



Bloques integrados básicos

547120 Introducción a ELN
mariomedina@udec.cl

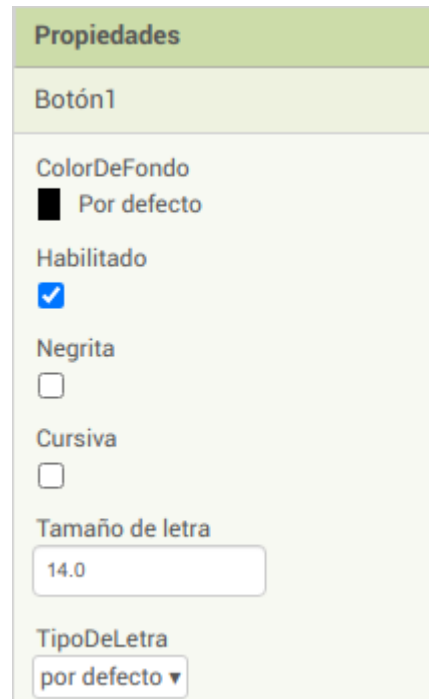


Bloques integrados

- En los ejemplos vistos hasta ahora, hemos usado algunos de los bloques integrados disponibles en AppInventor
 - Variables globales y locales
 - Bloques matemáticos
 - Bloques lógicos
 - Bloques condicionales

Objetos y propiedades

- En AppInventor, los diferentes componentes tienen *Propiedades*
 - Definir un valor en el *Diseñador* define el valor inicial de la propiedad
 - Muchas propiedades pueden ser leídas desde el programa
 - Muchas propiedades pueden ser modificadas desde el programa



The screenshot shows the 'Propiedades' (Properties) panel for a 'Botón1' (Button1) component. The panel has a green header with the title 'Propiedades'. Below the header, the component name 'Botón1' is displayed. The properties listed are:

- ColorDeFondo**: A color selection area showing a black square and the text 'Por defecto'.
- Habilitado**: A checkbox that is checked, indicated by a blue checkmark.
- Negrita**: An unchecked checkbox.
- Cursiva**: An unchecked checkbox.
- Tamaño de letra**: A text input field containing the value '14.0'.
- TipoDeLetra**: A dropdown menu showing 'por defecto' with a downward arrow.

Variables

- Una variable corresponde a una posición que tiene un nombre asociado, a través del cual es posible
 - acceder al valor almacenado
 - modificar el valor almacenado
- No está asociada a un componente específico



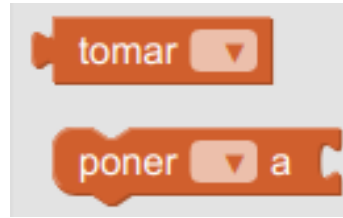
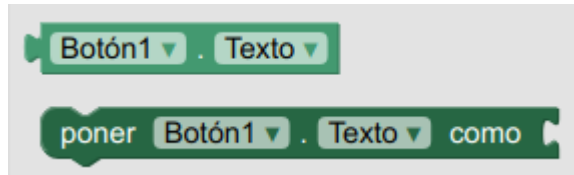
Inicialización

- Una propiedad es creada automáticamente al agregar un componente a la app
- Una variable debe ser creada explícitamente usando un bloque de inicialización, donde también se le da un nombre que debe ser único
 - Una variable puede ser inicializada con un valor numérico, un texto o una lista

inicializar global `nombre` como

Getters y Setters

- Tanto para propiedades como para variables, existen operaciones especiales para
 - leer el valor almacenado (*Getter*)
 - Modificar el valor almacenado (*Setter*)



Uso típico

- Muchas veces, una variable se usa para guardar un valor que indica la ocurrencia de un evento
- Variable se incrementa cada vez en 1

inicializar global Contador como 0

poner global Contador a  tomar global Contador + 1



Variables globales y locales

- Las variables globales existen fuera de cualquier componente o evento
 - Son accesibles desde cualquier otro manejador de evento o procedimiento
- Las variables locales existen sólo dentro de un manejador de evento o procedimiento

