

## Ficha de Trabalho 4

Nesta ficha de trabalho vamos criar uma pequena aplicação para uma pizzaria, a ser utilizada pelos funcionários que recebem encomendas por telefone.

Esta aplicação permite que os funcionários preencham os dados transmitidos pelos clientes, calculando o valor total a pagar, consoante as escolhas efetuadas pelos clientes.

Utilize o projeto disponibilizado pelo professor, o qual contém a classe `Encomenda` (dados de uma encomenda).

1. Insira o seguinte código no construtor da form para posicioná-la no centro do ecrã:

```
public FormPrincipal()
{
    InitializeComponent();

    // A janela deverá surgir no centro do ecrã
    this.StartPosition = FormStartPosition.CenterScreen;
}
```

2. Mude o tipo de letra da form para `Segoe UI` (propriedade `Font`).

3. Insira os controlos especificados na seguinte imagem (a azul está assinalado o nome do controlo) (não se esqueça de definir uma *access key* para a maior parte dos controlos):

4. Insira o seguinte código no evento `Load()` para inicializar os controlos:

```
private void Form1_Load(object sender, EventArgs e)
{
    this.Text = "Dados de Encomenda";

    // Inicializar as entradas
    Entradas.Items.Add("");
    Entradas.Items.Add("Pão de alho (2 unidades): €2,50");
    Entradas.Items.Add("Pão de alho supremo (2 unidades): €3,50");
    // Impedir que o utilizador possa editar o texto da combobox
    Entradas.DropDownStyle = ComboBoxStyle.DropDownList;
    // Quantidade das entradas
    QuantidadeEntrada.Minimum = 0;
    QuantidadeEntrada.Maximum = 4;
    QuantidadeEntrada.Enabled = false;

    // Inicializar as pizzas
    ListaPizzas.Items.Add("(selecione pizza)");
}
```

## Ficha de Trabalho 4

```

ListaPizzas.Items.Add("Pizza Margherita: €5,00");
ListaPizzas.Items.Add("Pizza Camponesa: €7,00");
ListaPizzas.Items.Add("Pizza Vegetariana: €6,80");
// Selecionar o 1º elemento da lista
ListaPizzas.SelectedIndex = 0;

ImagemPizza.BorderStyle = BorderStyle.FixedSingle;
ImagemPizza.SizeMode = PictureBoxSizeMode.StretchImage;

// Pré-selecionar entrega take away
EntregaTakeAway.Checked = true;
}

```

### 5. Defina a tab order dos controlos da form:

### 6. Uma característica distintiva desta aplicação é que à medida que o utilizador vai preenchendo os dados da encomenda, são atualizados os totais da encomenda (valor sem IVA e valor com IVA). Para tal, vamos criar um método chamado `AtualizarTotais()` o qual será chamado sempre que o utilizador modifica os dados da encomenda:

```
/// <summary>
/// Calcular e atualizar os totais com base na informação do formulário.
/// </summary>
private void AtualizarTotais()
{
    encomendaCliente.CalcularTotais();
    LabelTotalSemIva.Text = $"{encomendaCliente.TotalSemIva.ToString("C")}";
    LabelTotalComIva.Text = $"{encomendaCliente.TotalComIva.ToString("C")}";
}
```

Vamos agora inserir código em diferentes eventos para diferentes controlos, quando há mudança de informação ou quando é introduzida nova informação.

7. Vamos começar pela lista de entradas. Insira o seguinte código no evento `SelectedValueChanged()` do controlo `Entradas`:

```
/// <summary>
/// Mudança na entrada selecionada.
/// </summary>
private void Entradas_SelectedValueChanged(object sender, EventArgs e)
{
    switch (Entradas.SelectedItem.ToString())
    {
        case "":
            QuantidadeEntrada.Value = 0;
            QuantidadeEntrada.Enabled = false;
            break;

        case "Pão de alho (2 unidades): €2,50":
            encomendaCliente.Entrada = TipoEntrada.PaoAlho;
            encomendaCliente.QuantidadeEntrada = 1;
            QuantidadeEntrada.Value = 1;
            QuantidadeEntrada.Enabled = true;
            break;

        case "Pão de alho supremo (2 unidades): €3,50":
            encomendaCliente.Entrada = TipoEntrada.PaoAlhoSupremo;
            encomendaCliente.QuantidadeEntrada = 1;
            QuantidadeEntrada.Value = 1;
            QuantidadeEntrada.Enabled = true;
            break;
    }

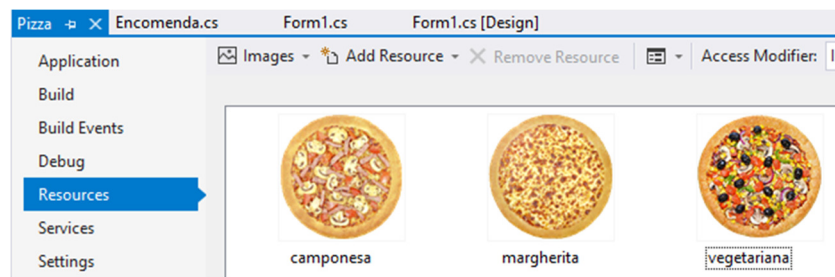
    AtualizarTotais();
}
```

