



Welcome to MIT App Inventor

2013

Desarrollo fácil y paso a paso de aplicaciones
para Android usando MIT App Inventor

Rafael Alberto

Moreno Parra

<http://darwin.50webs.com>

Contenido

Dedicatoria.....	4
Licencia de este libro	5
Licencia del software	5
Marcas registradas	5
Introducción	6
Paso 1. Tener una cuenta en Gmail	7
Paso 2. Ir a la página de MIT App Inventor.....	7
Paso 3. Preparando el PC para hacer uso de MIT App Inventor.....	8
Paso 3A. Descargando e instalando el JRE (Java Runtime Edition)	8
Paso 3B. Descargando e instalando el AppInventor Setup.....	14
Paso 3C. Instalando en Linux.....	16
Paso 4. Iniciando con MIT App Inventor	19
Paso 5. Iniciando con el tradicional "Hola Mundo"	22
Paso 6. Leer un par de números y luego sumarlos.....	42
Paso 7. Haciendo operaciones matemáticas más complejas	47
Paso 8. El SI condicional	50
Paso 9. Ciclos	52
Paso 10. Operaciones con cadenas (Básico).....	53
A. Longitud de cadenas.....	53
B. Chequear si una cadena está vacía	54
C. Concatenación de cadenas.....	54
D. Comparación de cadenas.....	55
E. Quitar espacios al inicio y al final	56
F. Convertir a mayúsculas	56
G. Convertir a minúsculas	56
H. Señala en qué posición se encuentra una determinada subcadena en la cadena	56
I. Retorna verdadero si encuentra la subcadena dentro de la cadena	57
J. Extraer una subcadena	58
K. Ejemplo: quitar los espacios de una cadena	58
L. Reemplazar una subcadena por otra	59
Paso 11. Operaciones con listas	61
A. Longitud de una lista	62
B. Seleccionar un ítem de una lista	62
C. Mostrar una lista en un "textbox multiline"	63
D. Reemplazar un ítem de una lista.....	64
E. Verifica si hay un ítem en una lista.....	64
F. Retorna en qué posición se encuentra un determinado ítem en la lista.....	65
G. Retorna si la lista está llena o vacía.....	65
H. Inserta un ítem en determinada posición de la lista	66
I. Quita un elemento de la lista.....	66
J. Copia una lista en otra	67
K. Adiciona una lista a otra lista.....	67
L. ¿Es determinado objeto una lista?.....	68
Paso 12. Procedimientos y Funciones.....	69
Paso 13. Gráficos	70
A. Dibujar puntos, líneas y círculos.....	72
B. Dibujar textos.....	72
C. Dibujar texto girado en un determinado ángulo	72
D. Tamaño y color del texto	73
E. Fondo del lienzo en un color determinado	73
F. Dibujando una línea con determinado ancho	73
G. Imagen de fondo	74

H. Dibujar círculos con clic de ratón (inicio de captura de eventos).....	76
I. Hacer un sencillo programa de dibujo (captura de eventos)	76
Paso 14. Animación	77
A. Usando un componente de tiempo	77
B. Hacer que una pelota rebote en los bordes (sin requerir componente de tiempo)	79
Paso 15. Bases de datos	81
Paso 16. La capacidad inalámbrica de MIT App Inventor	83
Paso 17. Tomando vídeos.....	87
Paso 18. Tomando fotos.....	88
Paso 19. Ver galería de imágenes.....	89
Paso 20. Usando el acelerómetro	90
Paso 21. Organización en pantalla	92
Paso 22. Múltiples pantallas.....	93
Anexo 1. Trabajando fuera de línea	97

Dedicatoria

Dedicado a mi familia: José Alberto (mi padre), María del Rosario (mi madre), Diana Pilar (mi hermana) y mis gatas: Sally y Tammy.



Los gatos son excelentes mascotas.
<http://www.mundogatos.com/>



Licencia del software

Todo el software desarrollado aquí tiene licencia LGPL "Lesser General Public License"



Marcas registradas

En este libro se hace uso de las siguientes tecnologías registradas:

MIT® AppInventor® Enlace: <http://www.appinventor.mit.edu/>

Oracle ® Java ® Enlace: <http://www.oracle.com/technetwork/java/index.html>

Microsoft ® Windows ® Enlace: <http://windows.microsoft.com/en-US/windows/home>

Introducción

Android es actualmente (Mayo de 2013) el sistema operativo para smartphones y tablets más famoso, con una enorme cuota de mercado.

Desarrollar software para esta plataforma permite llegar a una gran cantidad de público. Sin embargo, abordar un desarrollo directo con el lenguaje de programación Java, XML, emulación virtual y ambiente de dispositivo móvil puede ser intimidante. Esa es la razón por la que Google ideó una forma sencilla de desarrollar aplicaciones para Android: haciendo uso de programación gráfica y atraer así a los desarrolladores. Google, sin embargo, dejó de lado este proyecto el cual fue retomado por el MIT y se ha mantenido así.

El libro sigue el formato de llevar paso a paso al lector desde preparar el ambiente de desarrollo, pasando por el diseño de la pantalla en un Smartphone Android, los diferentes componentes visuales y no visuales, programación usando bloques como si fuese un rompecabezas, variables, uso del sí condicional, ciclos o bucles, hasta el uso de las capacidades propias de un Smartphone como la cámara, video o el acelerómetro.

MIT App Inventor oficialmente es una herramienta para desarrollo en línea por lo que requiere conexión permanente a Internet. Sin embargo, en el Anexo 1 se muestra como trabajar fuera de línea o sin conexión a Internet.

Paso 1. Tener una cuenta en Gmail

En <https://accounts.google.com/> puede crear una cuenta de correo, ¿ya la tiene? Entonces este paso es innecesario.

Paso 2. Ir a la página de MIT App Inventor

Diríjase a <http://www.appinventor.mit.edu/> . Allí inicia todo, pero antes hay que preparar el PC para poder usar esta herramienta.

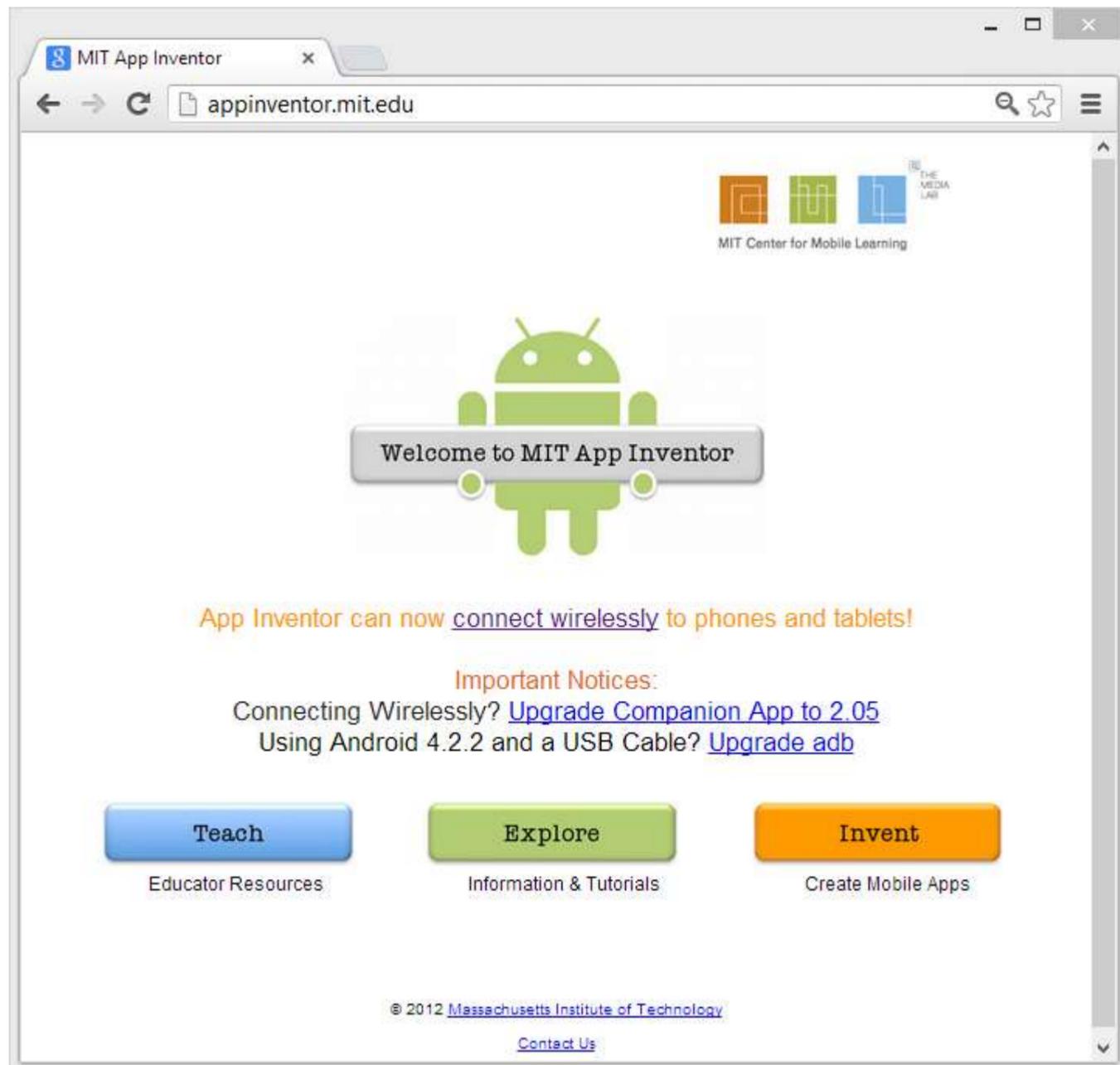


Imagen 1: Página inicial de MIT App Inventor

Paso 3. Preparando el PC para hacer uso de MIT App Inventor

Tiene Windows 7 o Windows 8 (32 o 64 bits), ¿Qué se debe instalar? Dos software:

1. JRE (Java Runtime Environment)
2. AppInventor Setup

Paso 3A. Descargando e instalando el JRE (Java Runtime Edition)

Con el navegador Google Chrome en 32bits, es necesario ir a esta ubicación: <http://java.com/es/download/index.jsp> y dar clic en "Descarga gratuita de Java"



Imagen 2: Descarga del JRE para Google Chrome de 32 bits

Luego se debe dar clic en "Aceptar e iniciar descarga gratuita"



Imagen 3: Advertencia cuando el sistema operativo del PC es Windows 8

Google Chrome descarga el instalador de Java (ver la parte inferior), luego se da clic en ese ejecutable



Imagen 4: Descargando JRE para Google Chrome



Imagen 5: Instalador de JRE en ejecución



Imagen 6: Descargando el resto de JRE



Imagen 7: JRE instalado

Para probar que efectivamente el JRE quedó correctamente instalado, diríjase a <http://www.java.com/en/download/testjava.jsp>, debe aparecer la siguiente imagen:



Imagen 8: Se da el permiso para ejecutar un applet Java dentro de Google Chrome

Al dar clic en Google Chrome en "Ejecutar siempre en este sitio" debe aparecer:

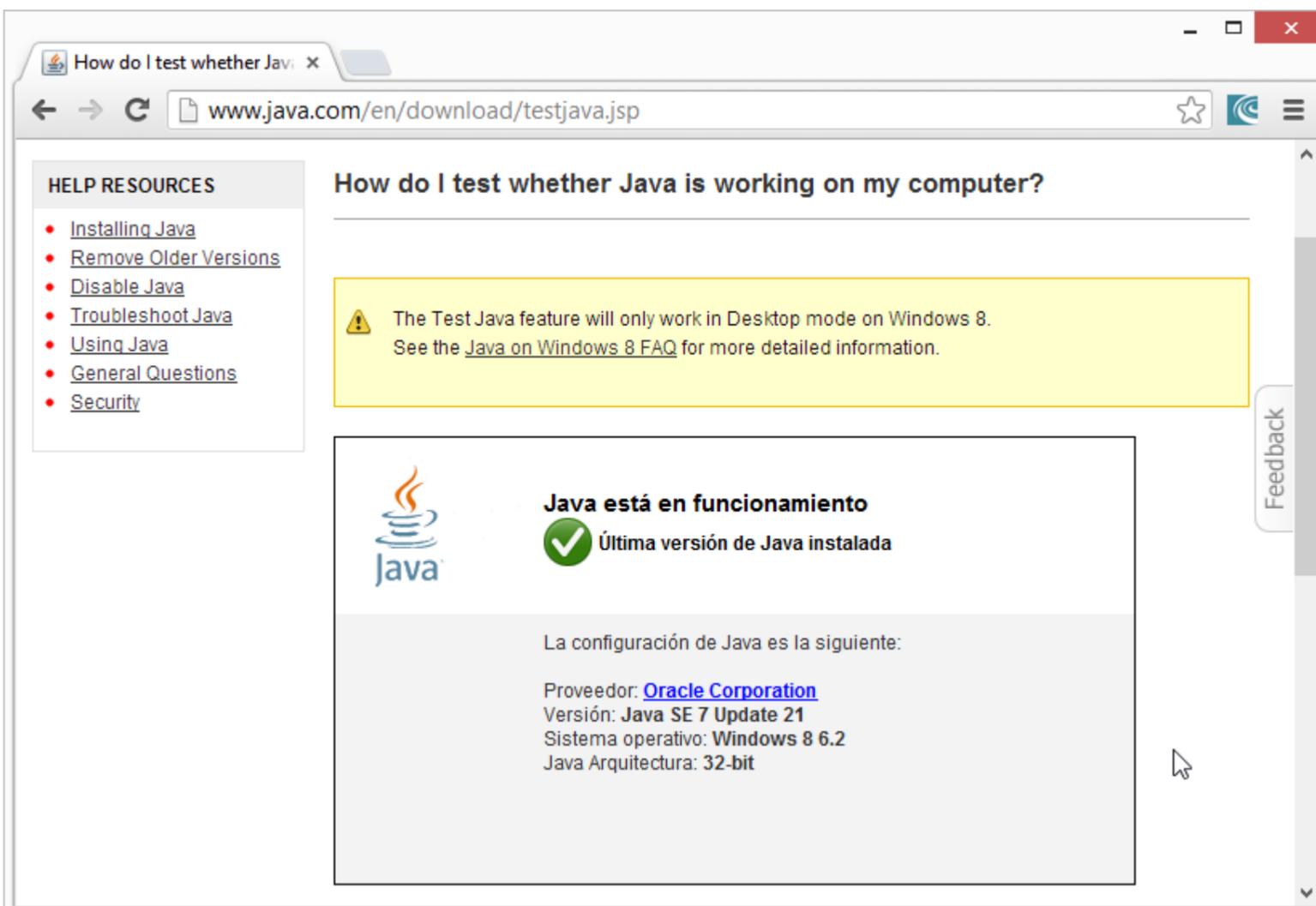


Imagen 9: Java correctamente instalado y ejecutando en el navegador

Y para hacer la prueba que puede ejecutar App Inventor, diríjase a este enlace:

<http://beta.appinventor.mit.edu/learn/setup/misc/JWSTest/AppInvJWSTest.html> , allí le debe dar clic al botón "Launch" que lanza un "Bloc de Notas" o "Notepad" funcional.

