

# Codificação 12: Armazenamento de Dados no teu Dispositivo

Armazenar dados no teu dispositivo é uma funcionalidade que pode ser útil para a tua aplicação, dependendo da solução do teu problema particular. Aprender a armazenar dados no teu dispositivo pode ajudar-te a marcar pontos na parte técnica da secção de Vídeo Demo da rubrica de julgamento.

Nesta aula, tu vais...

- Aprender a armazenar dados no teu dispositivo em App Inventor e Thinkable
- Aprender a usar o TinyDB no App Inventor
- Aprender a usar variáveis armazenadas no Thinkable

## Termos e Conceitos Chave

**Base de dados** - uma colecção organizada de informação

**Base de dados local** - uma colecção organizada de informação armazenada num dispositivo

**TinyDB** - O componente App Inventor que te permite armazenar informação num dispositivo

**Variáveis armazenadas** - Blocos no Thinkable que te permitem armazenar informação no teu dispositivo

**Tag-value pair** - uma forma de armazenar informação numa base de dados

## Inspiração - Guardar informação no teu aparelho

### Armazenamento Local

Nesta aula, aprenderás a armazenar dados no teu dispositivo, ou a usar o **armazenamento local**. Aqui estão alguns exemplos de armazenamento de dados que poderás usar todos os dias:

- Lista de contactos - guardas os números de telefone dos teus amigos para os poderes usar mais tarde
- Messaging app - a maioria das aplicações de mensagens armazenam automaticamente mensagens antigas para que as possas ler mais tarde
- Armário de arquivo - guardas e organizas a papelada aqui
- Fotos - guardas e organizas as tuas fotos no teu dispositivo móvel

Na tua aplicação, podes usar o armazenamento local para guardar informação entre os momentos em

que o teu utilizador fecha e reabre a sua aplicação.

As variáveis, de que tomaste conhecimento, permitem-te guardar informação apenas enquanto a tua aplicação está a funcionar. O que isso significa é que quando a aplicação fecha, a variável essencialmente "desaparece". Qualquer que seja o valor de uma variável, quando a aplicação fecha, perde-se. Por exemplo, se a aplicação for um jogo e o utilizador marcar 350 pontos então o valor da variável de pontuação é 350, quando o utilizador fecha a aplicação, esse valor desaparece. Quando o utilizador reabre a aplicação, a pontuação será redefinida para qualquer variável para a qual a variável é inicializada.

Por vezes podes querer guardar informação, por isso quando o utilizador volta à aplicação, essa informação permanece como da última vez que o utilizador executou a aplicação. Por exemplo, talvez queiras guardar a pontuação alta para um jogo. Ou se estiveres a seguir os teus passos diários de caminhada, ou o número de copos de água bebidos todos os dias, queres guardar essa informação cada vez que executas a aplicação. Então queres usar o armazenamento local, o que significa armazenar os dados no dispositivo móvel.

## **Bases de dados**

O armazenamento local é por vezes chamado de **base de dados local**. Uma **base de dados** é uma colecção organizada de informação. Aceder a uma base de dados é conhecido como *chamá-la*, ou *fazer uma chamada para ela*. Quando ligas para uma base de dados, podes obter informação, apagar informação, armazenar nova informação, ou actualizar informação. A chave é como um nome de variável, é como dizes à base de dados qual a informação que queres obter, armazenar, actualizar ou apagar.

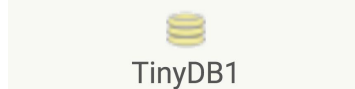
Estas duas secções seguintes mostram-te como guardar informação sobre o teu dispositivo no App Inventor e no Thinkable. As duas plataformas lidam com o armazenamento local de forma um pouco diferente, por isso vamos separar a aula. Se estás a usar Thinkable, salta para [Usar variáveis armazenadas em Thinkable](#).

