**Fatec Jundiaí - Deputado Ary Fossen**

Nicolas Sylverio Pereira – Gestão de TI - 171A13

TATI

proficiência

Jundiaí

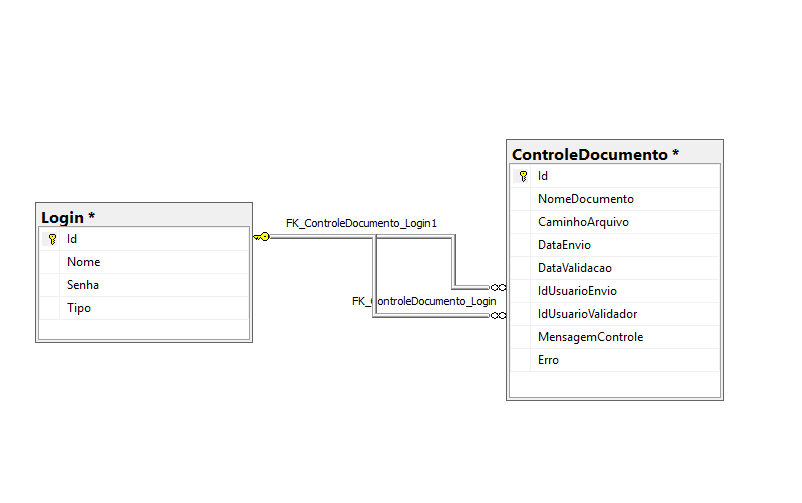
2019

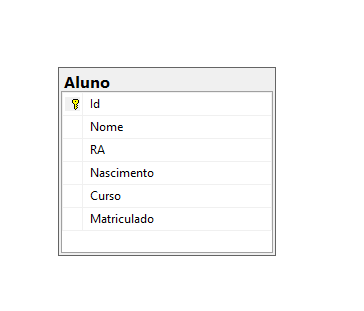
Projeto

O projeto consiste em um sistema de cadastro de alunos, com o CRUD completo criar, consultar, deletar e alterar. Também tem uma tela de consulta onde o usuário consegue consultar alunos, pelos filtros “todos”, “nome”, e “RA” e o resultado pode ser impresso em PDF como um relatório simples.

Também no sistema tem uma tela para enviar documentos para analise, gerando um protocolo de envio com data e nome do arquivo. Existe uma tela para consultar os relatórios enviados somente para os usuários que tem permissão tipo 1, nessa tela tem possibilidade de baixar o arquivo selecionado. E a Ultima tela desse sistema contém uma tela para consultar os status dos arquivos que o usuário enviou, seja ele enviado validado ou bloqueado.

Banco de Dados:





Foi utilizado o *Entity* *Framework* para consultas e gerenciamento do banco de dados, arquitetura *Code First* então montei os objetos e o *entity* cuida de criar as tabelas.

Para carregar as tabelas tem 2 maneiras:

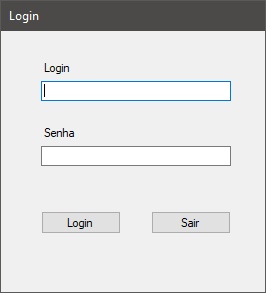
* Primeira é executando o sistema, na primeira ação que o sistema executa no banco de dados ele já cria o banco, pode demorar um pouco, mas somente na primeira vez.
* A segunda maneira é executa o comando *Update-Database* no package console do *visual studio* **(recomendado).** Esta segunda opção também executa um método *seed* criando um usuário padrão para executar o sistema.

Usuário 1: master, senha: 1234, tipo permissão 1

Usuário 2: usuário, senha 1234, tipo permissão 0.

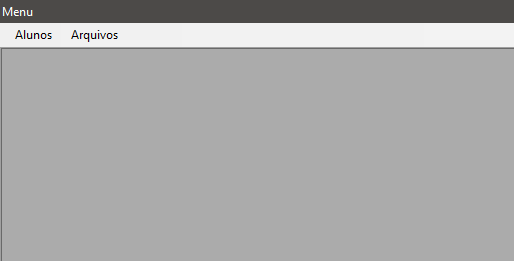
Telas

Login:

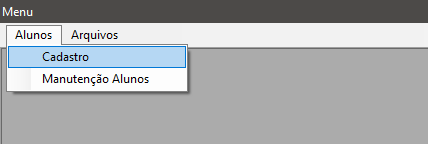


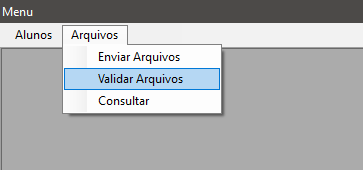
Tela para fazer o login e senha. Tela inicial do sistema

Menu:

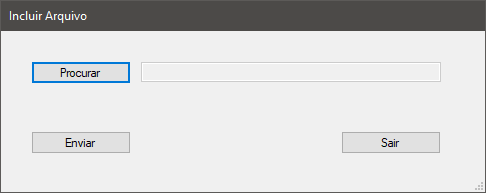


Menu principal do sistema, com os menus Alunos e Arquivos.





