;

}

}

}

}

### Menu.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="menu.aspx.cs" Inherits="WebApplication1.Views.menu2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<link href="../Styles/base.css" rel="stylesheet" />

<link href="../Styles/menu.css" rel="stylesheet" />

<title></title>

</head>

<body>

<form id="form1" runat="server" class="container">

<asp:Button ID="btnLogout" runat="server" Text="Sair" OnClick="btnLogout\_Click" />

<section class="card" data-tilt>

<h2>Editar</h2>

<p>pesquise por um contato e edite suas informações.</p>

<a href="editar\_contato.aspx" class="button">ir <img src="../Assets/arrow-right.svg" alt="ir" /></a>

</section>

<section class="card" data-tilt>

<h2>Novo contato</h2>

<p>cadastre novos contatos.</p>

<a href="novo\_contato.aspx" class="button">ir <img src="../Assets/arrow-right.svg" alt="ir" /></a>

</section>

<section class="card" data-tilt>

<h2>Contato</h2>

<p>Listar contatos cadastrados.</p>

<a href="pesquisar.aspx" class="button">ir <img src="../Assets/arrow-right.svg" alt="ir" /></a>

</section>

<section class="card" data-tilt>

<h2>Minha conta</h2>

<p>Altere seus dados cadastrais.</p>

<a href="alterar\_dados.aspx" class="button">ir <img src="../Assets/arrow-right.svg" alt="ir" /></a>

</section>

<script src="../Scripts/vanillatilt.js"></script>

<script type="text/javascript">

VanillaTilt.init(document.querySelectorAll(".card"), {

max: 25,

speed: 400,

glare: true,

"max-glare": 1,

});

</script>

</form>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication1.Views

{

public partial class menu2 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["login"] == null)

{

Response.Redirect("default.aspx");

}

}

protected void btnLogout\_Click(object sender, EventArgs e)

{

Session["login"] = null;

Response.Redirect("default.aspx");

}

}

}

### Novo\_contato.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="novo\_contato.aspx.cs" Inherits="WebApplication1.Views.novocontato" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<link href="../Styles/base.css" rel="stylesheet" />

<link href="../Styles/novo\_contato.css" rel="stylesheet" />

<title></title>

</head>

<body>

<form id="form1" runat="server" class="container">

<asp:Button ID="btnLogout" runat="server" Text="Sair" OnClick="btnLogout\_Click" />

<div class="form">

<div class="form-control">

<label for="novoNome">Nome:</label>

<asp:TextBox ID="novoNome" runat="server"></asp:TextBox>

</div>

<div class="form-control">

<label for="novoTelefone">Telefone:</label>

<asp:TextBox ID="novoTelefone" runat="server" MaxLength="11"></asp:TextBox>

</div>

<div class="form-control">

<label for="novoEmail">E-mail:</label>

<asp:TextBox ID="novoEmail" runat="server"></asp:TextBox>

</div>

<div class="form-control">

<label for="novoGrupo">Grupo:</label>

<asp:TextBox ID="novoGrupo" runat="server"></asp:TextBox>

</div>

<div class="btn-control form-control">

<asp:Button ID="btnSalvar" runat="server" Text="Salvar" OnClick="btnSalvar\_Click" />

<asp:Button ID="btnCancelar" runat="server" Text="Cancelar" OnClick="btnCancelar\_Click" />

</div>

</div>

</form>

<script>

const novoTelefone = document.getElementById("novoTelefone");

const novoEmail = document.getElementById("novoEmail");

novoTelefone.addEventListener("keyup", (event) => {

event.preventDefault();

let valor;

valor = novoTelefone.value.replace(/\D/g, "");

novoTelefone.value = valor;

});

novoEmail.addEventListener("change", (event) => {

event.preventDefault();

let pattern = /^[a-z0-9.]+@[a-z0-9]+\.[a-z]+(\.[a-z]+)?$/gi;

let emailEstaOk = pattern.test(novoEmail.value);

if (!emailEstaOk) {

alert("Formato de e-mail inválido! verifique os dados e insira novamente")

novoEmail.value = ""

}

});

</script>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using WebApplication1.Controllers;

using WebApplication1.Models;

namespace WebApplication1.Views

{

public partial class novocontato : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["login"] == null)

{

Response.Redirect("default.aspx");

}

}

protected void btnLogout\_Click(object sender, EventArgs e)

{

Session["login"] = null;

Response.Redirect("default.aspx");

}

protected void btnCancelar\_Click(object sender, EventArgs e)

{

novoNome.Text = "";

novoTelefone.Text = "";

novoEmail.Text = "";

novoGrupo.Text = "";

Response.Redirect("menu.aspx");

}

protected void btnSalvar\_Click(object sender, EventArgs e)

{

try

{

ContatoModel contato = new ContatoModel(novoNome.Text, novoTelefone.Text, novoEmail.Text, novoGrupo.Text, Session["login"].ToString());

ContatoController controller = new ContatoController();

bool resposta = controller.Inserir(contato);

if (resposta)

{

Response.Write("<script>alert('Novo contato inseridos!')</script>");

novoNome.Text = "";

novoTelefone.Text = "";

novoEmail.Text = "";

novoGrupo.Text = "";

}

else

{

Response.Write("<script>alert('Erro ao inserir novo contato!')</script>");

}

}

catch (Exception erro)

{

Console.WriteLine(erro);

//throw erro;

}

}

}

}

### Pesquisar.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="pesquisar.aspx.cs" Inherits="WebApplication1.Views.pesquisar2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<link href="../Styles/base.css" rel="stylesheet" />

<link href="../Styles/pesquisar.css" rel="stylesheet" />

<title></title>

</head>

<body>

<form id="form1" runat="server" class="container">

<asp:Button ID="btnLogout" runat="server" Text="Sair" OnClick="btnLogout\_Click" />