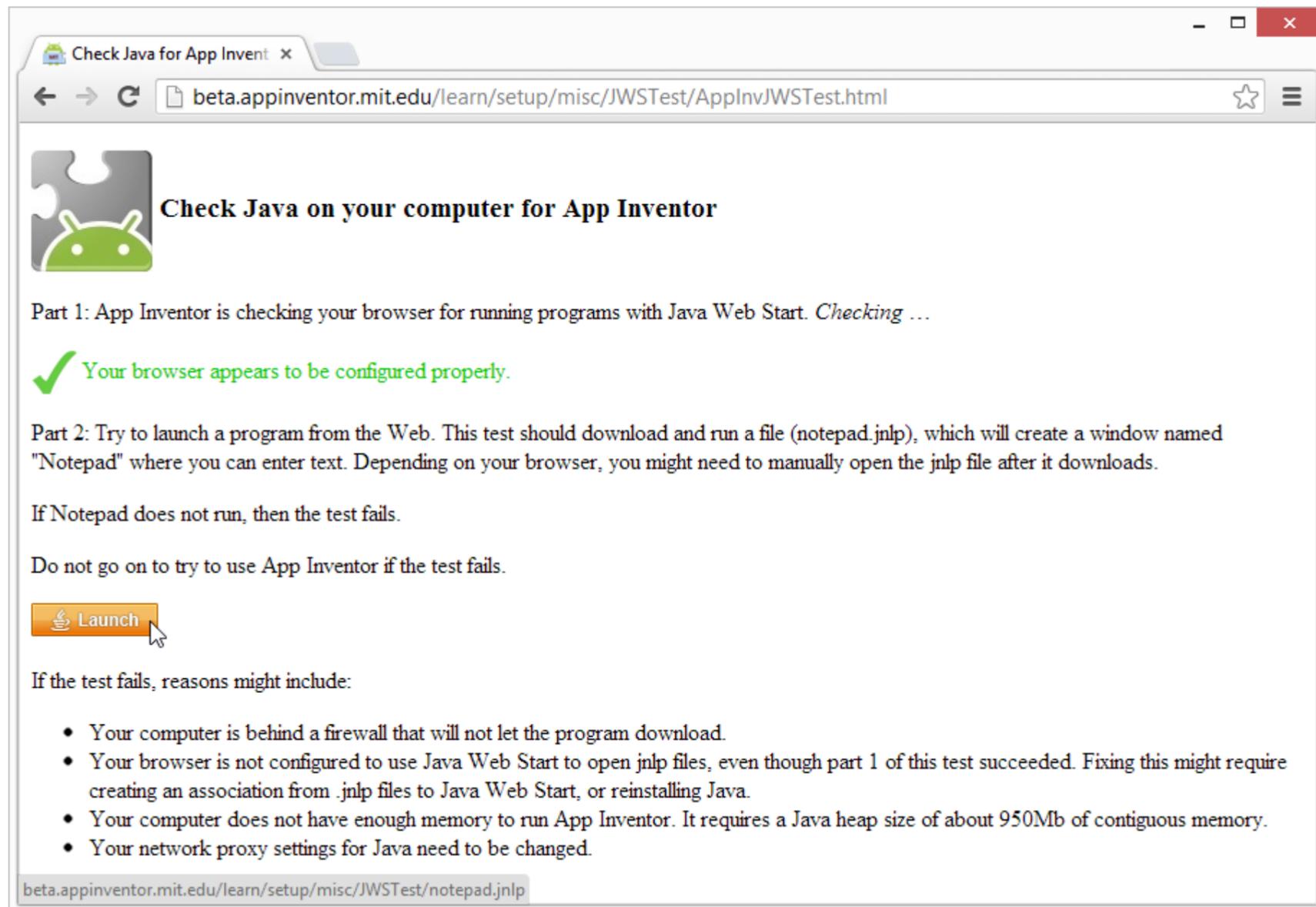


Imagen 10: De clic en el botón "Launch" para probar que el JRE está listo para ejecutar App Inventor

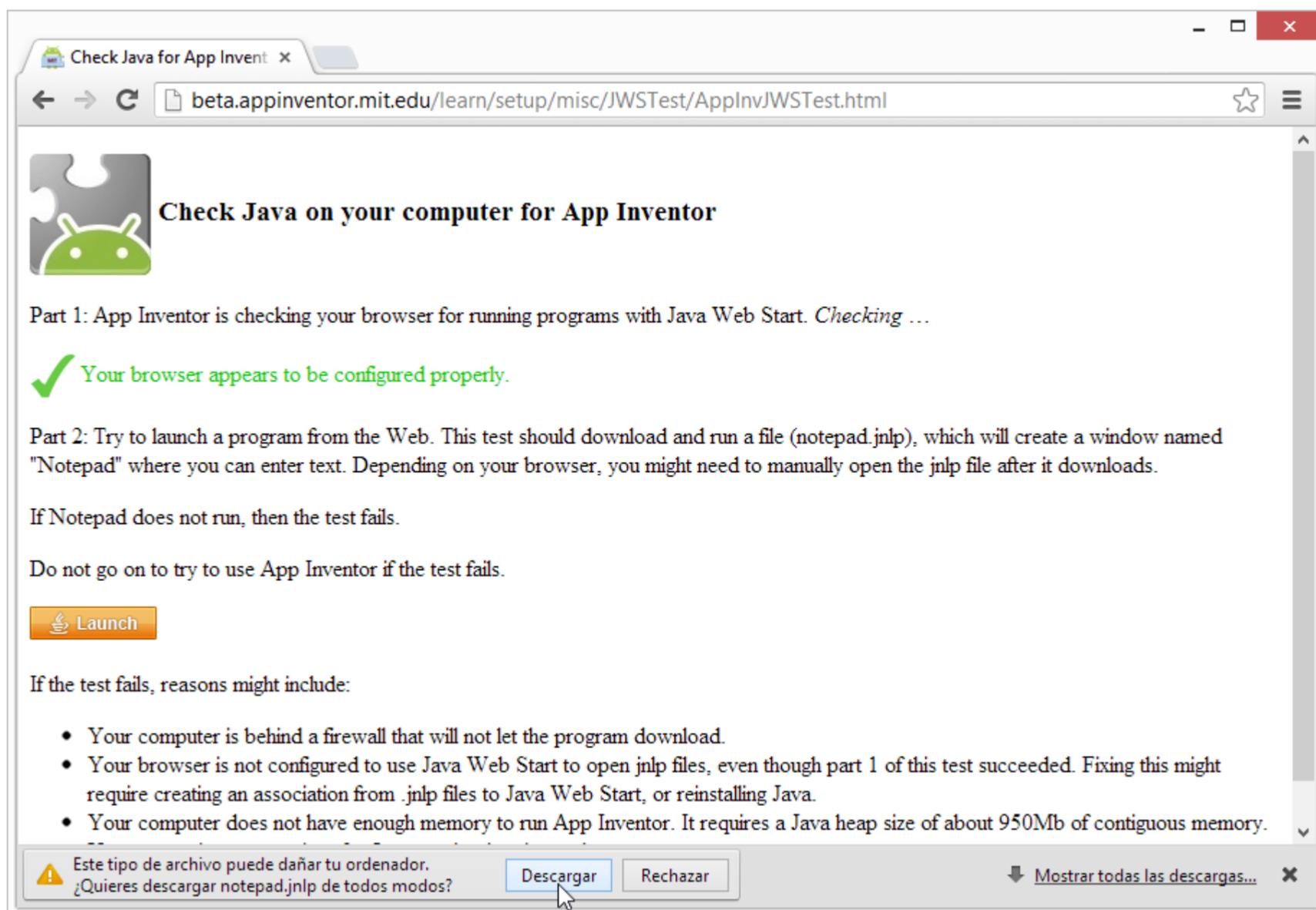


Imagen 11: Google Chrome pide permiso para descargar el programa notepad.jnlp

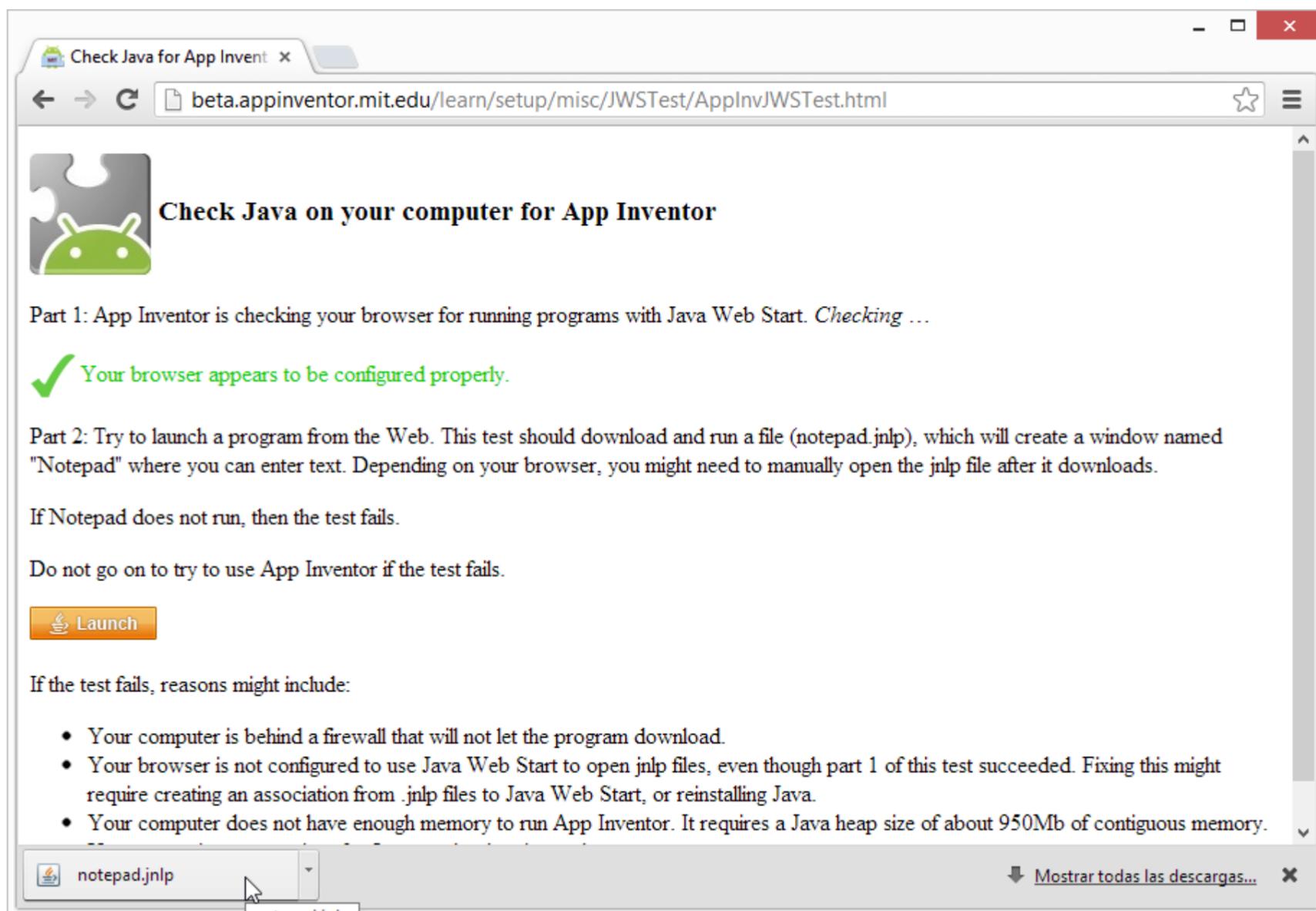


Imagen 12: Una vez descargado notepad.jnlp se da clic para ejecutarlo



Imagen 13: Permiso para ejecutar la aplicación

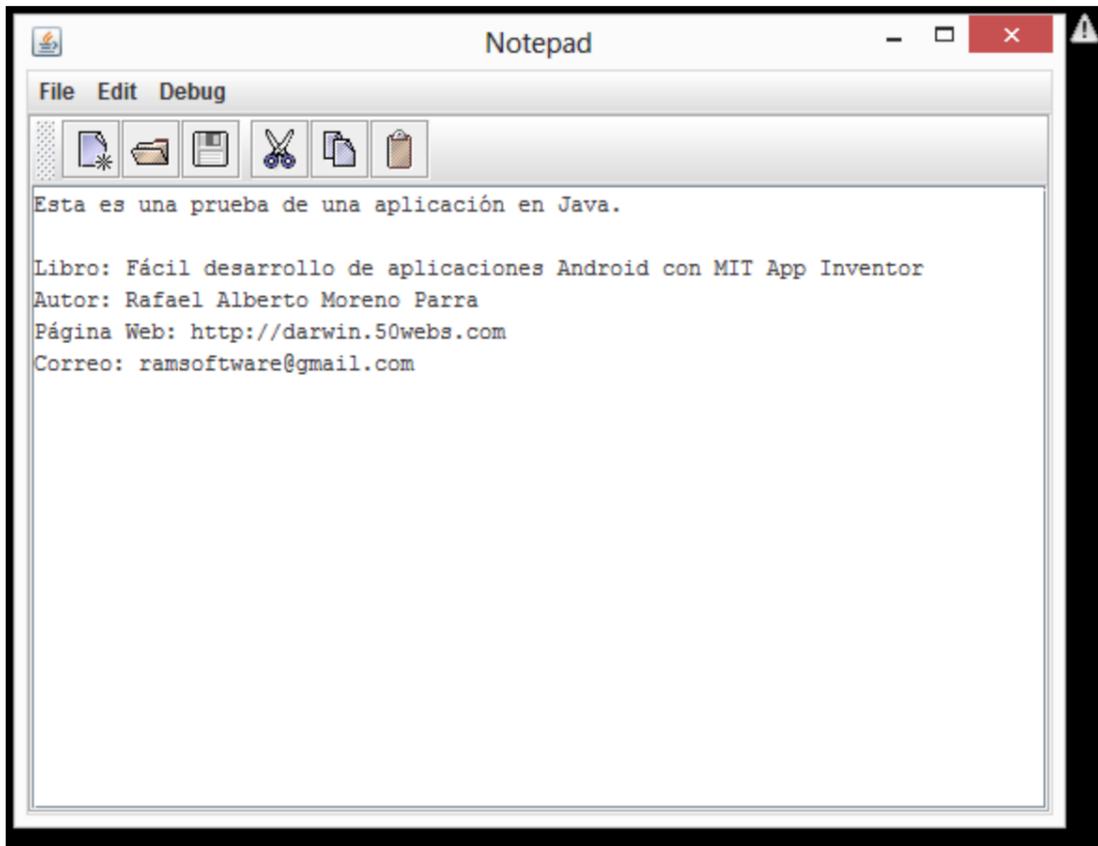


Imagen 14: Un "bloc de notas" o "Notepad" lanzado.

