Listas de datos

547120 Introducción a ELN

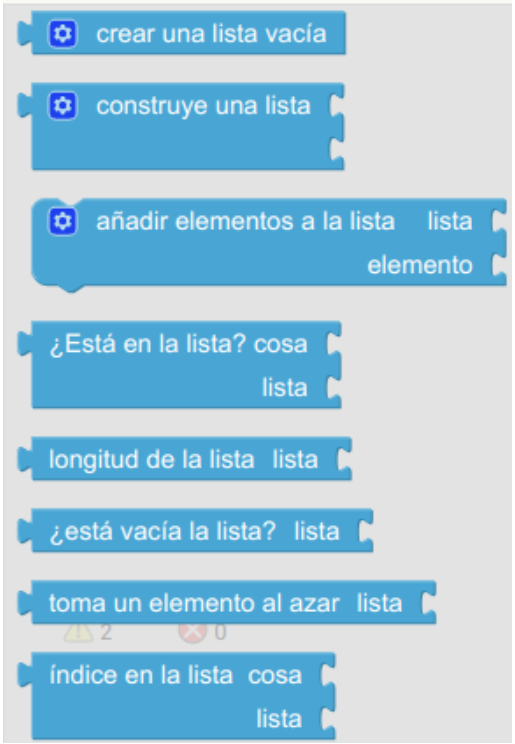
mariomedina@udec.cl

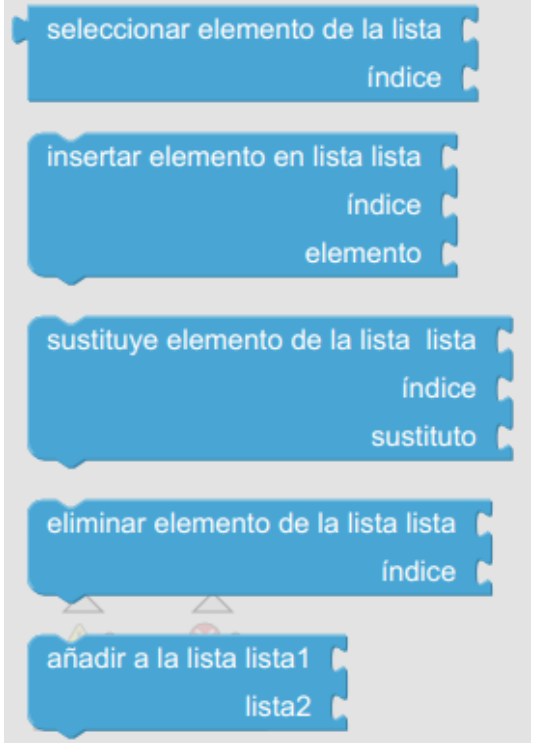
Listas

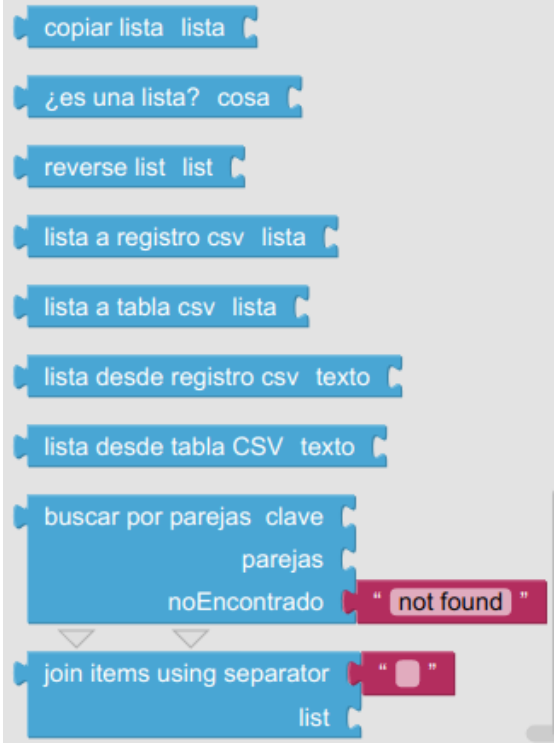
- En clases anteriores, hablamos sobre variables globales y locales
- Presentamos como crear variables globales y locales usando bloques de inicialización

- También es posible crear listas de elementos usando esos bloques de inicialización y bloques integrados para listas

Bloques integrados para Listas

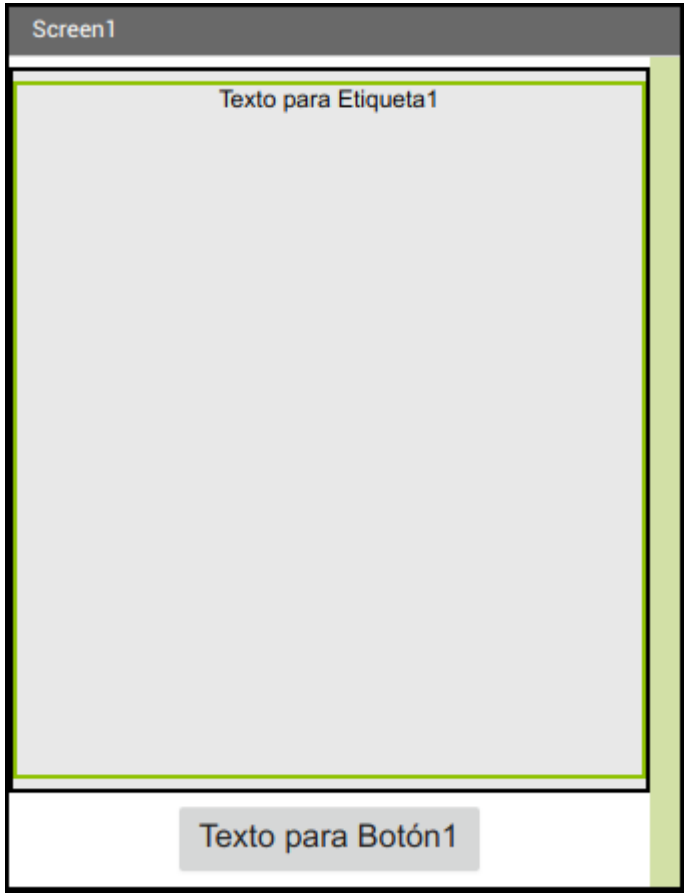






Ejemplo

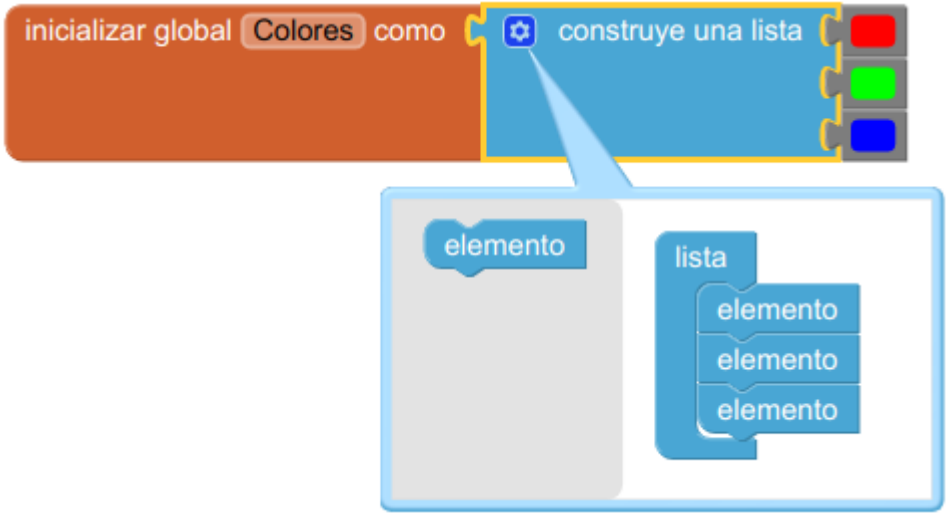
- Vamos a crear una app que cambie el color de la etiqueta por programa usando una lista de colores