<div class="form">

<div class="form-control btn-control">

<asp:TextBox ID="txbSearch" runat="server"></asp:TextBox>

<asp:Button ID="btnSearch" runat="server" Text="Pesquisar" OnClick="btnSearch\_Click" />

</div>

<div class="table-card">

<table>

<caption>Contato</caption>

<asp:Label ID="contatoID" runat="server" Text="" Visible="False"></asp:Label>

<thead>

</thead>

<tbody>

<tr>

<td>Nome:</td>

<td><asp:Label ID="contatoNome" runat="server" Text=""></asp:Label></td>

</tr>

<tr>

<td>Telefone:</td>

<td><asp:Label ID="contatoTelefone" runat="server" Text=""></asp:Label></td>

</tr>

<tr>

<td>E-mail:</td>

<td><asp:Label ID="contatoEmail" runat="server" Text=""></asp:Label></td>

</tr>

<tr>

<td>Grupo:</td>

<td><asp:Label ID="contatoGrupo" runat="server" Text=""></asp:Label></td>

</tr>

</tbody>

</table>

</div>

<div class="btn-control form-control">

<asp:Button ID="btnEdit" runat="server" Text="Editar" OnClick="btnEdit\_Click" />

<asp:Button ID="btnDelete" runat="server" Text="Excluir" OnClick="btnDelete\_Click" />

<asp:Button ID="btnBack" runat="server" Text="Voltar" OnClick="btnBack\_Click" />

</div>

</div>

</form>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using WebApplication1.Controllers;

namespace WebApplication1.Views

{

public partial class pesquisar2 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["login"] == null)

{

Response.Redirect("default.aspx");

}

}

protected void btnLogout\_Click(object sender, EventArgs e)

{

Session["login"] = null;

Response.Redirect("default.aspx");

}

protected void btnBack\_Click(object sender, EventArgs e)

{

Response.Redirect("menu.aspx");

}

protected void btnEdit\_Click(object sender, EventArgs e)

{

Response.Redirect("editar\_contato.aspx");

}

protected void btnDelete\_Click(object sender, EventArgs e)

{

List<string> resposta = new List<string>();

try

{

ContatoController controller = new ContatoController();

resposta = controller.Pesquisar(txbSearch.Text, Session["login"].ToString());

if (resposta[0] != "Contato não encontrado")

{

bool confirmacao = controller.Excluir(int.Parse(contatoID.Text));

if (confirmacao)

{

Response.Write("<script>alert('Contato excluido com sucesso!')</script>");

}

else

{

Response.Write("<script>alert('Erro ao excluir o contato')</script>");

}

}

}

catch (Exception erro)

{

Console.WriteLine(erro);

//throw erro;

}

}

protected void btnSearch\_Click(object sender, EventArgs e)

{

List<string> resposta = new List<string>();

try

{

ContatoController controller = new ContatoController();

resposta = controller.Pesquisar(txbSearch.Text, Session["login"].ToString());

if (resposta[0] == "Contato não encontrado")

{

contatoNome.Text = "";

contatoTelefone.Text = "";

contatoEmail.Text = "";

contatoGrupo.Text = "";

Response.Write("<script>alert('Contato não localizado!')</script>");

}

else

{

contatoID.Text = resposta[0].ToString();

contatoNome.Text = resposta[1].ToString();

contatoTelefone.Text = resposta[2].ToString();

contatoEmail.Text = resposta[3].ToString();

contatoGrupo.Text = resposta[4].ToString();

}

}

catch (Exception erro)

{

Console.WriteLine(erro);

//throw erro;

}

}

}

}

## Estilos

### Base.css

@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@500&display=swap");

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: "Poppins", sans-serif;

font-size: 1rem;

/\* border: 1px solid green; \*/

/\* outline: 1px solid red; \*/

border: none;

outline: none;

}

.container {

min-width: 1fr;

max-width: 1fr;

min-height: 100vh;

max-height: 100vh;

overflow: auto;

}

.form-control {

width: 100%;

}

.form-control + .form-control,

.form-control + .btn-control {

margin-top: 0.5rem;

}

.btn-control {

display: flex;

flex-direction: row;

}

input,

button {

width: 100%;

height: 30px;

text-align: center;

color: rgba(255, 255, 255, 0.8);

}

#btnLogout{

position: absolute;

top:10px;

right:10px;

width:auto;

padding:2px;

background-color:transparent;

}

#btnLogout:hover{

text-decoration:underline;

}

### Cadastro.css

.container {

display: flex;

justify-content: center;

align-items: center;

flex-wrap: wrap;

background: rgb(2, 0, 36);

background: linear-gradient( 320deg, rgba(2, 0, 36, 1) 0%, rgba(9, 9, 121, 1) 26%, rgba(0, 212, 255, 1) 100% );

}

.container:after {

content: attr(data-page);

-webkit-text-stroke-width: 1px;

-webkit-text-stroke-color: rgba(255, 255, 255, 0.3);

-webkit-text-fill-color: rgba(0, 0, 0, 0);

position: absolute;

top: 0;

left: 0;

transform: translateY(-25%);

font-size: 5rem;

}

.container .form {

padding: 30px;

border-radius: 15px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(20px);

}

.container .form label {

color: rgba(255, 255, 255, 0.5);

}

.container .form input {

border-radius: 5px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0);

backdrop-filter: blur(20px);

}

.container .form input:focus {

background: rgba(255, 255, 255, 0.1);

}

.container .form button {

background: transparent;

border-radius: 5px;

color: rgba(255, 255, 255, 0.5);

cursor: pointer;

transition: 0.2s;

}

.container .form button:hover {

background: rgba(255, 255, 255, 0.2);

border: 1px solid rgba(255, 255, 255, 0.2);

}

.container .form button:active {

animation: fade 0.1s;

}

@keyframes fade {

0% {

transform: scale(1.2);

opacity: 1;

