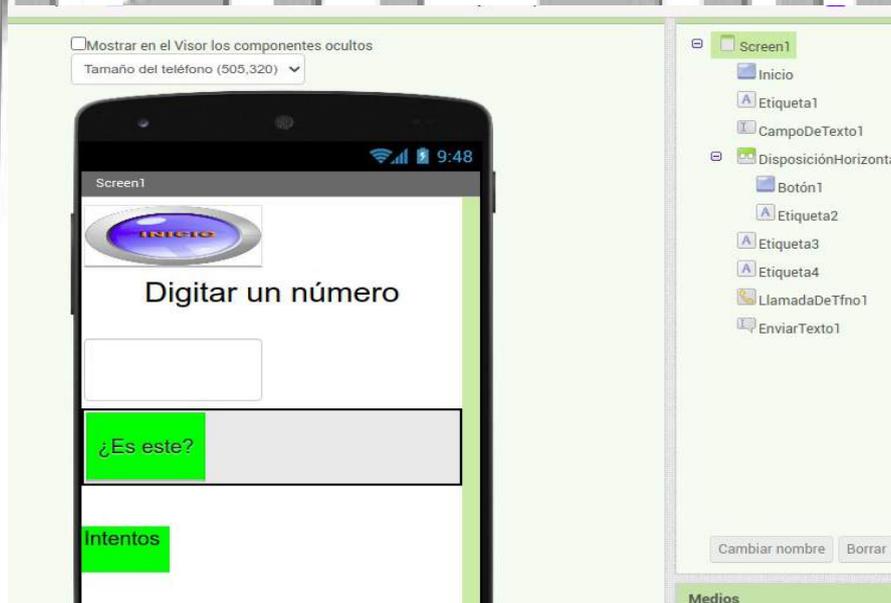


AppInventor

Teórico 3

Bloques (Programación)

Comenzaremos a programar la aplicación realizada en el teórico anterior llamada “Adivina el número” (Ver teórico 2).
Nuestra aplicación tenía el siguiente diseño:



Para explicar y entender lo que deseamos programar describiremos lo que se pretende con la misma.

Nuestra app presenta un botón (Inicio) que permitirá reiniciar una partida nueva, también hay un campo de texto que nos permitirá escribir un número y un segundo botón (¿Es este?) que comparará el número ingresado con el número creado al azar que debemos adivinar.

A la derecha del botón “¿Es este? Tendremos una etiqueta que nos dirá pistas (El número es mayor, El número es menor o Has acertado) y bien abajo otra que nos mostrará los intentos realizados.

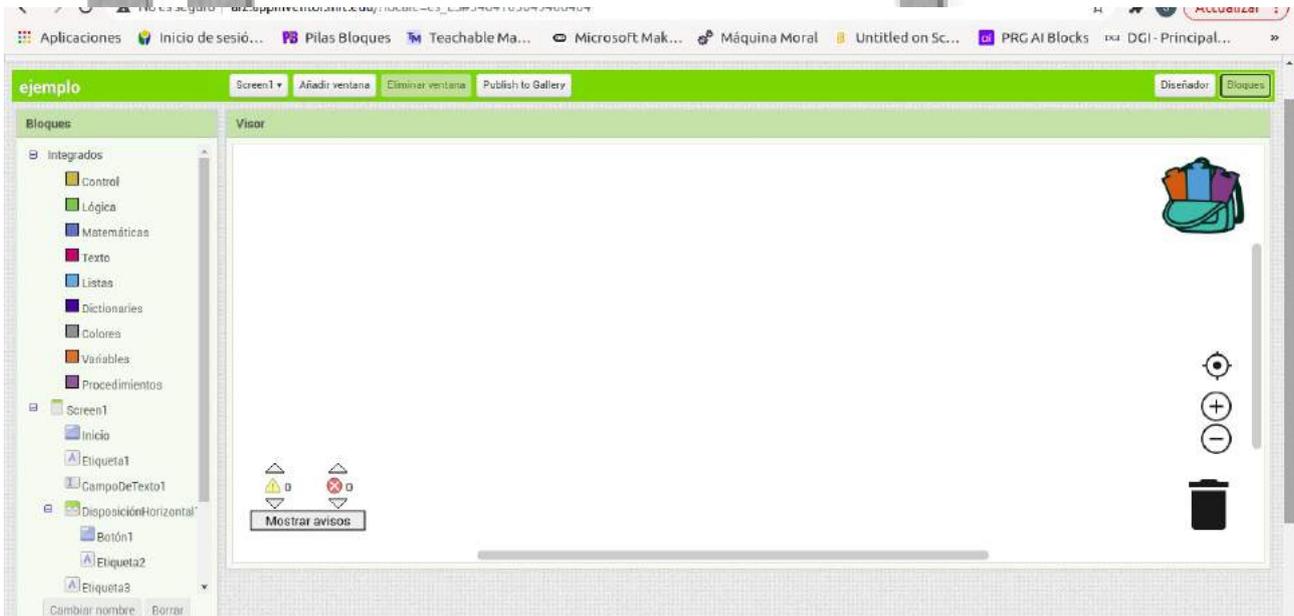
Comenzaremos a programar.