Paso 3B. Descargando e instalando el AppInventor Setup

Descargue directamente de este enlace el AppInventor Setup <http://beta.appinventor.mit.edu/learn/setup/setupwindows.html> , y siga estos pasos:

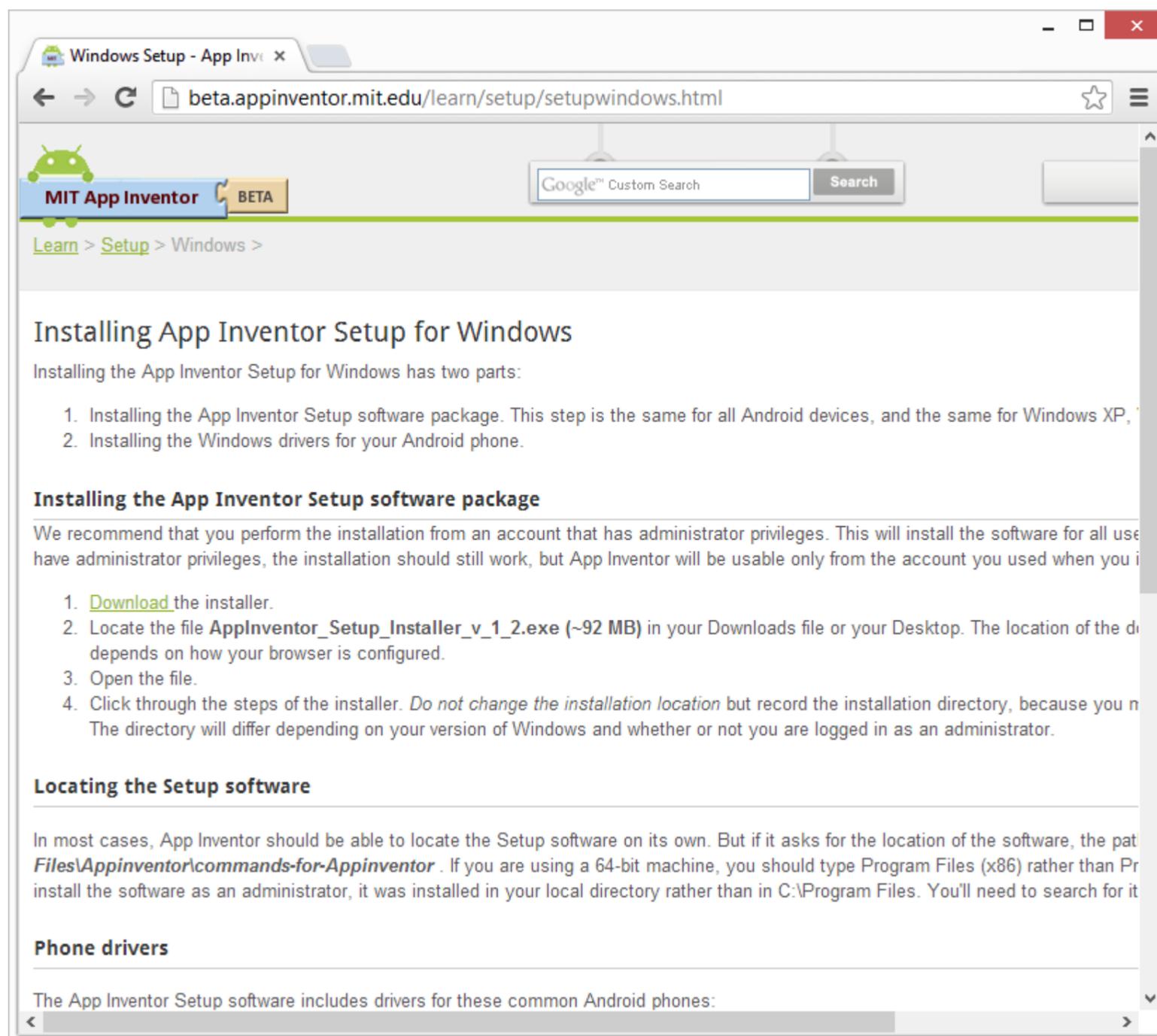


Imagen 15: Descargando el App Inventor Setup

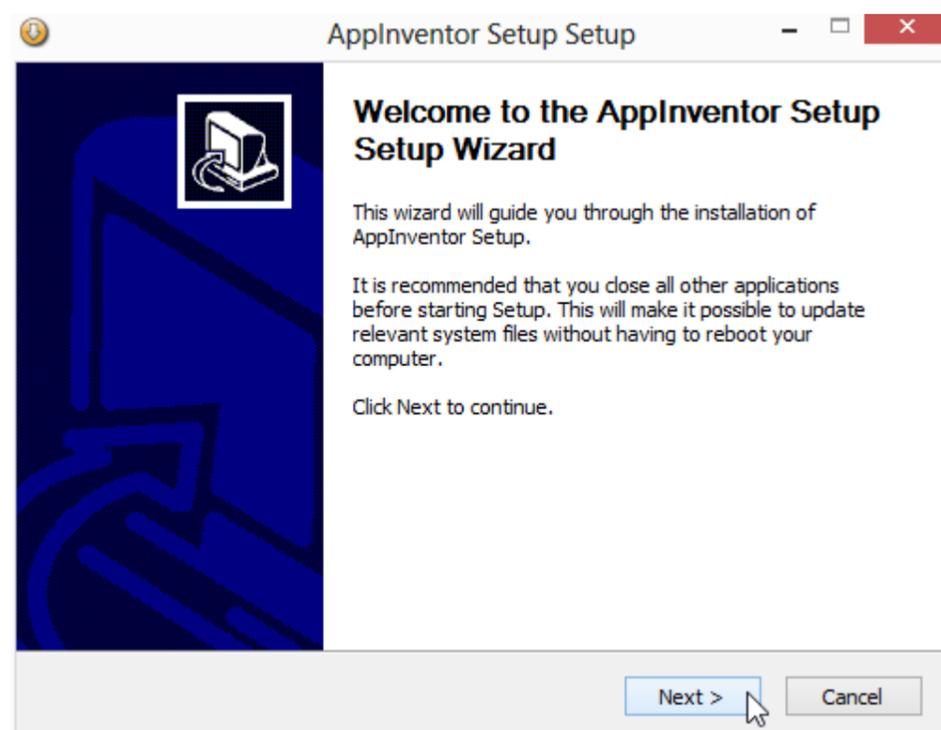


Imagen 16: Inicio de instalación

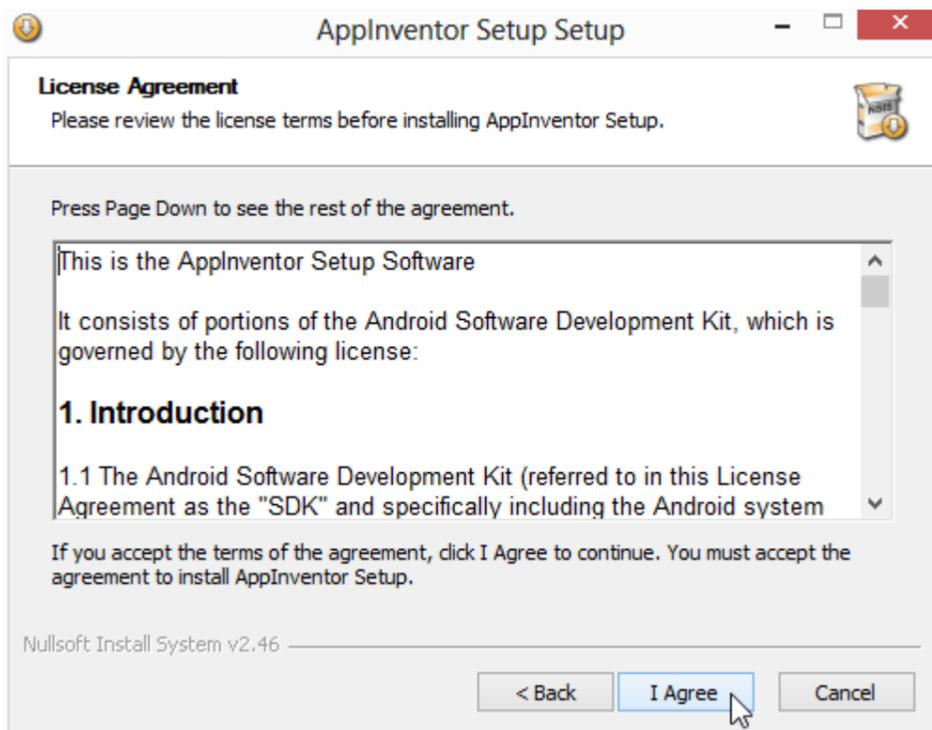


Imagen 17: Acuerdo de licencia

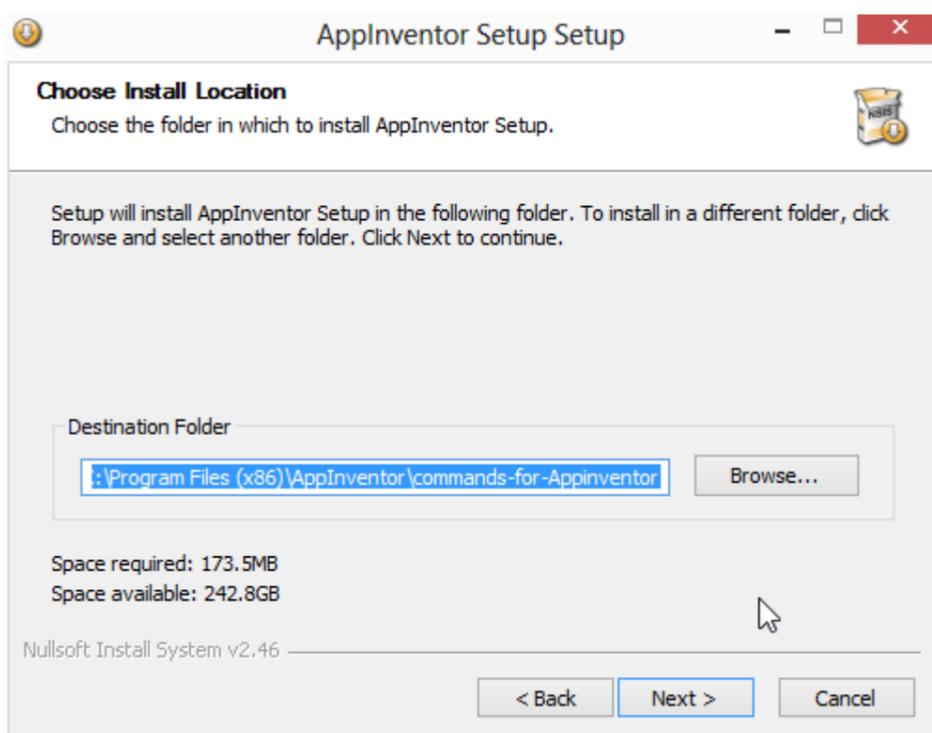


Imagen 18: En que carpeta será instalado

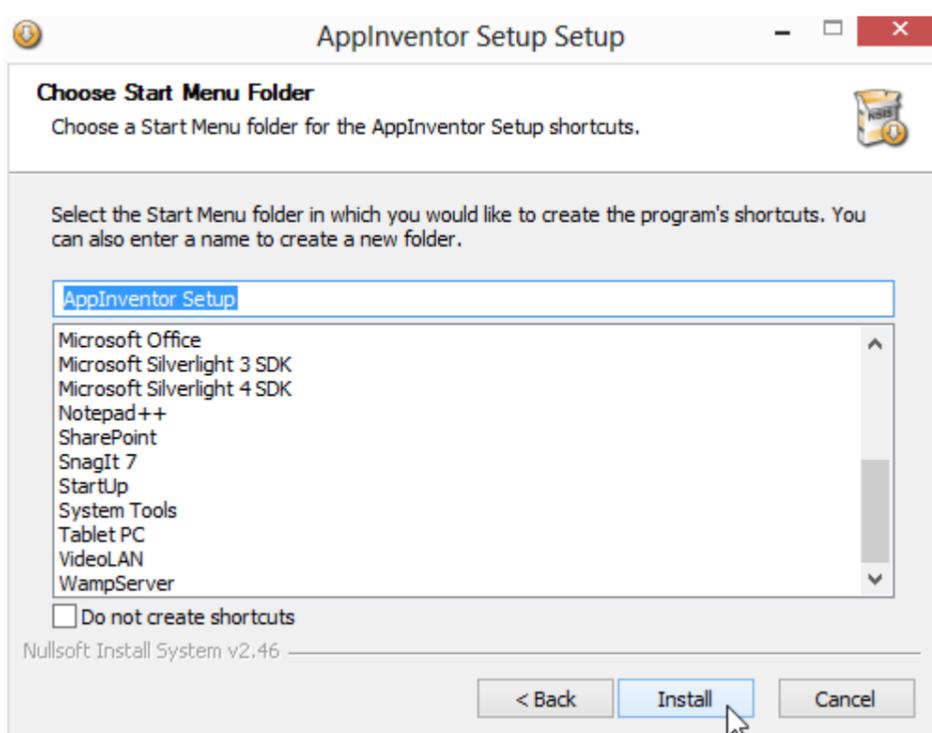


Imagen 19: Que grupo de programas se crea en la instalación

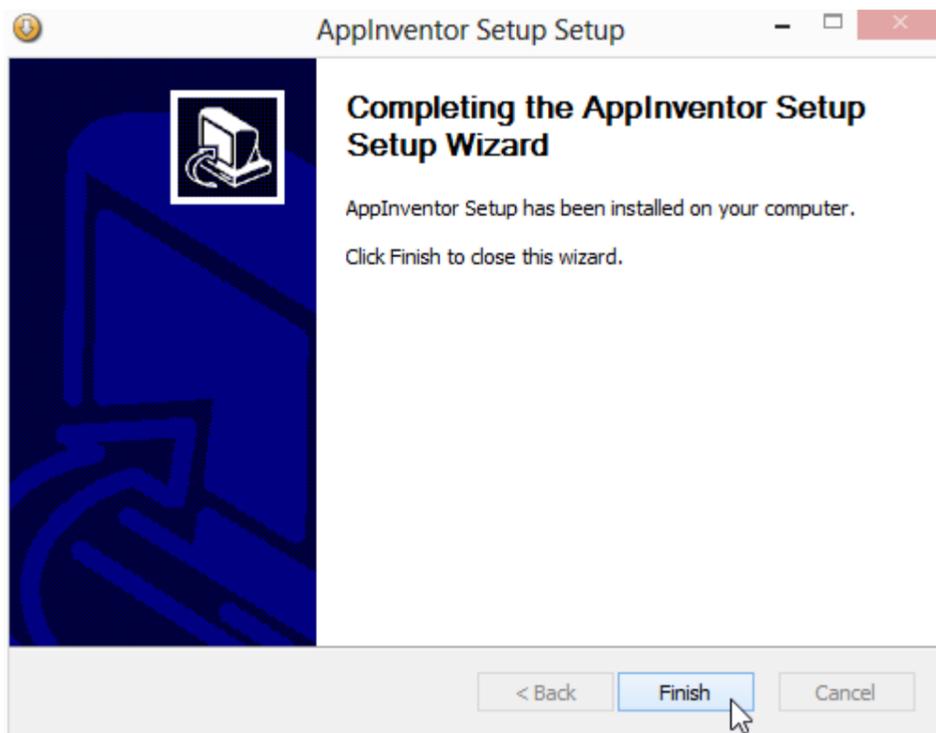


Imagen 20: AppInventor Setup instalado

Paso 3C. Instalando en Linux

En este libro se mostrará como instalar el ambiente para la distribución Linux Mint 14 Xfce (64 bits).



Imagen 21: Ya tiene Google Chrome instalado



Imagen 22: Las distribuciones por lo general ya tienen Java instalado. Una buena tarea es actualizar.