}

100% {

opacity: 0;

transform: scale(1);

}

}

### Editar\_contato.css

.container {

background: rgb(2, 0, 36);

background: linear-gradient( 320deg, rgba(2, 0, 36, 1) 0%, rgba(9, 9, 121, 1) 26%, rgba(0, 212, 255, 1) 100% );

}

.container:after {

content: attr(data-page);

-webkit-text-stroke-width: 1px;

-webkit-text-stroke-color: rgba(255, 255, 255, 0.3);

-webkit-text-fill-color: rgba(0, 0, 0, 0);

position: absolute;

top: 0;

left: 0;

transform: translateY(-25%);

font-size: 5rem;

}

.container,

.container .form {

display: flex;

align-items: center;

justify-content: center;

flex-direction: column;

}

.form {

padding: 30px;

border-radius: 15px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(20px);

}

.container .form label {

color: rgba(255, 255, 255, 0.5);

}

.container .form input {

border-radius: 5px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0);

backdrop-filter: blur(20px);

}

.container .form input:focus {

background: rgba(255, 255, 255, 0.1);

}

#btsearch {

width: 8rem;

}

.container button {

background: transparent;

border-radius: 5px;

color: rgba(255, 255, 255, 0.5);

cursor: pointer;

transition: 0.2s;

}

.container button:hover {

background: rgba(255, 255, 255, 0.2);

border: 1px solid rgba(255, 255, 255, 0.2);

}

.container button:active {

animation: fade 0.1s;

}

@keyframes fade {

0% {

transform: scale(1.2);

opacity: 1;

}

100% {

opacity: 0;

transform: scale(1);

}

}

### Editar\_dados.css

.container {

display: flex;

justify-content: center;

align-items: center;

flex-wrap: wrap;

background: rgb(2, 0, 36);

background: linear-gradient( 320deg, rgba(2, 0, 36, 1) 0%, rgba(9, 9, 121, 1) 26%, rgba(0, 212, 255, 1) 100% );

}

.container:after {

content: attr(data-page);

-webkit-text-stroke-width: 1px;

-webkit-text-stroke-color: rgba(255, 255, 255, 0.3);

-webkit-text-fill-color: rgba(0, 0, 0, 0);

position: absolute;

top: 0;

left: 0;

transform: translateY(-25%);

font-size: 5rem;

}

.container .form {

padding: 30px;

border-radius: 15px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(20px);

}

.container .form label {

color: rgba(255, 255, 255, 0.5);

}

.container .form input {

border-radius: 5px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0);

backdrop-filter: blur(20px);

}

.container .form input:focus {

background: rgba(255, 255, 255, 0.1);

}

.container .form button {

background: transparent;

border-radius: 5px;

color: rgba(255, 255, 255, 0.5);

cursor: pointer;

transition: 0.2s;

}

.container .form button:hover {

background: rgba(255, 255, 255, 0.2);

border: 1px solid rgba(255, 255, 255, 0.2);

}

.container .form button:active {

animation: fade 0.1s;

}

@keyframes fade {

0% {

transform: scale(1.2);

opacity: 1;

}

100% {

opacity: 0;

transform: scale(1);

}

}

### Index.css

.container {

background: rgb(2, 0, 36);

background: linear-gradient( 320deg, rgba(2, 0, 36, 1) 0%, rgba(9, 9, 121, 1) 26%, rgba(0, 212, 255, 1) 100% );

}

.container,

.container .form {

display: flex;

align-items: center;

justify-content: center;

flex-direction: column;

}

.form {

padding: 30px;

border-radius: 15px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(20px);

}

.form input {

border-radius: 5px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0);

backdrop-filter: blur(20px);

}

.form input:focus {

background: rgba(255, 255, 255, 0.1);

}

p {

color: rgba(255, 255, 255, 0.5);

}

a {

text-decoration: none;

color: rgba(255, 255, 255, 1);

}

p span a:hover {

font-size: 1.1rem;

}

button {

background: transparent;

border-radius: 5px;

color: rgba(255, 255, 255, 0.5);

cursor: pointer;

transition: 0.2s;

}

button:hover {

background: rgba(255, 255, 255, 0.2);

border: 1px solid rgba(255, 255, 255, 0.2);

}

button:active {

animation: fade 0.1s;

}

@keyframes fade {

0% {

transform: scale(1.2);

opacity: 1;

}

100% {

opacity: 0;

transform: scale(1);

}

}

### Menu.css

.container {

display: flex;

justify-content: center;

align-items: center;

flex-wrap: wrap;

background: rgb(2, 0, 36);

background: linear-gradient( 320deg, rgba(2, 0, 36, 1) 0%, rgba(9, 9, 121, 1) 26%, rgba(0, 212, 255, 1) 100% );

}

.card {

width: 230px;

height: 330px;

margin: 0.5rem;

padding: 2rem;

text-align: center;

display: flex;

justify-content: space-between;

flex-direction: column;

border-radius: 15px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(20px);

}

.card a {

text-decoration:none;

}

.card h2 {

font-size: 2.2rem;

color: rgba(255, 255, 255, 0.5);

cursor: default;

}

.card p {

color: rgba(255, 255, 255, 0.3);

text-align: end;

padding: 5px;

cursor: default;

}

.card .button {

background: transparent;

border-radius: 5px;

color: rgba(255, 255, 255, 0.5);

cursor: pointer;

display: flex;

justify-content: center;

align-items: center;

width: auto;

height: auto;

transition: 0.2s;

}

.card .button:hover {

background: rgba(255, 255, 255, 0.2);

border: 1px solid rgba(255, 255, 255, 0.2);

}

.card .button:active {

animation: fade 0.1s;

}

@keyframes fade {

0% {

transform: scale(1.2);

opacity: 1;

}

100% {

opacity: 0;

transform: scale(1);