Variables locales

- Puede usarse una variable local en un bloque de tipo *ejecutar* para calcular una suma



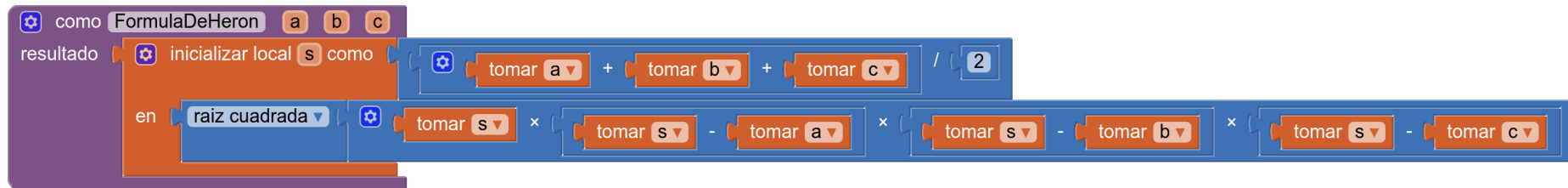


Variables locales

- Puede usarse una variable local para calcular una variable intermedia que sólo tiene sentido y uso dentro de un manejador de eventos ó procedimiento
- Se usa también cuando hay otras variables con el mismo nombre en otras partes del código

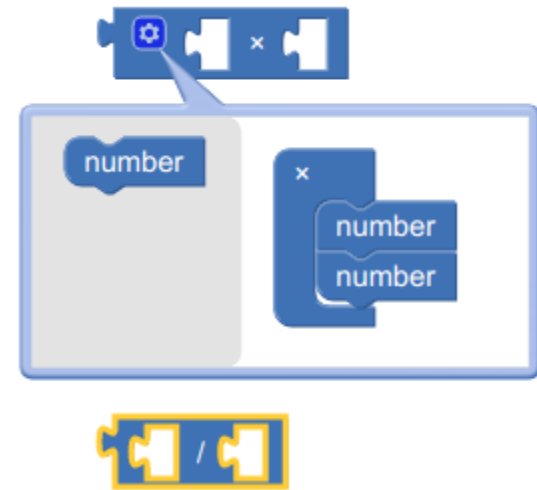
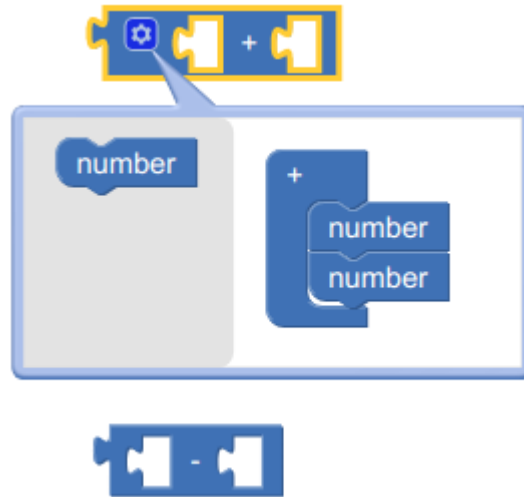
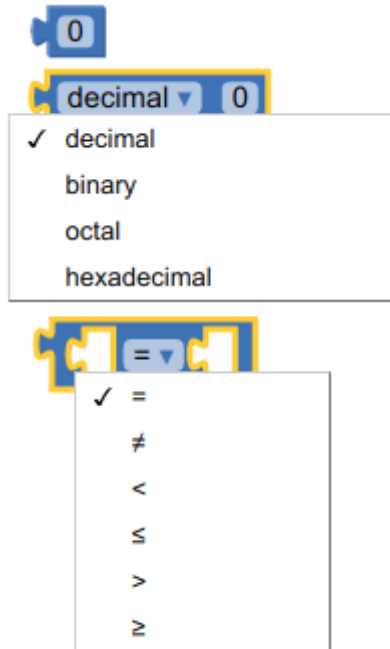
Variables locales

