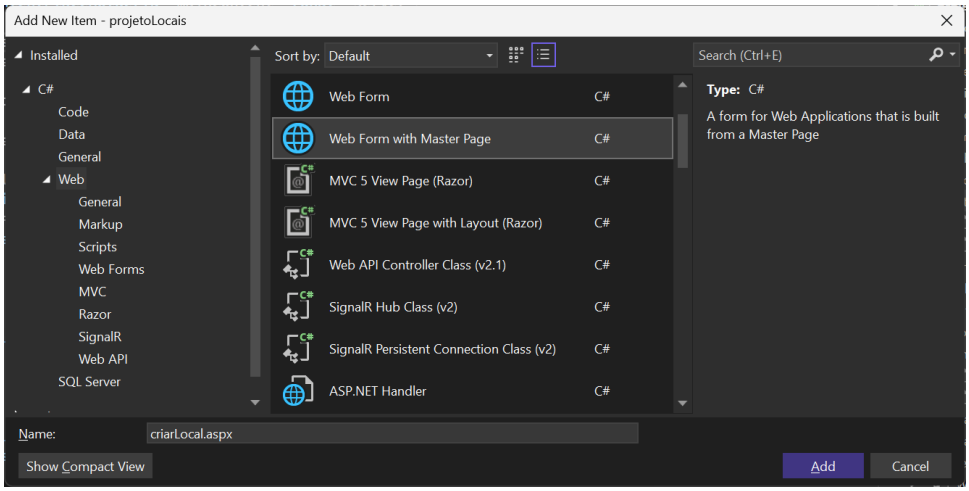
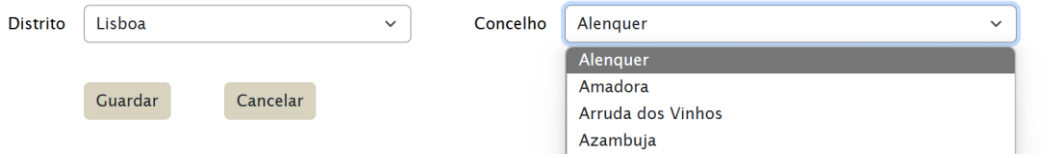
Projeto – Aldeias e locais históricos de Portugal

Edição de dados de um local.

Adicione à pasta **utilizador** um Web Form com o nome **editarLocal.aspx**;



Adicione ao formulário **editarLocal.aspx** controlos que permitam definir o nome, a descrição, a morada (se aplicável), a localidade e o distrito do local; a latitude e longitude poderão ser obtidas a partir da API https://positionstack.com/; considere ainda a possibilidade de gerir as fotos do local; será necessário um controlo do tipo **FileUpload** que permita copiar para o servidor o ficheiro selecionado; na tabela **Foto** deverá ser guardado o nome do ficheiro e uma legenda (opcional); considere a utilização de um DataList que permita ver as fotos associadas ao local e, se necessário, a modificação da legenda; para a definição do distrito e concelho considere a utilização de controlos DropDownList, a preencher com a informação das respetivas tabelas da base de dados; nota – a lista de concelhos deverá ser preenchida em função do distrito selecionado;



Nota – em termos de controlos, o formulário **editarLocal.aspx** é semelhante ao formulário **criarLocal.aspx**;

Crie um procedimento que permita efetuar o preenchimento do controlo **Distrito**;

```
void ObterDistritos() {

    using (SqlCommand commandDistritos = new SqlCommand
       ("SELECT Id, Nome FROM Distrito", connection))
    {
        connection.Open();
        SqlDataReader readerDistritos = commandDistritos.ExecuteReader();
        DataTable table = new DataTable();
        table.Load(readerDistritos);

        listDistrito.DataSource = table;
        listDistrito.DataTextField = "Nome";
        listDistrito.DataValueField = "Id";
        listDistrito.DataBind();

        connection.Close();
    }
}
```