}

}

### Novo\_contato.css

.container {

background: rgb(2, 0, 36);

background: linear-gradient( 320deg, rgba(2, 0, 36, 1) 0%, rgba(9, 9, 121, 1) 26%, rgba(0, 212, 255, 1) 100% );

}

.container,

.container .form {

display: flex;

align-items: center;

justify-content: center;

flex-direction: column;

}

.container:after {

content: attr(data-page);

-webkit-text-stroke-width: 1px;

-webkit-text-stroke-color: rgba(255, 255, 255, 0.3);

-webkit-text-fill-color: rgba(0, 0, 0, 0);

position: absolute;

top: 0;

left: 0;

transform: translateY(-25%);

font-size: 5rem;

}

.form {

padding: 30px;

border-radius: 15px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(20px);

}

.container .form input {

border-radius: 5px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0);

backdrop-filter: blur(20px);

}

.container .form label {

color: rgba(255, 255, 255, 0.5);

}

.container .form input:focus {

background: rgba(255, 255, 255, 0.1);

}

.container button {

background: transparent;

border-radius: 5px;

color: rgba(255, 255, 255, 0.5);

cursor: pointer;

transition: 0.2s;

}

.container button:hover {

background: rgba(255, 255, 255, 0.2);

border: 1px solid rgba(255, 255, 255, 0.2);

}

.container button:active {

animation: fade 0.1s;

}

@keyframes fade {

0% {

transform: scale(1.2);

opacity: 1;

}

100% {

opacity: 0;

transform: scale(1);

}

}

### Pesquisar.css

.container {

background: rgb(2, 0, 36);

background: linear-gradient( 320deg, rgba(2, 0, 36, 1) 0%, rgba(9, 9, 121, 1) 26%, rgba(0, 212, 255, 1) 100% );

}

.container:after {

content: attr(data-page);

-webkit-text-stroke-width: 1px;

-webkit-text-stroke-color: rgba(255, 255, 255, 0.3);

-webkit-text-fill-color: rgba(0, 0, 0, 0);

position: absolute;

top: 0;

left: 0;

transform: translateY(-25%);

font-size: 5rem;

}

.container > \* {

margin-top: 1rem;

}

.container,

.container .form {

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

}

.form,

.table-card {

padding: 30px;

width: 500px;

border-radius: 15px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(20px);

}

#btn-search {

width: 8rem;

}

.container .form input {

border-radius: 5px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.5);

background: rgba(255, 255, 255, 0);

backdrop-filter: blur(20px);

}

.container .form label {

color: rgba(255, 255, 255, 0.5);

}

.container .button {

background: transparent;

border-radius: 5px;

color: rgba(255, 255, 255, 0.5);

cursor: pointer;

transition: 0.2s;

}

.container .button:hover {

background: rgba(255, 255, 255, 0.2);

border: 1px solid rgba(255, 255, 255, 0.2);

}

.container .button:active {

animation: fade 0.1s;

}

@keyframes fade {

0% {

transform: scale(1.2);

opacity: 1;

}

100% {

opacity: 0;

transform: scale(1);

}

}

## Scripts

### Vanillatilt.js

var VanillaTilt = (function () {

'use strict';

/\*\*

\* Created by Sergiu Șandor (micku7zu) on 1/27/2017.

\* Original idea: https://github.com/gijsroge/tilt.js

\* MIT License.

\* Version 1.7.0

\*/

class VanillaTilt {

constructor(element, settings = {}) {

if (!(element instanceof Node)) {

throw ("Can't initialize VanillaTilt because " + element + " is not a Node.");

}

this.width = null;

this.height = null;

this.clientWidth = null;

this.clientHeight = null;

this.left = null;

this.top = null;

// for Gyroscope sampling

this.gammazero = null;

this.betazero = null;

this.lastgammazero = null;

this.lastbetazero = null;

this.transitionTimeout = null;

this.updateCall = null;

this.event = null;

this.updateBind = this.update.bind(this);

this.resetBind = this.reset.bind(this);

this.element = element;

this.settings = this.extendSettings(settings);

this.reverse = this.settings.reverse ? -1 : 1;

this.glare = VanillaTilt.isSettingTrue(this.settings.glare);

this.glarePrerender = VanillaTilt.isSettingTrue(this.settings["glare-prerender"]);

this.fullPageListening = VanillaTilt.isSettingTrue(this.settings["full-page-listening"]);

this.gyroscope = VanillaTilt.isSettingTrue(this.settings.gyroscope);

this.gyroscopeSamples = this.settings.gyroscopeSamples;

this.elementListener = this.getElementListener();

if (this.glare) {

this.prepareGlare();

}

if (this.fullPageListening) {

this.updateClientSize();

}

this.addEventListeners();

this.updateInitialPosition();

}

static isSettingTrue(setting) {

return setting === "" || setting === true || setting === 1;

}

/\*\*

\* Method returns element what will be listen mouse events

\* @return {Node}

\*/

getElementListener() {

if (this.fullPageListening) {

return window.document;

}

if (typeof this.settings["mouse-event-element"] === "string") {

const mouseEventElement = document.querySelector(this.settings["mouse-event-element"]);

if (mouseEventElement) {

return mouseEventElement;

}

}

if (this.settings["mouse-event-element"] instanceof Node) {

return this.settings["mouse-event-element"];

}

return this.element;

}

/\*\*

\* Method set listen methods for this.elementListener

\* @return {Node}

\*/

addEventListeners() {

this.onMouseEnterBind = this.onMouseEnter.bind(this);

this.onMouseMoveBind = this.onMouseMove.bind(this);

this.onMouseLeaveBind = this.onMouseLeave.bind(this);

this.onWindowResizeBind = this.onWindowResize.bind(this);

this.onDeviceOrientationBind = this.onDeviceOrientation.bind(this);

this.elementListener.addEventListener("mouseenter", this.onMouseEnterBind);

this.elementListener.addEventListener("mouseleave", this.onMouseLeaveBind);

this.elementListener.addEventListener("mousemove", this.onMouseMoveBind);

if (this.glare || this.fullPageListening) {

window.addEventListener("resize", this.onWindowResizeBind);

