

Ficha 5 – Windows Forms (.NET)

Registo de Contactos

Objetivo:

Criar uma aplicação Windows Forms em .NET onde o utilizador possa registar contactos com as seguintes informações:

- Nome
- E-mail
- Telefone

A aplicação deverá permitir adicionar, editar, eliminar e listar os contactos.

Passos do exercício:

1. Configuração Inicial

- Crie um novo projeto do tipo Windows Forms App em .NET no Visual Studio.
 - Abra o Visual Studio.
 - Vá a **Ficheiro** -> **Novo** -> **Projeto**.
 - Escolha o modelo **Windows Forms App (.NET)** e nomeie o projeto como **RegistoDeContactos**.

2. Estrutura da Interface

- Adicione os seguintes controlos ao formulário principal (**Form1**):
 - **TextBox** para introduzir o nome.
 - **TextBox** para introduzir o e-mail.
 - **TextBox** para introduzir o telefone.
 - **Button** para adicionar o contacto.
 - **Button** para editar o contacto selecionado.
 - **Button** para eliminar o contacto.
 - **Button** para limpar os campos.
 - **ListBox** para exibir a lista de contactos registados.

Sugestão para o layout:

- Coloque os campos de texto e os botões no topo.
- Abaixo, adicione a **ListBox** para mostrar os contactos.

3. Modelagem da Classe de Contacto

- No projeto, crie uma classe chamada **Contacto** para representar os dados de cada contacto.

```
public class Contacto
{
    public string Nome { get; set; }
```

```

    public string Email { get; set; }
    public string Telefone { get; set; }

    public override string ToString()
    {
        return $"{Nome} - {Email} - {Telefone}";
    }
}

```

4. Código para adicionar contactos

- No evento de clique do botão `Adicionar`, crie um novo objeto da classe `Contacto`, preencha-o com os dados fornecidos nos campos `TextBox` e adicione-o à lista de contactos (`ListBox`).

```

private void btnAdicionar_Click(object sender, EventArgs e)
{
    Contacto novoContacto = new Contacto
    {
        Nome = txtNome.Text,
        Email = txtEmail.Text,
        Telefone = txtTelefone.Text
    };

    lstContactos.Items.Add(novoContacto);
    LimparCampos();
}

private void LimparCampos()
{
    txtNome.Text = "";
    txtEmail.Text = "";
    txtTelefone.Text = "";
}

```

5. Edição de contactos

- Ao clicar num contacto na `ListBox`, os campos `TextBox` devem ser preenchidos com as informações do contacto selecionado, permitindo a sua edição.

```

private void lstContactos_SelectedIndexChanged(object sender,
EventArgs e)
{
    if (lstContactos.SelectedItem != null)
    {
        Contacto contactoSelecioneado =
(Contacto)lstContactos.SelectedItem;
    }
}

```

```

        txtNome.Text = contactoSelecioneado.Nome;
        txtEmail.Text = contactoSelecioneado.Email;
        txtTelefone.Text = contactoSelecioneado.Telefone;
    }
}
private void btnEditar_Click(object sender, EventArgs e)
{
    if (lstContactos.SelectedItem != null)
    {
        Contacto contactoSelecioneado =
(Contacto)lstContactos.SelectedItem;
        contactoSelecioneado.Nome = txtNome.Text;
        contactoSelecioneado.Email = txtEmail.Text;
        contactoSelecioneado.Telefone = txtTelefone.Text;
        lstContactos.Items[lstContactos.SelectedIndex] =
contactoSelecioneado;
        LimparCampos();
    }
}

```

6. Eliminação de contactos

- No evento de clique do botão Eliminar, remova o contacto selecionado da ListBox.

```

private void btnEliminar_Click(object sender, EventArgs e)
{
    if (lstContactos.SelectedItem != null)
    {
        lstContactos.Items.Remove(lstContactos.SelectedItem);
        LimparCampos();
    }
}

```

7. Testar e Refatorar

- Execute a aplicação e teste as funcionalidades de adicionar, editar e eliminar contactos.
- Verifique se os campos de texto são geridos corretamente ao adicionar e modificar contactos.