Envio Arquivo:



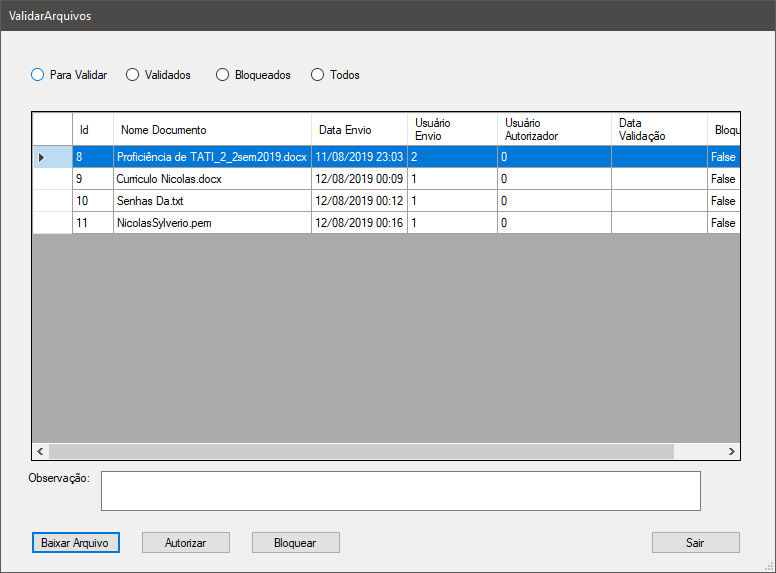
Ao clicar em procurar abre o arquivo explorer para selecionar o arquivo.

Após selecionar um arquivo valido, ele preenche o caminho do arquivo no textbox ao lado

Ao clicar em enviar ele envia o arquivo para uma pasta no sistema, armazena as informações no banco de dados, e emite um relaotio pdf com informações do envio, nome do arquivo, usuário que enviou e data.

Ao clicar em sair ele fecha a tela.

Validar Arquivos:



A tela tem permissão de acesso somente usuários tipo 1 podem acessá-lo.

A tela vem carregada com os arquivos que precisam de validação.

Existem 4 radiobuttons:

* Para Validar: Arquivos que precisam tomar uma ação.
* Validados: Arquivos que já foram autorizados.
* Bloqueados: Todos os arquivos que não foram autorizados.
* Todos: traz todos os arquivos.

Baixar Arquivo: Abre o Explorer para consultar a pasta selecionado e baixa o arquivo selecionado para validação.

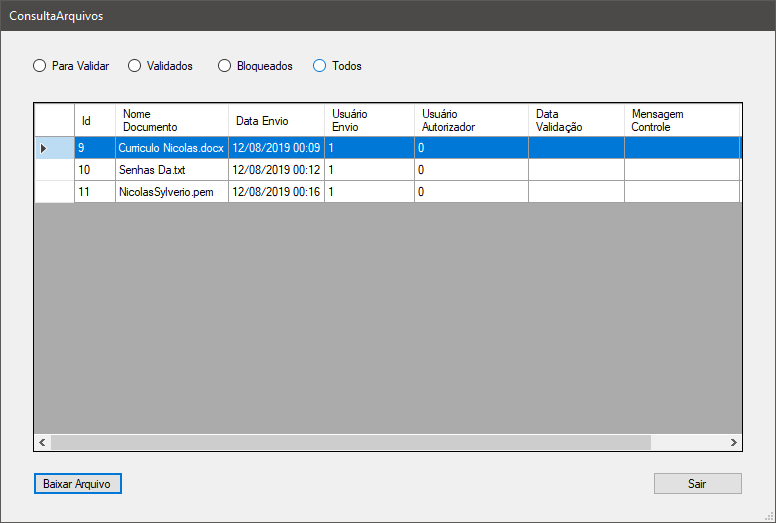
Observação: campo de texto para armazenar observações sobre validação ou sobre bloqueio.

Autorizar: Libera o arquivo, salvando usuário que autorizou data e observação.

Bloquear: Bloqueia Arquivo, salvando usuário que autorizou bloqueando e a observação.

Sair: Fecha a tela do sistema.

Consultar Arquivos:



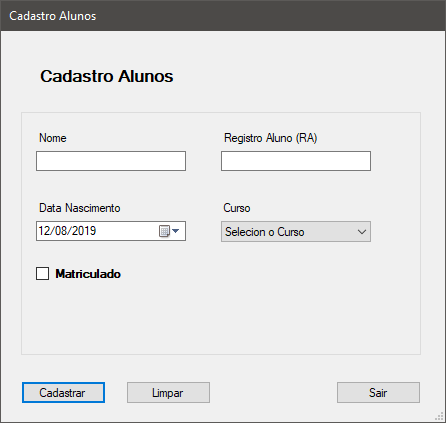
Tela para o usuário consultar a situação o arquivo que ele enviou.