}

if (this.gyroscope) {

window.addEventListener("deviceorientation", this.onDeviceOrientationBind);

}

}

/\*\*

\* Method remove event listeners from current this.elementListener

\*/

removeEventListeners() {

this.elementListener.removeEventListener("mouseenter", this.onMouseEnterBind);

this.elementListener.removeEventListener("mouseleave", this.onMouseLeaveBind);

this.elementListener.removeEventListener("mousemove", this.onMouseMoveBind);

if (this.gyroscope) {

window.removeEventListener("deviceorientation", this.onDeviceOrientationBind);

}

if (this.glare || this.fullPageListening) {

window.removeEventListener("resize", this.onWindowResizeBind);

}

}

destroy() {

clearTimeout(this.transitionTimeout);

if (this.updateCall !== null) {

cancelAnimationFrame(this.updateCall);

}

this.reset();

this.removeEventListeners();

this.element.vanillaTilt = null;

delete this.element.vanillaTilt;

this.element = null;

}

onDeviceOrientation(event) {

if (event.gamma === null || event.beta === null) {

return;

}

this.updateElementPosition();

if (this.gyroscopeSamples > 0) {

this.lastgammazero = this.gammazero;

this.lastbetazero = this.betazero;

if (this.gammazero === null) {

this.gammazero = event.gamma;

this.betazero = event.beta;

} else {

this.gammazero = (event.gamma + this.lastgammazero) / 2;

this.betazero = (event.beta + this.lastbetazero) / 2;

}

this.gyroscopeSamples -= 1;

}

const totalAngleX = this.settings.gyroscopeMaxAngleX - this.settings.gyroscopeMinAngleX;

const totalAngleY = this.settings.gyroscopeMaxAngleY - this.settings.gyroscopeMinAngleY;

const degreesPerPixelX = totalAngleX / this.width;

const degreesPerPixelY = totalAngleY / this.height;

const angleX = event.gamma - (this.settings.gyroscopeMinAngleX + this.gammazero);

const angleY = event.beta - (this.settings.gyroscopeMinAngleY + this.betazero);

const posX = angleX / degreesPerPixelX;

const posY = angleY / degreesPerPixelY;

if (this.updateCall !== null) {

cancelAnimationFrame(this.updateCall);

}

this.event = {

clientX: posX + this.left,

clientY: posY + this.top,

};

this.updateCall = requestAnimationFrame(this.updateBind);

}

onMouseEnter() {

this.updateElementPosition();

this.element.style.willChange = "transform";

this.setTransition();

}

onMouseMove(event) {

if (this.updateCall !== null) {

cancelAnimationFrame(this.updateCall);

}

this.event = event;

this.updateCall = requestAnimationFrame(this.updateBind);

}

onMouseLeave() {

this.setTransition();

if (this.settings.reset) {

requestAnimationFrame(this.resetBind);

}

}

reset() {

this.event = {

clientX: this.left + this.width / 2,

clientY: this.top + this.height / 2

};

if (this.element && this.element.style) {

this.element.style.transform = `perspective(${this.settings.perspective}px) ` +

`rotateX(0deg) ` +

`rotateY(0deg) ` +

`scale3d(1, 1, 1)`;

}

this.resetGlare();

}

resetGlare() {

if (this.glare) {

this.glareElement.style.transform = "rotate(180deg) translate(-50%, -50%)";

this.glareElement.style.opacity = "0";

}

}

updateInitialPosition() {

if (this.settings.startX === 0 && this.settings.startY === 0) {

return;

}

this.onMouseEnter();

if (this.fullPageListening) {

this.event = {

clientX: (this.settings.startX + this.settings.max) / (2 \* this.settings.max) \* this.clientWidth,

clientY: (this.settings.startY + this.settings.max) / (2 \* this.settings.max) \* this.clientHeight

};

} else {

this.event = {

clientX: this.left + ((this.settings.startX + this.settings.max) / (2 \* this.settings.max) \* this.width),

clientY: this.top + ((this.settings.startY + this.settings.max) / (2 \* this.settings.max) \* this.height)

};

}

let backupScale = this.settings.scale;

this.settings.scale = 1;

this.update();

this.settings.scale = backupScale;

this.resetGlare();

}

getValues() {

let x, y;

if (this.fullPageListening) {

x = this.event.clientX / this.clientWidth;

y = this.event.clientY / this.clientHeight;

} else {

x = (this.event.clientX - this.left) / this.width;

y = (this.event.clientY - this.top) / this.height;

}

x = Math.min(Math.max(x, 0), 1);

y = Math.min(Math.max(y, 0), 1);

let tiltX = (this.reverse \* (this.settings.max - x \* this.settings.max \* 2)).toFixed(2);

let tiltY = (this.reverse \* (y \* this.settings.max \* 2 - this.settings.max)).toFixed(2);

let angle = Math.atan2(this.event.clientX - (this.left + this.width / 2), -(this.event.clientY - (this.top + this.height / 2))) \* (180 / Math.PI);

return {

tiltX: tiltX,

tiltY: tiltY,

percentageX: x \* 100,

percentageY: y \* 100,

angle: angle

};

}

updateElementPosition() {

let rect = this.element.getBoundingClientRect();

this.width = this.element.offsetWidth;

this.height = this.element.offsetHeight;

this.left = rect.left;

this.top = rect.top;

}

update() {

let values = this.getValues();

this.element.style.transform = "perspective(" + this.settings.perspective + "px) " +

