

Código 5: Variáveis

Esta aula vai ajudar-te a ganhar pontos nas linhas da rubrica do Vídeo de Demonstração.

Nesta aula, vais aprender...

- O que são variáveis e como usá-las

Termos e Conceitos Chave

Variáveis - um pedaço de dado que pode mudar

Variáveis Locais - variáveis que só podem ser usadas dentro de certas funções

Global Variables - variáveis que podem ser usadas em qualquer parte do teu código

Aprende

Variáveis

Um dos conceitos mais importantes na codificação é uma **variável**. Uma variável é um nome para um pedaço de dado que pode mudar. Uma forma fácil de lembrar é que uma *variável* pode *variar*, ou *mudar de valor*.

Há muita informação que é importante para ti todos os dias e que muda de valor. Aqui estão alguns exemplos:

- A tua idade
- O tempo
- A data de hoje

Imagina uma variável como uma caixa da qual podes guardar alguma informação dentro e guardá-la. Antes de a guardares, colocas um nome na caixa para te lembrares do que está dentro dela. Quando precisas da informação, podes ir procurar a caixa **para** obteres o que **está** dentro dela. Também podes mudar o que está dentro da caixa mas manter o mesmo nome na mesma.

As variáveis são uma forma de te lembrares de alguma coisa na tua aplicação. Tal como a

caixa, as tuas variáveis vão precisar de um par de coisas.

1. Um nome para que a tua aplicação o possa encontrar
2. Alguma informação para guardar

As variáveis são úteis sempre que precisares da tua aplicação para "lembrar" alguma informação e quando essa informação pode mudar. Aqui estão alguns exemplos:

- A pontuação de um jogador num jogo
- A resposta de uma pessoa a uma pergunta de questionário
- Coisas que estão no carrinho de compras de alguém

Exemplo: Como usar variáveis para armazenar a pontuação num jogo

Para compreender melhor as variáveis e como usá-las, vamos caminhar através de um exemplo. Imagina que estás a criar um jogo onde o utilizador pode ganhar e perder pontos. Queres que o jogador consiga ver a sua pontuação, por isso colocas a pontuação num rótulo. No início do jogo, o jogador tem zero pontos, por isso adicionaste um bloco como este ao jogo.



No entanto, a pontuação do jogador irá mudar à medida que ela joga o jogo. Cada vez que ela ganha ou perde pontos, tens de actualizar a caixa de texto. Quando ela ganha mais pontos, adicionas um bloco como este.



Isso funciona, mas agora e da próxima vez que o jogador ganhar 10 pontos? Terás de actualizar o texto novamente para dizer 20. Então e se o jogador perder pontos? Terás de alterar a pontuação dela para 15. E se ela ganhar pontos de novo? Isto pode tornar-se confuso muito rapidamente.

Em cenários como este, podes criar uma **variável** para manter o registo da pontuação do jogador. Podes criar uma variável chamada *pontuação* que tem sempre o valor da pontuação actual do jogador. Enquanto ela joga o jogo, podes adicionar e subtrair pontos à *pontuação* e não precisarás de te preocupar em manter o registo da pontuação actual do jogador. Eis como funcionaria na App Inventor e Thunkable. Primeiro irias buscar este bloco à secção de variáveis.

App Inventor	Thunkable




Este bloco diz à aplicação que existirá uma variável e pede-te que dês um nome e um valor à variável. Eis como seria o bloco para um jogo em que a pontuação do jogador começasse a zero.