El primer paso será ir al botón “Bloques” ubicado en la parte superior derecha de la ventana.



Profesor: Marcelo Rebellato

Cuando presionamos la opción “Bloques” veremos la siguiente ventana:



Como se puede apreciar en la parte izquierda vemos una serie de paletas que trae el programa (Texto, Matemática, Lógica, Control, etc.) y debajo los componentes utilizados en la aplicación.

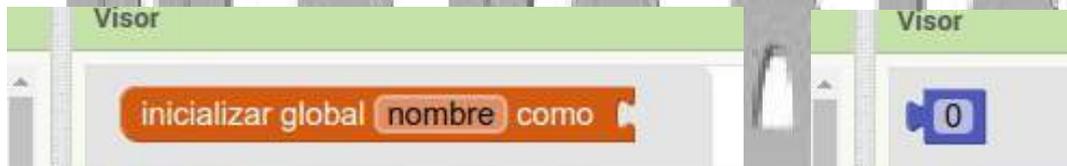
Para poder poner los bloques correspondientes a cada paleta o componente debemos hacer un clic sobre ellos y luego sobre el bloque deseado como se puede apreciar en la siguiente imagen donde realizamos clic sobre botón “Inicio” y luego el bloque “Cuando ‘Inicio’ clic”.



Profesor: Marcelo Rebellato

Comenzaremos nuestra programación creando las variables MIO (Guardará el número ingresado por el usuario) y Azar (Guardará el número que tenemos que adivinar, generado por la aplicación).

Para esto debemos ir a la paleta variable y matemática y utilizaremos los bloques siguientes:



Quedando la programación



Recordar poner el nombre de la variable dentro del bloque "Inicializar global como" y crear la variable "Intentos".

A continuación insertaremos de componente "Inicio" (Botón) el bloque "Cuando 'Inicio' clic - Ejecutar"



Programaremos el primer botón, recordemos que se quiere iniciar una nueva partida. Para ello cuando presionemos el mismo debe generar un número al azar y guardarlo en la variable "AZAR", luego lo mostrará en nuestra cuarta etiqueta que luego de terminada se ocultará, por último la variable intentos debe quedar en cero.



Profesor: Marcelo Rebellato

Es momento de programar el segundo botón (Permite comparar y verificar si adivinamos el número).

En este botón debemos hacer que lo digitado en el campo de texto se guarde en la variable "MIO", que sume 1 la variable "intentos", que muestre en la etiqueta 5 los intentos y por último que nos diga si el numero ingresado es mayor, menor o acertamos.