"rotateX(" + (this.settings.axis === "x" ? 0 : values.tiltY) + "deg) " +

"rotateY(" + (this.settings.axis === "y" ? 0 : values.tiltX) + "deg) " +

"scale3d(" + this.settings.scale + ", " + this.settings.scale + ", " + this.settings.scale + ")";

if (this.glare) {

this.glareElement.style.transform = `rotate(${values.angle}deg) translate(-50%, -50%)`;

this.glareElement.style.opacity = `${values.percentageY \* this.settings["max-glare"] / 100}`;

}

this.element.dispatchEvent(new CustomEvent("tiltChange", {

"detail": values

}));

this.updateCall = null;

}

/\*\*

\* Appends the glare element (if glarePrerender equals false)

\* and sets the default style

\*/

prepareGlare() {

// If option pre-render is enabled we assume all html/css is present for an optimal glare effect.

if (!this.glarePrerender) {

// Create glare element

const jsTiltGlare = document.createElement("div");

jsTiltGlare.classList.add("js-tilt-glare");

const jsTiltGlareInner = document.createElement("div");

jsTiltGlareInner.classList.add("js-tilt-glare-inner");

jsTiltGlare.appendChild(jsTiltGlareInner);

this.element.appendChild(jsTiltGlare);

}

this.glareElementWrapper = this.element.querySelector(".js-tilt-glare");

this.glareElement = this.element.querySelector(".js-tilt-glare-inner");

if (this.glarePrerender) {

return;

}

Object.assign(this.glareElementWrapper.style, {

"position": "absolute",

"top": "0",

"left": "0",

"width": "100%",

"height": "100%",

"overflow": "hidden",

"pointer-events": "none"

});

Object.assign(this.glareElement.style, {

"position": "absolute",

"top": "50%",

"left": "50%",

"pointer-events": "none",

"background-image": `linear-gradient(0deg, rgba(255,255,255,0) 0%, rgba(255,255,255,1) 100%)`,

"width": `${this.element.offsetWidth \* 2}px`,

"height": `${this.element.offsetWidth \* 2}px`,

"transform": "rotate(180deg) translate(-50%, -50%)",

"transform-origin": "0% 0%",

"opacity": "0",

});

}

updateGlareSize() {

if (this.glare) {

Object.assign(this.glareElement.style, {

"width": `${this.element.offsetWidth \* 2}`,

"height": `${this.element.offsetWidth \* 2}`,

});

}

}

updateClientSize() {

this.clientWidth = window.innerWidth

|| document.documentElement.clientWidth

|| document.body.clientWidth;

this.clientHeight = window.innerHeight

|| document.documentElement.clientHeight

|| document.body.clientHeight;

}

onWindowResize() {

this.updateGlareSize();

this.updateClientSize();

}

setTransition() {

clearTimeout(this.transitionTimeout);

this.element.style.transition = this.settings.speed + "ms " + this.settings.easing;

if (this.glare) this.glareElement.style.transition = `opacity ${this.settings.speed}ms ${this.settings.easing}`;

this.transitionTimeout = setTimeout(() => {

this.element.style.transition = "";

if (this.glare) {

this.glareElement.style.transition = "";

}

}, this.settings.speed);

}

/\*\*

\* Method return patched settings of instance

\* @param {boolean} settings.reverse - reverse the tilt direction

\* @param {number} settings.max - max tilt rotation (degrees)

\* @param {startX} settings.startX - the starting tilt on the X axis, in degrees. Default: 0

\* @param {startY} settings.startY - the starting tilt on the Y axis, in degrees. Default: 0

\* @param {number} settings.perspective - Transform perspective, the lower the more extreme the tilt gets

\* @param {string} settings.easing - Easing on enter/exit

\* @param {number} settings.scale - 2 = 200%, 1.5 = 150%, etc..

\* @param {number} settings.speed - Speed of the enter/exit transition

\* @param {boolean} settings.transition - Set a transition on enter/exit

\* @param {string|null} settings.axis - What axis should be disabled. Can be X or Y

\* @param {boolean} settings.glare - What axis should be disabled. Can be X or Y

\* @param {number} settings.max-glare - the maximum "glare" opacity (1 = 100%, 0.5 = 50%)

\* @param {boolean} settings.glare-prerender - false = VanillaTilt creates the glare elements for you, otherwise

\* @param {boolean} settings.full-page-listening - If true, parallax effect will listen to mouse move events on the whole document, not only the selected element

\* @param {string|object} settings.mouse-event-element - String selector or link to HTML-element what will be listen mouse events

\* @param {boolean} settings.reset - false = If the tilt effect has to be reset on exit

\* @param {gyroscope} settings.gyroscope - Enable tilting by deviceorientation events

\* @param {gyroscopeSensitivity} settings.gyroscopeSensitivity - Between 0 and 1 - The angle at which max tilt position is reached. 1 = 90deg, 0.5 = 45deg, etc..

\* @param {gyroscopeSamples} settings.gyroscopeSamples - How many gyroscope moves to decide the starting position.

\*/

extendSettings(settings) {

let defaultSettings = {

reverse: false,

max: 15,

startX: 0,

startY: 0,

perspective: 1000,

easing: "cubic-bezier(.03,.98,.52,.99)",

scale: 1,

speed: 300,

transition: true,

axis: null,

glare: false,

"max-glare": 1,

"glare-prerender": false,

"full-page-listening": false,

"mouse-event-element": null,

reset: true,

gyroscope: true,

gyroscopeMinAngleX: -45,

gyroscopeMaxAngleX: 45,

gyroscopeMinAngleY: -45,

gyroscopeMaxAngleY: 45,

gyroscopeSamples: 10

};

let newSettings = {};

for (var property in defaultSettings) {

if (property in settings) {

newSettings[property] = settings[property];

} else if (this.element.hasAttribute("data-tilt-" + property)) {

let attribute = this.element.getAttribute("data-tilt-" + property);

try {

newSettings[property] = JSON.parse(attribute);