Crie um procedimento que permita efetuar o preenchimento do controlo **Concelho** (em função do distrito selecionado);

```
void ObterConcelhos(int distrito)
{
    using (SqlCommand commandConcelhos = new SqlCommand
        ("SELECT Id, Nome FROM Concelho WHERE Distrito = @distrito", connection))
    {
        commandConcelhos.Parameters.AddWithValue("@distrito", distrito);
        connection.Open();
        SqlDataReader reader = commandConcelhos.ExecuteReader();
        DataTable table = new DataTable();
        table.Load(reader);

        listConcelho.DataSource = table;
        listConcelho.DataTextField = "Nome";
        listConcelho.DataValueField = "Id";
        listConcelho.DataBind();
        connection.Close();
    }
}
```

No evento **SelectIndexChanged** do controlo **Distrito** deverá ser feito o preenchimento do controlo Concelho, com os concelhos referentes ao distrito selecionado;

```
protected void listDistrito_SelectedIndexChanged(object sender, EventArgs e)
{
    int selectedDistrito = int.Parse(listDistrito.SelectedValue);
    ObterConcelhos(selectedDistrito);
}
```

As fotos referentes ao local deverão ser mostradas num controlo DataList; crie um procedimento que permita obter a lista de fotos a partir da base de dados e preencha o DataList;

```
void GetFotosLocal(string local)
{
    SqlCommand command = new SqlCommand();
    command.Connection = connection;
    command.CommandText =
        "SELECT Id, Ficheiro, Legenda FROM Foto WHERE Local = @local";

    //ID do local
    command.Parameters.AddWithValue("@local", local);

    connection.Open();
    SqlDataReader reader = command.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    reader.Close();
    connection.Close();

    listFotos.DataSource = table;
    listFotos.DataBind();
}
```

No evento **Load**:

1 – efetue o preenchimento dos controlos referentes ao nome, morada, localidade e descrição do local

```
protected void Page_Load(object sender, EventArgs e)
{
   if (!IsPostBack)
   {
       //1 - ler dados do local
       SqlCommand command = new SqlCommand();
       command.Connection = connection;
       command.CommandText = "  SELECT L.Nome, L.Morada, L.Localidade, L.Descricao, " +
            "CAST(L.Concelho AS NVARCHAR), CAST(C.Distrito AS NVARCHAR) " +
            "FROM Local L JOIN Concelho C ON L.Concelho = C.Id WHERE L.Id = @local";
       command.Parameters.AddWithValue("@local", Session["local"]);

       connection.Open();
       SqlDataReader reader = command.ExecuteReader();

       //variáveis a utilizar na seleção do concelho e respetivo distrito
       string idDistrito = "", idConcelho = "";

       while (reader.Read()) {

            nomeLocal.Text = reader.GetString(0);

            textNome.Text = reader.GetString(0);
            textMorada.Text = reader.GetValue(1)?.ToString() ?? string.Empty;

            textLocalidade.Text = reader.GetString(2);
            textDescricao.Text = reader.GetString(3);

            idConcelho = reader.GetString(4).ToString();
            idDistrito = reader.GetString(5).ToString();

       }
       reader.Close();
       connection.Close();
```

2 – efetue o preenchimento do controlo distrito e selecione o distrito referente ao local;

```
//2 - carregar distritos

ObterDistritos();

//selecionar o distrito
if (listDistrito.Items.FindByValue(idDistrito) != null)
{
   listDistrito.SelectedValue = idDistrito;
   listDistrito_SelectedIndexChanged(null, null); // atualiza os concelhos
}
```

3 – selecionar o concelho;

```
//3 - selecionar o concelho
if (listConcelho.Items.FindByValue(idConcelho) != null)
{
   listConcelho.SelectedValue = idConcelho;
}
```

4 – carregar as fotos do local;

```
//4 - carregar as fotos
GetFotosLocal(Session["local"].ToString());
```

