**PROJETO MERCURYMAIL**

**Discentes:** Bruna Cristina Mafra

Fabíola Oliveira

Igor Fernando Souza e Silva

**Docentes:**

Vânia de Oliveira Borges

Reni Aparecido Norberto Pinto

Sumário

[Layout e Descrição de Uso 2](#_Toc152841942)

[Classes e Métodos 3](#_Toc152841943)

[Banco de Dados 6](#_Toc152841944)

**Introdução**

O projeto MercuryMail foi concebido com o objetivo de oferecer uma experiência de excelência ao usuário. Com um layout amigável, claro e intuitivo, aliado a um backend eficiente, proporcionando não apenas facilidade de uso, mas também garantimos a segurança dos dados do usuário e a eficiência no acesso às funcionalidades do sistema.

# Layout e Descrição de Uso

O design do projeto foi concebido para oferecer aos usuários uma estética futurista, direta e agradável. Com um total de 8 formulários, garantimos uma experiência de navegação fácil, seja pelo mouse ou pelo teclado. Organizamos os elementos na tela de forma que possam ser acessados sequencialmente pela tecla "tab", proporcionando uma navegação prática para aqueles que optam por usar exclusivamente o teclado. O cursor do usuário é estrategicamente adaptado à interação com cada elemento do formulário, como botões, caixas de texto e campos somente de leitura, promovendo uma comunicação visual e intuitiva, atendendo assim aos requisitos fundamentais de uma interface bem elaborada.

Nossos formulários são compostos por:

* **Tela Inicial**

Uma mensagem de boas vindas ao usuário e dois botões. Um botão encaminha para os formulários de login e o outro para o formulário de cadastro.

* **Login**

No processo de login, o usuário interage com dois formulários distintos. No F\_Login, proporcionamos a flexibilidade de inserção de e-mail. Além disso, incluímos um link label - que direciona o usuário para o formulário F\_CriarConta. Completando o formulário, há um botão "Avançar" que, após a confirmação dos dados, encaminha o usuário para o segundo formulário de login.

No formulário de login 2, recepcionamos o usuário com seu nome um textbox de leitura, confirmando sua presença na plataforma. Adicionalmente, implementamos um textbox para a senha, com a opção de mascarar os dados e um checkbox para visualização da senha, se desejado. Um botão "Avançar" direciona o usuário para a página principal após a confirmação de todos os dados.

* **Criar Conta**

No formulário de "Criar Conta", o usuário efetua o cadastro dos dados necessários para acessar o Webmail. O formulário inclui 6 textboxes, sendo um para o nome, outro para o sobrenome, um destinado à criação do e-mail, um para a confirmação do e-mail, um para a definição da senha e o último para a confirmação da senha. A senha é verificada e só é cadastrada caso seja uma senha considerada forte. Ao inserir todos os dados corretamente, ao clicar no botão "Criar", o usuário será direcionado para a Página Inicial.

* **Página Inicial**

O Formulário de Página Inicial, identificado como Form1, serve como a porta de entrada para que o usuário explore as principais funcionalidades do Webmail. Este formulário é composto por cinco botões, cada um direcionando para uma funcionalidade-chave do Webmail (Nova Mensagem, Caixa de Entrada, Caixa de Saída, , Lixeira e Sair).

Cada funcionalidade possui um formulário específico; por exemplo, ao acionar o botão "Caixa de Entrada", o DataGridView exibirá os dados associados a essa opção. Ao interagir com um desses dados, um formulário de exibição será ativado, proporcionando as funcionalidades específicas correspondentes a cada opção. Cada botão vinculado ao DataGridView (Caixa de Entrada, Caixa de Saída, Spam e Lixeira) possui seu próprio formulário individual, oferecendo as ações específicas que serão exibidas. Os e-mail da lista do DataGridView podem ser expandidos ao lado. Os botões "Nova Mensagem" e "Sair" foram desenvolvidos para cumprir suas funções principais de maneira eficiente.

# Classes e Métodos

A linguagem de programação usada no back-end da aplicação é C#, aliada ao .NET, desenvolvida utilizando o Visual Studio Community 2022.

Foram criadas três classes principais para o funcionamento do programa: Usuario.cs, Email.cs e Conexao.cs.

A classe Conexao é responsável por toda a comunicação com o banco de dados. Ela possui as propriedades:

* Servidor (string);
* Database (string);
* Cadeia conexão (string);
* Um objeto do tipo SqlConnection nomeado “conexão”.