} catch (e) {

newSettings[property] = attribute;

}

} else {

newSettings[property] = defaultSettings[property];

}

}

return newSettings;

}

static init(elements, settings) {

if (elements instanceof Node) {

elements = [elements];

}

if (elements instanceof NodeList) {

elements = [].slice.call(elements);

}

if (!(elements instanceof Array)) {

return;

}

elements.forEach((element) => {

if (!("vanillaTilt" in element)) {

element.vanillaTilt = new VanillaTilt(element, settings);

}

});

}

}

if (typeof document !== "undefined") {

/\* expose the class to window \*/

window.VanillaTilt = VanillaTilt;

/\*\*

\* Auto load

\*/

VanillaTilt.init(document.querySelectorAll("[data-tilt]"));

}

return VanillaTilt;

}());

## Controladores

### ContatoController.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using WebApplication1.Models;

namespace WebApplication1.Controllers

{

public class ContatoController

{

public bool Inserir(ContatoModel contato)

{

MySqlConnection conn = new Conexao().conectar();

MySqlCommand cmd = new MySqlCommand();

cmd.CommandText = $"INSERT INTO Contacts" +

$"(contact\_name, contact\_phone, contact\_email, contact\_group, owner\_id)" +

$"VALUES (@nome, @telefone, @email, @grupo, @dono);";

cmd.Parameters.AddWithValue("@nome", contato.Nome);

cmd.Parameters.AddWithValue("@telefone", contato.Telefone);

cmd.Parameters.AddWithValue("@email", contato.Email);

cmd.Parameters.AddWithValue("@grupo", contato.Grupo);

cmd.Parameters.AddWithValue("@dono", contato.Dono);

cmd.Connection = conn;

try

{

int resposta = cmd.ExecuteNonQuery();

conn.Close();

conn.Dispose();

return (resposta > 0);

}

catch (Exception erro)

{

Console.WriteLine(erro);

throw erro;

}

}

public List<string> Pesquisar(string contato, string dono)

{

List<string> resposta = new List<string>();

MySqlConnection conn = new Conexao().conectar();

MySqlCommand cmd = new MySqlCommand();

cmd.CommandText = $"SELECT \* FROM Contacts WHERE contact\_name = @nome AND owner\_id = @dono;";

cmd.Parameters.AddWithValue("@nome", contato);

cmd.Parameters.AddWithValue("@dono", dono);

cmd.Connection = conn;

MySqlDataReader data = cmd.ExecuteReader();

if (data.Read())

{

resposta.Add(data[0].ToString());

resposta.Add(data[1].ToString());

resposta.Add(data[2].ToString());

resposta.Add(data[3].ToString());

resposta.Add(data[4].ToString());

resposta.Add(data[5].ToString());

conn.Close();

conn.Dispose();

return resposta;

}

resposta.Add("Contato não encontrado");

return resposta;

}

public bool Atualizar(ContatoModel contato)

{

MySqlConnection conn = new Conexao().conectar();

MySqlCommand cmd = new MySqlCommand();

cmd.CommandText = $"UPDATE Contacts SET contact\_name = @nome, contact\_phone = @telefone, contact\_email = @email, contact\_group = @grupo, owner\_id = @dono WHERE contact\_id = @id;";

cmd.Parameters.AddWithValue("@id", contato.Id);

cmd.Parameters.AddWithValue("@nome", contato.Nome);

cmd.Parameters.AddWithValue("@telefone", contato.Telefone);

cmd.Parameters.AddWithValue("@email", contato.Email);

cmd.Parameters.AddWithValue("@grupo", contato.Grupo);

cmd.Parameters.AddWithValue("@dono", contato.Dono);

cmd.Connection = conn;

try

{

int resposta = cmd.ExecuteNonQuery();

conn.Close();

conn.Dispose();

return (resposta > 0);

}

catch (Exception erro)

{

Console.WriteLine(erro);

throw erro;

}

}

public bool Excluir(int contato)

{

MySqlConnection conn = new Conexao().conectar();

MySqlCommand cmd = new MySqlCommand();

cmd.CommandText = $"DELETE FROM Contacts WHERE contact\_id = @contato;";

cmd.Parameters.AddWithValue("@contato", contato);

cmd.Connection = conn;

try

{

int resposta = cmd.ExecuteNonQuery();

conn.Close();

conn.Dispose();

return (resposta > 0);

}

catch (Exception erro)

{

Console.WriteLine(erro);

throw erro;

}

}

}

}

### UsuarioController.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using WebApplication1.Models;

namespace WebApplication1.Controllers

{

public class UsuarioController

{

public bool Inserir(UsuarioModel usuario)

{

MySqlConnection conn = new Conexao().conectar();

MySqlCommand cmd = new MySqlCommand();

cmd.CommandText = $"INSERT INTO Users (user\_name, user\_login, user\_password) values (@nome, @email, @senha);";

cmd.Parameters.AddWithValue("@nome", usuario.Nome);

cmd.Parameters.AddWithValue("@email", usuario.Email);

cmd.Parameters.AddWithValue("@senha", usuario.Senha);

cmd.Connection = conn;

int resposta = cmd.ExecuteNonQuery();

conn.Close();

conn.Dispose();

return (resposta > 0);

}

public bool Alterar(UsuarioModel usuario)

{

MySqlConnection conn = new Conexao().conectar();

MySqlCommand cmd = new MySqlCommand();

cmd.CommandText = $"UPDATE Users SET user\_name= @nome, user\_login = @email, user\_password= @senha WHERE user\_id = @id;";

cmd.Parameters.AddWithValue("@id", usuario.Id);

cmd.Parameters.AddWithValue("@nome", usuario.Nome);

cmd.Parameters.AddWithValue("@email", usuario.Email);

cmd.Parameters.AddWithValue("@senha", usuario.Senha);

cmd.Connection = conn;

int resposta = cmd.ExecuteNonQuery();

conn.Close();

conn.Dispose();

return (resposta > 0);