The image shows a sequence of code blocks for a button click event. The blocks are: 'cuando Botón1 .Clic' (yellow), 'ejecutar' (green), 'poner global MIO a CampoDeTexto1 . Texto' (green), 'poner global Intentos a tomar global Intentos + 1' (blue), 'poner Etiqueta4 . Texto como tomar global Intentos' (green), 'si tomar global AZAR > tomar global MIO' (blue), 'entonces poner Etiqueta2 . Texto como "El número a adivinar es mayor."' (green), 'si tomar global AZAR < tomar global MIO' (blue), 'entonces poner Etiqueta2 . Texto como "El número a adivinar es menor."' (green), 'si tomar global AZAR = tomar global MIO' (blue), 'entonces poner Etiqueta2 . Texto como "Adivinaste"' (green). Callouts identify: 'Componente: Campo de texto' (pointing to 'CampoDeTexto1 . Texto'), 'Bloques de: Variables' (pointing to 'global MIO' and 'global Intentos'), 'Bloques de: Matemática' (pointing to '+ 1'), 'Bloques de: Control' (pointing to 'si' and 'entonces' blocks), 'Bloques de: Texto' (pointing to the text strings in the 'entonces' blocks), and 'Bloque de Componente: Etiquetas' (pointing to 'Etiqueta2 . Texto').

Nuestro segundo botón quedaría programado de la siguiente manera.

A continuación veremos algunos ejemplos específicos que programaremos.

Hacer que un botón abra una página web

The image shows a sequence of code blocks for a button click event. The blocks are: 'cuando Botón1 .Clic' (yellow), 'ejecutar' (green), 'poner ActivityStarter1 . UriDelDato como "http://google.com"' (green), 'poner ActivityStarter1 . Acción como "android.intent.action.VIEW"' (green), and 'llamar ActivityStarter1 .IniciarActividad' (purple). Callouts identify: 'Bloques de: Texto' (pointing to the text strings in the 'poner' blocks), 'Bloques de componente: Botón' (pointing to 'Botón1 .Clic'), and 'Bloques de componente: ActivityStarter' (pointing to 'ActivityStarter1' blocks).

En el primer bloque de texto se digita la URL de la web y en el segundo el texto "android.intent.action.VIEW", respetando mayúsculas y minúsculas.

Hacer que un botón llame por teléfono

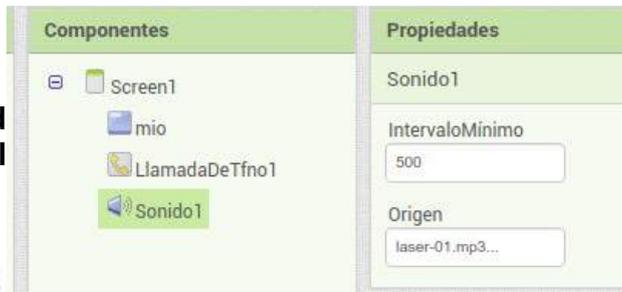
En primer lugar debemos tener los componentes no visibles “Llamada de tfno” de la paleta social y el componente “Sonido” de la paleta medio (No necesario), como se muestra en la imagen del diseño.

En el campo “Llamada de tfno” en la propiedad ”Numero de Teléfono” ingresar el mismo (Teléfono al cual llamaría cuando presionamos el botón).



Luego es el turno de poner el sonido que deseamos tener hasta que respondan.

En la propiedad origen subimos el archivo de sonido.



La programación quedaría de la siguiente manera:



Bloques de componente: Sonido

Bloques de componente: LlamadaDeTfno

Bloques de componente: Botón

Hacer que un botón que envíe un mensaje de texto

Para ello lo primero será poner dos campos de textos de la paleta “Interfaz de usuario” con una etiqueta cada uno indicando lo que se ingresará en ellos.

En el primero vamos a digitar el número de celular y en el segundo, el mensaje.

También arrastraremos de la paleta “Social” el componente “Enviar texto”, él mismo es no visible.



Ahora nos falta un botón en el final para enviar el mensaje, le llamaremos “Enviar”. Nuestro diseño quedará como lo muestra la siguiente imagen:



Profesor: Marcelo Rebellato

Como veremos a continuación la programación será sencilla.

