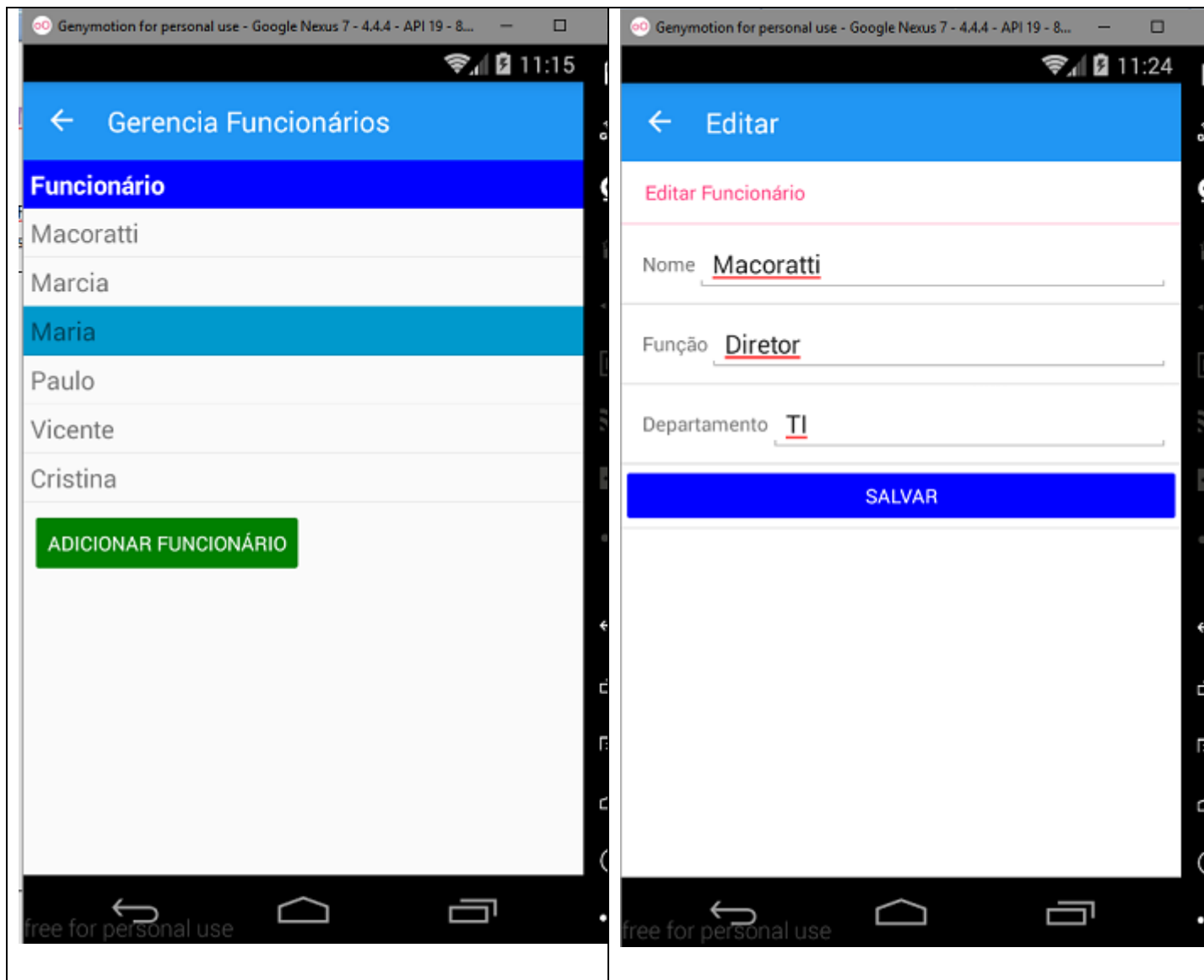


# Curso de Xamarin.Forms – Macoratti .net – SQLite : Exercícios

## Exercício - Manutenção de funcionários – CRUD SQLite

Criar uma aplicação Xamarin Forms para gerenciar informações de funcionários com as seguintes funcionalidades : **Adicionar, Editar, Deletar e Consultar funcionários.**



