

# RESUMO DE CONTIDOS



## Índice

1. Datos e variables.....	2
Os datos.....	2
2. Variables numéricas.....	3
Variables propias das placas.....	3
Operacións con variables numéricas.....	3
3. Variables lóxicas.....	4
Verdadeiro ou falso?.....	4
Operadores lóxicos.....	4
Operar con variables lóxicas e numéricas.....	4
4. Variables con texto.....	4
Traballar con texto en variables.....	4
Operacións con texto en variables.....	5
5. Listas.....	6
Que é unha lista.....	6
Operacións con listas.....	6
6. Aplicacións.....	7
Programas e aplicacións.....	7
Atribución dos recursos incorporados ao documento.....	7

# 1. Datos e variables

## Os datos

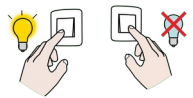
As placas controladoras reciben información do exterior que os programa procesan e transforman en ordes de actuación. Esa información recíbese em forma de **datos**. Os datos poden ser:

### Datos numéricos

Ao medir a temperatura, o nivel de ruído ou a intensidade da luz obtéñense valores numéricos. Estes datos representan cantidades e permiten comparar resultados, establecer límites e facer que o programa reaccione de distintas maneiras.



### Datos lóxicos



Cando se comproba se un botón está premido ou non, obtéñense datos lóxicos. Estes só poden ter dous valores: verdadeiro ou falso e empréganse para tomar decisións dentro do programa.

### Datos de texto



Os datos de texto, tamén chamados cadeas, permiten traballar con palabras e frases. Úsanse para gardar información escrita que pode empregarse posteriormente dentro do programa.

### Datos estruturados



Os datos estruturados son conxuntos de datos organizados nos que a orde é importante. A diferenza dos datos simples, permiten gardar varios valores xuntos. Un exemplo son as **listas**, onde cada elemento ocupa unha posición concreta con significado.

### Datos que cambian: as variables

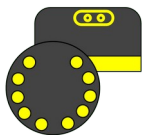


As variables son como espazos con nome onde se gardan datos para poder utilizalos e modificalos durante a execución dun programa.

Funcionan como caixas nas que o valor pode cambiar segundo o que ocorra.

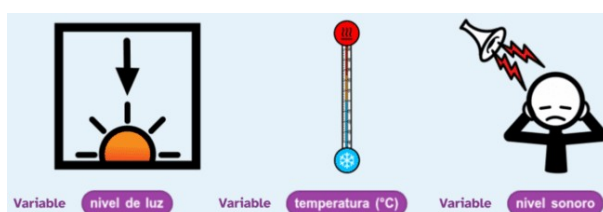
## 2. Variables numéricas

### Variables propias das placas



Os sensores das placas Adafruit PCE e micro:bit xeran datos numéricos que se poden ler a través das variables que MicroBlocks xa incorpora para cada sensor. Ademais, tamén é posible crear variables propias para gardar resultados de cálculos ou información que se precise lembrar posteriormente.

#### Luz, temperatura, son



#### As túas propias variables



### Operacións con variables numéricas

Para programar correctamente non abonda con crear variables, tamén é necesario saber utilízalas. Os operadores permiten modificar valores, realizar cálculos e comparar datos para comprobar condicións. A continuación tes os principais operadores cos que traballa MicroBlocks

- **Calcular.** operacións matemáticas e de programas en MicroBlocks que empregan sensores de inclinación.
- **Comparar.** comparacións matemáticas e programas en MicroBlocks que utilizan o sensor de luz.
- **Aumentar.** Empregando operadores podes usar variables propias para gardar datos, como contar cantas veces se activa un sensor.
- **Módulo.** O operador módulo úsase para obter o resto dunha división.

### 3. Variables lóxicas

#### Verdadeiro ou falso?

As variables lóxicas ou binarias só poden ter dous valores: verdadeiro ou falso. Representan estados, como aceso/apagado ou si/non, e exprésanse como *True* e *False* nas linguaxes de programación. En placas controladoras, por exemplo, un botón premido indica verdadeiro e, se non o está, falso; o mesmo ocorre co interruptor, que pode estar activado ou desactivado.

#### Operadores lóxicos

Os operadores lóxicos permiten combinar varias condicións para tomar decisións no programa. Os máis habituais son “e”, “ou” e “non”, cada un cun funcionamento distinto.



Significa que todas as condicións deben cumprirse á vez.



Significa que basta con que se cumpra unha condición.



Indica o contrario dunha condición.