App inventor	Thunkable

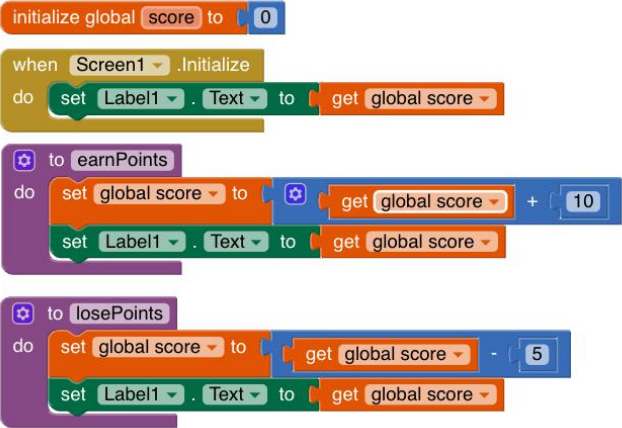
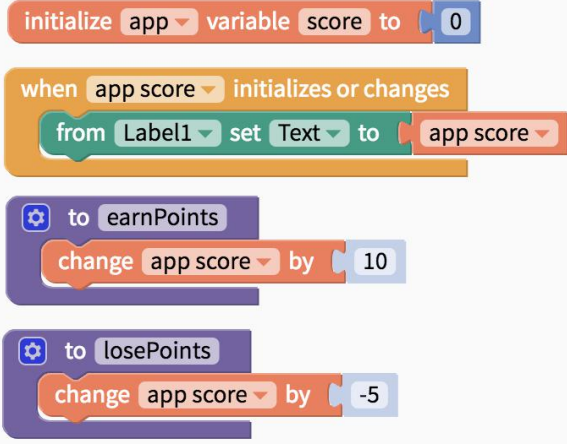
Com este bloco, estás a dizer à aplicação que haverá um dado chamado *pontuação* e que este irá mudar de valor à medida que o jogo for jogado. Cada vez que a aplicação arrancar, vai criar uma variável chamada *pontuação* e defini-la como zero. Agora podes fazer o que se chama "definir" e "obter" a tua variável de *pontuação*. "Obter" vai permitir-te ver qual é o valor da pontuação. Aqui está como dizes à tua aplicação para mostrar a pontuação numa caixa de texto.

App inventor	Thunkable

"Definir" permite-te atribuir um novo valor à pontuação do teu utilizador. Cada vez que o jogador ganha pontos podes dizer à aplicação para fazer isto.

App inventor	Thunkable
	 <p style="text-align: center;">ou</p> 

Isto diz-te para alterares o valor da pontuação para o que quer que seja agora e adiciona-lhe dez. Se a pontuação actual é zero, a nova pontuação seria 10. Se a pontuação actual é 25, a nova pontuação seria 35. Agora não tens de te preocupar em atribuir a pontuação de cada vez que o teu jogador ganha ou perde pontos. A aplicação apenas pegará na pontuação actual e adicionará 10 a ela.

App inventor	Thunkable
	

Não faz mal se ainda estás confuso com as variáveis! São realmente difíceis de compreender e a melhor maneira de as compreenderes é praticando a sua utilização. Vais poder praticar a sua utilização nas próximas aulas de codificação e desafios de codificação.

Actividade - Aplicativo de Contagem

Esta é uma aplicação que te permite contar por 1, 5 e 10.



No entanto, está incompleto! O contador nunca pode recomeçar. Arranja esta aplicação para que possas reiniciar o teu contador e recomeçar do zero.

[Obter o Código do App inventor](#)

[Obtém o Código de Afundações](#)

Reflexão

- Quais são algumas outras variáveis que consegues identificar na tua vida?
- Como usarias variáveis na tua aplicação?
- Como é que as variáveis podem beneficiar a tua aplicação?

Recursos adicionais - Variáveis Locais e Globais (App Inventor) e Variáveis App, Armazenadas, e Nuvem (Thunkable)

App Inventor e Thunkable tratam as variáveis de forma um pouco diferente, por isso basta leres a secção para qualquer plataforma que estejas a utilizar para codificar a tua aplicação.

Variáveis Locais e Globais - inventor de aplicações

Talvez tenhas reparado na secção acima que o bloco da variável dizia "global". Existem dois tipos de variáveis, **variáveis locais** e **variáveis globais**. Quando crias variáveis, terás de pensar se queres usar a

variável em vários locais do teu código, como a variável de *pontuação* de cima, ou se a queres usar num só local.