Creando una lista de colores

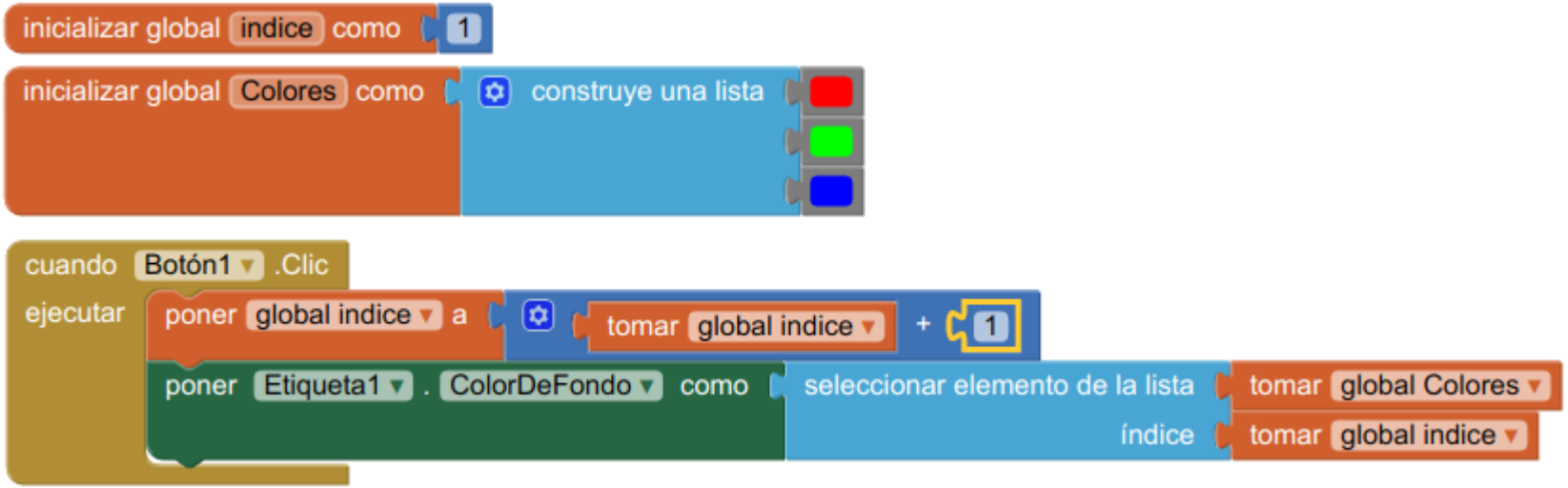
- Definir una variable global *Colores* como una lista de tres elementos

Índice	Color
1	Rojo
2	Verde
3	Azul



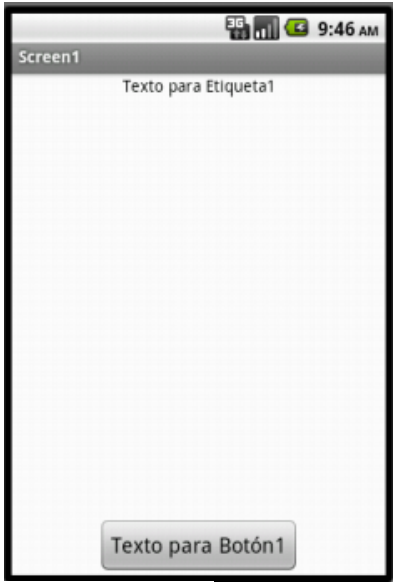
Cambiando el color

- Definir un manejador para el evento *Boton1.Clic* que cambie el color de la etiqueta
  - Requiere definir variable global *indice*

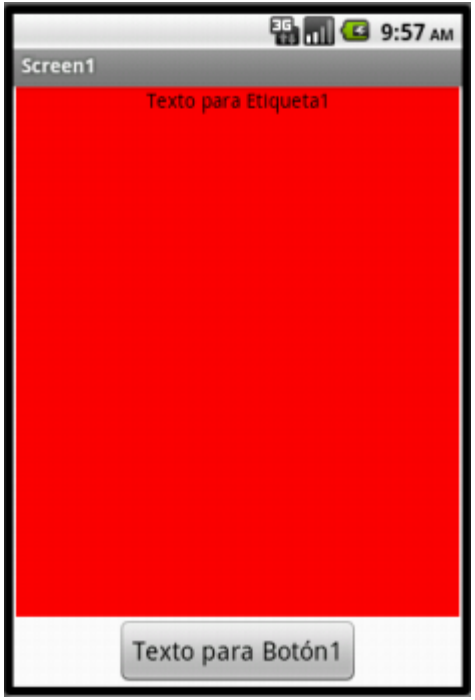


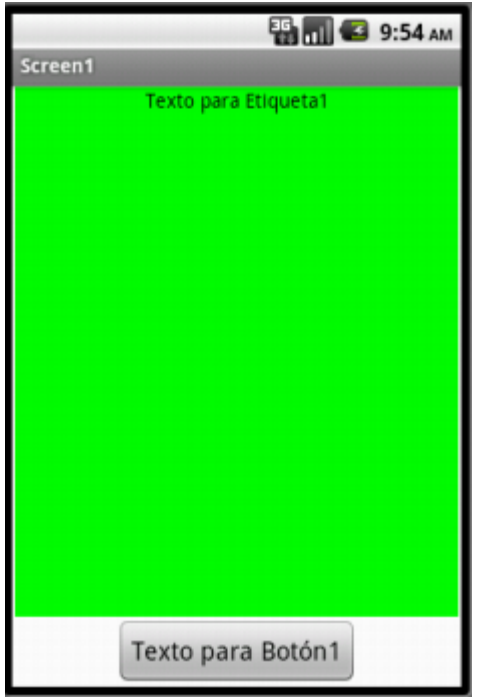
Definiendo el color inicial

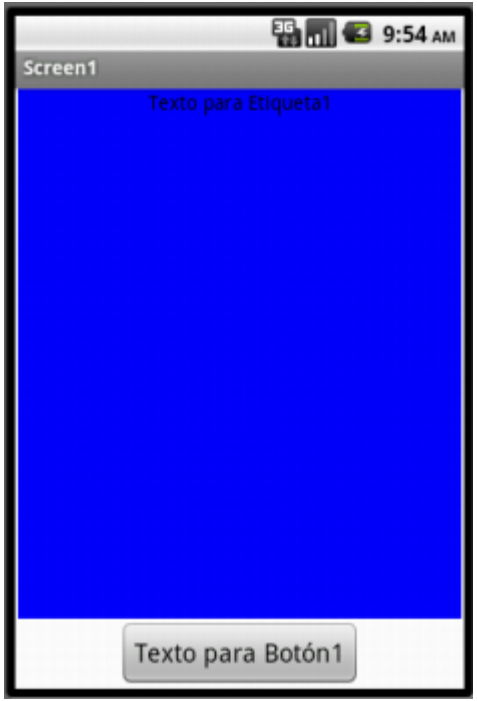
- Etiqueta comienza en blanco
- Definir un color inicial al inicializar la pantalla
  - Color inicial dado por valor inicial de la variable global *indice*



Cambiando el color

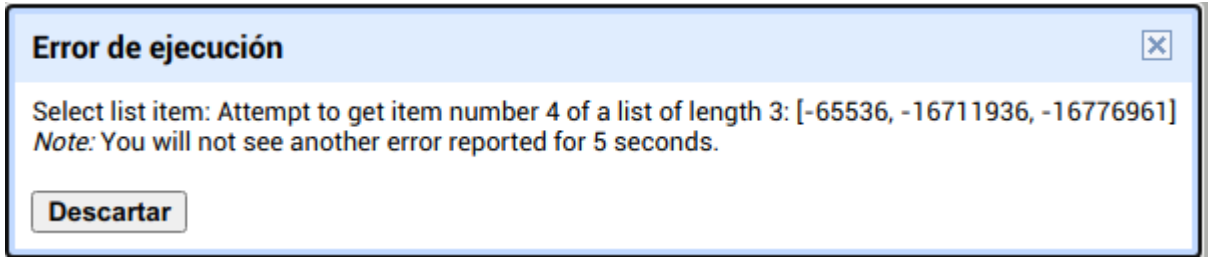






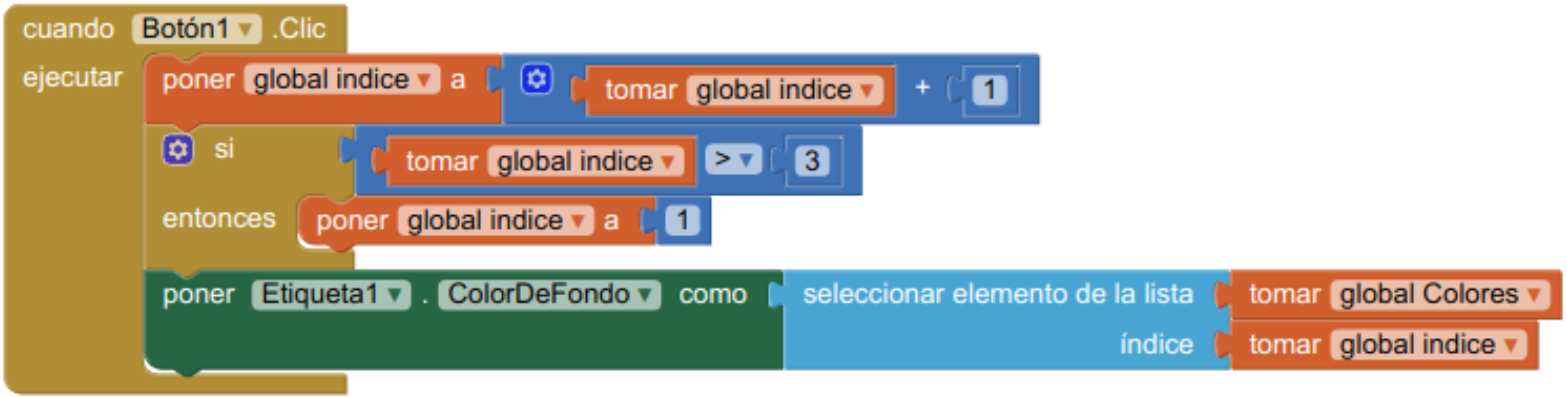
Fuera de rango

- Al presionar por tercera vez el botón, el índice a la lista queda fuera de rango
  - Acceso al cuarto elemento de una lista de tres elementos