- Ejemplo: Formula de Herón de Alejandría para el cálculo del área de un triangulo
- Área = $\sqrt{s(s-a)(s-b)(s-c)}$ donde s es el semiperímetro del triangulo $s = \frac{a+b+c}{2}$
 - s es una variable local a esta fórmula



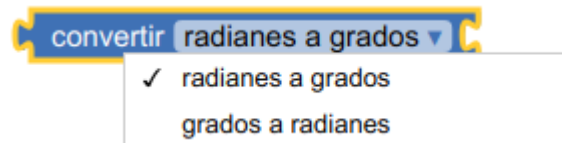
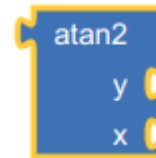
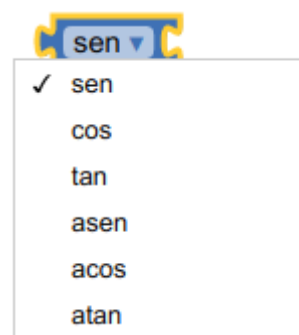
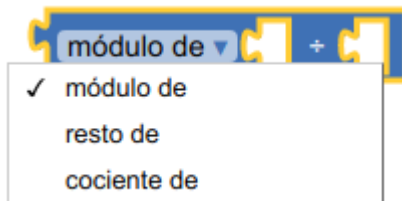
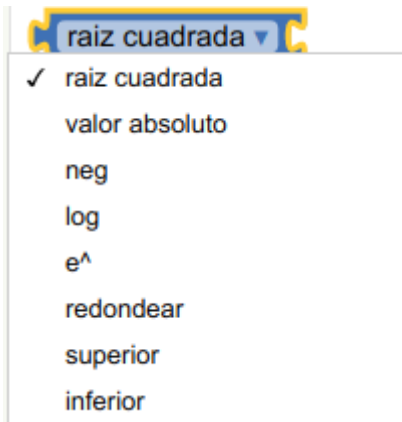
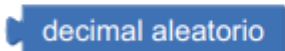
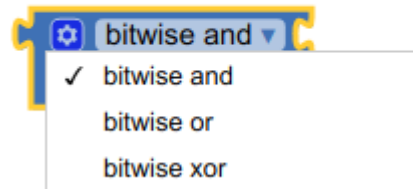
Bloques matemáticos

- AppInventor incluye bloques matemáticos básicos



Bloques matemáticos

- Exponente, números aleatorios, funciones trig.



Bloques matemáticos

- AppInventor también permite dar formato a números indicando cuantos decimales, y manejar números en diferentes bases (binario, decimal, hexadecimal)

dar formato decimal al número
decimales

¿es un número?

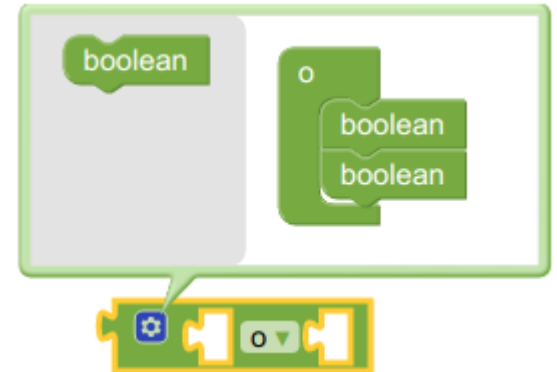
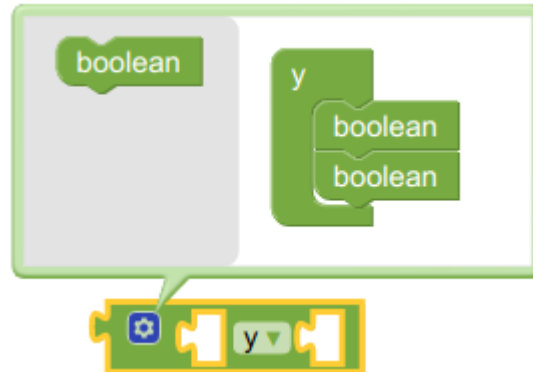
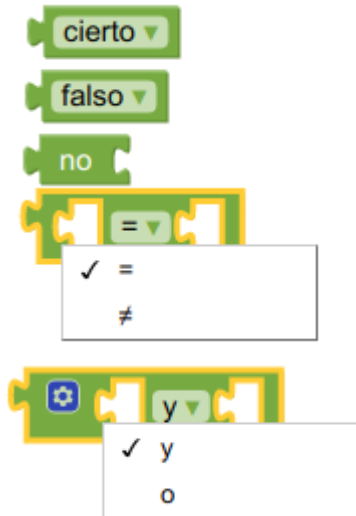
- ✓ ¿es un número?
- is base 10?
- is hexadecimal?
- is binary?