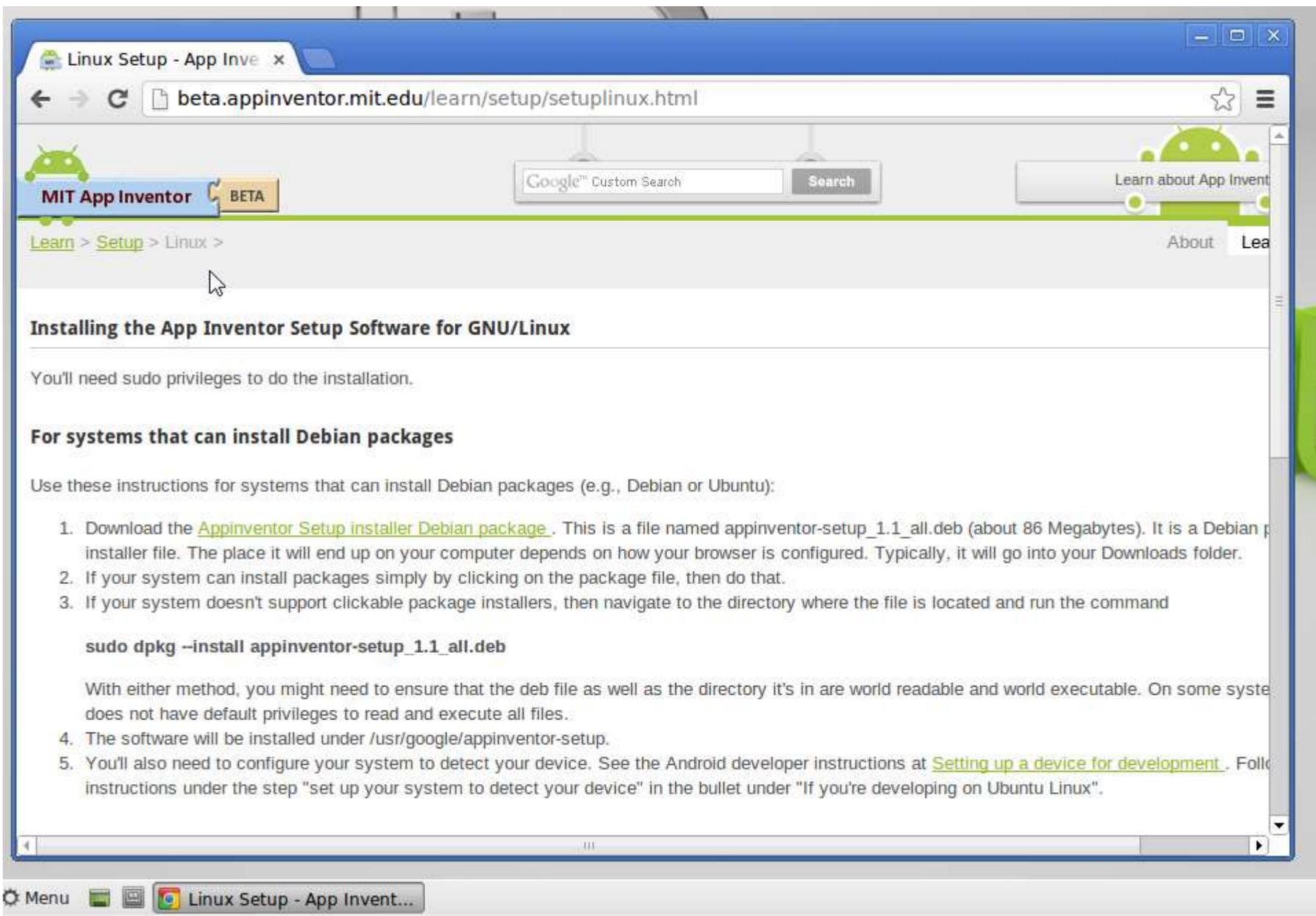


Imagen 23: Se descarga el MIT App Inventor Setup Installer para Linux

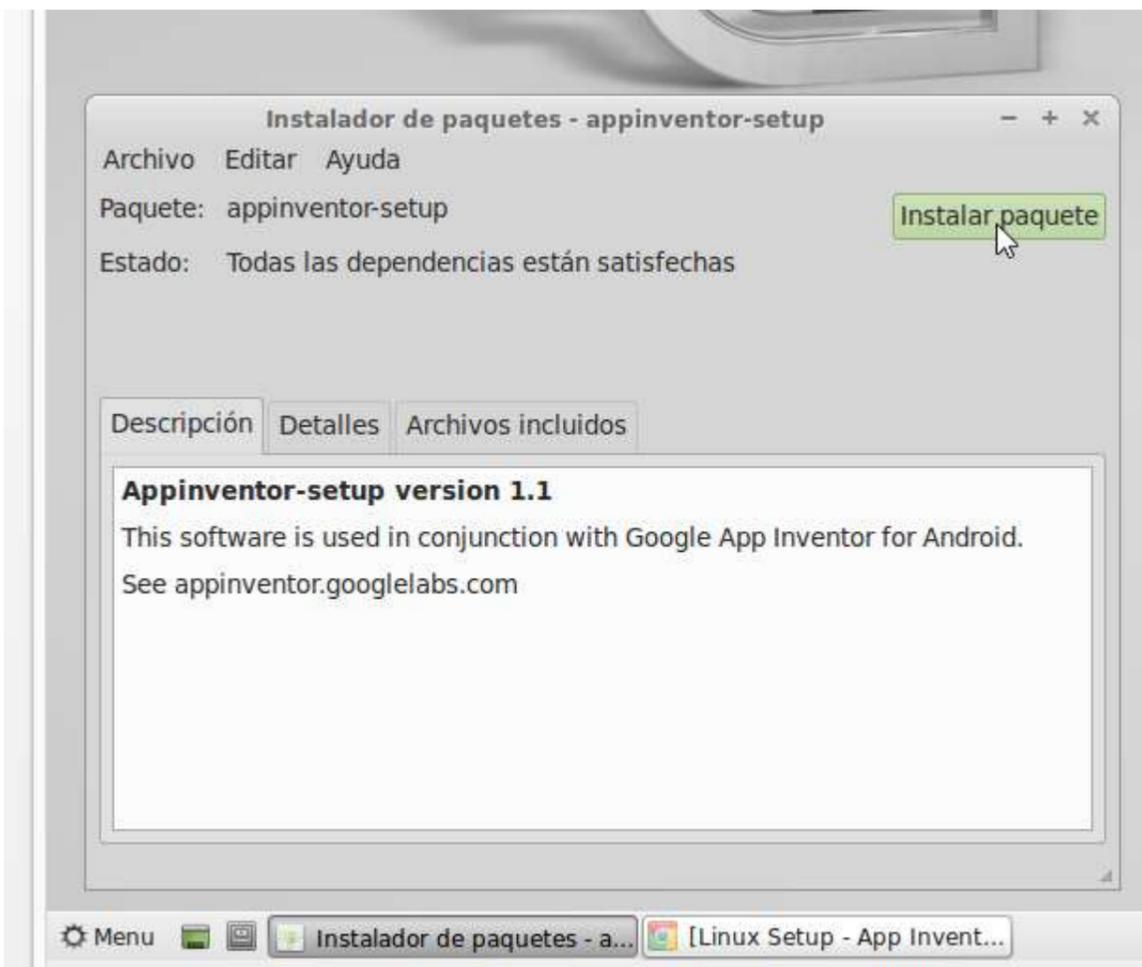


Imagen 24: Se instala el paquete

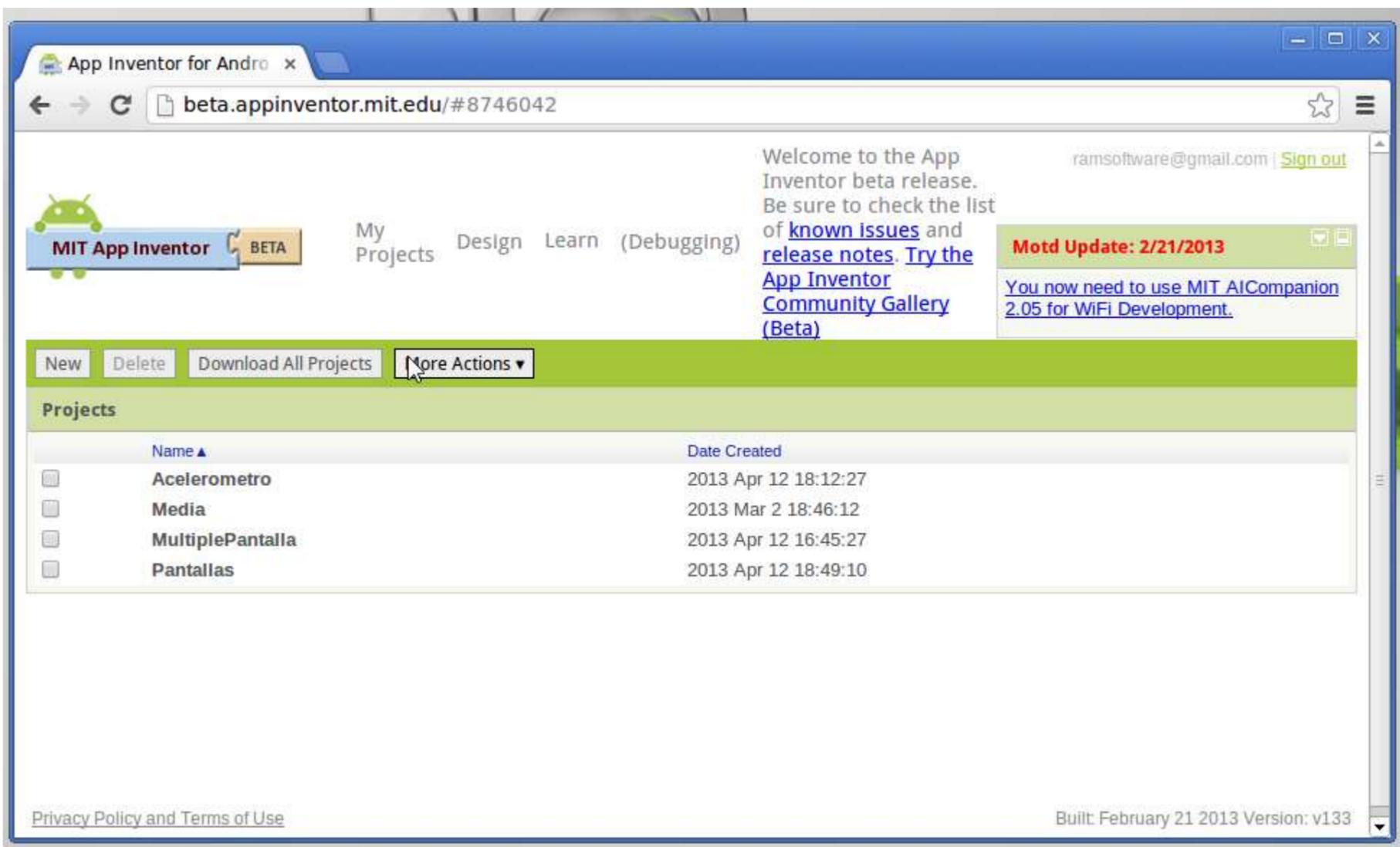


Imagen 25: Empezar a programar

Paso 4. Iniciando con MIT App Inventor

Ya puede dirigirse a <http://www.appinventor.mit.edu/> y dar clic en "Invent"

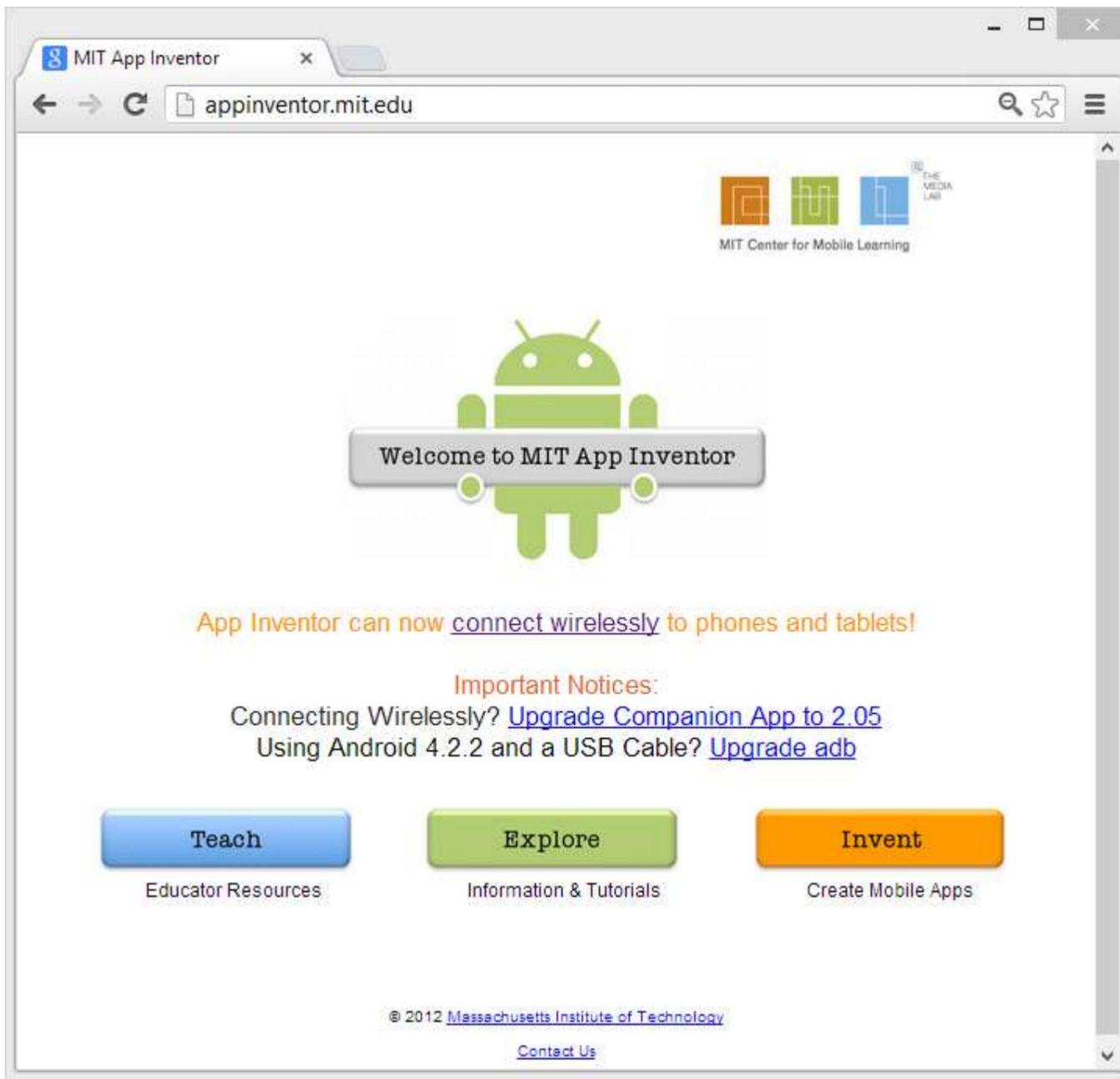


Imagen 26: De clic en "Invent"