#### Operar con variables lóxicas e numéricas

Ao combinar variables lóxicas e numéricas mediante operadores lóxicos, pódense crear programas máis completos que toman decisións segundo as accións dos botóns e os datos dos sensores.

### 4. Variables con texto

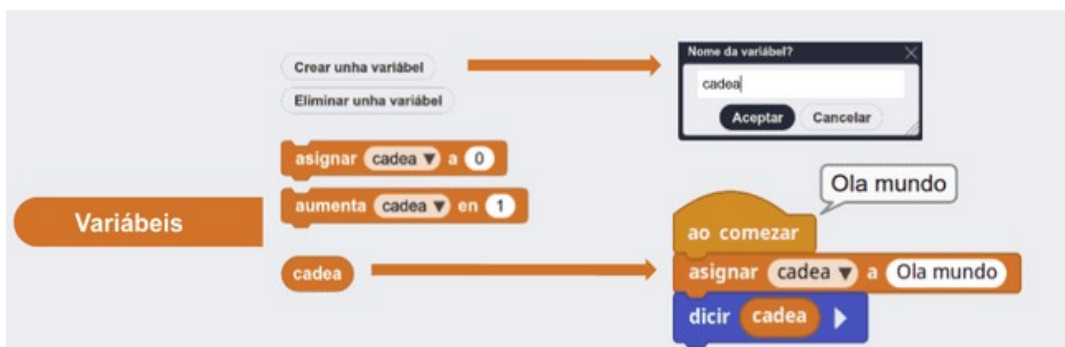
#### Traballar con texto en variables



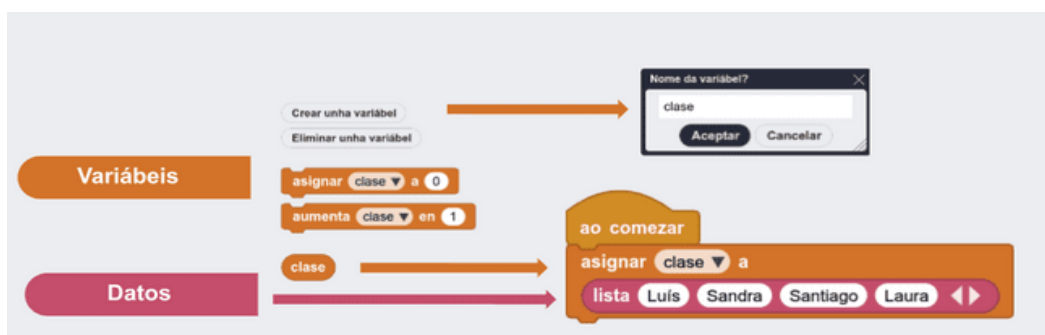
Nas variables pódese gardar texto, como letras, símbolos ou frases, que en programación se coñecen como cadeas (*strings*). Estas están formadas por varios caracteres unidos.

Un aspecto clave das cadeas é a súa lonxitude, é dicir, o número de caracteres que conteñen. En MicroBlocks pódese mostrar o seu valor ou a súa lonxitude mediante a instrución “**Dicir**”, que amosa o resultado ao inicio do programa.

É o mesmo que crear unha variable numérica: primeiro **indicas o nome da variable** e despois **asignas o seu valor inicial** coa orde correspondente. **A elipse co nome da variable** podes usala nas instrucións que necesites.



A variable lonxitude da cadea, dispoñible no bloque de “**Datos**”, permite coñecer o número de caracteres dun texto, incluídos os espazos. É moi utilizada en programación e resulta importante coñecela.



## Operacións con texto en variables

Cos textos pódense realizar operacións específicas como unir, identificar, modificar, converter, atopar e dividir.

**Unir.** Para escribir unha frase empregando o valor dunha variable, xa sexa numérica ou de texto, utilízase a operación de unir ou concatenar, que consiste en xuntar elementos nunha cadea.

**Identificar.** Para coñecer que letra ou carácter hai nunha posición concreta dunha cadea, pódese empregar a instrución “**elemento**”, indicando o número de orde correspondente.

**Modificar.** Unha cadea non pode modificarse parcialmente, é dicir, non se pode cambiar só un elemento. Con todo, si é posible substituíla completamente por un novo valor cando sexa necesario.

**Converter.** Ás veces é necesario converter datos **entre texto e número** para poder traballar con eles correctamente.

