

MANUAL UNIT MATCH.REGISTRY

(Passo a Passo)

The screenshot displays a web application window titled "UfraMatch" with a sub-tab "UfraMatch.Registry". The main content area has a light teal background. At the top left is a back arrow icon. In the center is a shield-shaped logo featuring a soccer ball and a banner that reads "Partidas". Below the logo, there are two stacked teal buttons labeled "HORA" and "DATA". Under these is a row of two light gray dropdown menus with a small "X" between them. At the bottom is a large teal button labeled "SALVAR".

UfraMatch UfraMatch.Registry X

<

Partidas

HORA

DATA

▼ X ▼


SALVAR

Sumário

1.	Uses Interface.....	3
2.	Métodos e Atributos Privados.....	3
3.	Métodos Públicos.....	3
4.	Variável do Frame	3
5.	Uses Implementation	4
6.	Constructor Create	4
7.	Método CarregarTeams	4
8.	Método Registrar	5
9.	Método VoltarTela	5
10.	Eventos Click do RectSalvar e RectVoltar	6

1. Uses Interface

```
interface
  uses
    System.SysUtils
    FMX.Types, FMX.
    FMX.Edit, FMX.C
    FMX.ListBox,
    UService.Intf;
```



2. Métodos e Atributos Privados

```
private
{ Private declarations }
FServiceTeam: IService;

procedure VoltarTela;
procedure CarregarTeams;
procedure Registrar;
public
```

3. Métodos Públicos

```
public
{ Public declarations }
constructor Create(aOwner: TComponent); override;
end;
```

4. Variável do Frame

```
var
  fraMatchRegistry: TfraMatchRegistry;

implementation
{$R *.fmx}
```

5. Uses Implementation

```
implementation  
  
{$R *.fmx}  
  
uses  
    UfraMatch,  
    UService.Team,  
    UService.Match,  
    UEntity.Teams,  
    UEntity.Matches;
```

6. Constructor Create

```
constructor TfraMatchRegistry.Create(aOwner: TComponent);  
begin  
    Inherited Create(aOwner);  
    Self.CarregarTeams;  
end;
```

7. Método CarregarTeams

```
procedure TfraMatchRegistry.CarregarTeams;  
var  
    xTeam: TTeam;  
begin  
    cmbTimeA.Items.Clear;  
    cmbTimeB.Items.Clear;  
  
    if not Assigned(FServiceTeam) then  
        FServiceTeam := TServiceTeam.Create;  
  
    FServiceTeam.Listar;  
    for xTeam in TServiceTeam(FServiceTeam).Teams do  
    begin  
        cmbTimeA.Items.AddObject(xTeam.Name, xTeam);  
        cmbTimeB.Items.AddObject(xTeam.Name, xTeam);  
    end;  
end;
```

8. Método Registrar

```
procedure TfraMatchRegistry.Registrar;
var
  xServiceMatch: IService;
  xHora: TTime;
  xData: TDate;
  xTimeAux: TTeam;
  xTimeA, xTimeB: TTeam;
begin
  if Trim(edtHora.Text) = EmptyStr then
    raise Exception.Create('Informe a Hora da Partida.');

  if Trim(edtData.Text) = EmptyStr then
    raise Exception.Create('Informe a Data da Partida.');

  if cmbTimeA.ItemIndex = -1 then
    raise Exception.Create('Informe o Time A da Partida.');

  if cmbTimeB.ItemIndex = -1 then
    raise Exception.Create('Informe o Time B da Partida.');

  if cmbTimeA.ItemIndex = cmbTimeB.ItemIndex then
    raise Exception.Create('Informe Times diferentes para a Partida.');

  xHora := StrToTime(Trim(edtHora.Text));
  xData := StrToDate(Trim(edtData.Text));

  xTimeAux := TTeam(cmbTimeA.Items.Objects[cmbTimeA.ItemIndex]);
  xTimeA := xTimeAux.Clone;

  xTimeAux := TTeam(cmbTimeB.Items.Objects[cmbTimeB.ItemIndex]);
  xTimeB := xTimeAux.Clone;

  xServiceMatch := TServiceMatch.Create(
    TMatch.Create(xData, xHora, xTimeA, xTimeB));

  xServiceMatch.Registrar;
  Self.VoltarTela;
end;
```

9. Método VoltarTela

```
procedure TfraMatchRegistry.VoltarTela;
begin
  if not Assigned(FraMatch) then
    FraMatch := TFraMatch.Create(Application);

  FraMatch.Align := TAlignLayout.Center;

  Self.Parent.AddObject(FraMatch);
  FreeAndNil(FraMatchRegistry);
end;
```

10. Eventos Click do RectSalvar e RectVoltar

```
procedure TfraMatchRegistry.rectSalvarClick(Sender: TObject);  
begin  
    Self.Registar;  
end;  
  
procedure TfraMatchRegistry.rectVoltarClick(Sender: TObject);  
begin  
    Self.VoltarTela;  
end;
```