Luego entre con su cuenta de Gmail

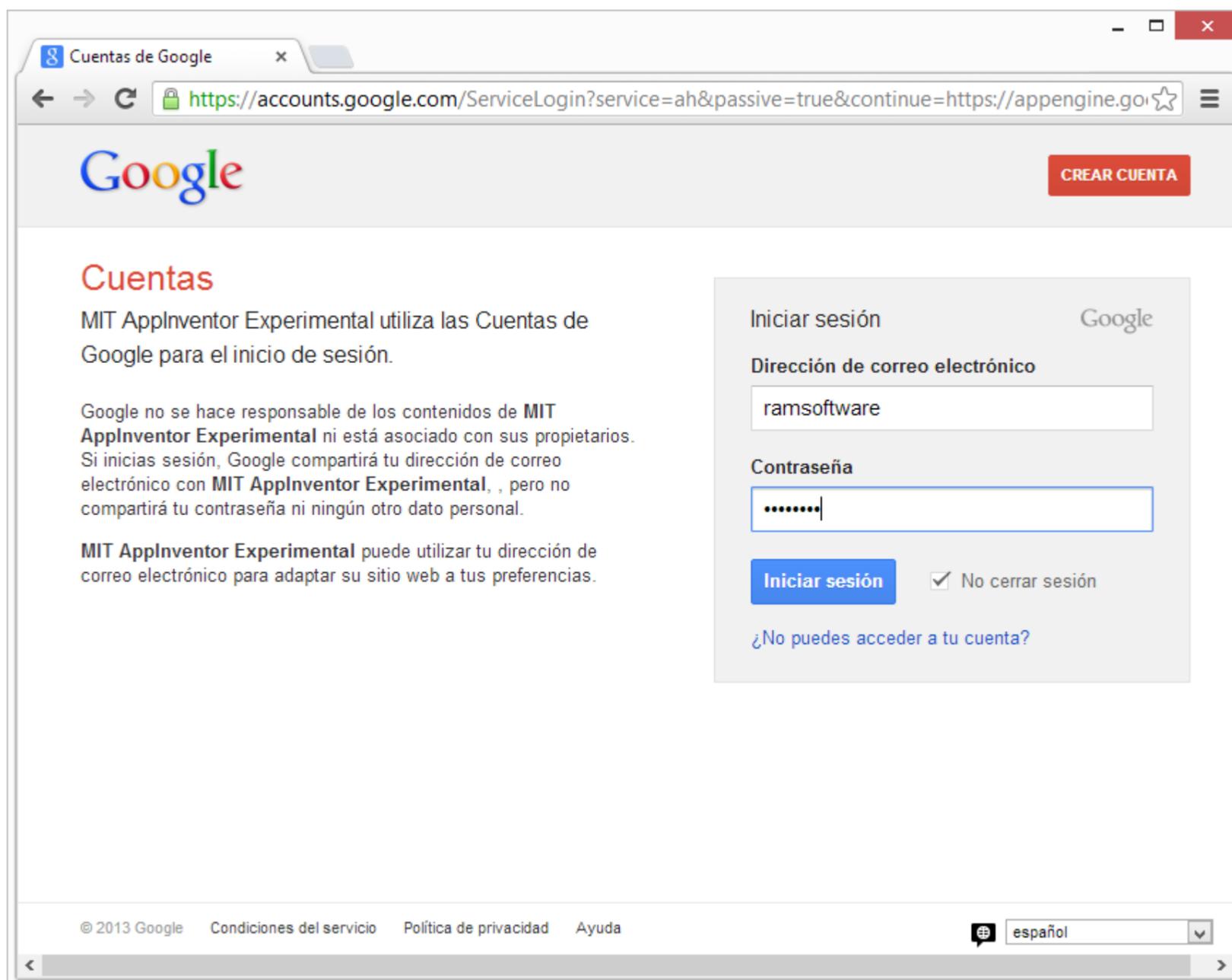


Imagen 27: Entre con su cuenta de Gmail



Imagen 28: Apruebe el uso de la cuenta en MIT AppInventor

Esta es la pantalla con que inicia

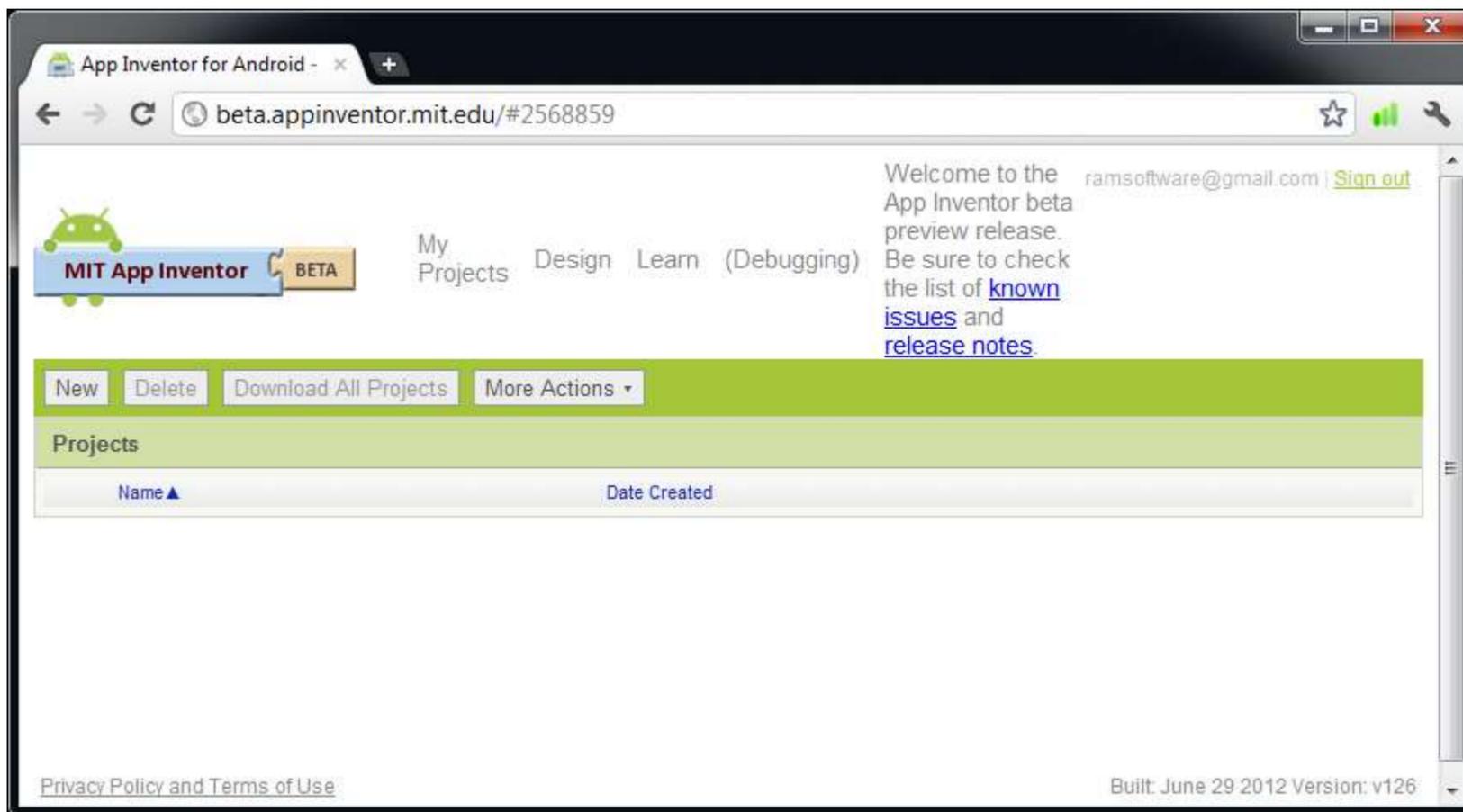


Imagen 29: Pantalla inicial que muestra los proyectos que ha hecho

Como es la primera vez que inicia en esta herramienta, todavía no tiene proyectos.

Paso 5. Iniciando con el tradicional "Hola Mundo"

Presione el botón "New" para crear un nuevo proyecto

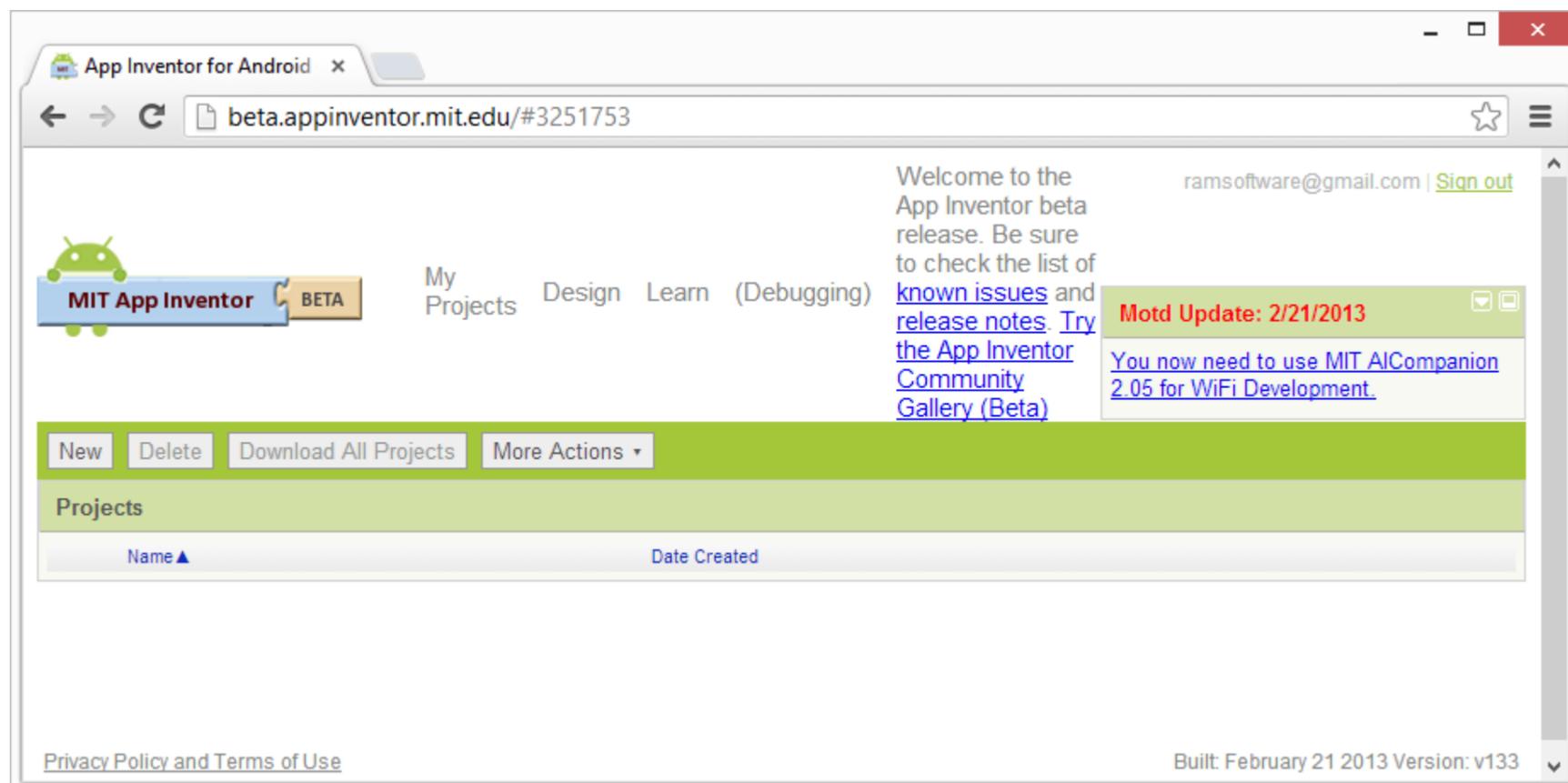


Imagen 30: Presione el botón "New" para generar un nuevo proyecto

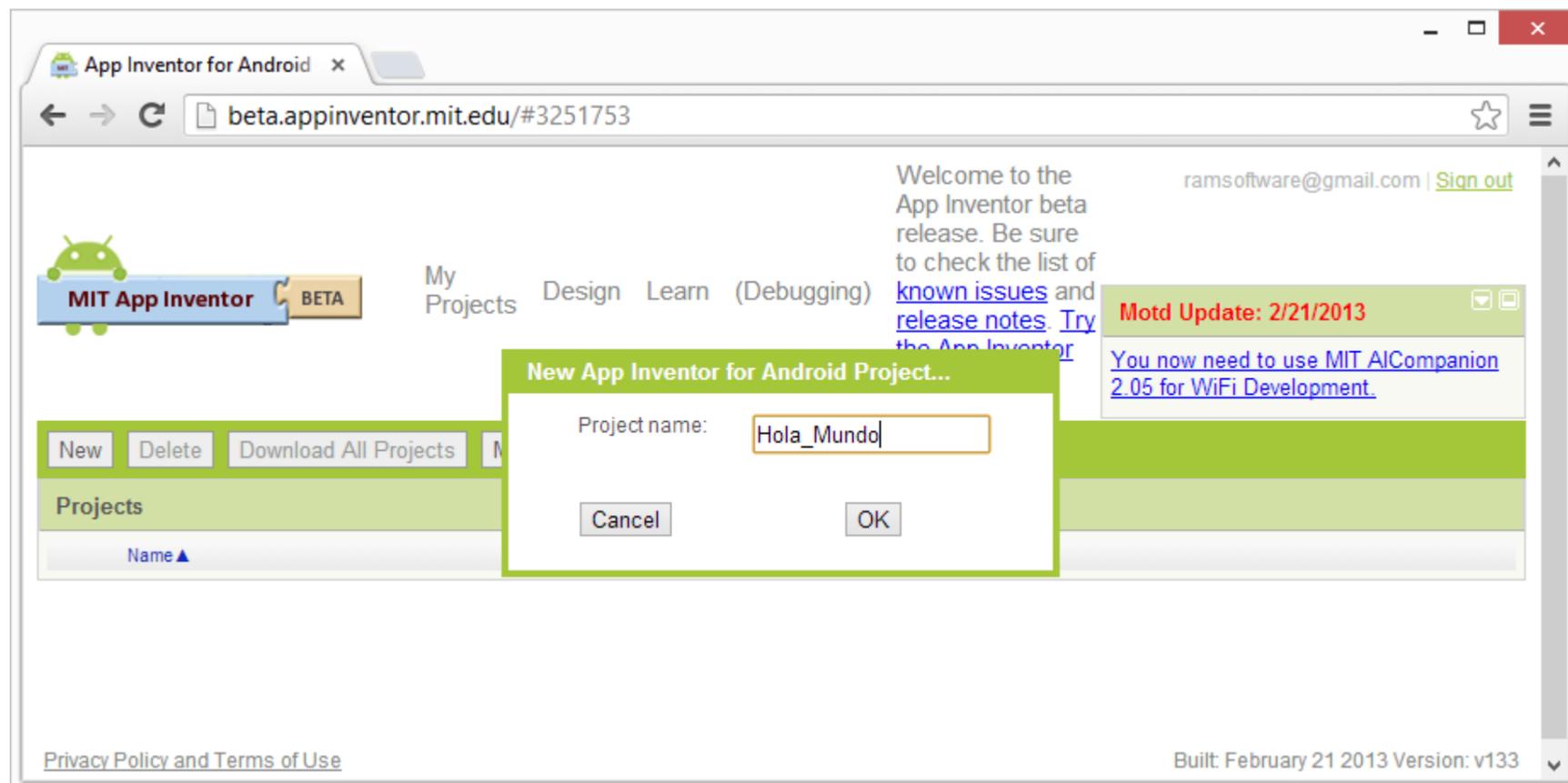


Imagen 31: Se le pone un nombre al proyecto (espacios no están permitidos)

Inmediatamente al crear un proyecto, se muestra la página de diseño de pantallas, la cual es muy parecida a otros IDE (Integrated Development Environment)