}

public List<string> Pesquisar(string usuario)

{

List<string> resposta = new List<string>();

MySqlConnection conn = new Conexao().conectar();

MySqlCommand cmd = new MySqlCommand();

cmd.CommandText = $"SELECT \* FROM Users where user\_login = @login;";

cmd.Parameters.AddWithValue("@login", usuario);

cmd.Connection = conn;

MySqlDataReader data = cmd.ExecuteReader();

if (data.Read())

{

resposta.Add(data[0].ToString());

resposta.Add(data[1].ToString());

resposta.Add(data[2].ToString());

resposta.Add(data[3].ToString());

conn.Close();

conn.Dispose();

return resposta;

}

resposta.Add("Usuário não encontrado");

return resposta;

}

}

}

## Modelos

### Conexao.cs

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

namespace WebApplication1.Models

{

public class Conexao

{

public MySqlConnection conectar()

{

string server = "127.0.0.1";

int port = 8000;

string database = "fatec\_aap";

string user = "root";

string password = "";

string stringDeConexao = $"Server={server};Port={port};Database={database};Uid={user};Pwd={password}";

var conexao = new MySqlConnection(stringDeConexao);

try

{

conexao.Open();

return conexao;

}

catch (Exception erro)

{

throw erro;

}

}

}

}

### ContatoModel.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebApplication1.Models

{

public class ContatoModel

{

private int id;

private string nome;

private string telefone;

private string email;

private string grupo;

private string dono;

public ContatoModel(string nome, string telefone, string email, string grupo, string dono)

{

this.nome = nome;

this.telefone = telefone;

this.email = email;

this.grupo = grupo;

this.dono = dono;

}

public int Id { get => id; set => id = value; }

public string Nome { get => nome; set => nome = value; }

public string Telefone { get => telefone; set => telefone = value; }

public string Email { get => email; set => email = value; }

public string Grupo { get => grupo; set => grupo = value; }

public string Dono { get => dono; set => dono = value; }

}

}

### UsuarioModel.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebApplication1.Models

{

public class UsuarioModel

{

private int id;

private string nome;

private string email;

private string senha;

public UsuarioModel(string nome, string email, string senha)

{

this.nome = nome;

this.email = email;

this.senha = senha;

}

public int Id { get => id; set => id = value; }

public string Nome { get => nome; set => nome = value; }

public string Email { get => email; set => email = value; }

public string Senha { get => senha; set => senha = value; }

}

}

# Cronograma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ATIVIDADE | 2021 | | | |
| mar | abr | mai | jun |
| Escolha do tema,definição do problema de pesquisa | x |  |  |  |
| Definição dos objetivos. Justificativa | x |  |  |  |
| Definição da metodologia | x |  |  |  |
| Pesquisa bibliográfica |  | x |  |  |
| Entrega do anteprojeto |  | x |  |  |
| Formulação da aplicação |  |  | x |  |
| Teste da aplicação |  |  | x |  |
| Elaboração do projeto completo |  |  |  | x |
| Entrega do projeto |  |  |  | x |

# Bibliografia

BRASIL. Executivo. **LEI n. 13709, de 14 de agosto de 2018. DISPÕE SOBRE A PROTEÇÃO DE DADOS PESSOAIS E ALTERA A LEI Nº 12.965, DE 23 DE ABRIL DE 2014 (MARCO CIVIL DA INTERNET).** Diário Judicial Eletrônico. Brasília, 14 de agosto de 2018. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/Lei/L13709.htm>. Acesso em: 14 jun. 2021.

BRASIL. Legislativo. **LEI n. 12965, de 23 de abril de 2014. ESTABELECE PRINCÍPIOS, GARANTIAS, DIREITOS E DEVERES PARA O USO DA INTERNET NO BRASIL**. Diário Judicial Eletrônico. Brasília, 23 de abril de 2014. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm>. Acesso em: 14 jun. 2021.

CASTELO BRANCO JÚNIOR, Eliseu; DE CASTRO MACHADO, Javam; DA SILVA MONTEIRO FILHO, José Maria. **Estratégias para Proteção da Privacidade de Dados Armazenados na Nuvem**. 2014. 75 p. Dissertação ( Doutorado em Ciências da Computação) - UNIVERSIDADE FEDERAL DO CEARÁ, 2014. Disponível em: https://www.inf.ufpr.br/sbbd-sbsc2014/sbbd/proceedings/artigos/pdfs/14.pdf. Acesso em: 20 mai. 2021.

GOULART, Gil Monteiro ; DE DAVID, Thomaz Delgado . O COMPARTILHAMENTO E ARMAZENAMENTO DE CONTEÚDOS NO ESPAÇO DIGITAL: UMA ANÁLISE ACERCA DA RESPONSABILIDADE DOS PROVEDORES EM FACE DA VIOLAÇÃO DE DIREITOS E GARANTIAS FUNDAMENTAIS. **Seminário internacional de demandas sociais e políticas públicas na sociedade contemporânea**, 2015.

RIBEIRO, Samuel de Oliveira; SILVA, Francisco Airton; VIEIRA, Alex Borges; GONÇALVES, Glauber Dias. **Análise de um Serviço Virtual de Armazenamento em Nuvem Ciente de Padrões de Acesso por Usuários.** *In*: ESCOLA REGIONAL DE COMPUTAÇÃO CEARÁ, MARANHÃO, PIAUÍ (ERCEMAPI), 7. , 2019, São Luís. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019 . p. 238-245.

SOMASUNDARAM, G.; SHRIVASTAVA, Alok; SERVICES, EMC Education. **Armazenamento e Gerenciamento de Informações: Como armazenar, gerenciar e proteger informações digitais**. Bookman Editora, 2009.