**GerenciaFuncionarios.xaml**

**MostraFuncionario.xaml**

Modelo de domínio representando pela classe : Funcionario (inclua na pasta Model)

```
[Table("Funcionarios")]
public class Funcionario
{
    [PrimaryKey, AutoIncrement]
    public int Id { get; set; }
    [NotNull]
    public string Nome { get; set; }
    [MaxLength(100)]
    public string Funcao { get; set; }
    [MaxLength(50)]
    public string Departamento { get; set; }
}
```

# Curso de Xamarin.Forms – Macoratti .net – SQLite : Exercícios

Roteiro para implementar as funcionalidades usando o SQLite:

- 1- Incluir o pacote sqlite-net-pcl no projeto via Nuget
- 2- Definir uma interface com os métodos de acesso ao SQLite
- 3- Implementar o acesso em cada plataforma
- 4- Registrar a implementação

Crie a interface em uma pasta **Persistencia**: ISQLiteDb

Depois crie uma classe **SQLiteDb** em cada projeto que implemente essa interface.

O projeto contém 4 paginas:

- 1- **GerenciaFuncionarios** – Defina um ListView com Header e Footer e uma ViewCell com um StackLayout para exibir o nome do funcionário e trate o evento **ItemSelected** do ListView. Inclua um Button para incluir um novo funcionário.

No construtor dessa página acesse todos os dados da tabela e exiba no List|View.

- 2- **AdicionarFuncionario** – Defina um **TableView** com 3 **EntryCell** para receber as informações para o nome, função e departamento e um botão Salvar
- 3- **MostrarFuncionario** - Defina um **Grid** com 6 linhas e 3 colunas conforme abaixo:

```
<Grid.RowDefinitions>
  <RowDefinition Height="*" />
  <RowDefinition Height="*" />
  <RowDefinition Height="*" />
  <RowDefinition Height="*" />
  <RowDefinition Height="*" />
  <RowDefinition Height="*" />
</Grid.RowDefinitions>

<Grid.ColumnDefinitions>
  <ColumnDefinition Width="10" />
  <ColumnDefinition Width="*" />
  <ColumnDefinition Width="*" />
</Grid.ColumnDefinitions>
```

Inclua dois botões de comando : **Editar e Deletar** e trate os eventos **Clicked**

## Curso de Xamarin.Forms – Macoratti .net – SQLite : Exercícios

- 4- **EditarFuncionario** – Defina um **TableView** com um **TableSection** usando 3 **EntryCell** para alterar os dados de nome, função e departamento e um **ViewCell** com um **StackLayout** com o botão **Salvar** . Trate o evento **Clicked**.

A navegação entre as páginas deve usar o método **Navigation.PushAsync**

Crie uma camada de acesso a dados na pasta **Persistencia** criando uma classe **DAL** com o seguinte código:

```
public class DAL
{
    SQLiteConnection dbConn;

    public DAL()
    {
        dbConn = DependencyService.Get<ISQLiteDb>().GetConnection();
        // cria a tabela
        dbConn.CreateTable<Funcionario>();
    }

    public List<Funcionario> GetTodosFuncionarios()
    {
        //return dbConn.Query<Funcionario>("Select * From Funcionario");
        return dbConn.Table<Funcionario>().ToList();
    }

    public int InserirFuncionario(Funcionario funcionario)
    {
        return dbConn.Insert(funcionario);
    }

    public int DeletarFuncionario(Funcionario funcionario)
    {
        return dbConn.Delete(funcionario);
    }

    public int EditarFuncionario(Funcionario funcionario)
    {
        return dbConn.Update(funcionario);
    }
}
```

Esta classe acessa cada implementação e cria ou acessa a tabela **Funcionarios** retornando um contexto do banco de dados. (**dbContext**)