Existem 4 radiobuttons:

* Para Validar: Arquivos que precisam tomar uma ação.
* Validados: Arquivos que já foram autorizados.
* Bloqueados: Todos os arquivos que não foram autorizados.
* Todos: traz todos os arquivos.

Baixar Arquivo: Abre o Explorer para consultar a pasta selecionado e baixa o arquivo selecionado para conferência.

Cadastro de alunos:



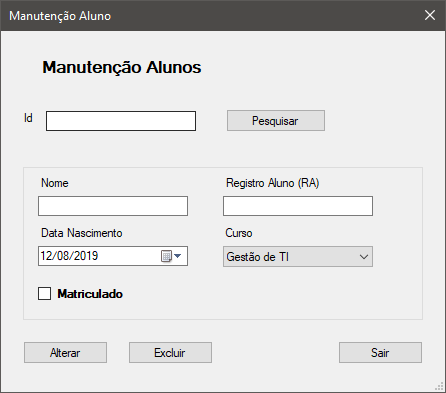
Cadastra um aluno no sistema.

Cadastrar: Cadastra de acordo com os dados selecionados.

Limpar: Limpa os dados dos campos da tela.

Sair: Fecha a tela.

Manutenção de Alunos:



Tela para alterar e excluir um aluno

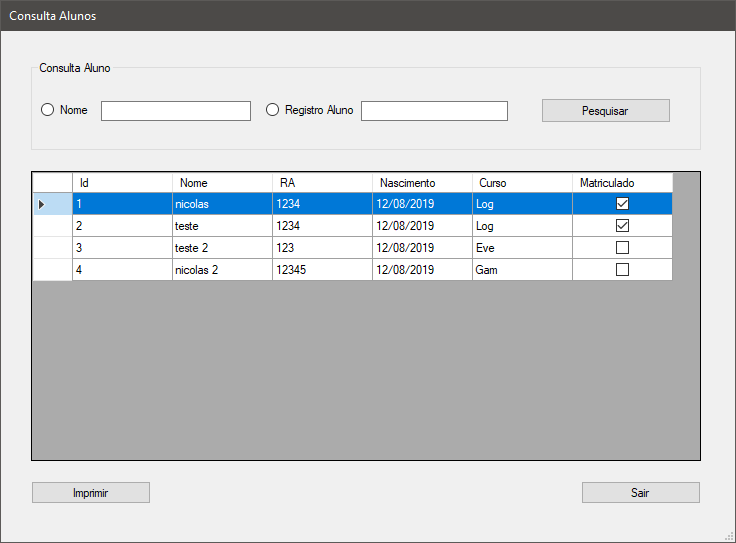
Ao Pesquisar um aluno pelo Id a tela é carregada com os dados do aluno e o usuário pode preencher os campos e alterá-lo ou clicar em excluir e exclui-lo.

Alterar: Altera os dados do aluno de acordo com os dados preenchidos.

Excluir: exclui o aluno selecionado.

Sair: Fecha o Sistema.

Consulta Alunos:



Tela para consultar os alunos e imprimir um relatório de alunos.

Pesquisar: Caso não tenha selecionado um nome retorna todos os alunos.

Imprimir: imprimir os dados da tela.

Sair: fecha a tela.

Código das Telas:

public partial class Login : Form

{

public Login()

{

InitializeComponent();

}

private void BtnSair\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void BtnLogin\_Click(object sender, EventArgs e)

{

try

{

var resultado = Logon.ValidarUsuario(txtLogin.Text, txtSenha.Text);

if (resultado)

{

var form = new Menu();

Visible = false;

form.ShowDialog(this);

}

else

{

MessageBox.Show("Usuario ou Senha Inválidos.", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

MessageBox.Show($"Erro Fatal: {ex}", "Aviso Sistema");

}

}

}

public partial class Menu : Form

{

public Menu()

{

InitializeComponent();

}

private void Menu\_Load(object sender, EventArgs e)

{

}

private void CadastraToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

if (Logon.Tipo != 1)

{

MessageBox.Show("Usuário sem permissão para acessar a rotina.", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

using (var form = new CadastroAlunos())

{

form.ShowDialog(this);

}

}

catch (Exception)

{

MessageBox.Show("Erro no Sistema", "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void EnviarArquivosToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

using (var form = new IncluirArquivo())

{

form.ShowDialog(this);

}

}

catch (Exception)

{

MessageBox.Show("Erro no Sistema", "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void ValidarArquivosToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

if (Logon.Tipo != 1)

{

MessageBox.Show("Usuário sem permissão para acessar a rotina.", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

using (var form = new ValidarArquivos())