The image shows a Scratch script starting with a 'when clicked' event block for a button. The script contains three 'do' blocks: 'set text of text field 1 to', 'set text of text field 2 to', and 'send message to text field 1'. Callout boxes identify the 'Enviar' button as a 'Component Block' and the text fields as 'Component Blocks'.

Bloque de Componente: Enviar

Bloque de Componente: Botón "Enviar".

Bloque de Componente: CampoDeTexto

Trabajar con listas

Primero veremos como crear una lista vacía.

The image shows two code blocks: 'initialize global variable Diamastriste as' and 'create an empty list'. Callout boxes identify these as 'Variable palette block' and 'List palette block'.

Bloque de paleta: Variable

Bloque de paleta: Lista

Intentaremos crear una lista con diferentes textos (Causas del día más triste), para luego poner uno de ellos al azar en una etiqueta. Para ello utilizaremos en nuestro diseño un botón que permitirá mostrar el texto.

The image shows a Scratch script for a button click. It includes three 'set text of' blocks for day, month, and year, each using a 'random number between' block. It then uses 'add elements to list' to create a list of four sad causes. Finally, it uses 'set text of label 4 to' with a 'pick random element from list' block. Callout boxes identify the 'Botón' component block, the 'Math palette block' for random numbers, the 'Text palette block' for the list items, and the 'Variable palette block' for the global variable.

Bloque del componente: Botón

Bloque de paleta: Matemática

Bloque de paleta: Texto

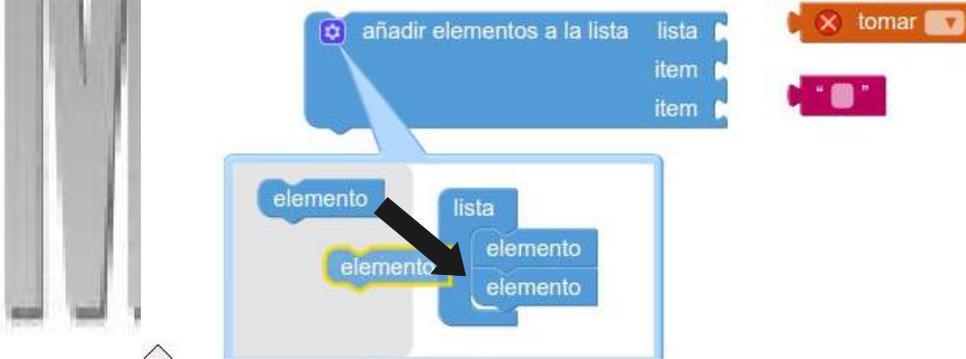
Bloque de paleta: Lista

Bloque de paleta: Variable

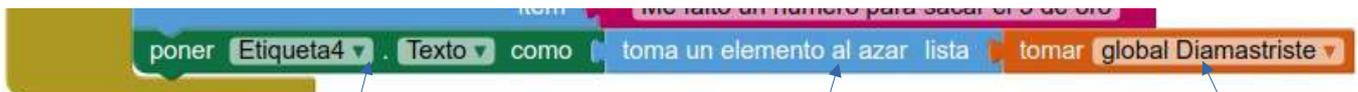
Profesor: Marcelo Rebellato

Como podemos ver el bloque de la paleta Lista “Añadir elementos a la lista” tiene en primer lugar un bloque de variable que permitirá guardar la causa para luego mostrar en una etiqueta, luego utilizando bloques con las comillas de la paleta texto se ingresarán las posibles causas.

El engranaje que se encuentra en la parte superior izquierda del bloque “Añadir elementos a la lista” permitirá aumentar los ítem la cantidad que necesitamos.



Simplemente debo arrastrar el elemento para dentro de la lista en la parte derecha. En el final mostraremos en una etiqueta un elemento al azar guardado en la variable.



Bloque del componente: Etiqueta

Bloque de paleta: Lista

Bloque de paleta: Variable

Rebellato