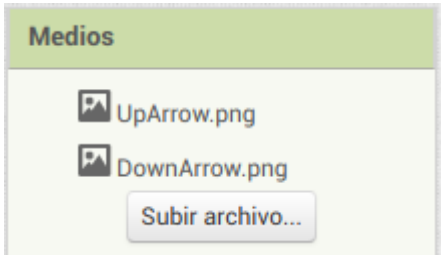
Fuera de rango

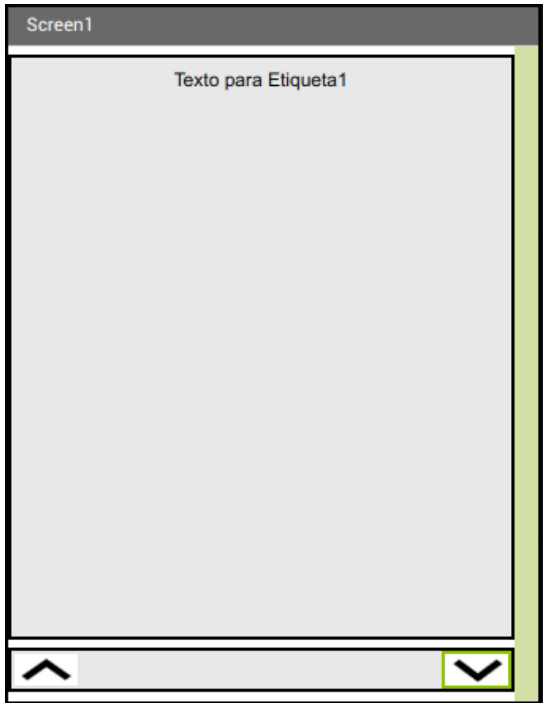
- Una manera de evitar salirse del rango es cautelar los límites con un condicional
  - Condicional cambia si se hace antes o después del incremento del índice



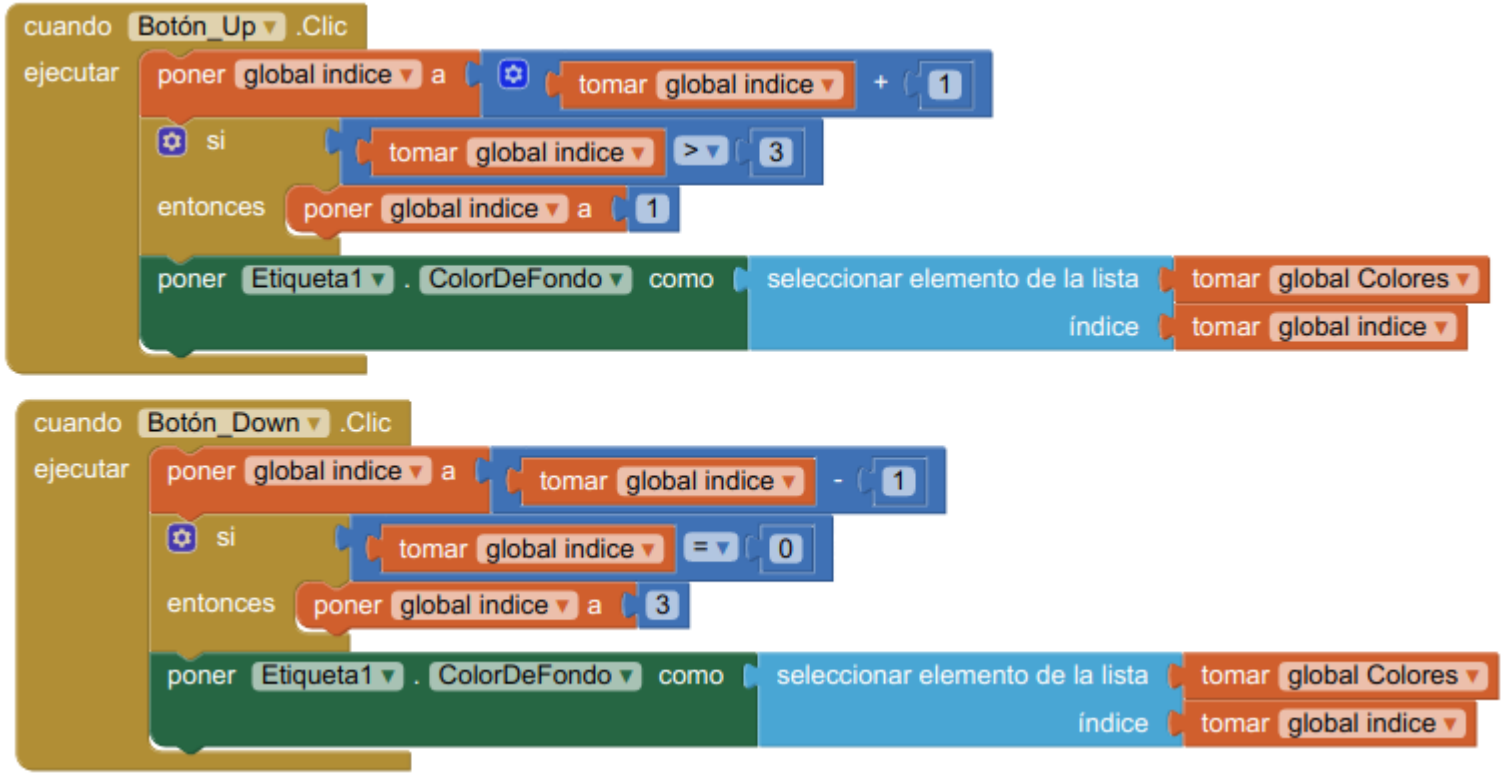
Agregando otro botón

- Agregar otro botón para recorrer la lista de colores en la otra dirección



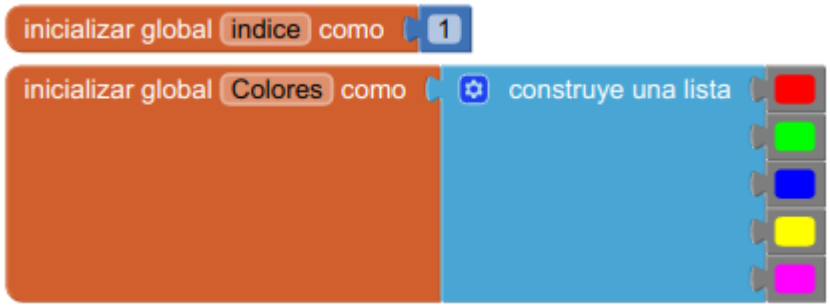


Dos botones


cuando Botón_Up .Clic
ejecutar
poner global indice a
tomar global indice + 1
si
tomar global indice > 3
entonces
poner global indice a 1
poner Etiqueta1 . ColorDeFondo como
seleccionar elemento de la lista
tomar global Colores
índice
tomar global indice
cuando Botón_Down .Clic
ejecutar
poner global indice a
tomar global indice - 1
si
tomar global indice = 0
entonces
poner global indice a 3
poner Etiqueta1 . ColorDeFondo como
seleccionar elemento de la lista
tomar global Colores
índice
tomar global indice