{

form.ShowDialog(this);

}

}

catch (Exception)

{

MessageBox.Show("Erro no Sistema", "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void ConsultarToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

using (var form = new ConsultaArquivos())

{

form.ShowDialog(this);

}

}

catch (Exception)

{

MessageBox.Show("Erro no Sistema", "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void ManutençãoAlunosToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

if (Logon.Tipo != 1)

{

MessageBox.Show("Usuário sem permissão para acessar a rotina.", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

using (var form = new ManutencaoAluno())

{

form.ShowDialog(this);

}

}

catch (Exception)

{

MessageBox.Show("Erro no Sistema", "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void ConsultaAlunosToolStripMenuItem\_Click(object sender, EventArgs e)

{

try

{

if (Logon.Tipo != 1)

{

MessageBox.Show("Usuário sem permissão para acessar a rotina.", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

using (var form = new ConsultaAlunos())

{

form.ShowDialog(this);

}

}

catch (Exception)

{

MessageBox.Show("Erro no Sistema", "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

public partial class IncluirArquivo : Form

{

private readonly ControleDocumentoRepository \_controleDocumentoRepository;

public IncluirArquivo()

{

InitializeComponent();

\_controleDocumentoRepository = new ControleDocumentoRepository(new Context());

if (\_controleDocumentoRepository == null) throw new ArgumentNullException($"Erro ao inicializar sistema. parametro: {nameof(\_controleDocumentoRepository)}");

}

private void BtnSair\_Click(object sender, EventArgs e)

{

Close();

}

private void BtnProcurar\_Click(object sender, EventArgs e)

{

using (var fileDialog = new OpenFileDialog())

{

fileDialog.InitialDirectory = @"C:\Users";

fileDialog.Filter = "txt files|\*.txt;\*.docx|csv files|\*.csv;\*.xlsx|pdf files|\*.pdf|Images files|\*.BMP;\*.JPG;\*.GIF;\*.PNG;\*.TIFF|All files (\*.\*)|\*.\*";

var file = fileDialog.ShowDialog(this);

if (file == DialogResult.OK && File.Exists(fileDialog.FileName))

{

txtArquivo.Text = fileDialog.FileName;

}

};

}

private void BtnEnviar\_Click(object sender, EventArgs e)

{

try

{

if (!string.IsNullOrWhiteSpace(txtArquivo.Text))

{

var arquivoSistema = Directory.GetCurrentDirectory();

if (arquivoSistema.IndexOf(Constants.SistemDebugPath) != 0)

{

arquivoSistema = arquivoSistema.Replace(Constants.SistemDebugPath, Constants.SistemFileSendPath);

}

else if (arquivoSistema.IndexOf(Constants.SistemReleasePath) != 0)

{

arquivoSistema = arquivoSistema.Replace(Constants.SistemReleasePath, Constants.SistemFileSendPath);

}

else

{

throw new Exception("Caminho do Arquivo do sistema não especificado.");

}

var fileNameArray = txtArquivo.Text.Split('\\');

var fileName = fileNameArray[fileNameArray.Length - 1];

var fileNameWithPath = arquivoSistema + "\\" + fileName;

if (File.Exists(fileNameWithPath))

{

throw new Exception($"Nome do Arquivo já Existente na Base ou Arquivo já Enviado. Altere o nome e verifique se este ja não foi enviado. Nome Arquivo: {fileName}");

}

var controle = new ControleDocumento

{

NomeDocumento = fileName,

IdUsuarioEnvio = Logon.Id,

DataEnvio = DateTime.Now,

CaminhoArquivo = fileNameWithPath,

Erro = false,

DataValidacao = null

};

\_controleDocumentoRepository.Add(controle);

File.Copy(txtArquivo.Text, fileNameWithPath);

MessageBox.Show($"Arquivo Enviado com Sucesso. Arquivo : {fileName}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Information);

GerarRelatorio(fileName, txtArquivo.Text.Replace(fileName, ""));

txtArquivo.Text = string.Empty;

}

}

catch (Exception ex)

{

MessageBox.Show($"Erro do Sistema {ex.Message}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void GerarRelatorio(string fileName, string caminho)

{

using (var document = new PdfDocument())

{

var page = document.AddPage();

var graphics = XGraphics.FromPdfPage(page);

var textFormatter = new XTextFormatter(graphics);

var font = new XFont("Calibri", 14);

textFormatter.Alignment = XParagraphAlignment.Center;

textFormatter.DrawString($"Arquivo Enviado Com Sucesso: {fileName} Login: {Logon.Usuario}, Data: {DateTime.Now.ToString("dd/MM/yyyy mm:ss")}", font, XBrushes.Black, new XRect(30, 60, page.Width - 60, page.Height - 60));

var path = $"{caminho}relatorio{DateTime.Now.ToString("yyyyMMddmmss")}.pdf";

document.Save(path);