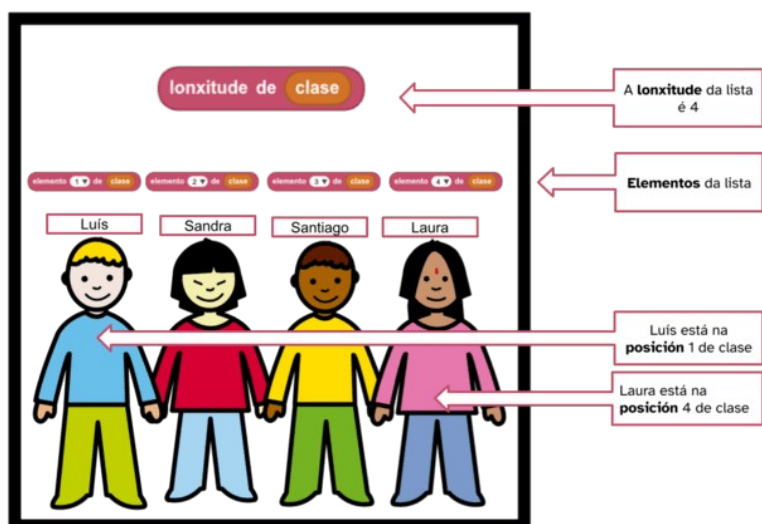
**Atopar.** Permite **buscar caracteres dentro dunha cadea**.

**Dividir.** Ás veces pode interesarche separar unha frase nas palabras que a forman para identificar a posición concreta de unha delas. Para iso úsase a orde “dividir”.

## 5. Listas

### Que é unha lista

Unha lista é un tipo de dato estruturado que permite gardar varios valores ordenados dentro dunha mesma variable. En MicroBlocks, pódese crear empregando a instrución “**lista**” do bloque “**Datos**”. Na seguinte imaxe explícase o que é unha lista.



- **Lonxitude:** é o número total de elementos que ten a lista
- **Elementos:** son os datos que se gardan na lista.
- **Posición:** é o lugar que ocupa cada elemento dentro da lista

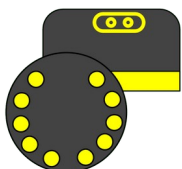
### Operacións con listas

Para comprender mellor as listas como datos organizados, é importante practicar operacións básicas como:

- **Enumerar:** mostrar os elementos dunha lista usando un separador.
- **Substituír:** cambiar un elemento coñecendo a súa posición.
- **Eliminar:** borrar un elemento segundo o seu número de orde.
- **Engadir:** incorporar novos elementos á lista.
- **Lista baleira:** crear unha lista sen elementos para engadir datos posteriormente.

## 6. Aplicacións

### Programas e aplicacións



Os programas permiten aprender a empregar instrucións, variables e sensores. Pola súa parte, unha aplicación permite integrar varios programas que traballan de maneira conxunta en dispositivos como a placa Adafruit PCE ou micro:bit.

#### Programa

- Realiza unha tarefa concreta.
- Resolve unha acción puntual.
- Pode ler un sensor, acender unha luz ou mostrar unha mensaxe.
- É máis pequeno e sinxelo.

#### Aplicación

- Está formada por varios programas que traballan conxuntamente.
- Resolve unha función máis complexa.
- Combina variables, condicións e subprogramas.
- Adáptase ao dispositivo no que se executa.

Para que a aplicación funcione correctamente, é necesario crear un programa principal que chame aos distintos subprogramas. Este programa debe incluír:

- Unha orde **“por sempre”**, que permite que a placa comprobe continuamente que botóns están a ser premidos (A, B, A+B...).
- Unha orde **“deter esta tarefa”** antes de pechar o bucle condicional que avalía os botóns. Deste xeito, cando se executa un subprograma, evítase que se activen os demais.

### Atribución dos recursos incorporados ao documento

Recursos incorporados por orde de aparición e páxina:

As infografías e imaxes pertencen ao Proxecto cREAgal e son de elaboración propia a partir de capturas de pantalla de MicroBlocks ([Licencia Pública de Mozilla v2.0](#)) e pictogramas ARASAAC, que son propiedade do Goberno de Aragón e foron creadas por Sergio Palao para ARASAAC (<https://www.arasaac.org>). Publícanse con [Licenza Creative Commons Recoñecemento Noncomercial Compartir igual 4.0](#)



“Resumo de contidos: Xogo programado reto superado”, do proxecto cREAgal, publícase coa [Licenza Creative Commons Atribución Non-comercial Compartir igual 4.0](#)