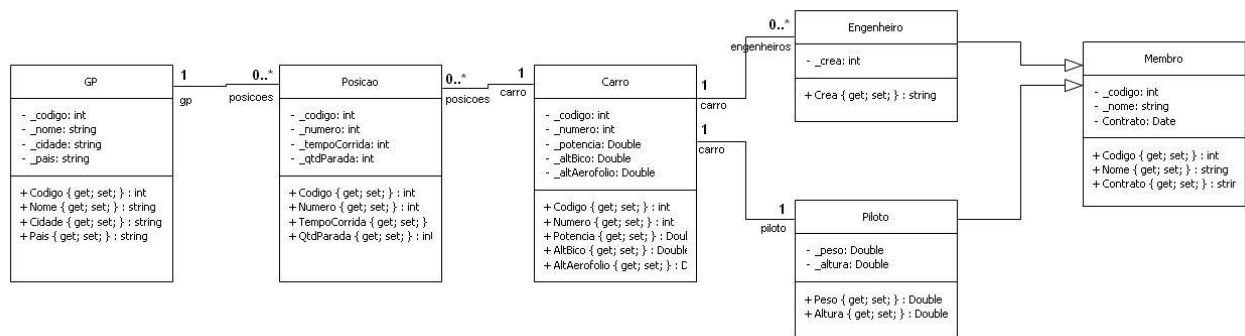


Associação 1:1

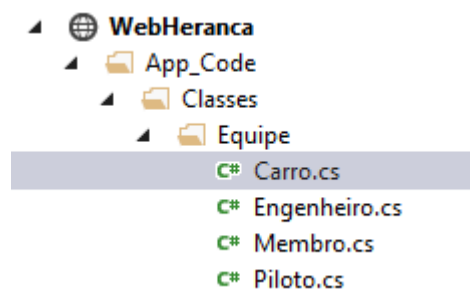
Abrir o Visual Studio

Abrir o projeto WebHeranca

Baseado no Diagrama de Classe a seguir, iremos criar as classes de Modelagem que compõem a Associação de Piloto com Carro.



Para isso, será criada a classe Carro, dentro de Classes/Equipe



- Defina o namespace
- Crie as propriedades Codigo, Numero, Potencia, AlturaBico, AlturaAerofolio.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebHeranca.Classes.Equipe
{
    /// <summary>
    /// Summary description for Carro
    /// </summary>
    public class Carro
    {
        public int Codigo { get; set; }
        public int Numero { get; set; }
    }
}
```

```

        public double Potencia { get; set; }
        public double AlturaBico { get; set; }
        public double AlturaAerofolio { get; set; }

        public Carro()
        {
            //
            // TODO: Add constructor logic here
            //
        }
    }
}

```

Definindo a associação entre as classes.

Adicione na classe Carro o relacionamento para Piloto. Nesse caso, todo Carro necessita de um Piloto.

```

namespace WebHeranca.Classes.Equipe
{
    /// <summary>
    /// Summary description for Carro
    /// </summary>
    public class Carro
    {
        public int Codigo { get; set; }
        public int Numero { get; set; }
        public double Potencia { get; set; }
        public double AlturaBico { get; set; }
        public double AlturaAerofolio { get; set; }

        //relacionamentos
        public Piloto Piloto { get; set; }

        public Carro()
        {
            //
            // TODO: Add constructor logic here
            //
        }
    }
}

```

Adicione na classe Piloto, o relacionamento para Carro.

```

namespace WebHeranca.Classes.Equipe
{
    /// <summary>
    /// Summary description for Piloto
    /// </summary>
    public class Piloto:Membro
    {
        public double Peso { get; set; }
        public double Altura { get; set; }

        //relacionamentos
        public Carro Carro { get; set; }

        public Piloto()
        {
            //

```

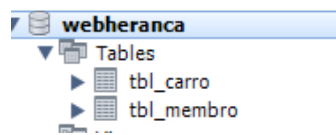
```