Process.Start(path);

}

}

}

public partial class ValidarArquivos : Form

{

private readonly ControleDocumentoRepository \_documentoRepository;

public ValidarArquivos()

{

InitializeComponent();

\_documentoRepository = new ControleDocumentoRepository(new Context());

if (\_documentoRepository == null) throw new ArgumentNullException($"Falha ao carregar componentes da tela. parametro: {nameof(\_documentoRepository)}");

}

private void ValidarArquivos\_Load(object sender, EventArgs e)

{

CarregarTela();

}

private void CarregarTela()

{

var lista = new BindingList<ControleDocumento>();

var documentos = \_documentoRepository

.GetAll()

.Where(x => x.DataValidacao == null && !x.Erro)

.Take(20);

foreach (var item in documentos)

{

lista.Add(item);

}

dtgArquivos.DataSource = lista;

}

private void BtnSair\_Click(object sender, EventArgs e)

{

Close();

}

private void BtnBaixar\_Click(object sender, EventArgs e)

{

try

{

var linha = dtgArquivos.SelectedRows[0].Cells;

var nomeArquivo = linha[1].Value.ToString();

var caminhoArquivo = linha[2].Value.ToString();

using (var folder = new FolderBrowserDialog())

{

var dialogResult = folder.ShowDialog(this);

if (dialogResult == DialogResult.OK)

{

var caminhoSelecionado = $"{folder.SelectedPath.ToString()}\\{DateTime.Now.ToString("yyyyMMddhhmmss")}-{nomeArquivo}";

File.Copy(caminhoArquivo, caminhoSelecionado);

MessageBox.Show($"Arquivo baixado no diretorio selecionado: {caminhoSelecionado}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

};

}

catch (Exception ex)

{

MessageBox.Show($"Erro ao baixar o arquivo. Erro: {ex.Message}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void BtnAutorizar\_Click(object sender, EventArgs e)

{

try

{

var linha = dtgArquivos.SelectedRows[0].Cells;

var idArquivo = Convert.ToInt32(linha[0].Value);

var nomeArquivo = linha[1].Value.ToString();

var caminhoArquivo = linha[2].Value.ToString();

var arquivoSistema = Directory.GetCurrentDirectory();

var documento = \_documentoRepository.GetById(idArquivo);

if (arquivoSistema.IndexOf(Constants.SistemDebugPath) != 0)

{

arquivoSistema = arquivoSistema.Replace(Constants.SistemDebugPath, Constants.SistemFileAutorizedPath + "\\" + nomeArquivo);

}

else if (arquivoSistema.IndexOf(Constants.SistemReleasePath) != 0)

{

arquivoSistema = arquivoSistema.Replace(Constants.SistemReleasePath, Constants.SistemFileAutorizedPath + "\\" + nomeArquivo);

}

else

{

throw new Exception("Caminho do Arquivo do sistema não especificado.");

}

if (documento.Erro) throw new Exception("Arquivo Bloqueado");

if (documento.DataValidacao != null) throw new Exception("Arquivo Já Válidado");

documento.DataValidacao = DateTime.Now;

documento.IdUsuarioValidador = Logon.Id;

documento.CaminhoArquivo = arquivoSistema;

documento.MensagemControle = txtObservacao.Text;

\_documentoRepository.SaveChanges();

File.Move(caminhoArquivo, arquivoSistema);

MessageBox.Show("Arquivo Autorizado com sucesso.", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Information);

CarregarTela();

}

catch (Exception ex)

{

MessageBox.Show($"Erro ao autorizar o arquivo. Erro: {ex.Message}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void BtnBloquear\_Click(object sender, EventArgs e)

{

try

{

var linha = dtgArquivos.SelectedRows[0].Cells;

var idArquivo = Convert.ToInt32(linha[0].Value);

var nomeArquivo = linha[1].Value.ToString();

var caminhoArquivo = linha[2].Value.ToString();

var arquivoSistema = Directory.GetCurrentDirectory();

var documento = \_documentoRepository.GetById(idArquivo);

if (arquivoSistema.IndexOf(Constants.SistemDebugPath) != 0)

{

arquivoSistema = arquivoSistema.Replace(Constants.SistemDebugPath, Constants.SistemFileBloquedPath + "\\" + nomeArquivo);

}

else if (arquivoSistema.IndexOf(Constants.SistemReleasePath) != 0)

{

arquivoSistema = arquivoSistema.Replace(Constants.SistemReleasePath, Constants.SistemFileBloquedPath + "\\" + nomeArquivo);

}

else

{

throw new Exception("Caminho do Arquivo do sistema não especificado.");

}

if (documento.Erro) throw new Exception("Arquivo Bloqueado");

if (documento.DataValidacao != null) throw new Exception("Arquivo Já Válidado");

documento.DataValidacao = DateTime.Now;

documento.IdUsuarioValidador = Logon.Id;

documento.CaminhoArquivo = arquivoSistema;

documento.Erro = true;

documento.MensagemControle = txtObservacao.Text;