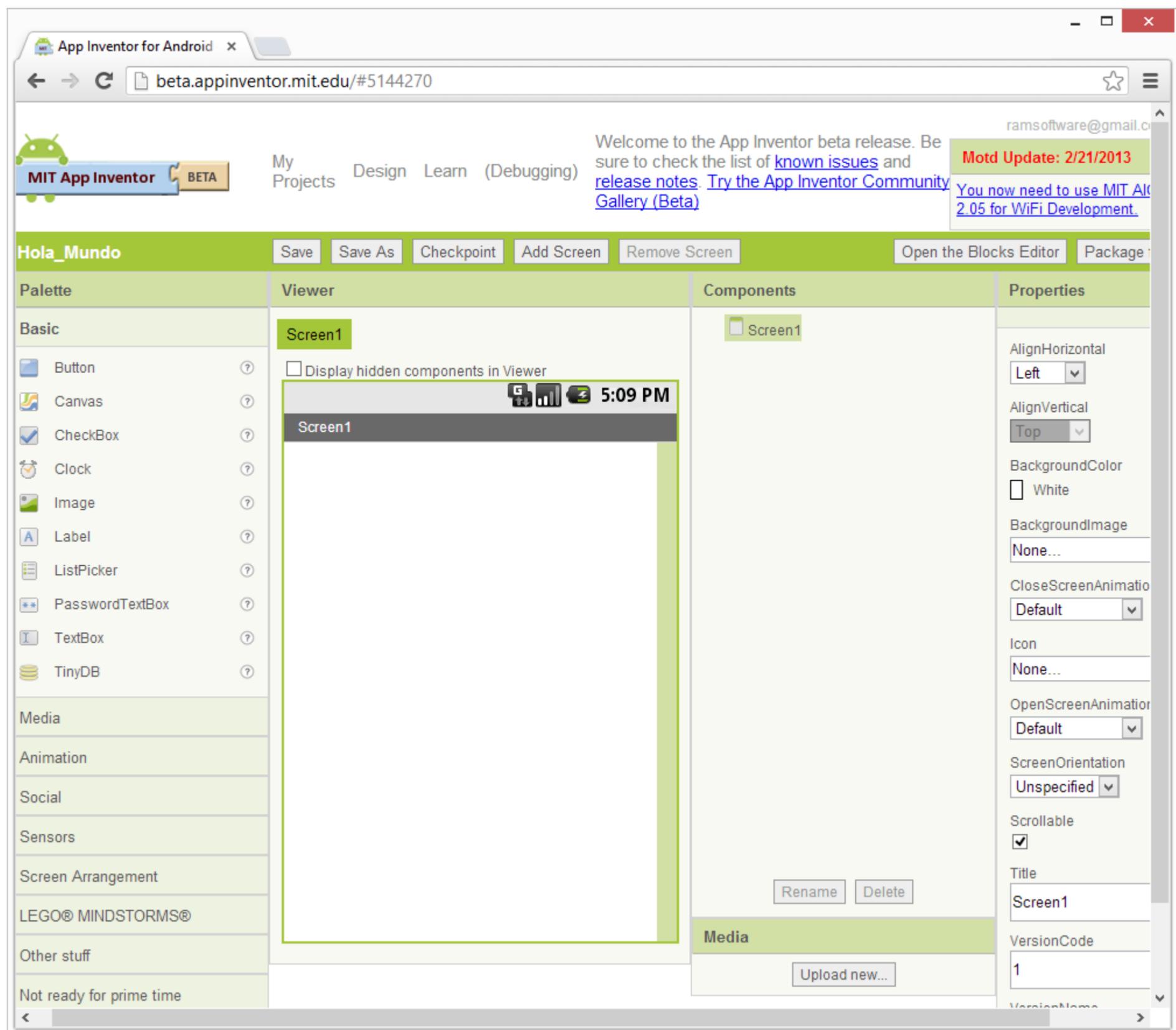


Imagen 32: Pantalla de diseño de pantalla.

Arrastre el control gráfico "TextBox" a la pantalla "Screen1" y observe lo que sucede

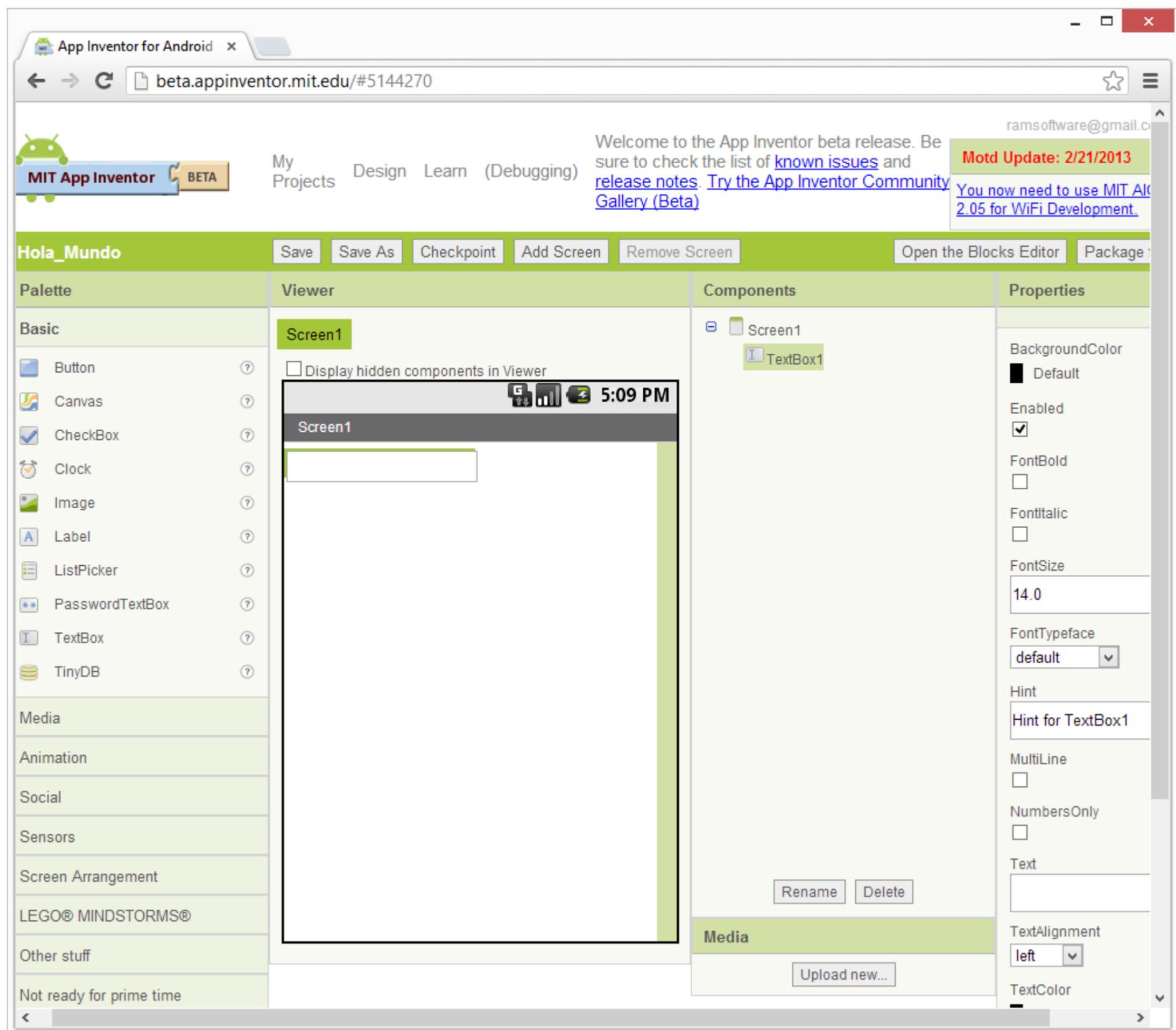


Imagen 33: Arrastre el control gráfico "TextBox" a la pantalla "Screen1"

Arrastre un "Button" a la pantalla "Screen1"

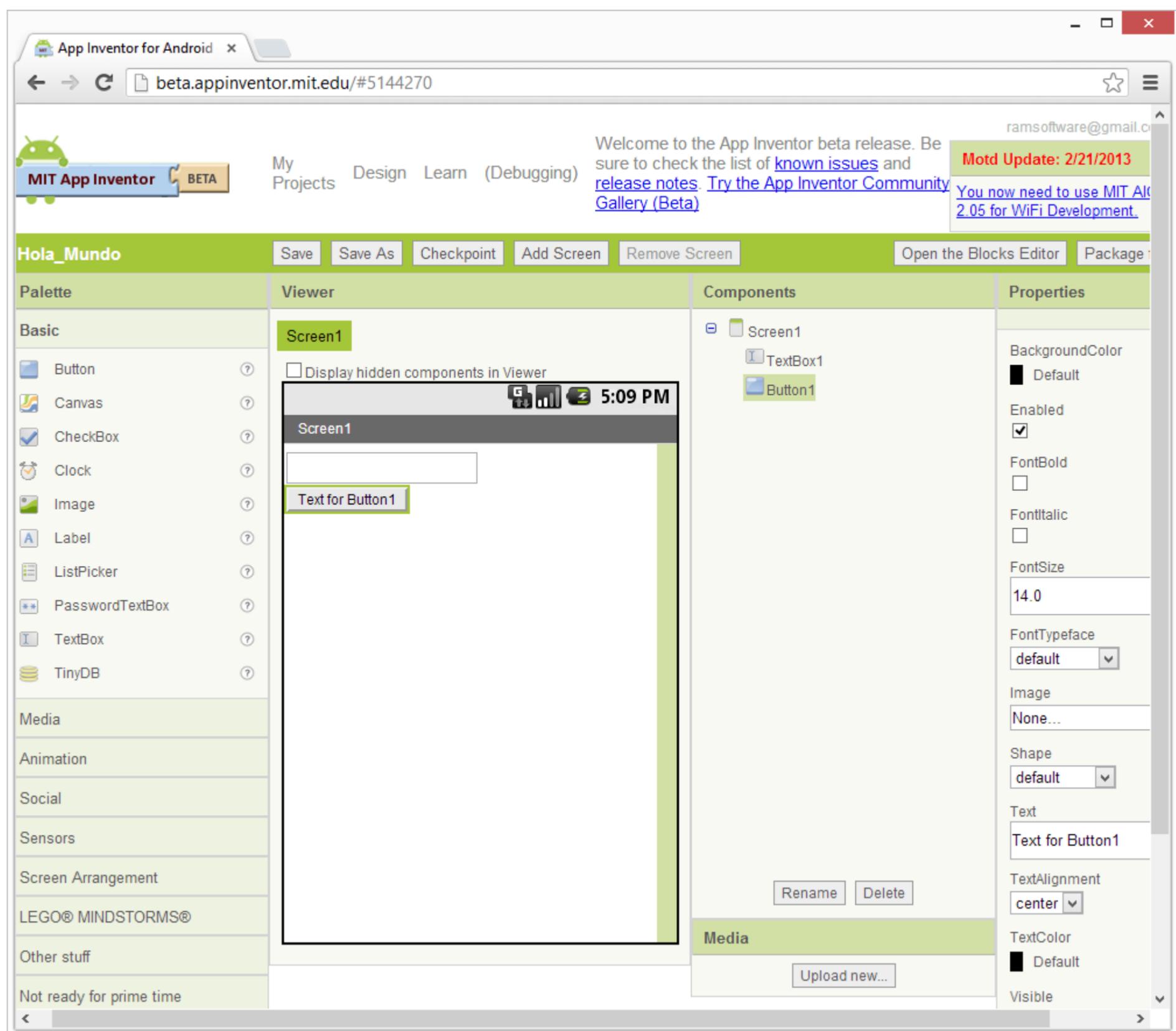


Imagen 34: Arrastre un "Button" a la pantalla "Screen1"

Una muy buena recomendación, es renombrar los objetos gráficos con un estándar. Se usará esta forma:

Objeto que usará	Prefijo y forma de nombrar	Ejemplos
Botón (Button)	btn + Letra mayúscula + resto de nombre	btnAceptar, btnPresionar, btnProcesar
Caja de texto (TextBox)	txt + Letra mayúscula + resto de nombre	txtAviso, txtNombre, txtApellido

Se presiona el botón "Rename" para cambiar cada objeto gráfico.