A classe possui os métodos:

* AbrirConexao: realiza a conexão com o banco de dados (implementado nos demais arquivos através da diretiva “using”);
* FecharConexao: encerra a conexão, se thouver sido previamente aberta.
* CriarBancoDeDados: Verifica a existência na máquina em questão, do banco de dados nomeado “WebMailDB”. Após a verificação, há dois caminhos:

Caso o banco já houver sido criado, exibe uma mensagem de confirmação no console. Se o banco não existe, realiza a criação do banco de dados e realiza a chamada da função CriarTabelas;

* CriarTabelas: cria as tabelas Usuario e Email no banco de dados “WebMailDB”.

A classe Usuario foi elaborada com as seguintes propriedades:

* Id (int);
* Nome (string);
* Sobrenome (string);
* Email (string);
* Senha (criptografada).

E os métodos :

* 2 Construtores: um vazio, e um que recebe por parâmetro os dados de: nome, sobrenome, email e senha (que são validados antes da chamada do construtor, no formulário de criação da conta).
* Cadastro: realiza a conexão com o banco de dados (através da instância da classe Conexao), insere os dados recebidos e validados no banco de dados, e realiza um SELECT no Id, após a inserção do registro, para obter o valor do ID gerado para aquele registro. Assim, todos os campos daquele objeto são preenchidos: Id, Nome, Sobrenome, Email e Senha criptografada.

A classe Email armazena os dados de cada e-mail gerado. Suas propriedades são:

* Id (int);
* IdRemetente (int);
* IdDestinatario (int);
* Assunto (string);
* Mensagem (string);

E os métodos:

* Construtor completo;
* ObterIdPorEmail: Realiza uma busca no banco de dados para obter o ID do destinatário ou remetente, para que não haja acesso direto do usuário à informação do ID.
* Enviar: realiza a inserção do e-mail no banco de dados, após seu envio por algum usuário.

Quanto aos formulários, o formulário F\_NovoEmail é o único que possui três construtores distintos, para três casos distintos, de sua instância:

1. Ele pode ser instanciado com o único parâmetro de ID do Usuário (mantendo sempre a “sessão ativa), quando o botão da tela principal “NOVO EMAIL” é clicado.
2. Ele pode ser instânciado com os parâmetros de: ID do Usuário e Destinatário, quando é exibido após o clique do usuário na opção “RESPONDER” em algum e-mail da caixa de entrada.
3. Ele pode ser instanciado com os parâmetros de: ID do Usuário, Assunto e Mensagem, quando é exibido após o clique do usuário na opção “ENCAMINHAR” de um e-mail exibido.

**Pontos importantes:**

* **Validação de campos ao criar conta:**

Ao criar uma nova conta, há a validação da força da senha inserida, validação se o e-mail inserido possui o caracter “@”, validação se as senhas do campo de confirmação são iguais, validação se os e-mails do campo de confirmação são iguais e a validação se todos os campos são diferentes de nulo (“ ”).

Quanto à validação da força da senha, é feita através dos métodos:

* **Criptografia:**

Utilizar a criptografia para armazenar senhas é uma prática essencial para garantir a segurança dos dados dos usuários. Usando a biblioteca System.Security.Cryptography, utilizamos o algoritmo de hash SHA-256 para criar um hash seguro das senhas antes de armazená-las no banco de dados.

* **Criação de sessões:**

Simulamos a existência de sessões ao passar como parâmetro para cada novo formulário, o ID da sessão do usuário logado, após sua autenticação no MercuryMail.

**Diagrama de Classes:**

Texto

Descrição gerada automaticamente

# Banco de Dados

Segue o **Modelo Conceitual:**

# 

**Modelo Lógico:**

Diagrama

Descrição gerada automaticamente

**Modelo Físico:**

**CREATE DATABASE WebMailDB**

CREATE TABLE Usuario (

Id INT IDENTITY(1, 1) PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL,

senha VARCHAR(100) NOT NULL

);

CREATE TABLE Email (

Id INT IDENTITY(1, 1) PRIMARY KEY,

IdRemetente INT NOT NULL,

IdDestinatario INT NOT NULL,

Assunto VARCHAR(50) NOT NULL,

Mensagem VARCHAR(200) NOT NULL,

LixeiraRemetente BIT DEFAULT (0) NOT NULL,

LixeiraDestinatario BIT DEFAULT (0) NOT NULL,

Lido BIT DEFAULT (0) NOT NULL,

FOREIGN KEY (IdRemetente) REFERENCES Usuario(Id),

FOREIGN KEY (IdDestinatario) REFERENCES Usuario(Id)

);