\_documentoRepository.SaveChanges();

File.Move(caminhoArquivo, arquivoSistema);

MessageBox.Show("Arquivo bloqueado com sucesso.", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Information);

CarregarTela();

}

catch (Exception ex)

{

MessageBox.Show($"Erro ao bloquear o arquivo. Erro: {ex.Message}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void Pesquisar()

{

try

{

var lista = new BindingList<ControleDocumento>();

var documentos = \_documentoRepository

.GetAll();

if (rdbValidar.Checked)

{

documentos = documentos

.Where(x => !x.Erro && x.DataValidacao == null)

.Take(20);

}

else if (rdbValidados.Checked)

{

documentos = documentos

.Where(x => !x.Erro && x.DataValidacao != null)

.Take(20);

}

else if (rdbBloqueados.Checked)

{

documentos = documentos

.Where(x => x.Erro)

.Take(20);

}

else

{

documentos = documentos

.Take(20);

}

foreach (var item in documentos)

{

lista.Add(item);

}

dtgArquivos.DataSource = lista;

}

catch (Exception ex)

{

MessageBox.Show($"Erro ao pesquisar os arquivos. Erro: {ex.Message}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RdbTodos\_Click(object sender, EventArgs e)

{

Pesquisar();

}

private void RdbBloqueados\_Click(object sender, EventArgs e)

{

Pesquisar();

}

private void RdbValidados\_Click(object sender, EventArgs e)

{

Pesquisar();

}

private void RdbValidar\_Click(object sender, EventArgs e)

{

Pesquisar();

}

}

public partial class ConsultaArquivos : Form

{

private readonly ControleDocumentoRepository \_documentoRepository;

public ConsultaArquivos()

{

InitializeComponent();

\_documentoRepository = new ControleDocumentoRepository(new Context());

if (\_documentoRepository == null) throw new ArgumentNullException($"Falha ao carregar componentes da tela. parametro: {nameof(\_documentoRepository)}");

}

private void ConsultaArquivos\_Load(object sender, EventArgs e)

{

CarregarTela();

}

private void CarregarTela()

{

var lista = new BindingList<ControleDocumento>();

var documentos = \_documentoRepository

.GetAll()

.Where(x => x.DataValidacao == null && !x.Erro && x.IdUsuarioEnvio == Logon.Id)

.Take(20);

foreach (var item in documentos)

{

lista.Add(item);

}

dtgArquivos.DataSource = lista;

}

private void Pesquisar()

{

try

{

var lista = new BindingList<ControleDocumento>();

var documentos = \_documentoRepository

.GetAll();

if (rdbValidar.Checked)

{

documentos = documentos

.Where(x => !x.Erro && x.DataValidacao == null && x.IdUsuarioEnvio == Logon.Id)

.Take(20);

}

else if (rdbValidados.Checked)

{

documentos = documentos

.Where(x => !x.Erro && x.DataValidacao != null && x.IdUsuarioEnvio == Logon.Id)

.Take(20);

}

else if (rdbBloqueados.Checked)

{

documentos = documentos

.Where(x => x.Erro && x.IdUsuarioEnvio == Logon.Id)

.Take(20);

}

else

{

documentos = documentos

.Where(x => x.IdUsuarioEnvio == Logon.Id)

.Take(20);

}

foreach (var item in documentos)

{

lista.Add(item);

}

dtgArquivos.DataSource = lista;

}

catch (Exception ex)

{

MessageBox.Show($"Erro ao pesquisar os arquivos. Erro: {ex.Message}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void BtnSair\_Click(object sender, EventArgs e)

{

Close();

}

private void BtnBaixar\_Click(object sender, EventArgs e)

{

try

{

var linha = dtgArquivos.SelectedRows[0].Cells;

var nomeArquivo = linha[1].Value.ToString();

var caminhoArquivo = linha[2].Value.ToString();

using (var folder = new FolderBrowserDialog())

{

var dialogResult = folder.ShowDialog(this);

if (dialogResult == DialogResult.OK)

{

var caminhoSelecionado = $"{folder.SelectedPath.ToString()}\\{DateTime.Now.ToString("yyyyMMddhhmmss")}-{nomeArquivo}";

File.Copy(caminhoArquivo, caminhoSelecionado);

MessageBox.Show($"Arquivo baixado no diretorio selecionado: {caminhoSelecionado}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

};

}

catch (Exception ex)

{

MessageBox.Show($"Erro ao baixar o arquivo. Erro: {ex.Message}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void RdbBloqueados\_Click(object sender, EventArgs e)

{

Pesquisar();

}

private void RdbTodos\_Click(object sender, EventArgs e)

{

Pesquisar();

