



GUÍA DE PRÁCTICAS 03

APPINVENTOR – ESTRUCTURAS CONDICIONALES

1. Objetivos:

- Utilizar las estructuras condicionales en AppInventor.
- Realizar aplicaciones en AppInventor.

2. Recursos y materiales:

- Computador con conexión a internet y puertos correspondientes.
- Correo personal en Gmail.
- Aplicativo AppInventor.

3. Información teórica:

Estructura SI-ENTONCES-SINO (IF-ELSE)

Es la que permite que el flujo de control de un programa bifurque a una de dos posibles acciones. Está compuesta por una decisión lógica, de la cual salen dos posibles caminos o estructuras de secuencia. Una secuencia de órdenes debe efectuarse cuando el resultado de la decisión sea verdadero y la otra cuando sea falso. Posteriormente se unifican los dos caminos para que se cumpla así con la característica de una única salida desde la estructura. Uno de los caminos siempre es positivo y el otro negativo; o uno de los caminos es verdadero y el otro falso.

Control en AppInventor:



Prueba una condición dada, si el resultado es verdadero, realiza las acciones de la secuencia de bloques -entonces; de lo contrario, realiza las acciones en la secuencia de bloques -si no.

Se tiene también las opciones de un solo lado, o los bloques anidados.

4. Procedimientos

PARTE 1: Estructuras condicional.

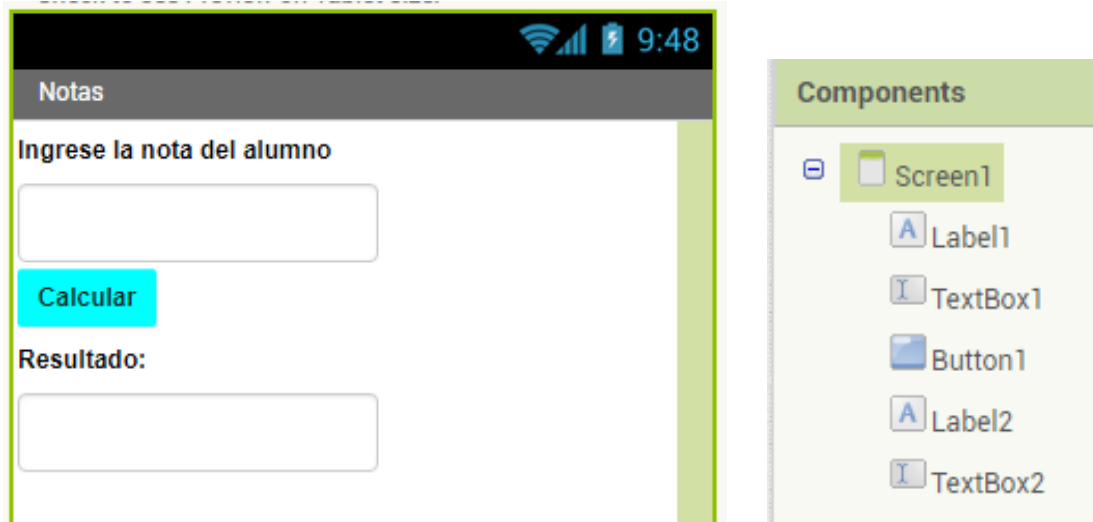
1 Aplicaciones con Estructuras Condicionales.



ENUNCIADO:

Programa que solicita el ingreso de la nota promedio de un estudiante y al presionar el botón calcular se muestra si está aprobado o desaprobado. En el sistema vigesimal si se tiene una nota mayor o igual a 10.5 se considera aprobado, en caso contrario estaría desaprobado.

Implementa la siguiente interface, considera los componentes y los atributos de cada objeto según la siguiente figura:



2 Escribir código y compilar.

Selecciona la sección “**Bloks**” para activar la sección de código e implementa el siguiente código:

Se tiene el evento Click para el botón **Button1** y dentro de ello la estructura condicional de dos lados, se evalúa la condición si la nota es mayor o igual a 10.5 se muestra en resultado el texto “APROBADO” en caso contrario se muestra el texto “DESAPROBADO”.

```

when Button1 .Click
do
  if TextBox1 . Text > 10.5
  then set TextBox2 . Text to " APROBADO "
  else set TextBox2 . Text to " DESAPROBADO "
  
```



Ejecute la aplicación: En el menú, seleccione “Connect” y la opción “Al Companion”, utilice la aplicación **MIT AI2 Companion** en su dispositivo móvil y ejecute la aplicación.