        // TODO: Add constructor logic here
        //
    }
}
}

```

Alterando o Banco de Dados

- No MySQL Query Browser ou MySQL Workbench, abra o schema (database) webheranca
- Crie a tabela tbl_carro
 - A regra para criação da tabela para Associação 1:1 é:
 - Cada classe da associação 1:1, deve estar numa tabela.
 - Em qualquer uma das tabelas, deve ficar a FK.
 - Nesse exemplo, preferi colocar a FK (mem_codigo) na tabela tbl_carro.



Observe a tabela

The screenshot shows the 'tbl_carro' table structure in MySQL Workbench. The table is located in the 'webheranca' schema and uses the 'InnoDB' engine with 'utf8 - default collation'. The table has six columns: 'car_codigo', 'car_numero', 'car_potencia', 'car_alturabico', 'car_alturaaerofolio', and 'mem_codigo'. The 'car_codigo' column is the primary key. The 'mem_codigo' column is a foreign key.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
car_codigo	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
car_numero	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
car_potencia	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
car_alturabico	DECIMAL(5,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
car_alturaaerofolio	DECIMAL(5,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
mem_codigo	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Criando a classe que terá os métodos de acesso ao banco de dados relacionados a classe Carro

- Botão direito na pasta Persistencia
- Adicione Nova Classe
- Nome da classe: CarroBD.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using FATEC; //para acesso a classe Mapped
using WebHeranca.Classes.Equipe; //para acesso a classe de modelagem
using System.Data; // para acesso ao DataSet

```

```

namespace WebHeranca.Persistencia.Equipe
{
    /// <summary>
    /// Summary description for CarroBD
    /// </summary>
    public class CarroBD
    {

```

```

//métodos
//insert
public int Insert(Carro carro)
{
    int retorno = 0;

    try
    {
        System.Data.IDbConnection objConexao;
        System.Data.IDbCommand objCommand;
        string sql = "INSERT INTO tbl_carro(car_numero, car_potencia, car_alturabico,
car_alturaaerofolio, mem_codigo) VALUES (?numero, ?potencia, ?bico, ?aerofolio, ?piloto)";

        objConexao = Mapped.Connection();
        objCommand = Mapped.Command(sql, objConexao);

        objCommand.Parameters.Add(Mapped.Parameter("?numero", carro.Numero));
        objCommand.Parameters.Add(Mapped.Parameter("?potencia", carro.Potencia));
        objCommand.Parameters.Add(Mapped.Parameter("?bico", carro.AlturaBico));
        objCommand.Parameters.Add(Mapped.Parameter("?aerofolio",
carro.AlturaAerofolio));
        objCommand.Parameters.Add(Mapped.Parameter("?piloto", carro.Piloto.Codigo));

        objCommand.ExecuteNonQuery();
        objConexao.Close();
        objCommand.Dispose();
        objConexao.Dispose();
    }
    catch (MySql.Data.MySqlClient.MySqlException)
    {
        retorno = -1;
    }
    catch (Exception)
    {
        retorno = -2;
    }

    return retorno;
}

//select all

//select

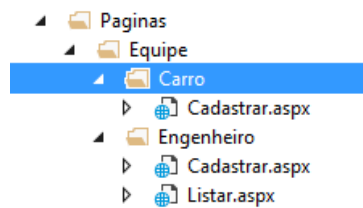
//update

//delete

//construtor
public CarroBD()
{
    //
    // TODO: Add constructor logic here
    //
}
}
}

```

Adicione uma pasta dentro de Paginas/Equipe, chamada Carro. Dentro dela, adicione um WebForm chamado Cadastrar.aspx.



Crie a tela como:

Cadastro de Carro

Piloto:

Número:

Potência:

Altura do Bico:

Altura do Aerofólio:

[blMensagem]

Antes de trabalhar no código dessa tela, é possível perceber que o Piloto necessita ser selecionado. Para isso, crie a classe PilotoBD em Persistencia/Equipe (caso você ainda não tenha criado).

- Defina o namespace
- Adicione os namespaces que serão usados
- Adicione o método SelectAll
- Adicione o método Select

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using FATEC; //para acesso a classe Mapped
using WebHeranca.Classes.Equipe; //para acesso a classe de modelagem
using System.Data; // para acesso ao DataSet
using WebHeranca.Util; //para acesso ao enum
```

```
/// <summary>
/// Summary description for PilotoBD
/// </summary>
public class PilotoBD
{
    //métodos
    //insert

    //selectall
    public DataSet SelectAll()
    {
        DataSet ds = new DataSet();
```

```

System.Data.IDbConnection objConexao;
System.Data.IDbCommand objCommand;
System.Data.IDataAdapter objDataAdapter;

objConexao = Mapped.Connection();
objCommand = Mapped.Command("SELECT * FROM tbl_membro WHERE mem_tipo = ?tipo",
objConexao);
objCommand.Parameters.Add(Mapped.Parameter("?tipo", MEMBRO.PILOTO));

objDataAdapter = Mapped.Adapter(objCommand);
objDataAdapter.Fill(ds);

objConexao.Close();

objCommand.Dispose();
objConexao.Dispose();

return ds;
}

//select
public Piloto Select(int id)
{
    Piloto obj = null;