}

private void RdbValidados\_Click(object sender, EventArgs e)

{

Pesquisar();

}

private void RdbValidar\_Click(object sender, EventArgs e)

{

Pesquisar();

}

}

public partial class CadastroAlunos : Form

{

private readonly AlunoRepository \_alunoRepository;

public CadastroAlunos()

{

InitializeComponent();

\_alunoRepository = new AlunoRepository(new Context());

if (\_alunoRepository == null) throw new ArgumentNullException($"Falha ao carregar componentes da tela. parametro: {nameof(\_alunoRepository)}");

}

private void CadastroAlunos\_Load(object sender, EventArgs e)

{

dtpDataNascimento.Value = DateTime.Now.Date;

cmbCursos.DataSource = new BindingList<object>

{

new {Id = 0, Descricao = "Selecione o tipo"},

new {Id = 1, Descricao = "Gestão de Ti"},

new {Id = 2, Descricao = "Analise Desenvolvimento Sistemas"},

new {Id = 3, Descricao = "Eventos"},

new {Id = 4, Descricao = "Logistica"},

new {Id = 5, Descricao = "Gestão Ambiental"}

};

cmbCursos.DisplayMember = "Descricao";

cmbCursos.ValueMember = "Id";

}

private void BtnCadastrar\_Click(object sender, EventArgs e)

{

try

{

if (!ValidarCampos()) throw new Exception("Campos Obrigatórios não preenchidos ou selecionados.");

var aluno = new Aluno

{

Nome = txtNome.Text,

RA = txtRa.Text,

Nascimento = dtpDataNascimento.Value,

Curso = (Cursos)cmbCursos.SelectedValue,

Matriculado = ckbMatriculado.Checked

};

\_alunoRepository.Add(aluno);

LimparTela();

MessageBox.Show

(

"Aluno Cadastrado Com Sucesso",

"Aviso Sistema",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

catch (Exception ex)

{

MessageBox.Show

(

$"Erro ao Cadastrar Aluno: {ex.Message}",

"Aviso Sistema",

MessageBoxButtons.OK,

MessageBoxIcon.Error

);

}

}

private void BtnSair\_Click(object sender, EventArgs e)

{

Close();

}

private void BtnLimpar\_Click(object sender, EventArgs e)

{

LimparTela();

}

private void LimparTela()

{

try

{

txtNome.Text = string.Empty;

txtRa.Text = string.Empty;

dtpDataNascimento.Value = DateTime.Now.Date;

cmbCursos.SelectedIndex = 0;

ckbMatriculado.Checked = false;

txtNome.Focus();

}

catch (Exception ex)

{

MessageBox.Show

(

$"Erro ao Limpar Tela: {ex.Message}",

"Aviso Sistema",

MessageBoxButtons.OK,

MessageBoxIcon.Error

);

}

}

private bool ValidarCampos()

{

return !string.IsNullOrWhiteSpace(txtNome.Text)

&& !string.IsNullOrWhiteSpace(txtRa.Text);

}

}

public partial class ConsultaAlunos : Form

{

private readonly AlunoRepository \_alunoRepository;

public ConsultaAlunos()

{

InitializeComponent();

\_alunoRepository = new AlunoRepository(new Context());

if (\_alunoRepository == null) throw new ArgumentNullException($"Erro ao inicializar sistema. parametro: {nameof(\_alunoRepository)}");

}

private void btnPesquisar\_Click(object sender, EventArgs e)

{

try

{

var listaAlunos = new List<Aluno>();

if (rdbNome.Checked)

{

listaAlunos = CarregarAlunoPorNome().ToList();

}

else

{

listaAlunos = CarregarAlunoPorRa().ToList();

}

var dataGridData = new BindingList<Aluno>(listaAlunos);

if (!dataGridData.Any())

{

MessageBox.Show(this, "Nenhum aluno encontrado.", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvAluno.Rows.Clear();

return;

}

dgvAluno.DataSource = dataGridData;

}

catch (Exception exception)

{

Console.WriteLine(exception);

throw;

}

}

private void btnSair\_Click(object sender, EventArgs e)

{

Close();

}

private IEnumerable<Aluno> CarregarAlunoPorNome()

{

return \_alunoRepository

.GetAll()

.Where(x => x.Nome.Contains(txtNome.Text));

}

private IEnumerable<Aluno> CarregarAlunoPorRa()

{

return \_alunoRepository

.GetAll()

.Where(x => x.RA.Contains(txtRa.Text)); throw new Exception();

}

private void BtnImprimir\_Click(object sender, EventArgs e)

{

using (var folder = new FolderBrowserDialog())

{

var dialogResult = folder.ShowDialog(this);

if (dialogResult == DialogResult.OK)

{

GerarRelatorio(folder.SelectedPath.ToString());

MessageBox.Show($"Arquivo baixado no diretorio selecionado: {folder.SelectedPath.ToString()}", "Aviso Sistema", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