convert number base 10 to hex

- ✓ base 10 to hex
- hex to base 10
- base 10 to binary
- binary to base 10

Bloques lógicos

- Variables lógicas o Booleanas sólo pueden tomar dos valores, V ó F



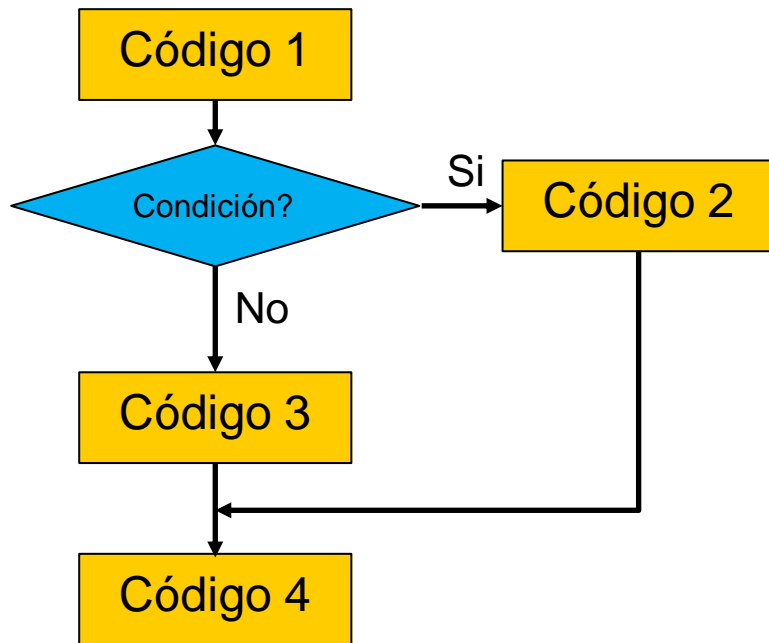
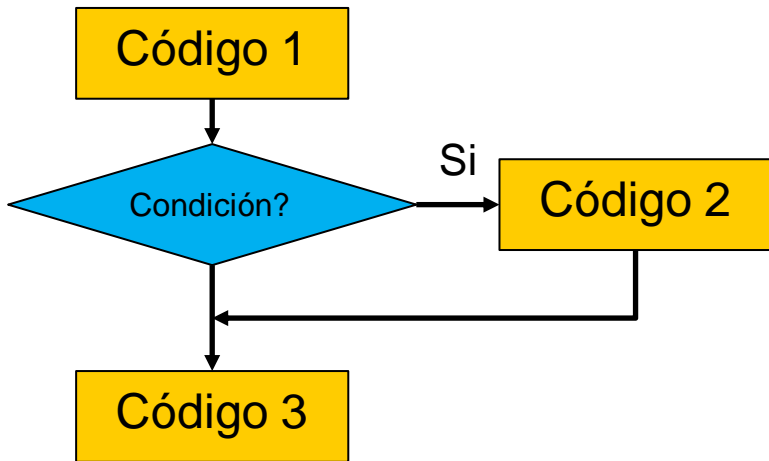
Bloques de control



- Bloques condicionales
 - Si se cumple la condición, entonces hacer algo
 - Si no se cumple la condición, entonces hacer otra cosa
 - Si no se cumple la condición, entonces verificar otra condición
- Condición debe ser evaluable a cierto ó falso

Bloques condicionales

- A veces dibujados como un diagrama de flujo



Bloques condicionales

- Pueden ser anidados