3 Implemente una nueva aplicación.

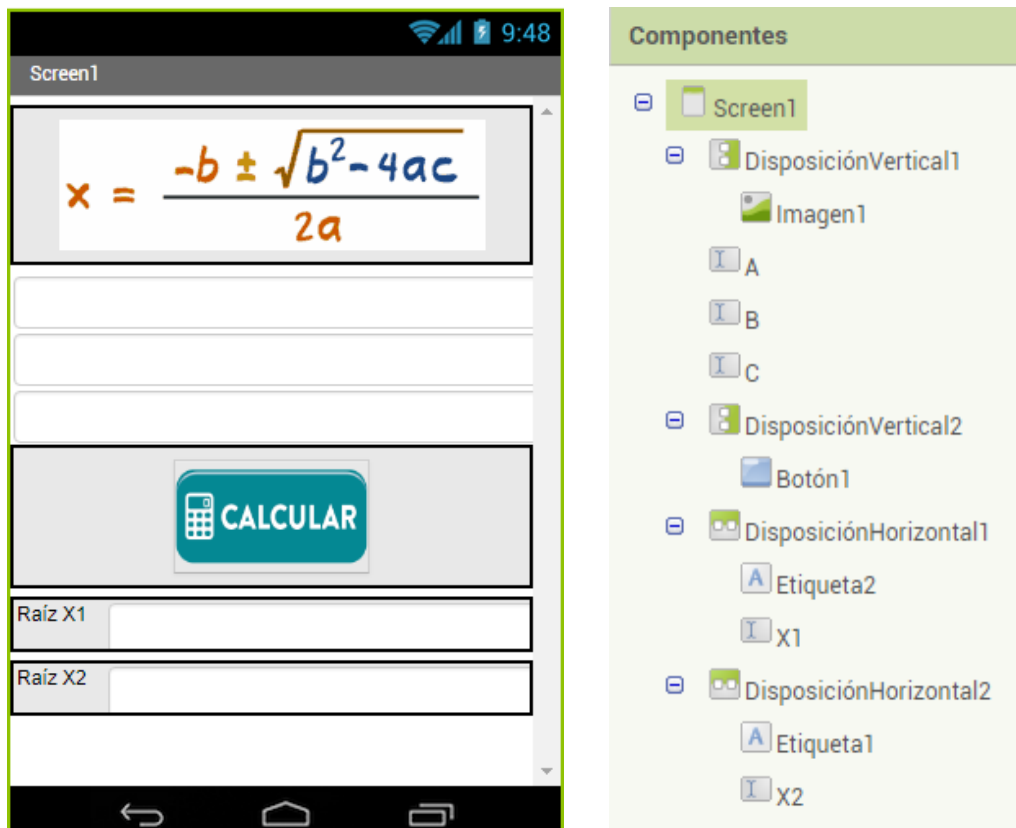
ENUNCIADO:

Implementar una aplicación que halle las raíces que den solución a una ecuación de segundo grado de la forma:

$$Ax^2 + Bx + C = 0$$

Utilizando la fórmula general.

Para ello cree un nuevo proyecto de nombre “cuadratica” e implemente la siguiente interface: **Modifique los nombres de los componentes** según se indica en la figura:



4 Implementa el siguiente código en el sector de Bloques:

a) Inicialice las variables A, B, C y Disc.



```

initialize global A to 0
initialize global B to 0
initialize global C to 0
initialize global Disc to 0
    
```

b) El evento Click del Botón

```

when Botón1 . Click
do
  set global A to A . Text
  set global B to B . Text
  set global C to C . Text
  set global Disc to (get global B ^ 2 - 4 * get global A * get global C)
  if (get global Disc < 0)
  then
    set X1 . Text to "No tiene raíces reales"
  else
    set X1 . Text to ((-1 * get global B + square root (get global Disc)) / (2 * get global A))
    set X2 . Text to ((-1 * get global B - square root (get global Disc)) / (2 * get global A))
    
```

5 Ahora implementamos una nueva aplicación.

ENUNCIADO:

Aplicación que genera un número aleatorio, el cual se tiene que adivinar:

Tenemos que crear un interfaz para para que la aplicación se relacione con el usuario, le pregunte un número, y le vaya dando pistas, diciendo si es demasiado alto, demasiado bajo, o si finalmente ha acertado el número secreto que se genera aleatoriamente.

a) Implemente la siguiente interface en MIT App Inventor. Considere los siguientes objetos y sus propiedades:



- b) Ingresar el siguiente código, que crea la variable numero y le asigna el valor de un número generado aleatoriamente en el rango de 1 a 10.

```
initialize global numero to random integer from 1 to 10
```

- c) Implemente el código para el Boton1 que permite realizar el juego.

```
when Button1 .Click
do
  if
    get global numero = TextBox1 . Text
  then
    set Label2 . Text to "Acertaste..."
    set Label1 . Text to get global numero
  else if
    get global numero > TextBox1 . Text
  then
    set Label2 . Text to "Te falta un poco..."
  else if
    get global numero < TextBox1 . Text
  then
    set Label2 . Text to "Te pasaste un poco..."
```



Implemente el código para el Boton2 que permite iniciar un nuevo juego

```

when Button2 .Click
do
  set global numero to random integer from 1 to 10
  set Label1 . Text to "?"
  set Label2 . Text to "Adivina el número"
  set TextBox1 . Text to ""
  
```

Construya la aplicación APK y ejecútelo en el simulador BlueStacks o en el dispositivo móvil.

Completar:

- 1) Implementar para que luego de 30 segundos de jugar salga un mensaje de Game Over.
- 2) Implementar un contador de intentos realizados, y que solo permita realizar 3 intentos, luego de los cuales debe decir Game Over.