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    System.Data.IDataReader objDataReader;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command("SELECT * FROM tbl_membro WHERE mem_codigo = ?codigo",
objConexao);
    objCommand.Parameters.Add(Mapped.Parameter("?codigo", id));

    objDataReader = objCommand.ExecuteReader();
    while (objDataReader.Read())
    {
        obj = new Piloto();
        obj.Codigo = Convert.ToInt32(objDataReader["mem_codigo"]);
        obj.Nome = Convert.ToString(objDataReader["mem_nome"]);
        obj.Contrato = Convert.ToString(objDataReader["mem_contrato"]);
        obj.Peso = Convert.ToDouble(objDataReader["mem_peso"]);
        obj.Altura = Convert.ToDouble(objDataReader["mem_altura"]);
    }

    objDataReader.Close();
    objConexao.Close();

    objCommand.Dispose();
    objConexao.Dispose();
    objDataReader.Dispose();

    return obj;
}

//update

//delete

//construtor
public PilotoBD()
{

```

```

        //
        // TODO: Add constructor logic here
        //
    }
}

```

De volta a tela Paginas/Equipe/Carro/Cadastrar.aspx

Acesse o código fonte da tela (Cadastrar.aspx.cs)

Importe os namespaces que serão usados. Coloque-os no começo do código:

```

using WebHeranca.Classes.Equipe;
using WebHeranca.Persistencia.Equipe;
using System.Data;

```

Crie um método para Carregar todos os pilotos. Este método será usado no Page_Load da página.

```

private void CarregaPilotos()
{
    PilotoBD bd = new PilotoBD();
    DataSet ds = bd.SelectAll();
    ddlPiloto.DataSource = ds.Tables[0].DefaultView;
    ddlPiloto.DataTextField = "mem_nome";
    ddlPiloto.DataValueField = "mem_codigo";
    ddlPiloto.DataBind();
    ddlPiloto.Items.Insert(0, "Selecione");
}

```

Crie um método para Limpar os campos após o cadastro.

```

private void LimparCampos()
{
    txtNumero.Text = "";
    txtPotencia.Text = "";
    txtAlturaBico.Text = "";
    txtAlturaAerofolio.Text = "";

    //remove seleção do ddl
    for (int i = 0; i < ddlPiloto.Items.Count; i++)
    {
        ddlPiloto.Items[i].Selected = false;
    }
    //coloca o "Selecione" selecionado
    ddlPiloto.Items[0].Selected = true;
}

```

No evento Page_Load

```

protected void Page_Load(object sender, EventArgs e)
{
    //carrega somente a primeira vez
    if (!Page.IsPostBack)
    {
        CarregaPilotos();
        ddlPiloto.Focus();
    }
}

```

No evento click do Botão

```
protected void btnSalvar_Click(object sender, EventArgs e)
{
    PilotoBD pilotobd = new PilotoBD();
    Piloto piloto = pilotobd.Select(Convert.ToInt32(ddlPiloto.SelectedItem.Value));

    Carro carro = new Carro();
    carro.Numero = Convert.ToInt32(txtNumero.Text);
    carro.Potencia = Convert.ToDouble(txtPotencia.Text);
    carro.AlturaBico = Convert.ToDouble(txtAlturaBico.Text);
    carro.AlturaAerofolio = Convert.ToDouble(txtAlturaAerofolio.Text);
    carro.Piloto = piloto;

    CarroBD carrobd = new CarroBD();
    int retorno = carrobd.Insert(carro);

    switch (retorno)
    {
        case 0:
            LimparCampos();
            ddlPiloto.Focus();
            lblMensagem.Text = "Cadastro realizado com sucesso";
            break;
        case 1:
            //Erro no banco de dados
            lblMensagem.Text = "Não foi possível realizar o cadastro.";
            break;
        case 2:
            //Erro geral
            lblMensagem.Text = "Não foi possível realizar o cadastro.";
            break;
        default:
            break;
    }
}
```

Coloque essa página para ser a primeira a ser executada.

Cadastre alguns pilotos no banco de dados.