Agregando más colores

- Agregar más colores a la lista

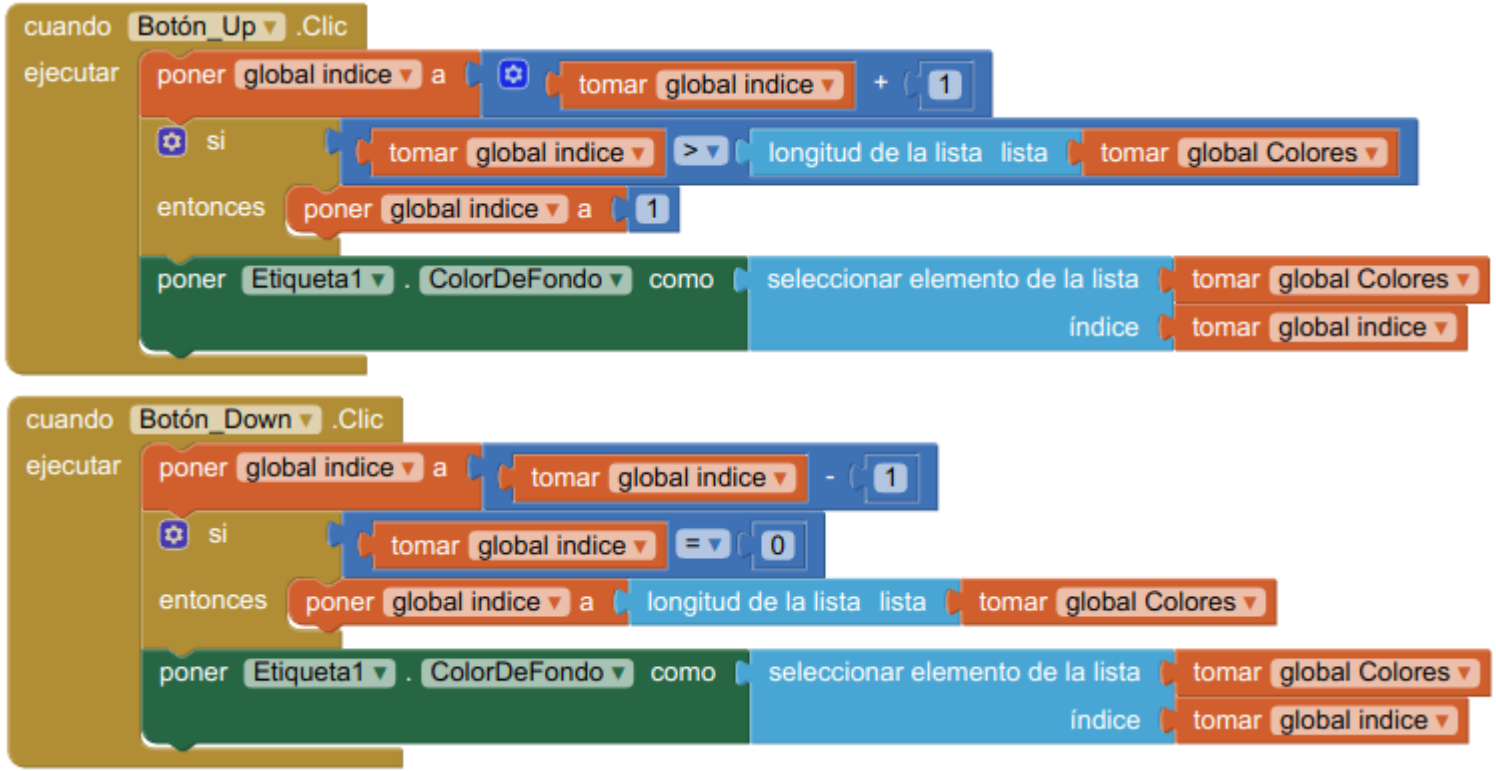


- ¿Pero, por que los nuevos colores no aparecen en pantalla?

Agregando más colores

- Los límites definidos en el código para el índice son rígidos
  - Mejor calcular los límites del índice en base al número de elementos de la lista
  - Si el índice es menor a 1, nuevo índice debe ser el número de elementos de la lista
  - Si índice es mayor que el número de elementos de la lista, nuevo índice debe ser 1

Agregando más colores



Mostrando el nombre del color

- Hasta ahora, sólo hemos cambiado el color de la etiqueta en base a una lista de colores
- ¿Cómo mostrar en pantalla el nombre del color?

Índice	Color	Nombre
1	Rojo	«Rojo»
2	Verde	«Verde»
3	Azul	«Azul»
4	Amarillo	«Amarillo»
5	Lila	«Lila»

Mostrando el nombre del color

- Una solución posible es tener dos listas y recorrerlas con el mismo índice

Índice	Color
1	Rojo
2	Verde
3	Azul
4	Amarillo
5	Lila

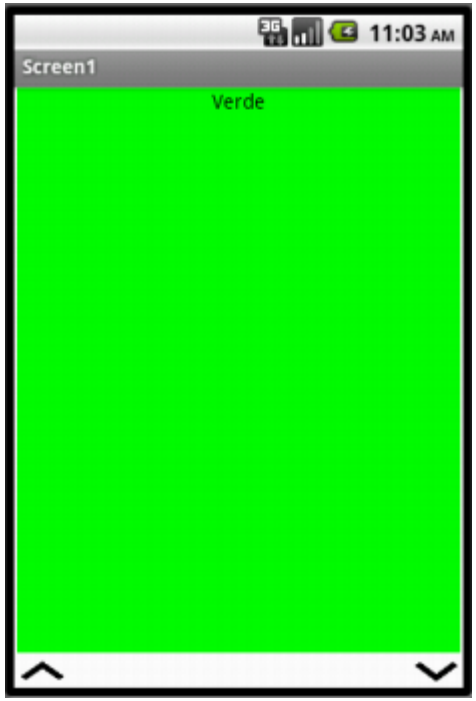
Índice	Nombre
1	«Rojo»
2	«Verde»
3	«Azul»
4	«Amarillo»
5	«Lila»

Mostrando el nombre del color

- Nueva lista *Nombres*

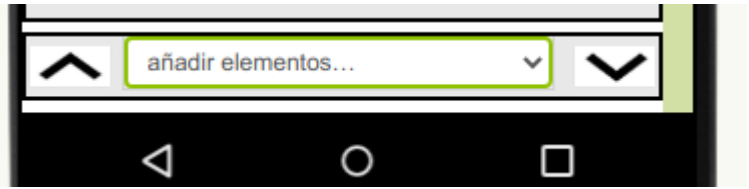
Mostrando el nombre del color

- Modificar código de botones *Up* y *Down* para que modifiquen el texto de la Etiqueta



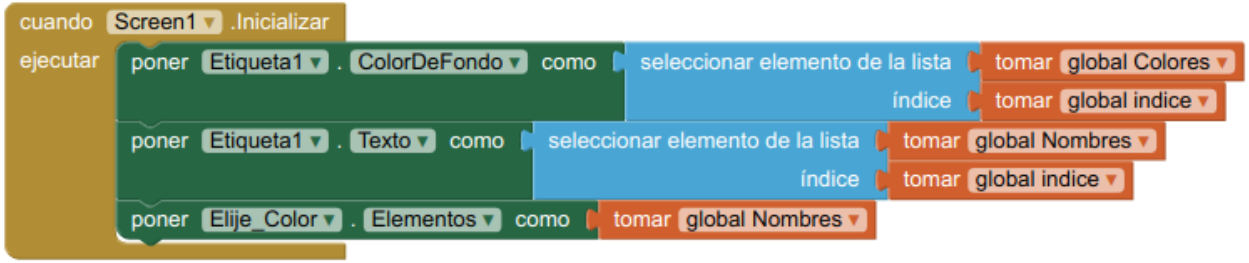
Escoger el color

- Nuestra app permite cambiar el color de la etiqueta recorriendo una lista de colores
  - Hacer que el usuario pueda escoger el color directamente de la lista mediante un *Desplegable*



Inicializando el desplegable

- Definir la lista de *Nombres* como los elementos a mostrar por el objeto *Desplegable*