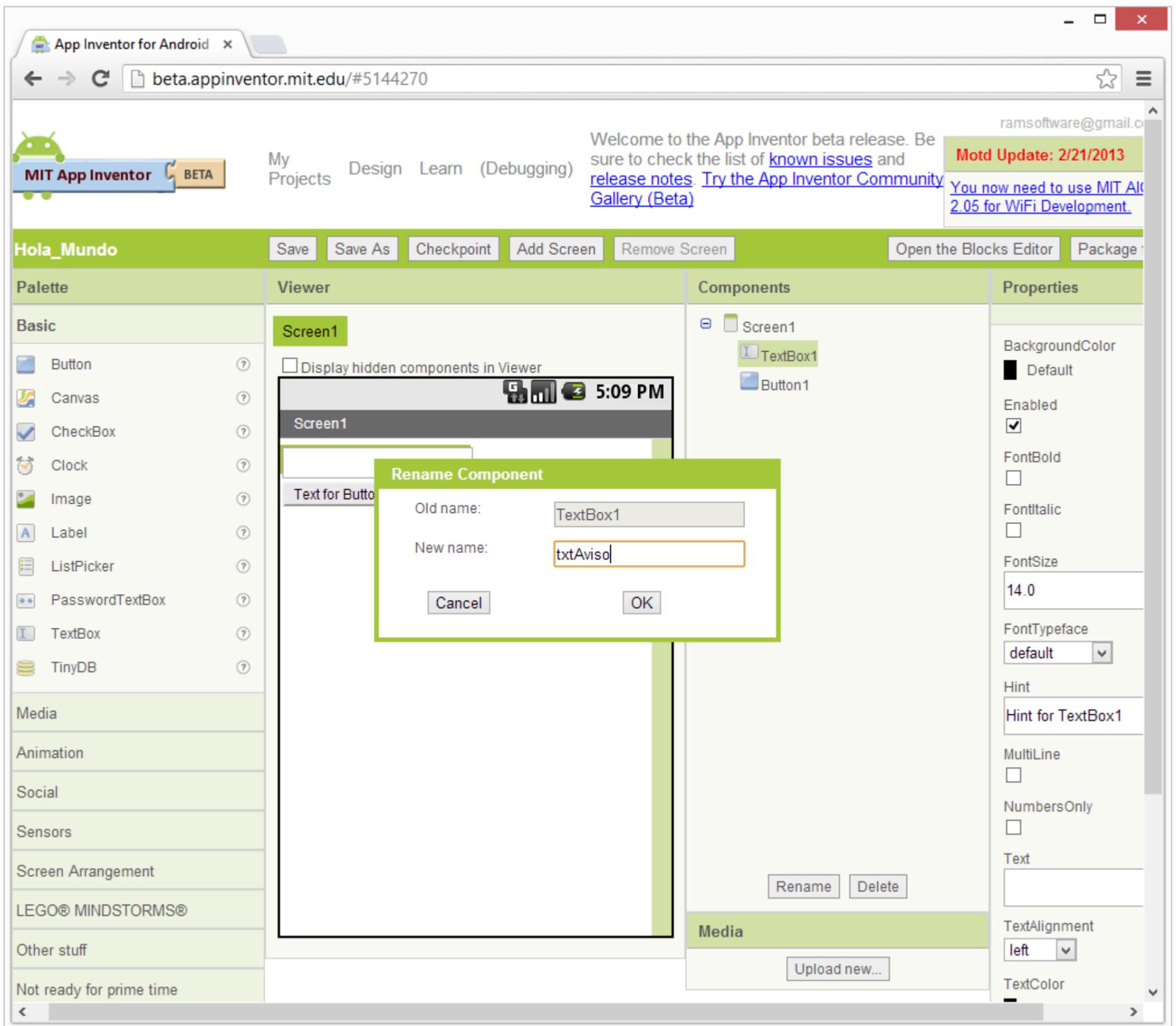


Imagen 35: Renombra el objeto visual TextBox

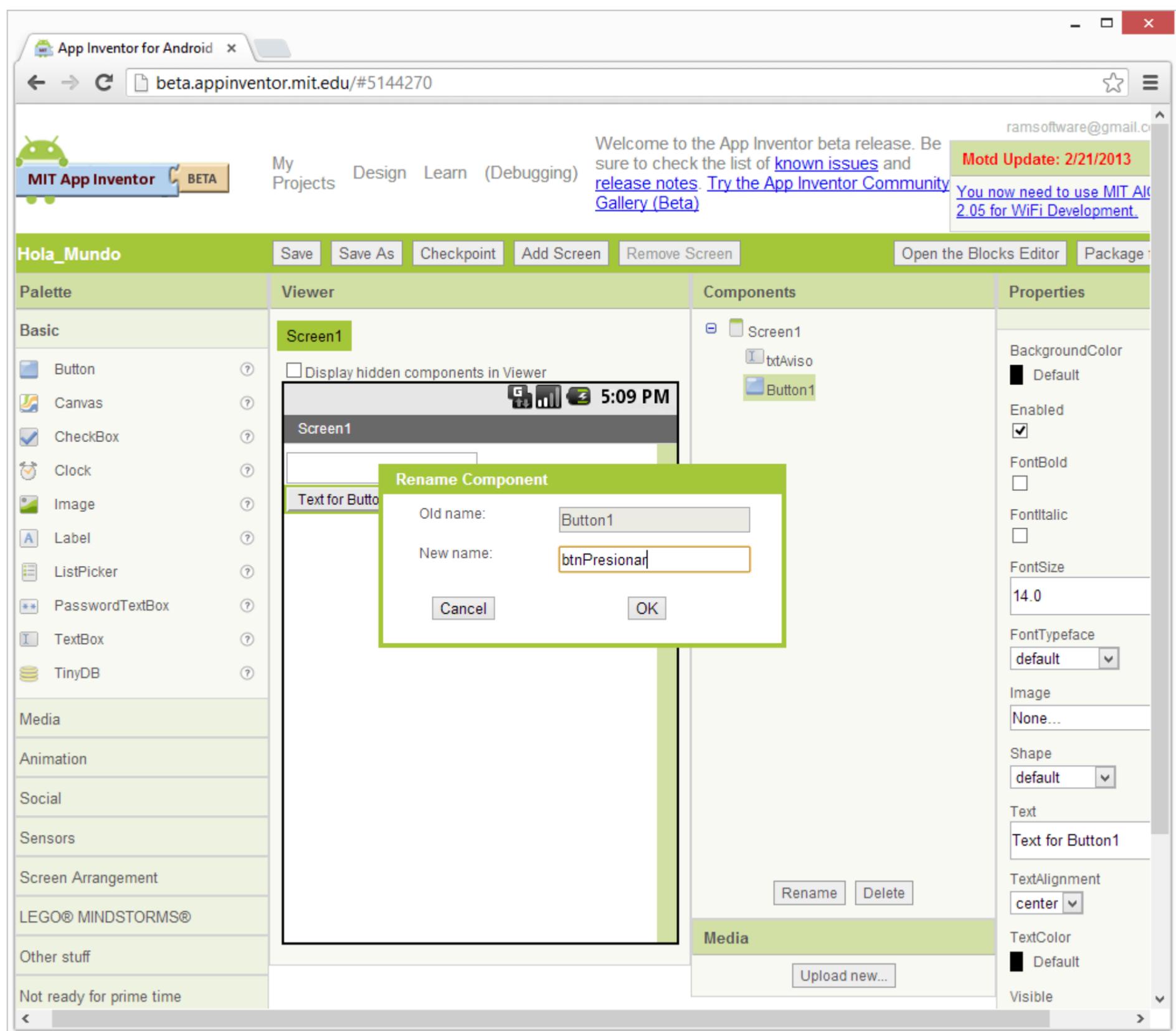


Imagen 36: Renombra el objeto visual "Button"

El siguiente paso es cambiar las propiedades (que se encuentran a la derecha) de cada objeto visual utilizado hasta lograr el efecto deseado.