O controlo **Guardar** deverá atualizar na tabela **Local** o nome, descrição, morada (se aplicável), localidade e o concelho do local (opcionalmente, a latitude e a longitude); comece por criar o Stored Procedure que fará o UPDATE na tabela:

```
CREATE PROCEDURE LocalEditar
	@id 			INT,
	@nome 		NVARCHAR(300),
	@descricao 		NVARCHAR(max),
	@morada 		NVARCHAR(350),
	@localidade 		NVARCHAR(350),
	@concelho 		INT,
	@latitude 		VARCHAR(20),
	@longitude 		VARCHAR(20)
)
AS
	UPDATE Local SET
		Nome = @nome,
		Descricao = @descricao,
		Morada = @morada,
		Localidade = @localidade,
		Concelho = @concelho,
		Latitude = @latitude,
		Longitude = @longitude
	WHERE
		Id = @id
```

No evento Click do controlo **Guardar** execute o Stored Procedure **LocalEditar** (definindo os valores para parâmetros do procedimento);

```
SqlCommand command = new SqlCommand();
command.Connection = connection;
//definição do Stored Procedure que cria o registo na base de dados
command.CommandText = "LocalEditar";
command.CommandType = CommandType.StoredProcedure;
//definição dos valores a alterar na tabela
command.Parameters.AddWithValue("@id", Session["local"]);

command.Parameters.AddWithValue("@nome", textNome.Text);
command.Parameters.AddWithValue("@descricao", textDescricao.Text);
if (textMorada.Text == "")
    command.Parameters.AddWithValue("@morada", DBNull.Value);
else
    command.Parameters.AddWithValue("@morada", textMorada.Text);
command.Parameters.AddWithValue("@localidade", textLocalidade.Text);
command.Parameters.AddWithValue("@concelho", listConcelho.SelectedValue);

command.Parameters.AddWithValue("@latitude", DBNull.Value);
command.Parameters.AddWithValue("@longitude", DBNull.Value);

connection.Open();
command.ExecuteNonQuery();
connection.Close();
```

O controlo **Guardar foto** deverá fazer o upload do ficheiro selecionado para a pasta **imagens** e colocar, com a legenda, a informação na tabela **Foto**; considere a utilização do procedimento **LocalFotoCriar** existente na base de dados; a ação **Guardar foto** só poderá acontecer se o utilizador definiu um ficheiro e se o ficheiro selecionado for do tipo jpg, jpeg, png, gif ou tiff; se estas condições não se verificarem, o upload não deve acontecer;

```
//ficheiro - nome do controlo FileUpload
if (ficheiro.HasFile)
{
    SqlCommand command = new SqlCommand();
    command.Connection = connection;
    command.CommandText = "LocalFotoCriar";
    command.CommandType = CommandType.StoredProcedure;

    command.Parameters.AddWithValue("@local", ViewState["idLocal"]);
    command.Parameters.AddWithValue("@legenda", textLegenda.Text);

    //tipos de ficheiros admitidos
    string[] ext = { ".jpg", ".jpeg", ".png", ".gif", ".tiff" };

    bool ok = false;
    //obter extensão do ficheiro
    string extensao = System.IO.Path.GetExtension(ficheiro.FileName).ToString();
    //verificar se a extensão se encontra no Array de ficheiros admitidos
    foreach (string item in ext)
        if (extensao == item)
            ok = true;
    //se o tipo de ficheiro está correto
    if (ok)
    {
        //gerar Guid para evitar nomes repetidos
        Guid g = Guid.NewGuid();
        string fileName = $"{g}-{ficheiro.FileName}";
        ficheiro.SaveAs(Server.MapPath("../imagens/") + fileName);

        //definir parâmetro
        command.Parameters.AddWithValue("@ficheiro", "imagens/" + fileName);

        //tipo de ficheiro correto - colocar informação na base de dados
        connection.Open();
        command.ExecuteNonQuery();
        connection.Close();
        //atualizar DataList referente às fotos
        GetFotosLocal(Session["local"].ToString());

    }
    else
    {
        //ficheiro inválido - cancelar execução do procedimento
        Response.Write("<script>alert('Selecione um ficheiro do tipo \".jpg\", \".jpeg\", "
             + "\".png\", \".gif\" ou \".tiff.');</script>");
        return;
    }

}
else
{
    //não foi selecionado um ficheiro - cancelar execução do procedimento
    Response.Write("<script>alert('Selecione um ficheiro do tipo \".jpg\", \".jpeg\", "
       + "\".png\", \".gif\" ou \".tiff.');</script>");
    return;
}
```