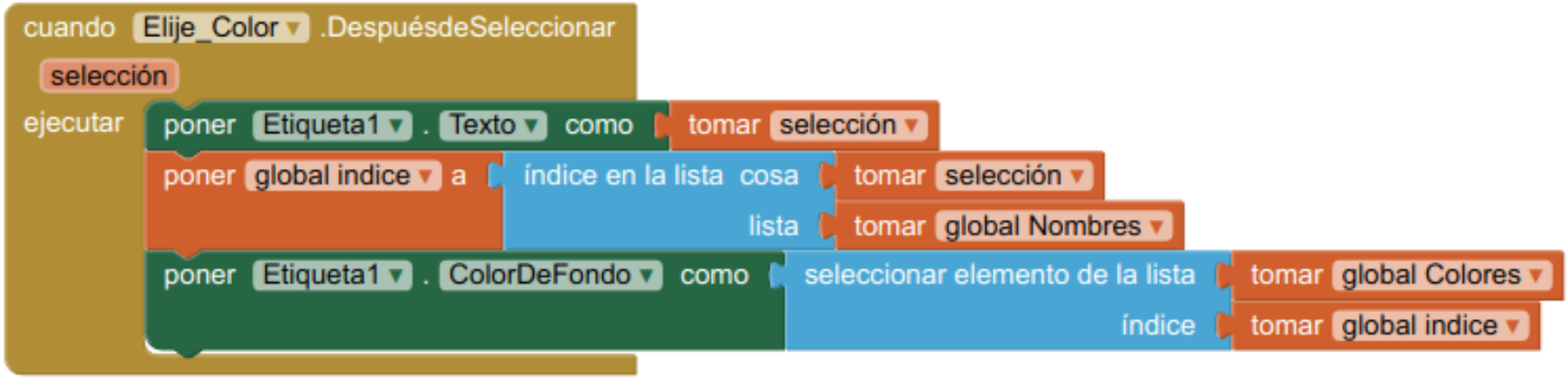


Usando el desplegable

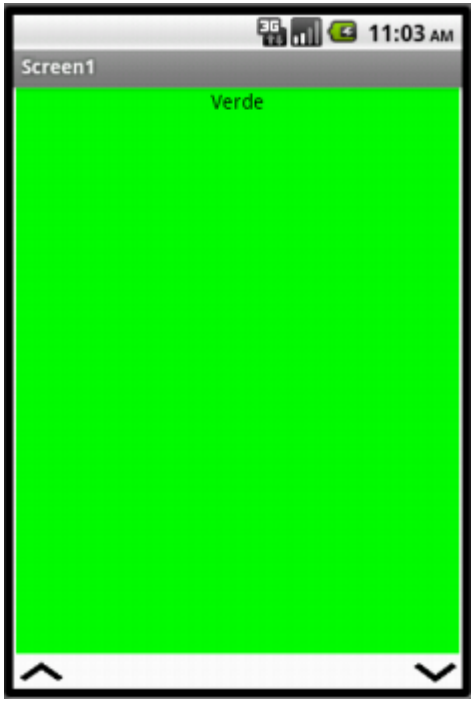
- Una vez seleccionado un color, actualizar el índice, pintar la pantalla y mostrar el nombre
- Pero, el desplegable retorna el nombre del color
  - ¿Cómo saber el color correspondiente al nombre?
  - Buscar el índice del nombre seleccionado en la *Lista de Nombres*
  - Usar ese índice para encontrar el color en la *Lista de Colores*

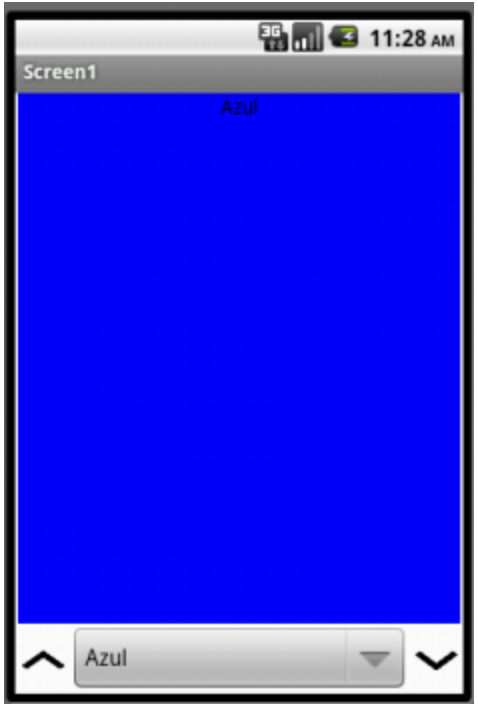
Usando el desplegable

- *seleccion* contiene el nombre del color escogido
  - Se usa para encontrar el índice del nombre en Nombres y ese índice se usa en lista de Colores



Usando el desplegable





Agregando un nuevo color

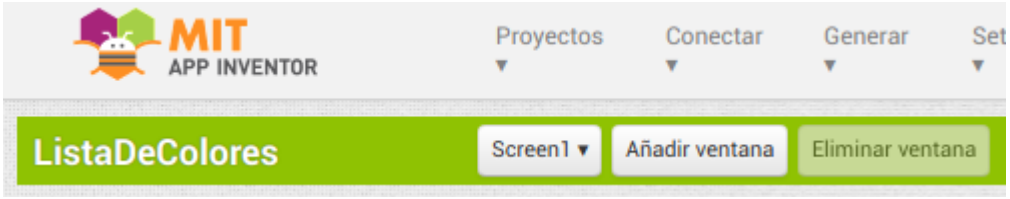
- Hasta ahora, nuestra aplicación permite que el usuario escoja el color de la etiqueta de una lista de colores predeterminada
  - ¿Qué hay que hacer para que el usuario pueda definir sus propios colores?
  - ¿Cómo se define un color en AppInventor?
  - ¿Qué es el modelo RGB?

El modelo RGB

- Modelo de color basado en la síntesis aditiva de colores
  - Representa un color mediante la suma de los tres colores de luz primarios rojo (R), verde (G) y azul (B)
- En App Inventor, cada color primario puede tomar un valor entre 0 y 255
  - Eso da un total de 16,777,216 colores

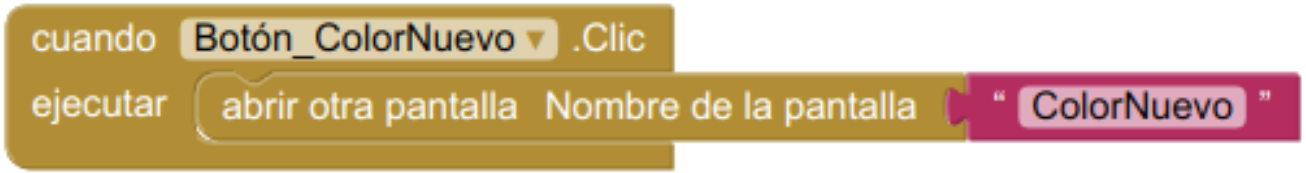
Definiendo un nuevo color

- Para definir un nuevo color, usaremos una nueva ventana llamada «ColorNuevo»
- En AppInventor, esta ventana es independiente de la ventana inicial *Screen1*
  - No comparte bloques ni componentes



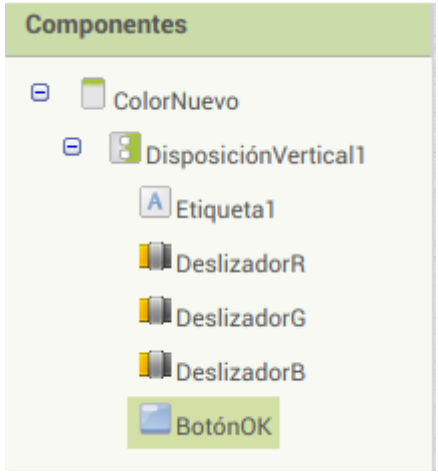
Invocando a la nueva ventana

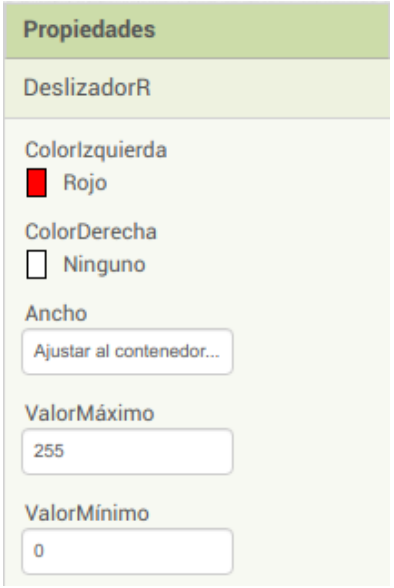
- Agregaremos además un botón *ColorNuevo* que invoque a esta nueva ventana
  - Nombre de la nueva ventana es de tipo Texto

