

## Faça como eu fiz

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) Para alterar ou excluir um cliente, primeiramente exiba-o. Logo, implemente o método `Buscar`, na classe `Fichario`:

```
public string Buscar(string Id)
{
    status = true;
    try
    {
        if (!(File.Exists(diretorio + "\\\" + Id + ".json")))
        {
            status = false;
            mensagem = "Identificador não existente: " + Id;
        }
        else
        {
            string conteudo = File.ReadAllText(diretorio + "\\\" + Id + ".json");
            status = true;
            mensagem = "Busca efetuada com sucesso. Identificador: " + Id;
            return conteudo;
        }
    }
    catch (Exception ex)
    {
        status = false;
        mensagem = "Erro ao buscar o conteúdo do identificador: " + ex.Message;
    }
    return "";
}
```

2) Implemente também os métodos `Apagar` e `Alterar`:

```
public void Apagar (string Id)
{
    status = true;
    try
    {
        if (!(File.Exists(diretorio + "\\\" + Id + ".json")))
        {
            status = false;
            mensagem = "Identificador não existente: " + Id;
        }
        else
        {
            File.Delete(diretorio + "\\\" + Id + ".json");
            status = true;
            mensagem = "Exclusão efetuada com sucesso. Identificador: " + Id;
        }
    }
    catch (Exception ex)
```

```

    {
        status = false;
        mensagem = "Erro ao buscar o conteúdo do identificador: " + ex.Message;
    }

    public void Alterar(string Id, string jsonUnit)
    {
        status = true;
        try
        {
            if (!(File.Exists(diretorio + "\\\" + Id + ".json")))
            {
                status = false;
                mensagem = "Alteração não permitida porque o identificador não existe: " + Id;
            }
            else
            {
                File.Delete(diretorio + "\\\" + Id + ".json");
                File.WriteAllText(diretorio + "\\\" + Id + ".json", jsonUnit);
                status = true;
                mensagem = "Inclusão efetuada com sucesso. Identificador: " + Id;
            }
        }
        catch (Exception ex)
        {
            status = false;
            mensagem = "Conexão com o Fichario com erro: " + ex.Message;
        }
    }
}

```

3) Agora, faça a chamada desses métodos, de acordo com o clique nos respectivos botões da barra de ferramentas. Volte para o modo Designer do formulário **Frm\_CadastroCliente\_UC**.

4) Inclua uma rotina que escreva o conteúdo de uma classe **Cliente** no formulário. Logo, implemente:

```

void EscreveFormulario(Cliente.Unit C)
{
    Txt_Codigo.Text = C.Id;
    Txt_NomeCliente.Text = C.Nome;
    Txt_NomeMae.Text = C.NomeMae;

    if (C.NaoTemPai == true)
    {
        Chk_TemPai.Checked = true;
        Txt_NomePai.Text = "";
    }
    else
    {
        Chk_TemPai.Checked = false;
        Txt_NomePai.Text = C.NomePai;
    }

    if (C.Genero == 0)
    {
        Rdb_Masculino.Checked = true;
    }
}

```

```

    }
    if (C.Genero == 1)
    {
        Rdb_Feminino.Checked = true;
    }
    if (C.Genero == 3)
    {
        Rdb_Indefinido.Checked = true;
    }

    Txt_CPF.Text = C.Cpf;
    Txt_CEP.Text = C.Cep;
    Txt_Logradouro.Text = C.Logradouro;
    Txt_Complemento.Text = C.Complemento;
    Txt_Cidade.Text = C.Cidade;
    Txt_Bairro.Text = C.Bairro;
    Txt_Telefone.Text = C.Telefone;
    Txt_Profissao.Text = C.Profissao;

    if (C.Estado == "")
    {
        Cmb_Estados.SelectedIndex = -1;
    }
    else
    {
        for (int i=0; i <= Cmb_Estados.Items.Count -1; i++)
        {
            if (C.Estado == Cmb_Estados.Items[i].ToString())
            {
                Cmb_Estados.SelectedIndex = i;
            }
        }
    }
}

Txt_RendaFamiliar.Text = C.RendaFamiliar.ToString();
}

```

5) No evento **Click** do botão **Abrir**, leia o **Id** do **TextBox** do formulário, e passe-o para o método **Buscar** , da classe **Fichario** . Isso retornará os dados do cliente que serão preenchidos no formulário:

```

private void abrirToolStripButton_Click(object sender, EventArgs e)
{
    if (Txt_Codigo.Text == "")
    {
        MessageBox.Show("Código do Cliente vazio.", "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        Fichario F = new Fichario("C:\\WindowsForms\\Curso\\CursoWindowsForms\\Fichario");
        if (F.status)
        {
            string clienteJson = F.Buscar(Txt_Codigo.Text);
            Cliente.Unit C = new Cliente.Unit();
            C = Cliente.DesSerializedClassUnit(clienteJson);
            EscreveFormulario(C);
        }
    }
}

```

```

    else
    {
        MessageBox.Show("ERR: " + F.mensagem, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
}

```

6) Com os dados exibidos na tela, você pode verificar se o cliente a ser excluído foi o cliente buscado. Assim, você pode clicar no botão **Excluir**, da barra de ferramentas, e executar o código abaixo, chamando o correspondente método da classe:

```

private void ApagatoolStripButton_Click(object sender, EventArgs e)
{
    if (Txt_Codigo.Text == "")
    {
        MessageBox.Show("Código do Cliente vazio.", "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        Fichario F = new Fichario("C:\\WindowsForms\\Curso\\CursoWindowsForms\\Fichario");
        if (F.status)
        {
            string clienteJson = F.Buscar(Txt_Codigo.Text);
            Cliente.Unit C = new Cliente.Unit();
            C = Cliente.DesSerializedClassUnit(clienteJson);
            EscreveFormulario(C);

            Frm_Questao Db = new Frm_Questao("icons8_question_mark_961", "Você quer excluir o cliente?");
            Db.ShowDialog();
            if (Db.DialogResult == DialogResult.Yes)
            {
                F.Apagar(Txt_Codigo.Text);
                if (F.status)
                {
                    MessageBox.Show("OK: " + F.mensagem, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    LimparFormulario();
                }
                else
                {
                    MessageBox.Show("ERR: " + F.mensagem, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }
        else
        {
            MessageBox.Show("ERR: " + F.mensagem, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

7) Para o botão **Alterar**, capture o conteúdo do formulário, transforme a classe em JSON e envie-a para o método **Alterar**, da classe **Fichario**:

```
private void salvarToolStripButton_Click(object sender, EventArgs e)
{
    if (Txt_Codigo.Text == "")
    {
        MessageBox.Show("Código do Cliente vazio.", "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        try
        {
            Cliente.Unit C = new Cliente.Unit();
            C = LeituraFormulario();
            C.ValidaClasse();
            C.ValidaComplemento();
            string clienteJson = Cliente.SerializedClassUnit(C);
            Fichario F = new Fichario("C:\\WindowsForms\\Curso\\Fichario");
            if (F.status)
            {
                F.Alterar(C.Id, clienteJson);
                if (F.status)
                {
                    MessageBox.Show("OK: " + F.mensagem, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
                else
                {
                    MessageBox.Show("ERR: " + F.mensagem, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            else
            {
                MessageBox.Show("ERR: " + F.mensagem, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        catch (ValidationException Ex)
        {
            MessageBox.Show(Ex.Message, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        catch (Exception Ex)
        {
            MessageBox.Show(Ex.Message, "ByteBank", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

8) Salve e teste a aplicação.