};

}

private void GerarRelatorio(string caminho)

{

using (var document = new PdfDocument())

{

var page = document.AddPage();

var graphics = XGraphics.FromPdfPage(page);

var textFormatter = new XTextFormatter(graphics);

var font = new XFont("Calibri", 14);

textFormatter.Alignment = XParagraphAlignment.Center;

var listaAlunos = new List<Aluno>();

if (rdbNome.Checked)

{

listaAlunos = CarregarAlunoPorNome().ToList();

}

else

{

listaAlunos = CarregarAlunoPorRa().ToList();

}

var iterador = 2;

foreach (var item in listaAlunos)

{

textFormatter.DrawString($"nome: {item.Nome} Ra: {item.RA}, Data Nascimento: {item.Nascimento.ToString("dd/MM/yyyy")} Curso: {item.Curso.ToString()}", font, XBrushes.Black, new XRect(iterador \* 15, iterador \* 30, page.Width - iterador \* 30, page.Height - iterador \* 30));

iterador++;

}

var path = $"{caminho}\\relatorioAlunos-{DateTime.Now.ToString("yyyyMMddmmss")}.pdf";

document.Save(path);

Process.Start(path);

}

}

private void ConsultaAlunos\_Load(object sender, EventArgs e)

{

}

}

public partial class ManutencaoAluno : Form

{

private readonly AlunoRepository \_alunoRepository;

public ManutencaoAluno()

{

InitializeComponent();

\_alunoRepository = new AlunoRepository(new Context());

if (\_alunoRepository == null) throw new ArgumentNullException($"Falha ao carregar componentes da tela. parametro: {nameof(\_alunoRepository)}");

}

private void ManutencaoAluno\_Load(object sender, EventArgs e)

{

dtpDataNascimento.Value = DateTime.Now.Date;

var comboList = new BindingList<object>

{

new { Id = 0, Descricao = "Selecion o Curso" },

new { Id = 1, Descricao = "Gestão de TI" },

new { Id = 2, Descricao = "Análise e Desenv. de Sistemas" },

new { Id = 3, Descricao = "Eventos" },

new { Id = 4, Descricao = "Logistica" },

new { Id = 5, Descricao = "Gestão Ambiental" }

};

cmbCursos.DataSource = comboList;

cmbCursos.ValueMember = "Descricao";

}

private void BtnPesquisar\_Click(object sender, EventArgs e)

{

try

{

var aluno = \_alunoRepository.GetById(Convert.ToInt32(txtId.Text));

if (aluno == null) throw new Exception("Aluno não encontrado.");

txtNome.Text = aluno.Nome;

txtRa.Text = aluno.RA;

dtpDataNascimento.Value = aluno.Nascimento;

cmbCursos.SelectedIndex = (int)aluno.Curso;

ckbMatriculado.Checked = aluno.Matriculado;

}

catch (Exception ex)

{

MessageBox.Show

(

$"Erro ao Pesquisar Aluno: {ex.Message}",

"Aviso Sistema",

MessageBoxButtons.OK,

MessageBoxIcon.Error

);

}

}

private void LimparTela()

{

try

{

txtNome.Text = string.Empty;

txtRa.Text = string.Empty;

dtpDataNascimento.Value = DateTime.Now.Date;

cmbCursos.SelectedIndex = 0;

ckbMatriculado.Checked = false;

txtNome.Focus();

}

catch (Exception ex)

{

MessageBox.Show

(

$"Erro ao Limpar Tela: {ex.Message}",

"Aviso Sistema",

MessageBoxButtons.OK,

MessageBoxIcon.Error

);

}

}

private void BtnExcluir\_Click(object sender, EventArgs e)

{

try

{

if (\_alunoRepository.GetById(Convert.ToInt32(txtId.Text)) == null) throw new Exception("Aluno não encontrado");

\_alunoRepository.Remove(Convert.ToInt32(txtId.Text));

LimparTela();

}

catch (Exception ex)

{

MessageBox.Show

(

$"Erro ao Excluir Aluno: {ex.Message}",

"Aviso Sistema",

MessageBoxButtons.OK,

MessageBoxIcon.Error

);

}

}

private void BtnSair\_Click(object sender, EventArgs e)

{

\_alunoRepository.Dispose();

Close();

}

private void BtnCadastrar\_Click(object sender, EventArgs e)

{

try

{

var aluno = \_alunoRepository.GetById(Convert.ToInt32(txtId.Text));

aluno.Nome = txtNome.Text;

aluno.RA = txtRa.Text;

aluno.Nascimento = dtpDataNascimento.Value;

aluno.Curso = (Cursos)cmbCursos.SelectedIndex;

aluno.Matriculado = ckbMatriculado.Checked;

\_alunoRepository.SaveChanges();

LimparTela();

}

catch (Exception)

{

}

}

}