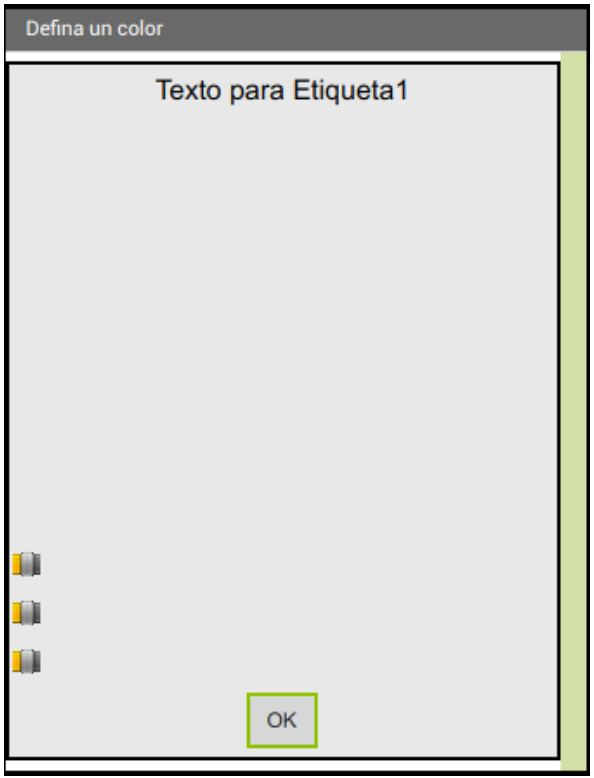


Pantalla de selección de color

- Etiqueta para mostrar el color
- Deslizadores R, G y B
- Botón OK

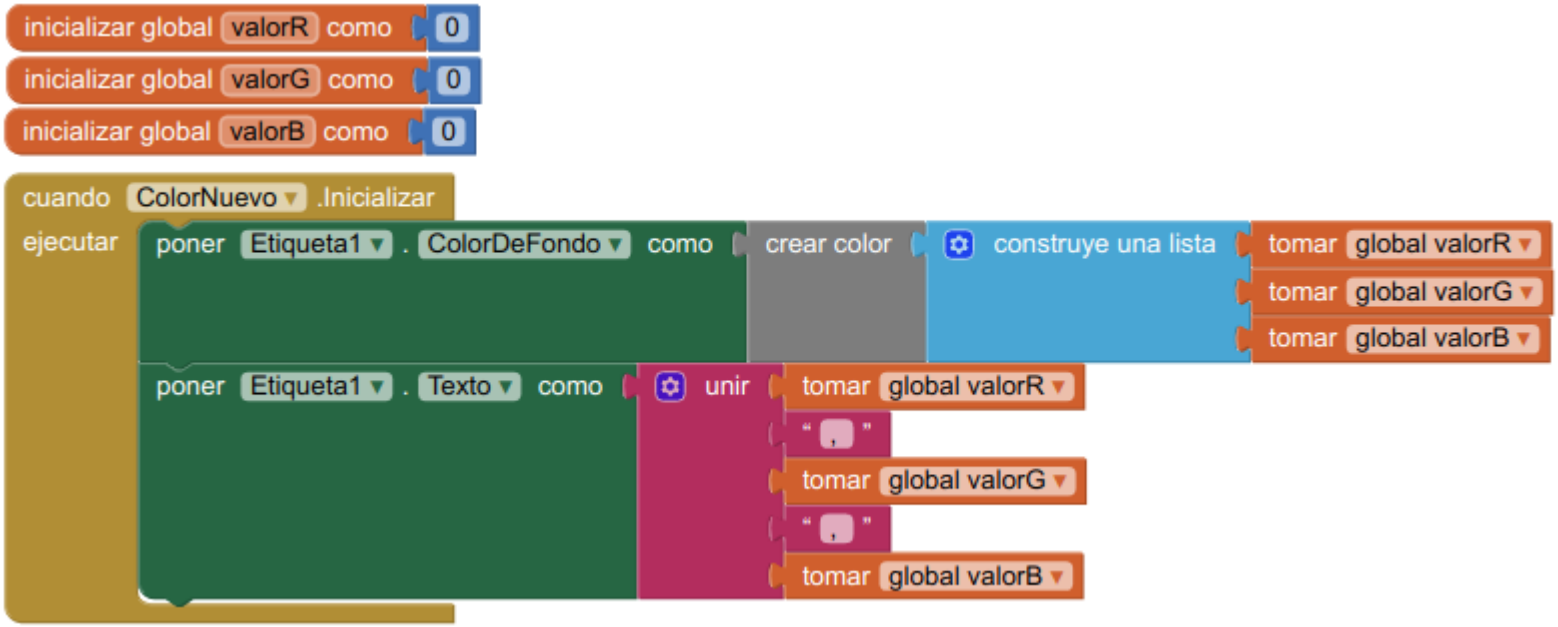






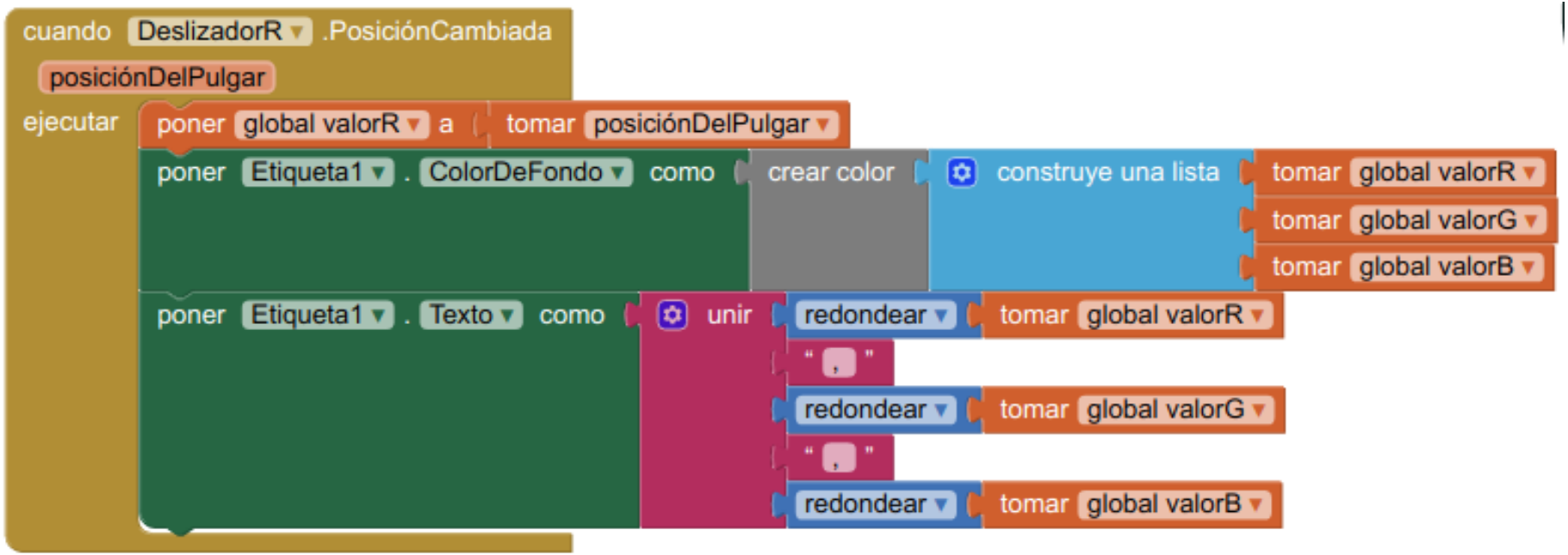
Inicialización de pantalla

- Define variables globales y color inicial



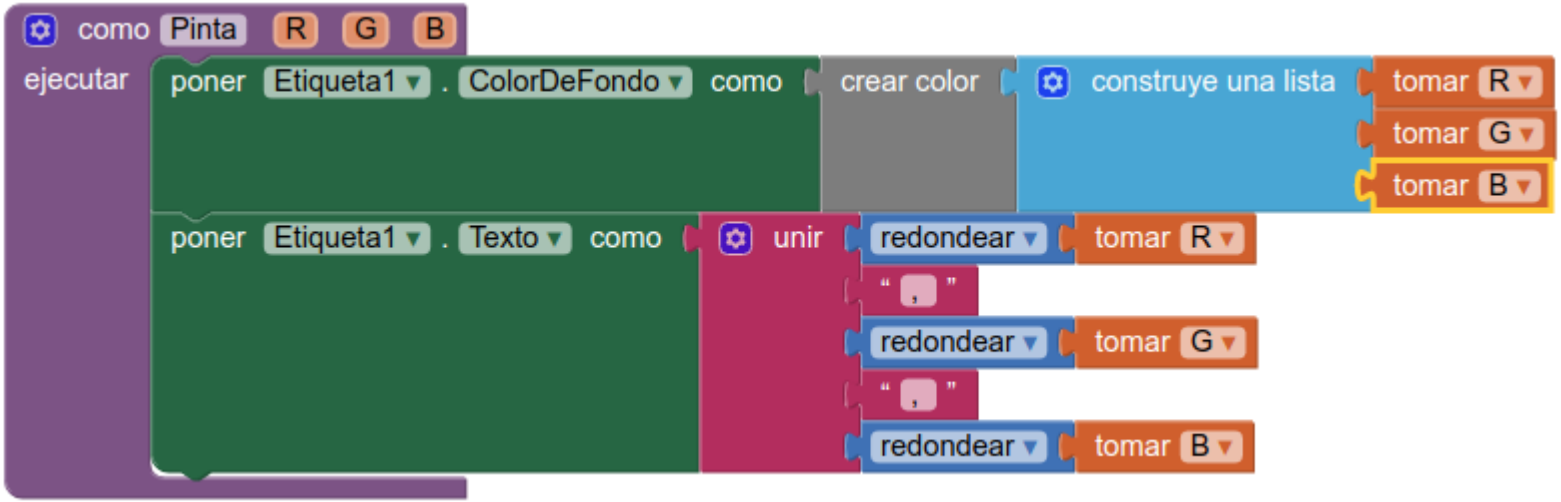
Leyendo valores de los *Sliders*

- Sliders definen valores para R, G y B

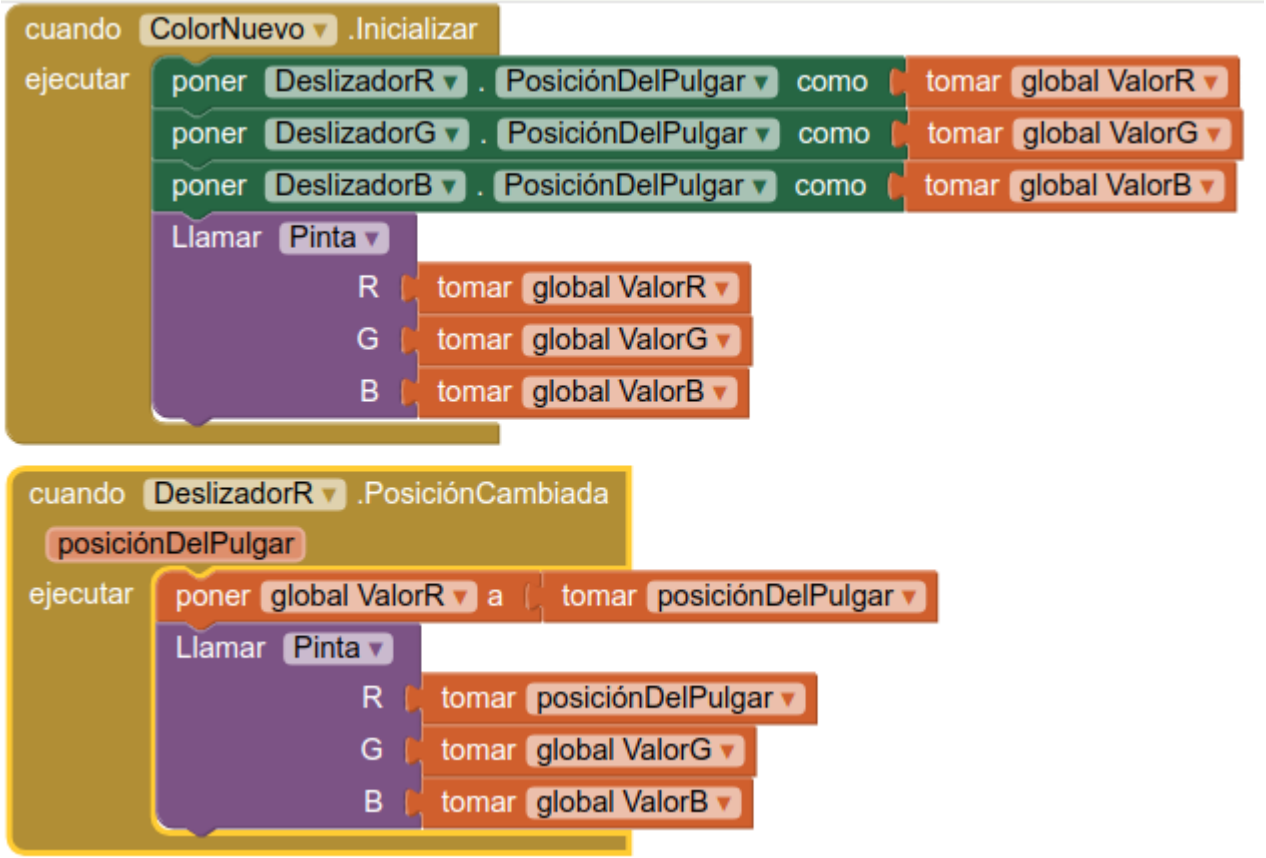


Creando un procedimiento