## **Utilização de um pequeno componente DB no App Inventor**

Na App Inventor, o componente **TinyDB** permite-te guardar dados no telemóvel do utilizador que estará lá cada vez que ela abrir a tua aplicação. É importante saber que o TinyDB só te permite armazenar dados *localmente*. Isto significa que dois utilizadores não podem partilhar dados com o mesmo TinyDB. Quando um utilizador armazena dados no TinyDB, estes só estarão disponíveis no seu telemóvel, e no de mais ninguém. Na próxima aula, aprenderás como partilhar informação entre telemóveis.

Para usares o TinyDB, arrasta-o para o teu ecrã através do *Designer*. Podes encontrá-lo no menu de



## Non-visible components



Armazenamento no App Inventor. Irá aparecer como um componente não visível e terá este aspecto:

Podes falar com o teu TinyDB ou Armazém Local fazendo *chamadas* para ele. Quando fazes uma chamada, podes *guardar* coisas e *obter* coisas a partir delas. O TinyDB usa **pares de tags-valor**. Cada peça de informação no teu armazenamento é conhecida como um **valor**, e a **etiqueta** é o nome. Este é um nome que vais usar para recuperar os dados. É muito parecido com um nome de variável. Se usares o mesmo nome de tag para armazenar dados duas vezes, o TinyDB irá sobrepor, ou apagar, os dados antigos com os novos dados. Por isso, funciona exactamente como actualizar ou definir o valor de uma variável.

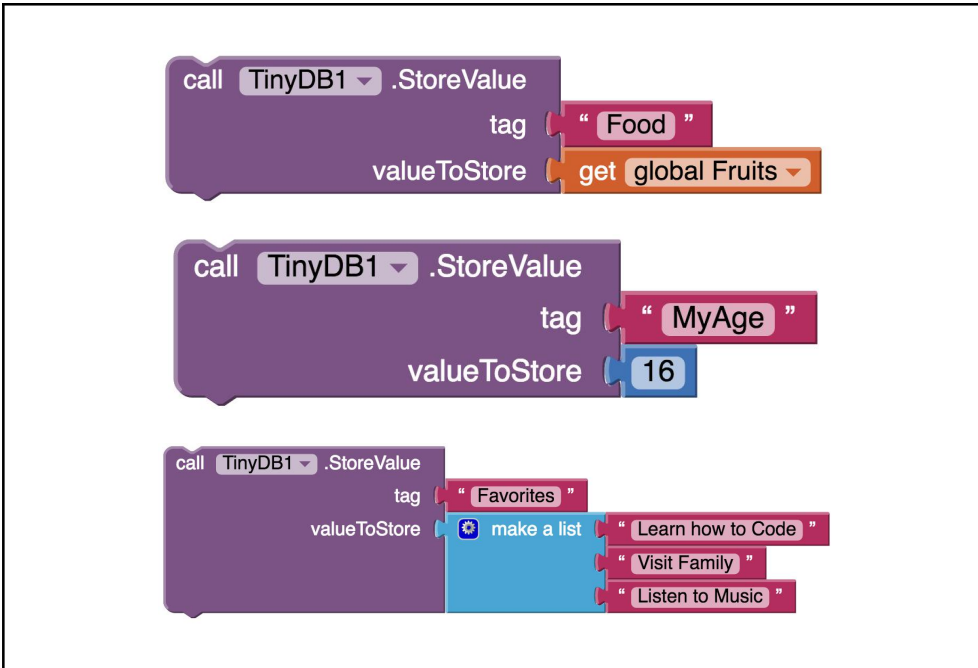
Aqui estão exemplos de como fazer chamadas para o TinyDB. O **StoreValue** é como o bloco de variáveis **definido**. **GetValue** é como o bloco da variável **get**.

<b>Guardar ou Guardar</b>	 A Scratch code block for TinyDB1's StoreValue method. The block is purple and has two input fields: 'tag' with the value 'Food' and 'valueToStore' with the value 'get global Fruits'.
<b>Obter</b>	 A Scratch code block for TinyDB1's GetValue method. The block is purple and has two input fields: 'tag' with the value 'Food' and 'valueIfTagNotThere' with the value 'create empty list'.