8. De seguida, insira o seguinte código no evento `ValueChanged()` do controlo `QuantidadeEntrada` quando o utilizador modifica a quantidade de entradas:

```
/// <summary>
/// Mudança na quantidade de entradas selecionadas.
/// </summary>
private void QuantidadeEntrada_ValueChanged(object sender, EventArgs e)
{
    encomendaCliente.QuantidadeEntrada = (int)QuantidadeEntrada.Value;
    AtualizarTotais();
}
```

## 9. Adicione as imagens das pizzas aos *resources* (recursos) do projeto:

- clique em **Project** → **Pizza Properties...**
- nas várias opções que surgem, clique em **Resources**
- clique em **Add Resource** → **Add Existing File...**
- selecione os ficheiros **camponesa.png**, **margherita.png** e **vegetariana.png** (terá de adicionar os ficheiros, um de cada vez)



Grave o projeto, antes de prosseguir e feche os *resources*.

## 10. Vamos tratar do evento `SelectedValueChanged()` no controlo `ListaPizzas`. Este evento ocorre quando o utilizador clica num elemento da lista. No caso desta aplicação, tem como consequência uma mudança na imagem que é mostrada, bem como uma atualização dos totais da encomenda:

```
/// <summary>
/// Selecionar uma pizza.
/// </summary>
private void ListaPizzas_SelectedValueChanged(object sender, EventArgs e)
{
    // Determinar qual o item da lista que foi selecionado pelo utilizador
    switch (ListaPizzas.SelectedItem)
    {
        case "(selecione pizza)":
            encomendaCliente.Pizza = TipoPizza.Nulo;
            ImagemPizza.Image = null;
            break;

        case "Pizza Margherita: €5,00":
            encomendaCliente.Pizza = TipoPizza.Margherita;
            ImagemPizza.Image = Pizza.Properties.Resources.margherita;
            break;

        case "Pizza Camponesa: €7,00":
            encomendaCliente.Pizza = TipoPizza.Camponesa;
            ImagemPizza.Image = Pizza.Properties.Resources.camponesa;
            break;

        case "Pizza Vegetariana: €6,80":
            encomendaCliente.Pizza = TipoPizza.Vegetariana;
            ImagemPizza.Image = Pizza.Properties.Resources.vegetariana;
            break;
    }

    AtualizarTotais();
}
```

11. O utilizador pode selecionar zero ou várias bebidas. Para tal dispõe de vários controlos do tipo `CheckBox`. Vamos inserir código no evento `CheckedChanged()` das várias checkboxes:

```
/// <summary>
/// Selecionar (ou não) água.
/// </summary>
private void BebidaAgua_CheckedChanged(object sender, EventArgs e)
{
    encomendaCliente.BebidaAgua = BebidaAgua.Checked;
    AtualizarTotais();
}

/// <summary>
/// Selecionar (ou não) Coca-Cola.
/// </summary>
private void BebidaCocaCola_CheckedChanged(object sender, EventArgs e)
{
    encomendaCliente.BebidaCocaCola = BebidaCocaCola.Checked;
    AtualizarTotais();
}

/// <summary>
/// Selecionar (ou não) sumo sem gás.
/// </summary>
private void BebidaSumoSemGas_CheckedChanged(object sender, EventArgs e)
{
    encomendaCliente.BebidaSumoSemGas = BebidaSumoSemGas.Checked;
    AtualizarTotais();
}

/// <summary>
/// Selecionar (ou não) sumo com gás.
/// </summary>
private void BebidaSumoComGas_CheckedChanged(object sender, EventArgs e)
{
    encomendaCliente.BebidaSumoComGas = BebidaSumoComGas.Checked;
    AtualizarTotais();
}
```

12. A última informação da encomenda diz respeito ao método de entrega, podendo o utilizador efetuar a sua escolha utilizando dois controlos `RadioButton`. Insira o seguinte código nos eventos `CheckedChanged()` dos controlos `EntregaTakeAway` e `EntregaCasa`:

```
/// <summary>
/// Tipo de entrega: entrega take away.
/// </summary>
private void EntregaTakeAway_CheckedChanged(object sender, EventArgs e)
{
    if (EntregaTakeAway.Checked)
    {
        encomendaCliente.Entrega = MetodoEntrega.TakeAway;
        AtualizarTotais();
    }
}

/// <summary>
/// Tipo de entrega: entrega em casa.
```

```

/// </summary>
private void EntregaCasa_CheckedChanged(object sender, EventArgs e)
{
    if (EntregaCasa.Checked)
    {
        encomendaCliente.Entrega = MetodoEntrega.Casa;
        AtualizarTotais();
    }
}

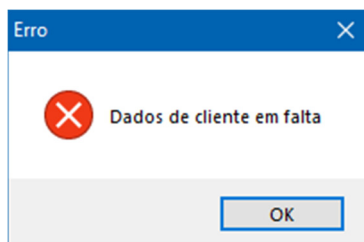
```

