



Neste artigo vou apresentar o conceito relacionado com **Shared Preferences** para persistir dados em uma aplicação **Xamarin Android**.

Curso C# Vídeo Aulas
Do básico ao intermediário

Por um preço justo

A maioria dos aplicativos Android precisa salvar dados, mesmo que seja apenas para salvar informações sobre o estado do aplicativo durante `onPause()` para que o progresso do usuário não seja perdido.

A maior parte dos aplicativos não triviais também precisa salvar configurações do usuário e alguns aplicativos precisam gerenciar grandes quantidades de informação em arquivos e bancos de dados.

Quando você tem que persistir uma quantidade pequena de dados não vale a pena criar um banco de dados para realizar essa tarefa pois o Android oferece um recurso chamado **SharedPreferences** que permite armazenar pares **chave-valor (key-value)** de tipos de dados simples em um arquivo de preferências compartilhadas.

Caso você tenha uma coleção pequena de dados para salvar, pode usar as **APIs** [SharedPreferences](#).

Um objeto [SharedPreferences](#) indica um arquivo que contém pares **chave-valor(key-value)** e fornece métodos simples para ler e gravar.

Cada arquivo [SharedPreferences](#) é gerenciado pelo framework e pode ser privado ou compartilhado.

Afim de usar **Shared Preferences** você tem obter uma instância da interface **ISharedPreferences**; Sendo que uma **sharedpreference** pode ser específica a uma Activity ou ser global para todas as atividades da aplicação.

Para obter um nível de preferência a nível de aplicação você chama o método `getSharedPreferences()` passando o nome da preferência e o modo de operação; Este método retorna uma instância **SharedPreferences** que aponta para o arquivo que contém o valor das preferências.

Exemplo:

```
ISharedPReferences dadosLocais = Application.Context.getSharedPreferences("MeuArquivo", Android.Content.FileCreationMode.Private);
```

O primeiro parâmetro - **MeuArquivo** - é o nome do arquivo e o segundo parâmetro - **Android.Content.FileCreationMode.Private** - é o modo de acesso.

Nota : Se você deseja criar um arquivo único de preferência específico para uma atividade, você pode usar **Activity.GetPreferences** para inicializar a interface **ISharedPreferences**.

A seguir temos os principais modos de acesso :

- **PRIVATE** - Permite o acesso somente pela aplicação;
- **MODE_APPEND** - Anexa as novas preferências com as já existentes;
- **MODE_ENABLE_WRITE_AHEAD_LOGGING** - Flag de abertura de banco de dados;
- **MODE_MULTI_PROCESS** - Irá verificar as modificações das preferências mesmo se a instância de *sharedpreferences* já foi carregada;
- **MODE_WORLD_READABLE** - Permite que outra aplicação leia as preferências;
- **MODE_WORLD_WRITABLE** - Permite que outra aplicação escreva nas preferências;

Você pode salvar informações em *sharedpreferences* chamando o método **Edit()** para obter uma instância de **ISharedPreferencesEditor** sendo que isso engloba todas as alterações feitas no valor das preferências.

Para salvar o valor das preferências usamos o método **Commit()** ou **Apply()**.

Exemplo :

```
ISharedPreferencesEditor editor = prefs.Edit();
editor.PutInt("chave1", 10);
editor.PutString("chave2", "Macoratti .net");
editor.Apply();
```

```
ISharedPreferencesEditor editor =
prefs.Edit();
editor.PutInt("chave1", 10);
editor.PutString("chave2", "Macoratti
.net");
editor.Commit();
```

Para ler dados armazenados nas *sharedpreferences* podemos usar os métodos : **GetString()**, **GetInt()**, **GetFloat()**, **etc.**, fornecendo a chave para recuperar o valor.

Exemplo :

```
var valor1 = prefs.GetInt ("chave1", 0);  
var valor2 = prefs.GetFloat ("chave2", null);  
var valor3 = prefs.GetString ("chave3",  
null);
```

Lembrando que você nunca deve armazenar dados pessoais ou senhas nos arquivos de preferência.

Na segunda parte do artigo vou mostrar como usar esse recurso.

E disse-lhe Jesus: Eu vim a este mundo para juízo, a fim de que os que não vêm vejam, e os que vêm sejam cegos.
João 9:39

[Veja os Destaques e novidades do SUPER DVD Visual Basic \(sempre atualizado\) : clique e confira !](#)

Quer migrar para o VB .NET ?

- Veja mais sistemas completos para a plataforma .NET no [Super DVD .NET](#) , confira...
- [Curso Básico VB .NET - Vídeo Aulas](#)

Quer aprender C# ??

- Chegou o [Super DVD C#](#) com exclusivo material de suporte e vídeo aulas com curso básico sobre C#.
- [Curso C# Basico - Video Aulas](#)

Quer aprender os conceitos da Programação Orientada a objetos ?

- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW

Quer aprender o gerar relatórios com o ReportViewer no VS 2013 ?

- [Curso - Gerando Relatórios com o ReportViewer no VS 2013 - Vídeo Aulas](#) NEW

Referências:

- [Seção VB .NET do Site Macoratti.net](#)
- [Super DVD .NET - A sua porta de entrada na plataforma .NET](#)
- [Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#](#)
- [Super DVD C# - Recursos de aprendizagens e vídeo aulas para C#](#)
- [Seção C# do site Macoratti.net](#)
- [Seção ASP .NET do site Macoratti .net](#)
- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW
- [Macoratti .net | Facebook](#)

- [macoratti - YouTube](#)
- [Jose C Macoratti \(@macorati\) | Twitter](#)
- [Xamarin - Desenvolvimento Multiplataforma com C# ... - Macoratti.net](#)
- [Xamarin - Apresentando Xamarin.Forms - Macoratti.net](#)
- [Xamarin.Forms - Olá Mundo - Criando sua primeira ... - Macoratti.net](#)
- [Xamarin.Forms - Olá Mundo - Anatomia da aplicação - Macoratti.net](#)
- <https://developer.xamarin.com/api/type/Android.App.AlertDialog/>
- [Xamarin Android - Tratando eventos de forma declarativa](#)

[José Carlos Macoratti](#)