Acima, guardámos a nossa lista de *frutas* com a etiqueta "Comida". O segundo bloco mostra como recuperar a lista de *frutas* do armazenamento. A etiqueta precisa de ser escrita exactamente como estava quando os dados foram guardados, incluindo todas as letras maiúsculas. Além disso, certifica-te sempre de que o valorIfTagNotThere corresponde ao tipo de dados. Por exemplo, a Food está a armazenar uma lista de frutas, por isso se a aplicação ainda não tiver armazenado nada na etiqueta da Food, queres que seja definida para uma lista vazia.

Para melhor compreenderes isto, vamos dar um exemplo. Digamos que precisavas de armazenar três coisas numa base de dados. Uma é a lista de *Frutas*, uma delas é da tua idade, e a outra é uma lista das

tuas coisas favoritas a fazer. Fazes três chamadas para a base de dados desta forma:



Tens agora três entradas na tua base de dados, e é assim que elas se parecem:

<p>Nome da etiqueta: "Comida".</p> <p>Valor:</p> <ul style="list-style-type: none"><li>● "Maçãs" (índice = 1)</li><li>● "Bananas" (índice = 2)</li><li>● "Laranjas" (índice = 3)</li></ul>	<p>Nome da etiqueta: "MyAge"</p> <p>Valor: 16</p>	<p>Nome da etiqueta: "Favoritos".</p> <p>Valor:</p> <ul style="list-style-type: none"><li>● "Aprende a Codificar" (índice = 1)</li><li>● "Visit Family" (índice = 2)</li><li>● "Ouve a Música" (índice = 3)</li></ul>
--	---	---

Nota: Para uma actualização sobre como funcionam as listas, referência [Codificação 6: Listas](#)

Agora, quando queres recuperar as tuas coisas favoritas, fazes uma chamada como esta:



Quando usares este bloco irás obter uma lista {Aprender a Codificar, Visitar a Família, Ouvir Música}. Agora vamos dizer que queres recuperar a tua idade, por isso fazes uma chamada como esta:



Quando usas este bloco apenas recibes 0 de volta. Isto é porque a etiqueta "myage" não existe na tua base de dados! Uma vez que a tua base de dados não reconheceu a etiqueta, por defeito mostra-te o bloco junto ao "valueIfTagNotThere", que é o valor zero. Podes fazer valorIfTagNotThere tudo o que quiseres, mas faz sentido mantê-lo do mesmo tipo de dados que o que estás a armazenar. Por isso, fazer um valor numérico faz sentido. Usa sempre exactamente a mesma ortografia e capitalização quando armazenares e obtiveres valores de tag do TinyDB ou não conseguirá obter a tua informação.

Nota: No App Inventor, se criares uma aplicação com múltiplos ecrãs, os componentes e variáveis não poderão falar uns com os outros entre cada ecrã. Podes utilizar o TinyDB para transferir informação de um ecrã da tua aplicação para outro. Visita esta página para mais informação: [Ecrãs MIT](#).

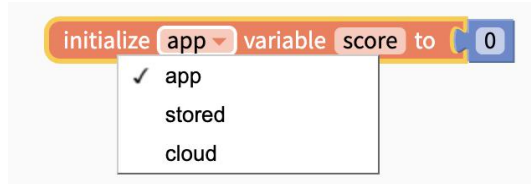
### Usando Variáveis Armazenadas no Thinkable

Thinkable simplificou o armazenamento de dados localmente em dispositivos móveis, diferenciando os seus tipos de variáveis em **variáveis de aplicação**, **variáveis armazenadas** e **variáveis de nuvem**.