Vamos agora tratar dos 3 botões da form:

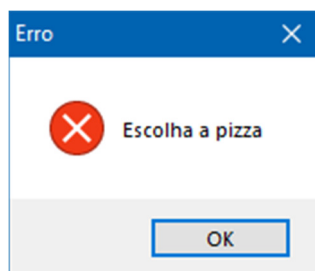
- Validar (**BotaoValidar**): valida os dados do formulário
- Anular (**BotaoAnular**): anula os dados inseridos (ou seja, limpa os campos)
- Fechar (**BotaoFechar**): fecha a aplicação

13. Quando o utilizador clica no botão **BotaoValidar**, deverão ser efetuadas duas validações:

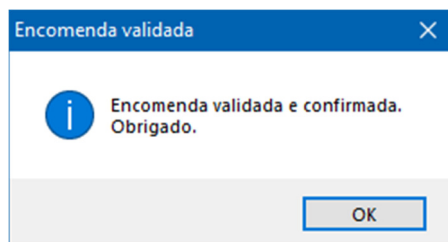
- 13.1. Verificar se o utilizador preencheu os dados pessoais do cliente (nome, morada e contacto). Caso não tenha preenchido todos os campos, deverá surgir a seguinte janela:



- 13.2. Verificar se o utilizador escolheu uma pizza. Caso não o tenha feito deverá surgir a seguinte janela:



- 13.3. Se os dados estiverem válidos deverá surgir a seguinte janela:





Insira e complete o seguinte código:

```
/// <summary>
/// Validar a encomenda.
/// </summary>
private void BotaoValidar_Click(object sender, EventArgs e)
{
    bool encomendaValida = true;

    if Completar: verificar se os campos pessoais estão preenchidos
    {
        Completar: mostrar mensagem

        encomendaValida = false;
    }

    if (ListaPizzas.SelectedIndex == 0)
    {
        Completar: mostrar mensagem

        encomendaValida = false;
    }

    if (encomendaValida)
    {
        Completar: mostrar mensagem

        // Efetuar reset ao formulário: iniciar nova encomenda
        LimparCampos();
    }
}
```

14. Copie e complete o seguinte código, para o método `LimparCampos()`, o qual limpa todos os campos:

```
/// <summary>
/// Limpar os dados do formulário.
/// </summary>
private void LimparCampos()
{
    // Dados do cliente
    Nome.Completar = Completar;
    Morada.Completar = Completar;
    Contacto.Completar = Completar;

    // Entradas
    Entradas.SelectedIndex = Completar;
    QuantidadeEntrada.Value = Completar;

    // Pizza
    ListaPizzas.SelectedIndex = Completar;

    // Bebidas
    BebidaAgua.Checked = Completar;
    BebidaCocaCola.Checked = Completar;
    BebidaSumoSemGas.Checked = Completar;
}
```

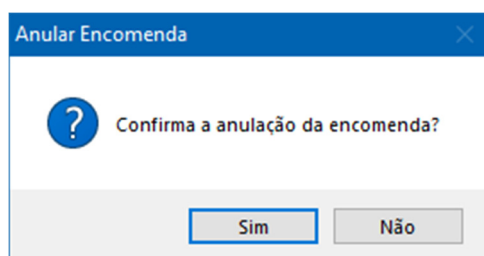


```
BebidaSumoComGas.Checked = Completar;

// Tipo de entrega
EntregaTakeAway.Checked = Completar;
}
```

Dica: encontra algum do código no evento `Load()`

15. O botão `BotaoAnular`, permite ao utilizador cancelar o preenchimento da encomenda atual e limpar todos os campos, sendo primeiro pedida uma confirmação:



Copie e complete o seguinte código:

```
/// <summary>
/// Anular a encomenda.
/// </summary>
private void BotaoAnular_Click(object sender, EventArgs e)
{
    // Confirmar que o utilizador pretende anular a encomenda
    if (Completar: mostrar mensagem == DialogResult.Completar)
    {
        Completar: limpar os campos

        Completar: atualizar os totais
    }
}
```

16. E para finalizar, o botão `BotaoFechar`: este botão deverá fechar a aplicação. Insira o código apropriado no evento `Click()` do controlo `BotaoFechar`.