



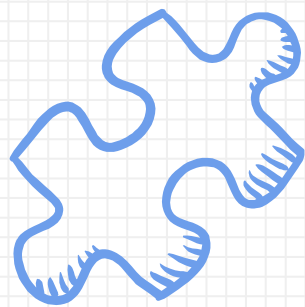
Desarrollo de APPs sin saber Programación

Profesores

Jezabel Molina Gil, Cándido Caballero Gil, Moisés Lodeiro

Santiago





Bloques

Todo el “código” está representado por bloques, los cuales son como piezas de un puzzle que se pueden arrastrar a un tablero

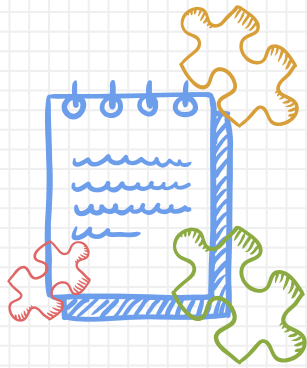
Otro modo de denominar a un bloque/s es “función/es”. Este término es el que se suele usar en lenguajes de programación

Bloques de Texto



Los bloques de texto son utilizados principalmente para dotar de texto a elementos (campos de texto, etiquetas, etc). También tiene funciones como [unir textos, recortar, comprobar si está vacío, comprobar longitudes, etc..]





Elementos y bloques

Cada elemento de la interfaz, a su vez, tiene sus propios bloques que aparecen al hacer clic sobre el elemento en los bloques

A screenshot of a visual programming interface. On the left, a 'Bloques' (Blocks) panel shows a tree view of components: 'Integrados' (Control, Lógica, Matemáticas, Texto, Listas, Colores, Variables, Procedimientos), 'Screen1' (containing 'TextBox1', 'boton1', and 'Label1'), and 'Cualquier componente'. The 'boton1' component is highlighted with a red box. Below this are buttons for 'Cambiar nombre' and 'Borrar'. On the right, a 'Visor' (Viewer) panel shows a list of blocks for the selected 'boton1' component. The blocks are organized into two sections: 'Medios' (Media) and 'Estilos' (Styles). The 'Medios' section includes blocks for 'ColorDeFondo' (Color), 'Habilitado' (Enabled), and 'Negrita' (Bold). The 'Estilos' section includes blocks for 'Cursiva' (Italic). Each block has a 'poner' (set) or 'ejecutar' (execute) action, and some have a 'como' (as) dropdown menu. The blocks are color-coded: yellow for actions and green for style setters.

Al igual que los elementos de la interfaz, los bloques también se pueden arrastrar al panel (en blanco)

La mochila sirve para guardar bloques y poder utilizarlos en todos nuestros proyectos sin necesidad de volver a crearlos

Bloques

- Integrados
 - Control
 - Lógica
 - Matemáticas
 - Texto
 - Listas
 - Colores
 - Variables
 - Procedimientos
- Screen1
 - Botón1
 - Imagen1
 - Lienzo1
 - Cámara1
 - Reloj1
- Cualquier componente

Cambiar nombre Borrar

Visor

0

say for 2 secs

+

-

x

/

^

entero aleatorio entre 1 y 100

decimal aleatorio

semilla aleatoria es

La mochila sirve para guardar bloques y poder utilizarlos en todos nuestros proyectos sin necesidad de volver a crearlos

Las tuercas que aparecen en algunos elementos indica que ese elemento tiene opciones de configuración