As variáveis globais podem ser usadas em qualquer lugar. Como viste acima, a variável de *pontuação* precisava de ser usada várias vezes. Tinhas de a usar quando o jogador ganhava pontos, quando o jogador perdia pontos, e de a mostrar na caixa de texto. Podia também ser usada para uma tabela de pontuação alta. Nunca poderás reutilizar um nome quando usares uma variável global. Se o fizeste, a tua aplicação não saberá qual a variável a utilizar. Felizmente a aplicação Inventor não te permitirá criar duas variáveis com o mesmo nome. Se tiveres dúvidas quanto ao tipo de variável a utilizar, utiliza global. Uma coisa a lembrar é que cada ecrã na App Inventor tem de ter as suas próprias variáveis. Uma variável global pode ser acedida em qualquer lugar dentro de um ecrã.

Getting Global Variables

As variáveis locais só podem ser usadas num ponto do teu código. Se precisares de uma variável para manteres o registo de algo num só local, usa uma variável local. Por exemplo, podes querer usar uma variável apenas para um *Button.Click* event, ou dentro de um procedimento que tenhas feito. A variável local só poderá ser usada dentro do bloco "variável local" e o App Inventor não te deixará usá-la fora dele. Ao contrário das variáveis globais, poderás reutilizar os nomes das variáveis locais.

Setting Local Variables

No desafio de codificação, verás exemplos de quando usar variáveis globais e quando usar variáveis locais. Para saberes mais sobre variáveis globais e locais visita o site do MIT: [Variáveis Globais e Locais](#).

App, Stored, e Cloud Variables - Thinkable

O Thinkable trata as variáveis de forma diferente do que o App Inventor. Tem três categorias de variáveis.



As variáveis App são semelhantes às variáveis globais no App Inventor. Podem ser usadas em qualquer parte da aplicação, e em qualquer ecrã. Como viste acima, a variável de *pontuação* precisava de ser usada várias vezes. Tiveste de a utilizar quando o jogador ganhou pontos, quando o jogador perdeu pontos, e para a mostrar na caixa de texto. Também podia ser usada para uma tabela de pontuação alta. Nunca poderás reutilizar um nome quando usares uma variável global. Se o fizeste, a tua aplicação não saberá qual a variável a utilizar. Felizmente o Thinkable não te permitirá criar duas variáveis com o mesmo nome. Se tiveres dúvidas quanto ao tipo de variável a usar, usa uma variável da aplicação.



As variáveis armazenadas permitem-te armazenar uma variável no teu dispositivo. Quando fechas uma aplicação no teu dispositivo, todas as variáveis são apagadas da memória do dispositivo. Assim, na

próxima vez que abrires a aplicação, a variável é recriada. É por isso que adicionas uma variável à tua aplicação usando o bloco de variáveis inicializável. Esta inicializa a variável para qualquer valor inicial que lhe dês. As variáveis armazenadas não são apagadas quando fechas a aplicação. Armazena o valor actual no dispositivo, e quando reabres a aplicação, ela recupera o valor. Nota que o bloco de inicialização das variáveis armazenadas não requer um valor inicial ou inicial. O valor inicial é recuperado a partir do dispositivo. Claro que, se é a primeira vez que estás a executar a aplicação, terás de definir o valor para a variável algures na tua aplicação.

A screenshot of a Thunkable variable block. The block is a horizontal bar with a light orange background. It contains the text "initialize" followed by a dropdown menu showing "stored" with a downward arrow, then "variable" and "age".

initialize stored variable age

As variáveis da nuvem são semelhantes às variáveis armazenadas, excepto que as variáveis são armazenadas na nuvem. Para usar as variáveis da nuvem, terás de adicionar um componente Firebase à tua aplicação. As tuas variáveis de nuvem serão armazenadas na base de dados do Firebase. Quando o utilizador fechar e reabrir a aplicação, o valor actual da variável será recuperado a partir do Firebase.

A screenshot of a Thunkable variable block. The block is a horizontal bar with a light orange background. It contains the text "initialize" followed by a dropdown menu showing "cloud" with a downward arrow, then "variable" and "age".

initialize cloud variable age

Para mais informação sobre os diferentes tipos de variáveis Thunkables, visita o website [Thunkable, Variables](#).