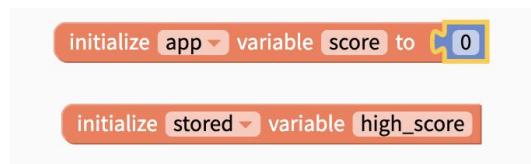
**As variáveis da aplicação** permitem armazenar informação na aplicação. O que isso significa é que quando a aplicação fecha, a variável essencialmente "desaparece". O que isso significa é que qualquer que seja o valor de uma variável quando a aplicação fecha, esta perde-se. Por exemplo, se a aplicação for um jogo e o utilizador marcar 350 pontos, o valor da variável de pontuação é 350, quando o utilizador fecha a aplicação, esse valor desaparece. Quando o utilizador reabre a aplicação, a pontuação será redefinida para qualquer variável para a qual a variável é inicializada. Isso significa que, se quiseres que a aplicação mantenha o valor, terás de usar uma **variável armazenada**. O valor seria guardado no telemóvel do utilizador, por isso quando ela reabrir a aplicação, o valor é o mesmo de quando a fechou pela última vez. Isto pode ser útil se, por exemplo, quiseres manter um registo da tua pontuação alta para um jogo.

**As variáveis da nuvem** permitem que a tua aplicação armazene dados na nuvem. Discutiremos as variáveis da nuvem muito mais em profundidade no Coding 13, por isso não falaremos mais sobre isso aqui.

As variáveis armazenadas funcionam muito parecidas com as variáveis da aplicação. Quando arrastas o bloco de **inicialização de variáveis**, o tipo padrão é app. No entanto, existe um menu pendente onde podes definir o tipo de variável como armazenada ou nebulosa.



Nota que quando inicializas uma variável de aplicação, deves dar-lhe um valor inicial. Para variáveis armazenadas, não há valor inicial, porque o seu valor inicial é puxado de onde está armazenado no



telefone.

Isso significa que terás de definir o valor da variável algures na tua aplicação. Por exemplo, podes definir a pontuação alta para qualquer que seja a pontuação actual do jogo, para que seja guardada entre as jogadas.



O Thunkable tem outro componente chamado LocalDB, que te permite guardar dados de tabelas na tua aplicação. É útil se quiseres incluir uma tabela de informação dentro da tua aplicação. No entanto, actualmente não funciona para armazenar informação localmente no teu dispositivo móvel, por isso é melhor ficares com o uso de variáveis armazenadas.

### Actividade - Lista de afazeres

Esta Lista de afazeres permite ao utilizador criar uma lista de afazeres que será lembrada cada vez que abrir e fechar a sua aplicação. No entanto, o código fonte tem um bug. A aplicação funciona para adicionar tarefas mas as tarefas eliminadas continuam a aparecer. Consegues descobrir como corrigi-lo?

[Código Fonte da App Inventor](#)

[Código Fonte Thunkable](#)

**Dica:** Tens de adicionar este bloco para que sempre que o utilizador adicionar ou

apagar algo da sua lista, também actualize a base de dados. Consegues descobrir qual deve ser a etiqueta e o valorToStore ou o valor?

### App Inventor



### Thunkable



Se estás presa, vê este vídeo da aplicação Inventor da Technovation Jennifer John: [Jennifer John's App Inventor To-dodo List Video](#)

## Reflexão

Agora que já aprendeste a armazenar dados no teu telemóvel, deves fazer uma tempestade de ideias com a tua equipa onde possas usar isto na tua aplicação. A próxima aula de codificação irá ensinar-te a armazenar informação na nuvem que não só é acessível no telemóvel do utilizador, mas em qualquer lugar onde ele entre na sua conta.

## Recursos Adicionais

### App Inventor e TinyDB

Este tutorial mostra-te como usar o TinyDB para localizar objectivos localmente no teu telemóvel.

- [Parte 1](#)
- [Parte 2](#)
- [Parte 3](#)

### Variáveis Armazenadas em Thunkable

Este [tutorial](#) é uma aplicação básica para guardar uma nota.