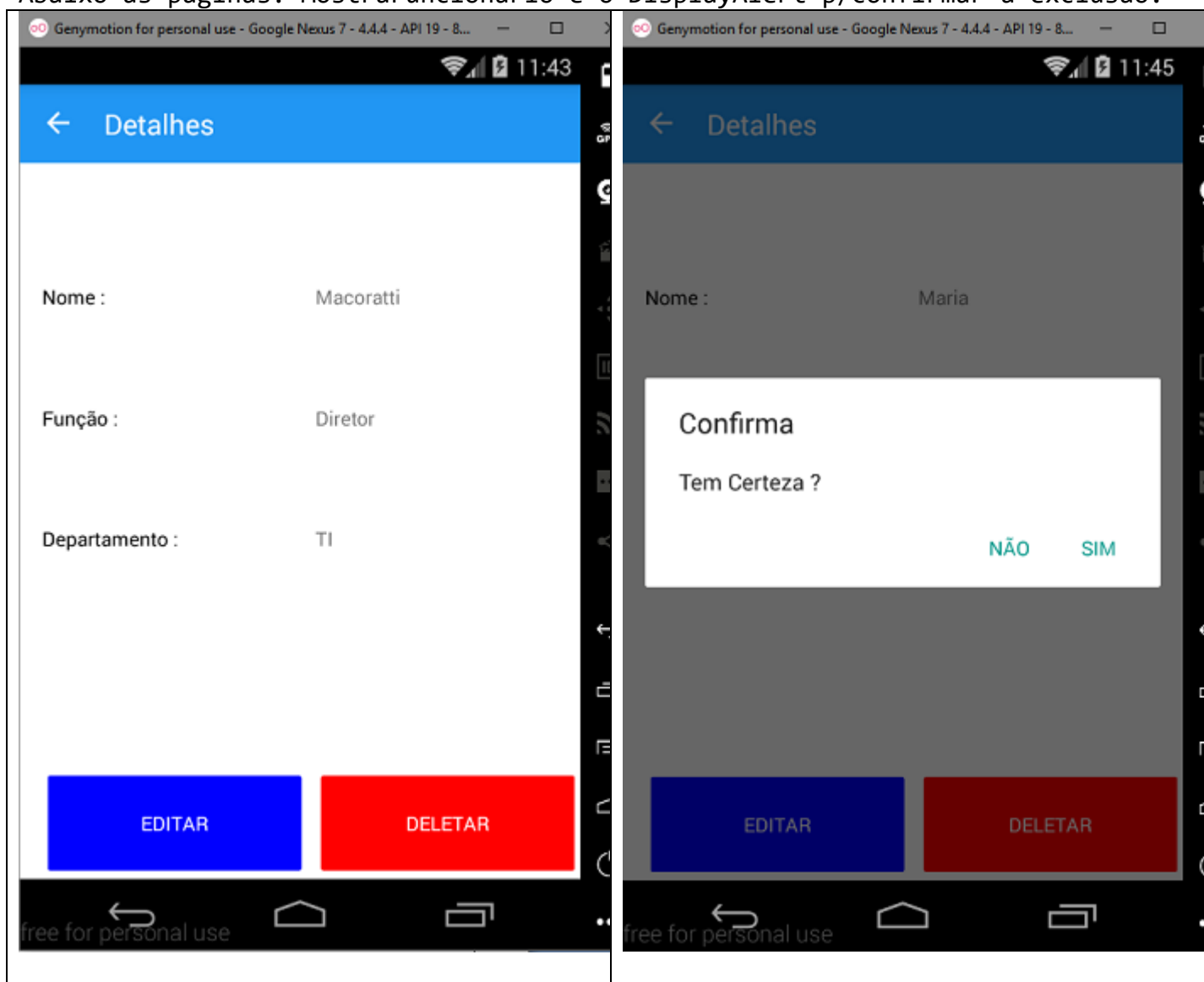
## Curso de Xamarin.Forms – Macoratti .net – SQLite : Exercícios

No arquivo **App.cs** defina o código para obter uma instância da conexão com o banco de dados em uma variável estática:

```
static DAL db;
public static DAL dbContext
{
    get
    {
        if (db == null)
        {
            db = new DAL();
        }
        return db;
    }
}
```

Depois para acessar os métodos da camada de acesso a dados basta usar:  
**App.dbContext.EditarFuncionario(funci);**

Abaixo as páginas: MostraFuncionario e o DisplayAlert p/confirmar a exclusão.



## Curso de Xamarin.Forms – Macoratti .net – SQLite : Exercícios