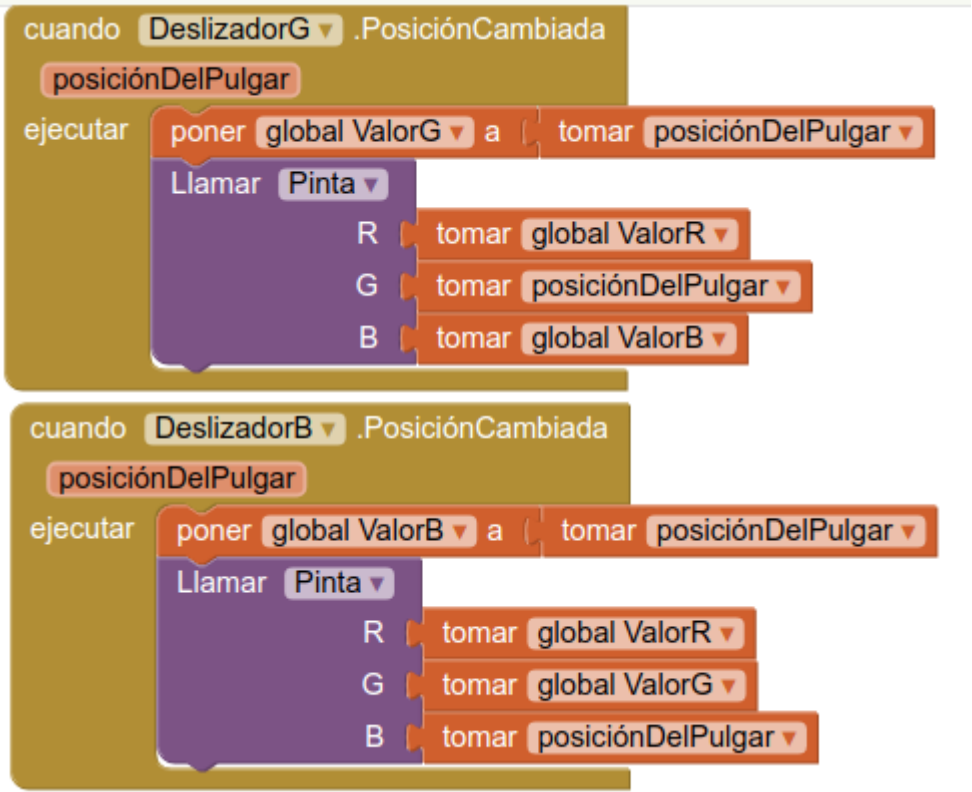
- Evita repetir código
- Reduce las modificaciones si hay que hacer cambios



Usando el procedimiento

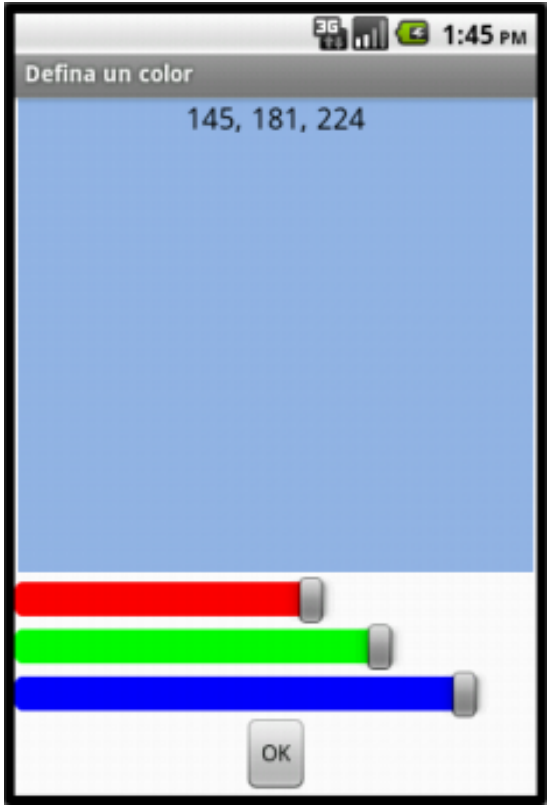


Usando el procedimiento



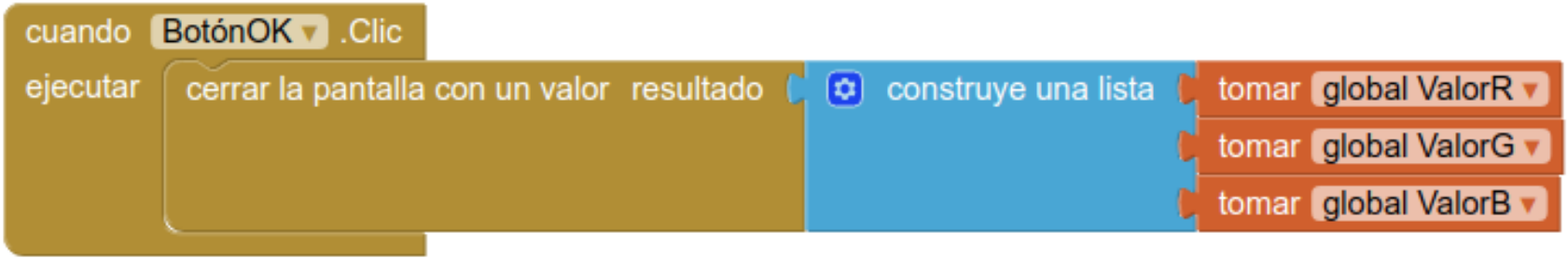
Escogiendo un color

- Una vez que el usuario presiona el botón, tenemos que agregar el nuevo color a la lista de colores de la pantalla anterior con el nombre «Mi Color»
- ¡Pero, las pantallas no comparten componentes ni bloques!