```
INSERT INTO `webheranca`.`tbl_membro` (`mem_nome`, `mem_contrato`, `mem_tipo`,
`mem_peso`, `mem_altura`) VALUES ('Rubens Barrichello', '1982631', '1', '72.0', '1.65');
INSERT INTO `webheranca`.`tbl_membro` (`mem_nome`, `mem_contrato`, `mem_tipo`,
`mem_peso`, `mem_altura`) VALUES ('Felipe Massa', '1218368', '1', '68.00', '1.62');
```

Execute a aplicação e cadastre um Carro.

Obs.: Não está sendo considerado se já existe um carro para o Piloto. Essa verificação pode ser feita, dependendo da regra de negócio. Mais a frente é mostrada como pode ser feita.

Listar Carros

Crie um novo método em CarroBD

```
public DataSet SelectAll()
{
    DataSet ds = new DataSet();

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    System.Data.IDataAdapter objDataAdapter;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command("SELECT * FROM tbl_carro", objConexao);

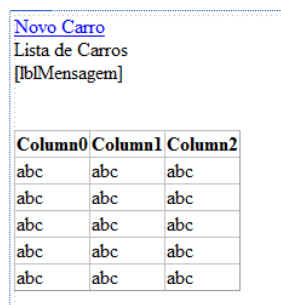
    objDataAdapter = Mapped.Adapter(objCommand);
    objDataAdapter.Fill(ds);

    objConexao.Close();

    objCommand.Dispose();
    objConexao.Dispose();

    return ds;
}
```

Crie uma nova página, dentro de Paginas/Equipe/Carro, chamada Listar.aspx



Configure o Hyperlink para chamar a página de Cadastro de Carros.

Acesse o código fonte da tela (Listar.aspx.cs)

Importe os namespaces

```
using WebHeranca.Persistencia.Equipe;
using System.Data;
```

No evento Page_Load

```
protected void Page_Load(object sender, EventArgs e)
{
    CarroBD bd = new CarroBD();
    DataSet ds = bd.SelectAll();

    //verifica a quantidade de carros no dataset
    int quantidade = ds.Tables[0].Rows.Count;
    if (quantidade > 0)
```

```

{
    grvCarros.DataSource = ds.Tables[0].DefaultView;
    grvCarros.DataBind();
    lblMensagem.Text = "Existem " + quantidade + " carros cadastrados";
}
else
{
    lblMensagem.Text = "Nenhum carro cadastrado";
}
}

```

Coloque a página Listar.aspx para ser a primeira a ser executada.

Execute a aplicação

Serão mostrados os carros mas sem formatação do GridView.
Além disso, irá mostrar o código do piloto e não as informações dele.

Altere o GridView para deixar apenas as colunas Número do carro, Potência do carro e Nome do Piloto.

Na edição do GridView, desmarque a opção “Auto generate fields”

Selecione um BoundField e clique em Add.

Faça isso 3 vezes.

Cada BoundField corresponde a um campo que será exibido.

Na caixa Selected Fields, selecione o primeiro BoundField e altere as propriedades:

- HeaderText – cabeçalho da coluna
- DataField – campo no banco de dados que será exibido.

Repita isso para os 3 campos.

HeaderText	DataField
Carro	car_numero
Potência	car_potencia
Piloto	mem_nome

Em CarroBD, crie um novo método.

```

public DataSet SelectAllWithPilots()
{
    DataSet ds = new DataSet();

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    System.Data.IDataAdapter objDataAdapter;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command("SELECT * FROM tbl_carro INNER JOIN tbl_membro ON
tbl_carro.mem_codigo=tbl_membro.mem_codigo ORDER BY car_numero", objConexao);

```

```

        objDataAdapter = Mapped.Adapter(objCommand);
        objDataAdapter.Fill(ds);

        objConexao.Close();

        objCommand.Dispose();
        objConexao.Dispose();

        return ds;
    }

```

Acesse o código fonte da tela (Listar.aspx.cs)
 Altere o Page_Load (apenas o código em destaque).

```

protected void Page_Load(object sender, EventArgs e)
{
    CarroBD bd = new CarroBD();
    DataSet ds = bd.SelectAllWithPilots();

    //verifica a quantidade de carros no dataset
    int quantidade = ds.Tables[0].Rows.Count;
    if (quantidade > 0)
    {
        grvCarros.DataSource = ds.Tables[0].DefaultView;
        grvCarros.DataBind();
        lblMensagem.Text = "Existem " + quantidade + " carros cadastrados";
    }
    else
    {
        lblMensagem.Text = "Nenhum carro cadastrado";
    }
}

```

Execute a aplicação.

Exercícios

1. Faça a alteração de um Carro.
2. Faça a exclusão de um Carro.