A opção **Selecionar** permite selecionar uma foto do local; coloca em ViewState o **ID** da foto e no controlo **Legenda** a frase associada à foto;


```
protected void lnkDetalhes_Command(object sender,
   System.Web.UI.WebControls.CommandEventArgs e)
{
   if (e.CommandArgument != null)
   {
      //colocar em ViewState o Id da foto selecionada
      ViewState["idFoto"] = e.CommandArgument.ToString();

      //selecionar legenda da foto selecionada
      SqlCommand commandFoto = new SqlCommand();
      commandFoto.Connection = connection;
      commandFoto.CommandText = "SELECT Legenda FROM Foto WHERE Id = @id";

      commandFoto.Parameters.AddWithValue("@id", ViewState["idFoto"]);

      connection.Open();
      SqlDataReader reader = commandFoto.ExecuteReader();
      while (reader.Read())
      {
         textLegenda.Text = reader[0].ToString();
      }
      reader.Close();
   }
}
```

O controlo **Editar legenda** deve editar o valor da legenda da foto selecionada; depois de alterada a legenda na base de dados, deve ser atualizada a DataList;

```
protected void buttonEditarLegenda_Click(object sender, EventArgs e)
{
    SqlCommand commandFoto = new SqlCommand();
    commandFoto.Connection = connection;
    commandFoto.CommandText = "UPDATE Foto SET Legenda = @legenda WHERE Id = @id";
    commandFoto.CommandType = CommandType.Text;
    //ID da foto definido quando a foto é selecionada
    commandFoto.Parameters.AddWithValue("@id", ViewState["idFoto"]);
    commandFoto.Parameters.AddWithValue("@legenda", textLegenda.Text);
    connection.Open();
    commandFoto.ExecuteNonQuery();
    connection.Close();

    textLegenda.Text = string.Empty;

    //atualizar DataList fotos do local
    GetFotosLocal(Session["local"].ToString());
}
```

O controlo **Eliminar foto** deve eliminar a foto selecionada; é necessário remover a informação da tabela **Foto** e eliminar o ficheiro; depois de eliminada a foto, deve ser atualizada a DataList;

```
protected void buttonEliminarFoto_Click(object sender, EventArgs e)
{
    //eliminar ficheiro
    SqlCommand commandFoto = new SqlCommand();
    commandFoto.Connection = connection;
    commandFoto.CommandText = "SELECT Ficheiro FROM Foto WHERE Id = @id";

    //ID da foto definido quando a foto é selecionada
    commandFoto.Parameters.AddWithValue("@id", ViewState["idFoto"]);

    connection.Open();
    //obter nome do ficheiro a eliminar
    SqlDataReader reader = commandFoto.ExecuteReader();
    while (reader.Read())
    {
       string ficheiro = Server.MapPath("../" + reader[0].ToString());
       //eliminar ficheiro
       if (File.Exists(ficheiro))
           File.Delete(ficheiro);
    }
    reader.Close();

    //eliminar dados na tabela
    commandFoto.Parameters.Clear();

    commandFoto.CommandText = "DELETE Foto WHERE Id = @id";
    commandFoto.CommandType = CommandType.Text;
    commandFoto.Parameters.AddWithValue("@id", ViewState["idFoto"]);
    commandFoto.ExecuteNonQuery();

    connection.Close();

    textLegenda.Text = string.Empty;
    //atualizar DataList fotos do local
    GetFotosLocal(Session["local"].ToString());
}
```

O controlo **Cancelar** (referente à definição de fotos do local) deve limpar os controlos referentes ao ficheiro e respetiva legenda;

```
protected void buttonCancelarFoto_Click(object sender, EventArgs e)
{
    textLegenda.Text = string.Empty;
    //o ficheiro selecionado é removido –
    //o valor do FileUpload não é mantido em ViewState
}
```

O controlo **Cancelar** (referente ao local) deve reencaminhar o utilizador para o Web Form **areaPessoal.aspx**;

```
protected void buttonCancelar_Click(object sender, EventArgs e)
{
    Response.Redirect("areaPessoal.aspx");
}
```