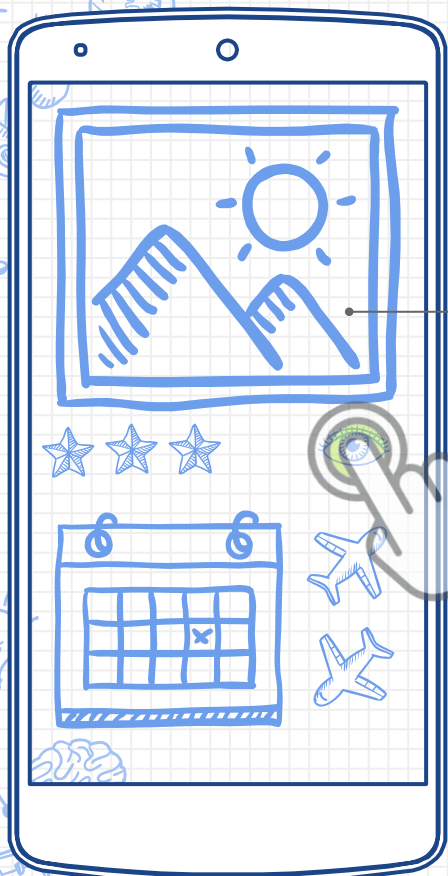
number

+

number

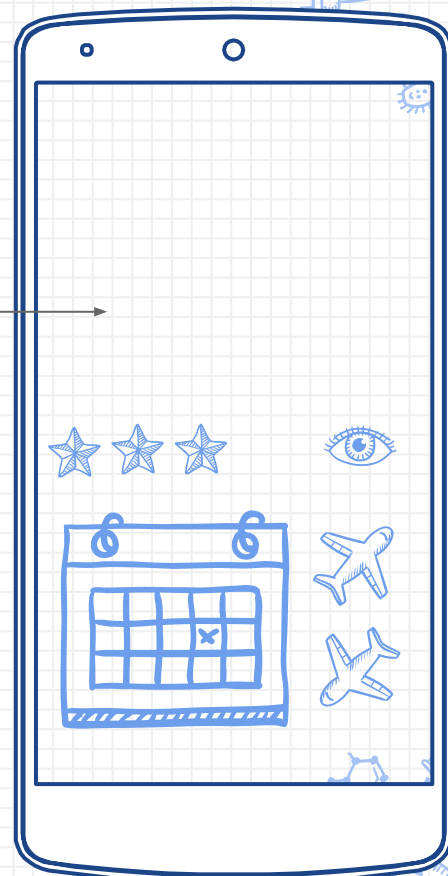
number

Los elementos gráficos y los bloques están enlazados, generalmente mediante eventos de Control



```
cuando Button1 .Clic
ejecutar
  si Imagen1 . Visible
  entonces poner Imagen1 . Visible como falso
  si no poner Imagen1 . Visible como cierto
```

“Cuando se haga clic en el botón (o icono), ocultar la imagen si está visible y viceversa”

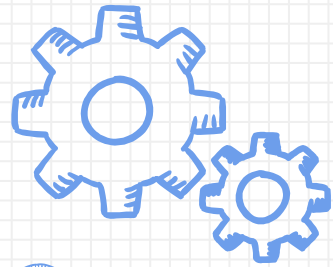


Calcula tu IMC

$$\frac{\text{peso}}{\text{altura} * \text{altura}} = \frac{\text{peso}}{\text{altura}^2}$$

La altura debe estar en **m**, el peso en **kg**

- ✗ La interfaz debe ser como la de la imagen
- ✗ Para el cuadrado de la altura usa un bloque de matemáticas
- ✗ Calcular el IMC al pulsar el botón “Calcular”
- ✗ Si el IMC es ≥ 25 , deberá mostrar un mensaje de sobrepeso. En caso contrario mostrar “OK”



10 minutos

Screen1

Peso (en kg)

Altura (en m)

Calcular

Tu IMC es:

Todos los caminos llevan a Roma. Un problema puede (y suele) tener más de una solución

Consejo: Haz uso del bloque “unir” de texto para juntar dos o más textos

Consejo: El mensaje puede ser una etiqueta sin texto (a rellenar a posteriori)


```

cuando boton_calcular .Clic
ejecutar poner calculoimc . Texto como unir " Tu IMC es: "
      (campopeso . Texto / altura . Texto ^ 2)
si (campopeso . Texto / altura . Texto ^ 2) >= 25
entonces poner texto . Texto como " ¡Sobrepeso! "
si no poner texto . Texto como " OK "

```

Esta opción es tan válida como otra cualquiera PERO no es recomendable porque repite cálculos

```

inicializar global IMC como 0
cuando boton_calcular .Clic
ejecutar poner global IMC a (campopeso . Texto / altura . Texto ^ 2)
      poner calculoimc . Texto como unir " Tu IMC es: "
      tomar global IMC
si (tomar global IMC) >= 25
entonces poner texto . Texto como " ¡Sobrepeso! "
si no poner texto . Texto como " OK "

```

Bloques inconexos pero "IMC" se puede usar al presionar el botón

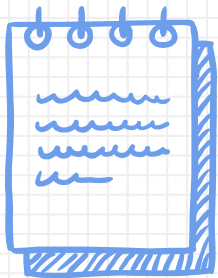
Mejor usar variables (si pueden ser locales, mejor). La diferencia es que la global la puedes usar en cualquier bloque mientras que la local sólo dentro de su bloque

Aquí IMC sólo se puede usar dentro del bloque naranja de la variable. Si intentamos usarlo fuera no nos dejará (ni siquiera mostrará la variable como existente)

```

cuando boton_calcular .Clic
ejecutar inicializar local IMC como (campopeso . Texto / altura . Texto ^ 2)
en poner calculoimc . Texto como unir " Tu IMC es: "
      tomar IMC
si (tomar IMC) >= 25
entonces poner texto . Texto como " ¡Sobrepeso! "
si no poner texto . Texto como " OK "

```



Los iconos utilizados en esta presentación han sido utilizados con licencia abierta y han sido extraídos de la página Flatlcon y de los propios de la presentación.