Pasando datos entre ventanas

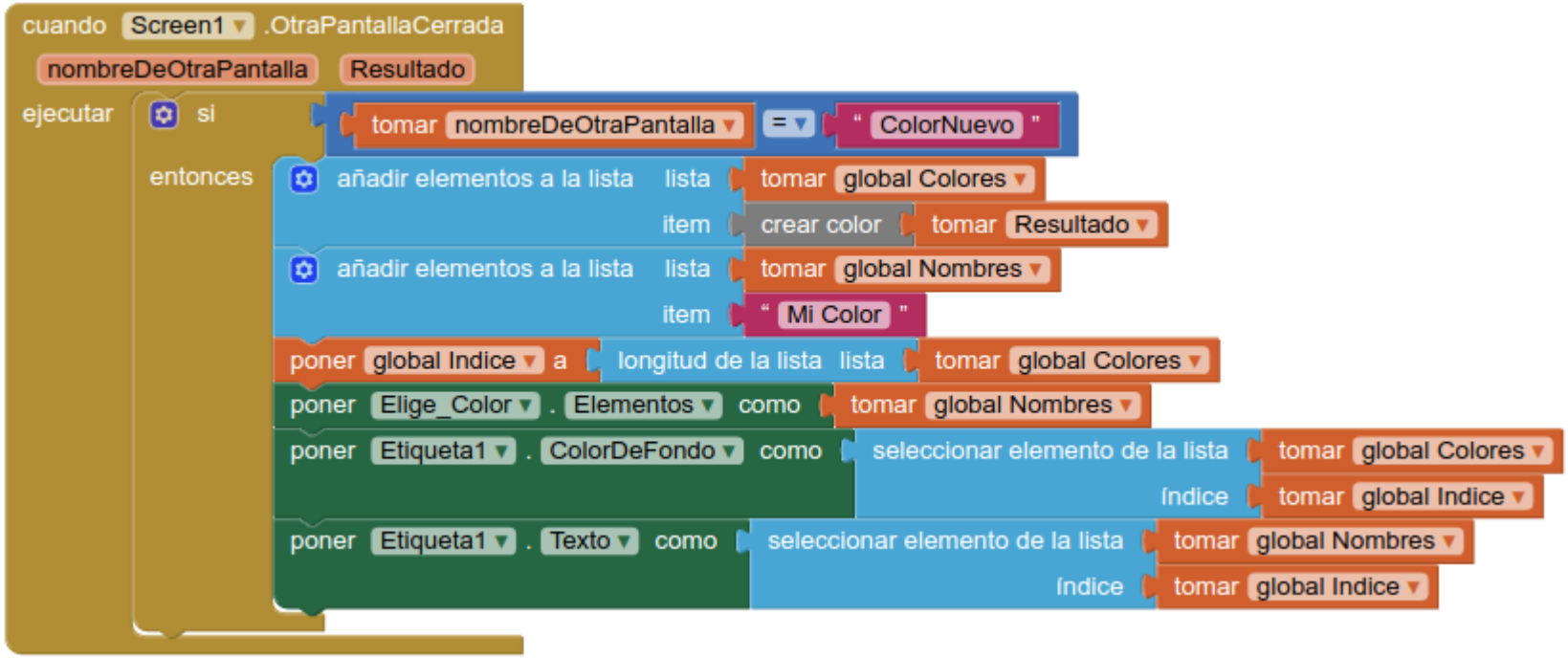
- La solución es que la ventana *ColorNuevo* le envíe el color nuevo a la ventana *Screen1* al presionar el botón OK
  - Valores R, G, B del color nuevo deben retornarse como una lista



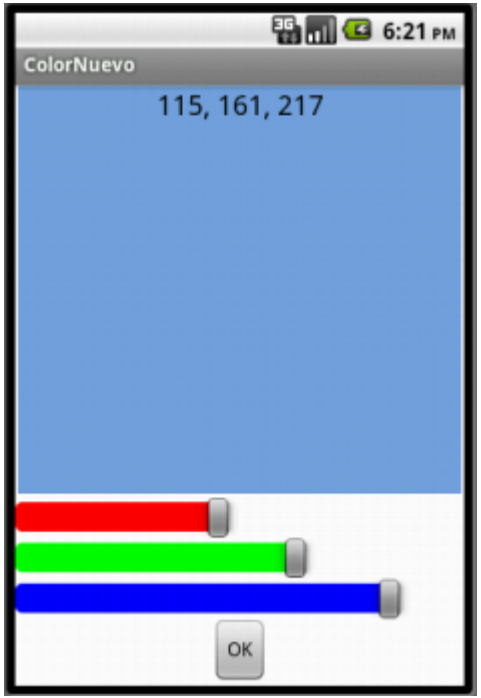
Recibiendo el nuevo color

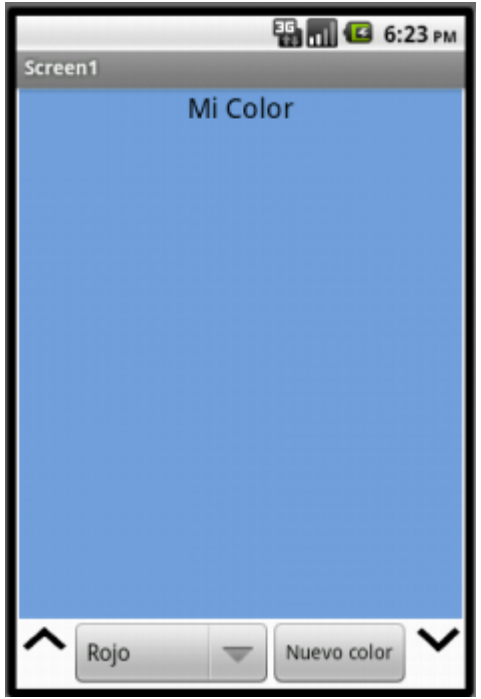
- La ventana *Screen1* recibe un resultado al cerrarse la ventana *ColorNuevo*
- Agregar el nuevo color a la lista de colores
- Agregar además el nombre «MiColor» a la lista de nombres
  - Usarlos como el nuevo color de fondo y el nuevo texto a desplegar
  - Actualizar el índice

Recibiendo el nuevo color



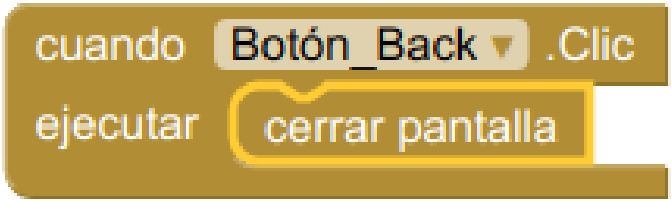
Mostrando el nuevo color





Retornando sin crear un color

- Debemos darle al usuario la chance de salir de la ventana "Color Nuevo" sin definir un nuevo color
  - Agregar otro botón *Back*



Posibles mejoras

- Aplicación sólo define valores R, G y B del color
- Solicitar que el usuario además defina el valor de transparencia (α) para su color
  - Modelo RGBA usa un cuarto valor Alpha para indicar la transparencia de un pixel donde 0: transparente, 255: completamente opaco

Posibles mejoras

- Aplicación desarrollada permite guardar sólo un nuevo color, bajo el nombre "Mi Color"
- Solicitar que el usuario, además de los valores R, G y B, ingrese un nuevo nombre para su color mediante un Notificador
- Agregar más colores y nombres a las listas respectivas