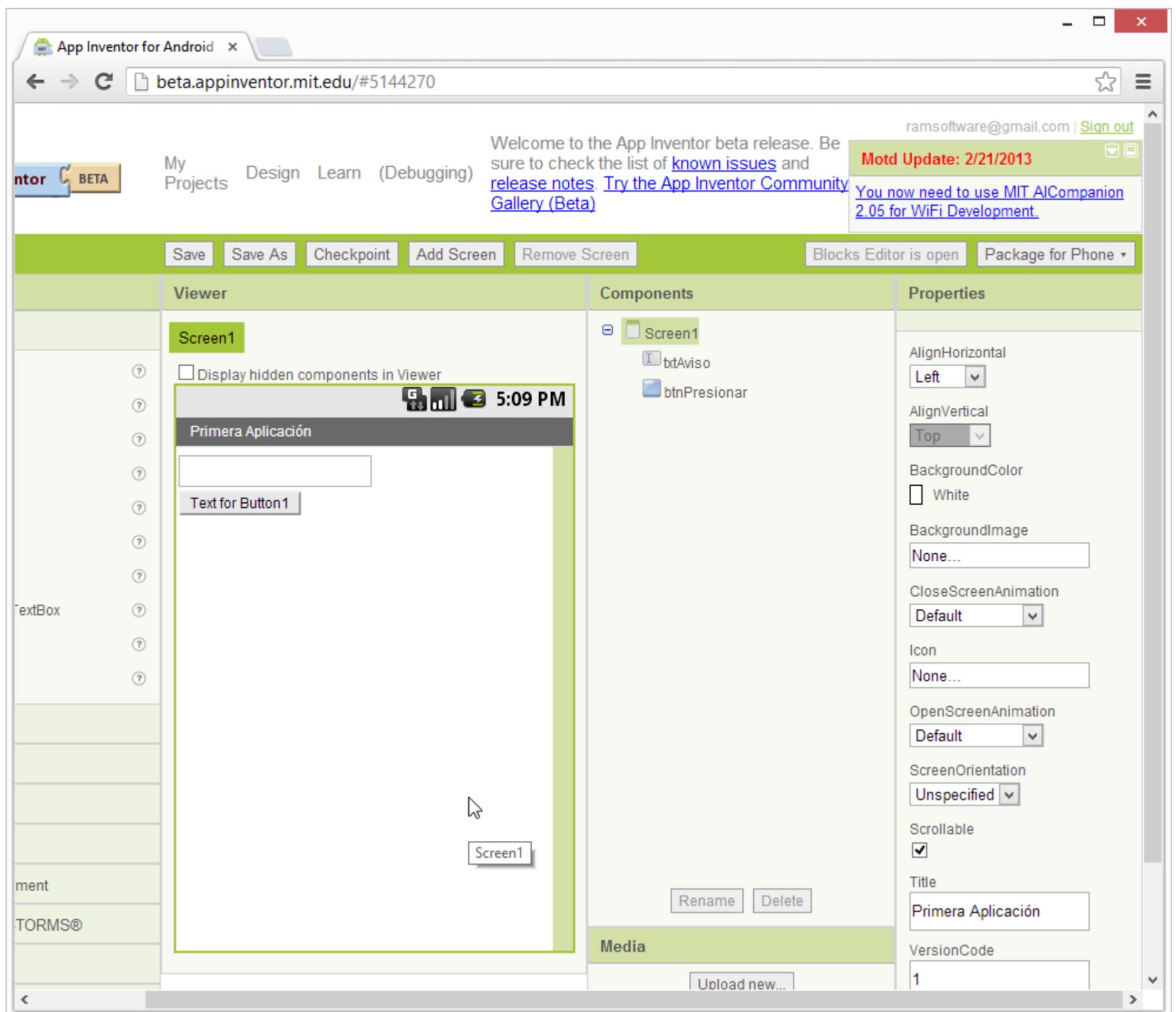


Imagen 37: Cambia la propiedad "título" de la pantalla

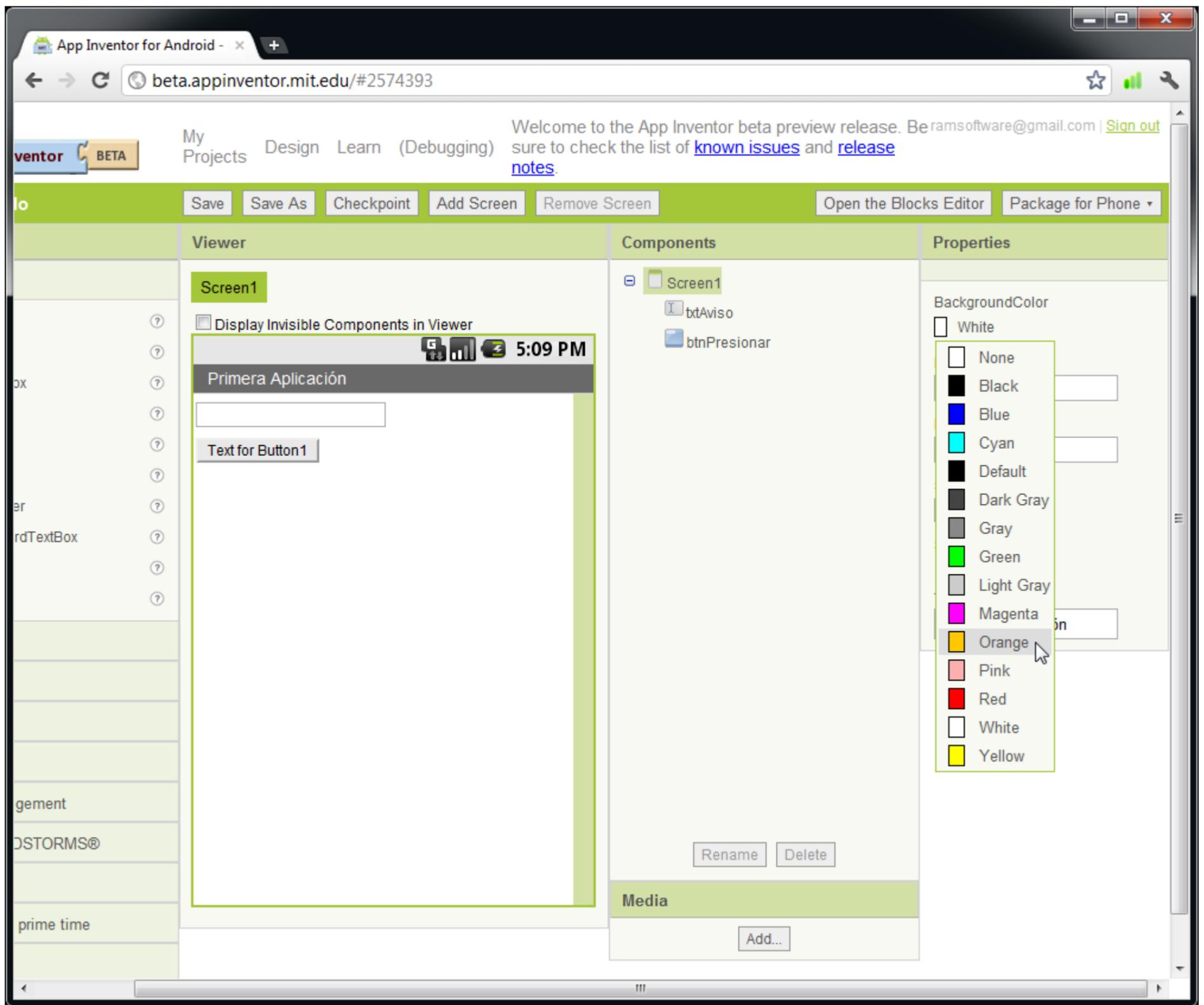


Imagen 38: Cambia el color de fondo de la pantalla

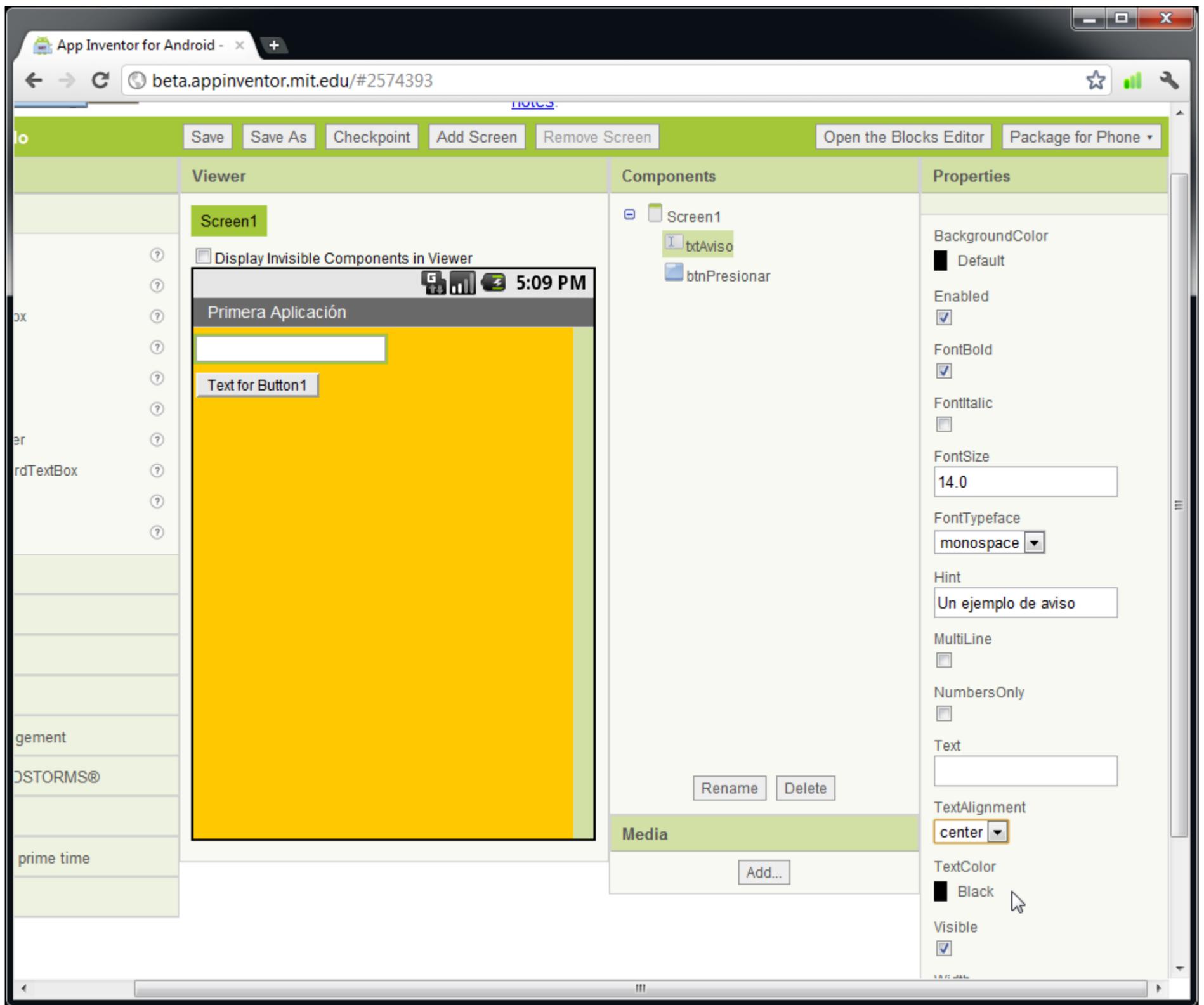


Imagen 39: Cambia las propiedades del TextBox

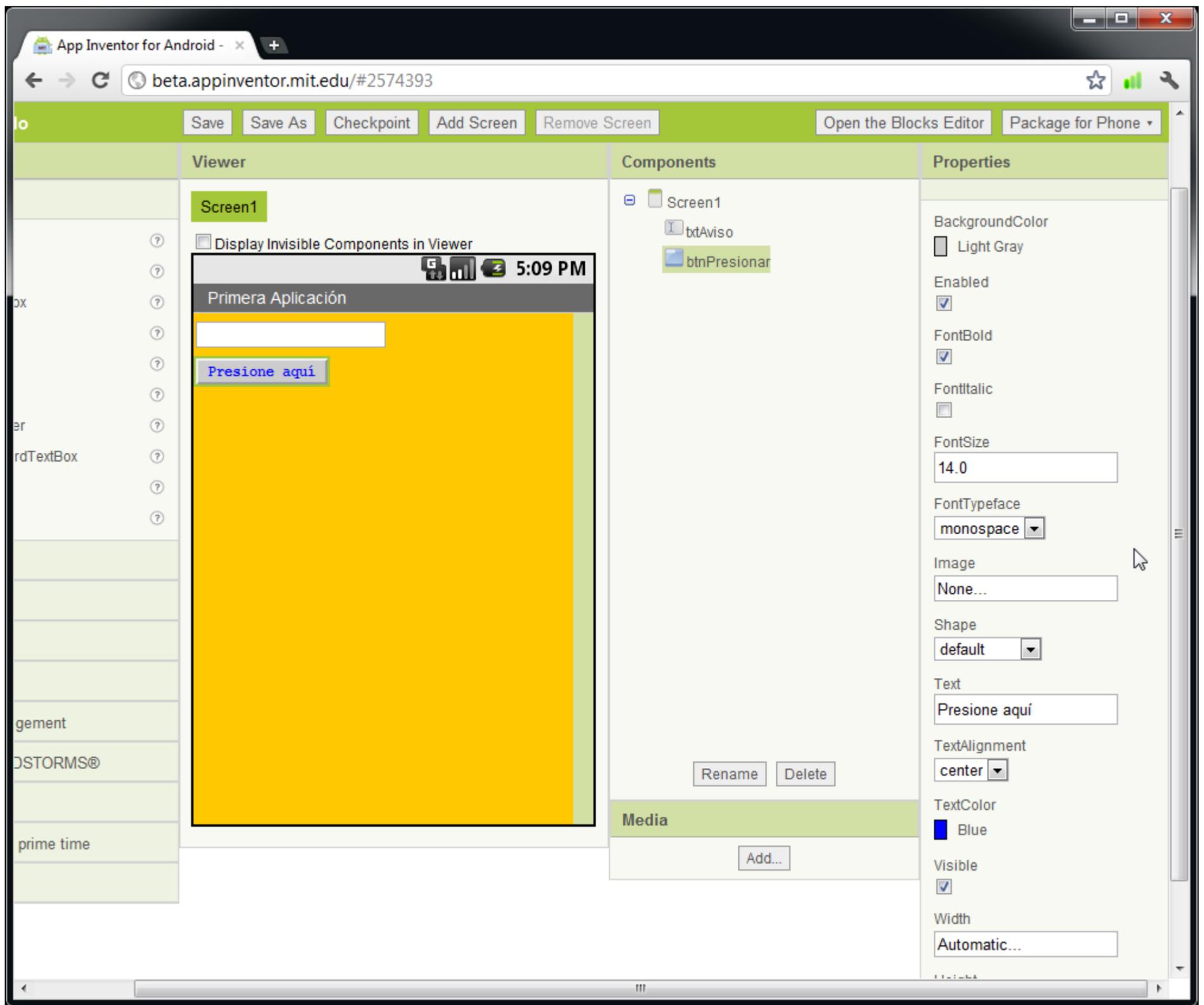


Imagen 40: Cambia las propiedades del botón

Es momento de hacer el software: El objetivo es que al presionar el botón, aparezca el texto "Hola Mundo" en el TextBox. Luego se debe presionar el botón "Open the Blocks Editor"

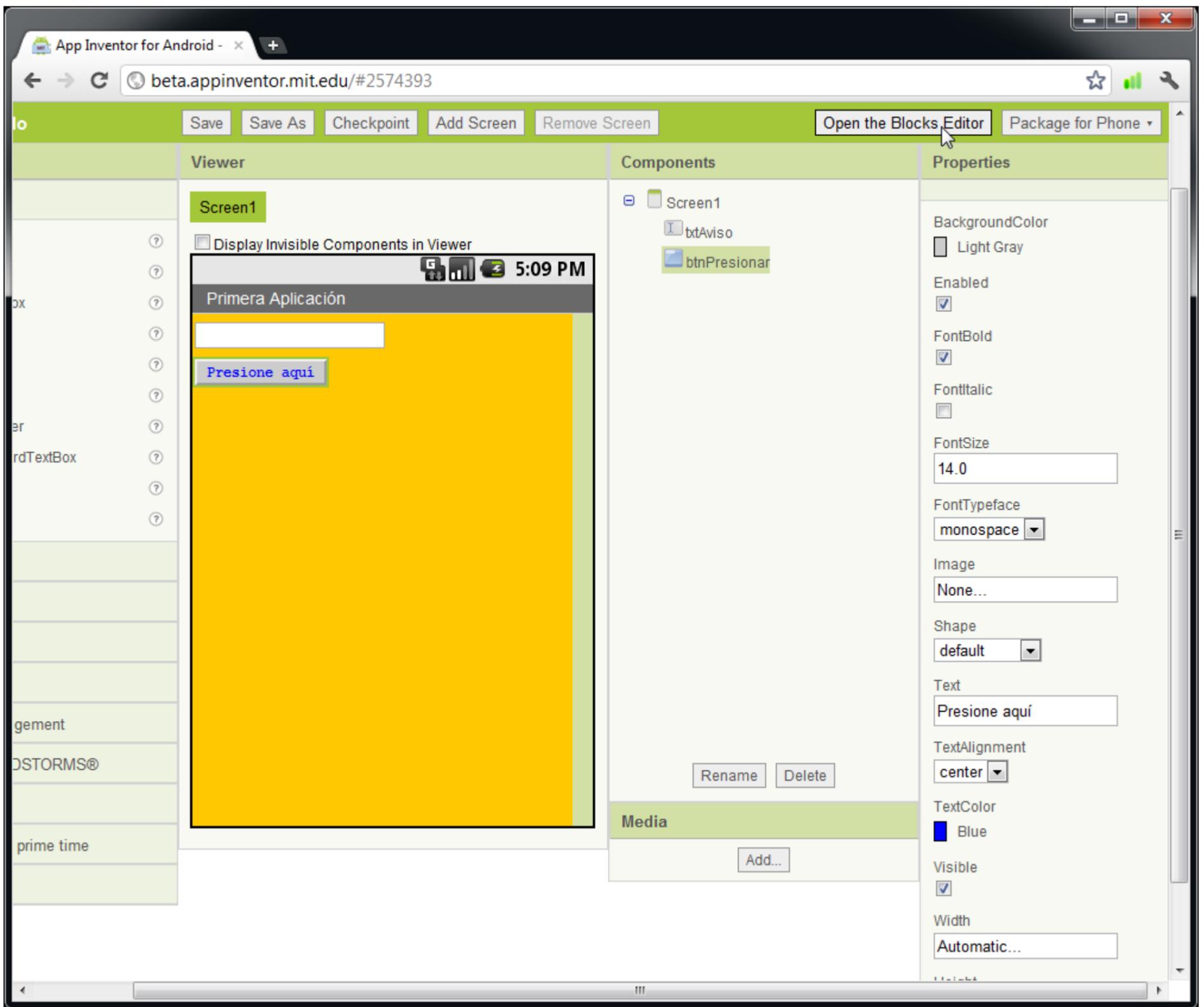


Imagen 41: Presione el botón "Open the Blocks Editor" para empezar a programar

Dependiendo del navegador, hay un aviso de advertencia, en el caso de Google Chrome es sólo es presionar "Descargar" y darle clic al archivo para que ejecute. Saldrá este aviso:



Imagen 42: Aviso de advertencia que se ejecutará una aplicación, de clic en "Ejecutar"

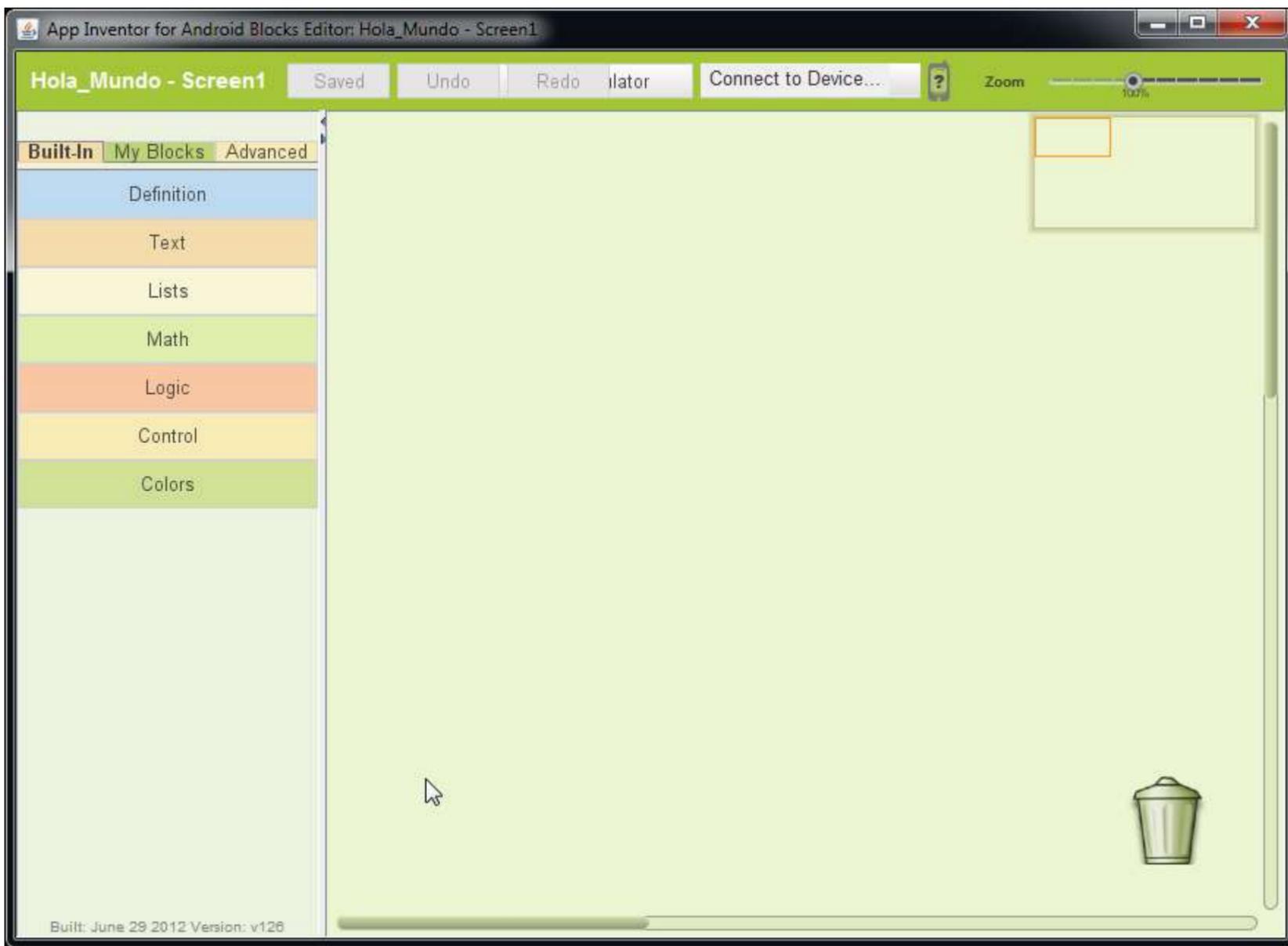


Imagen 43: Pantalla donde se programa la aplicación.

En "My Blocks" aparecen los objetos visuales definidos; allí radica la importancia de nombrarlos bien con estándares:

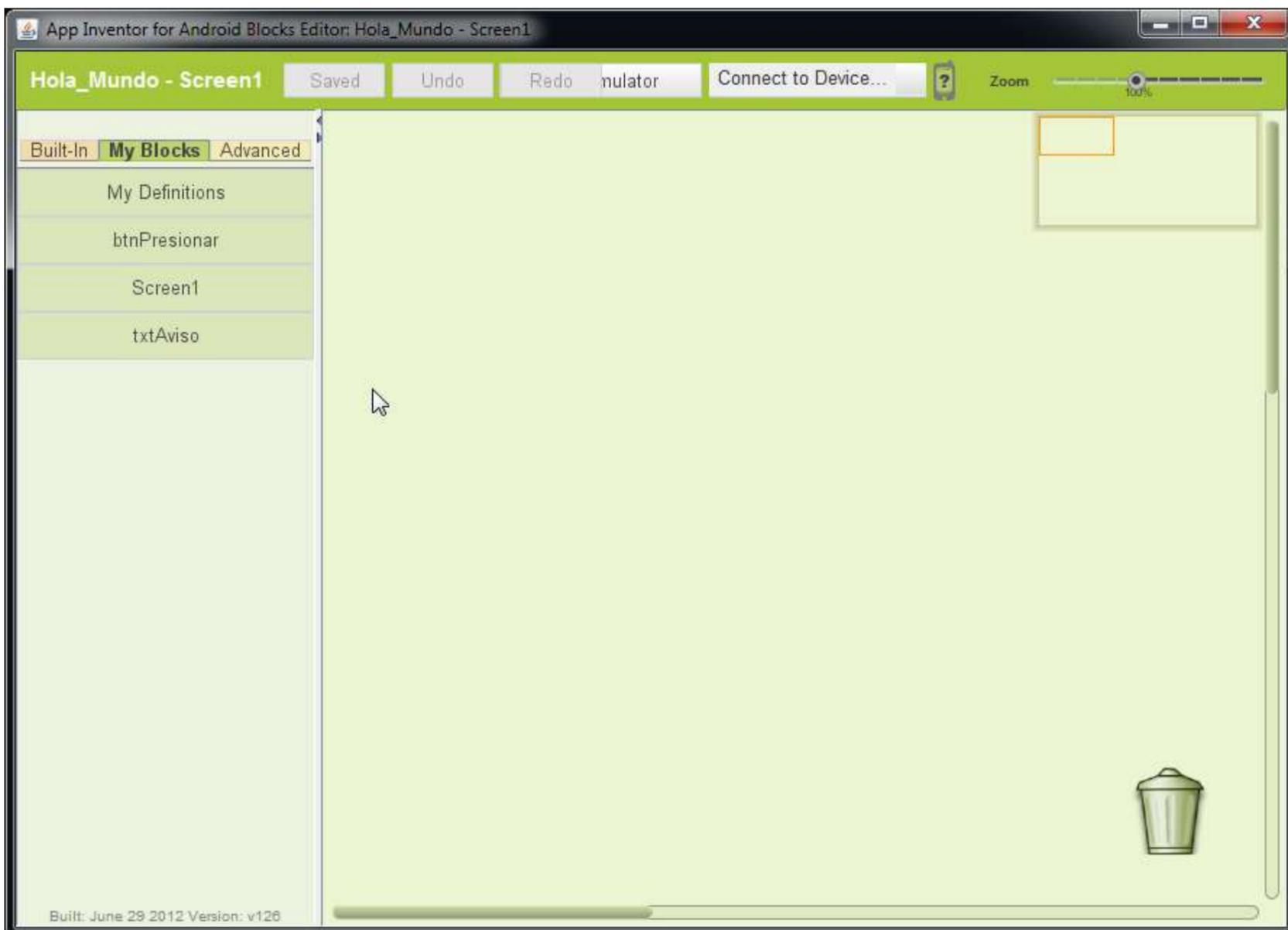


Imagen 44: Objetos visuales aparecen en "My Blocks"

El desarrollo es con piezas de rompecabezas que encajan. Se da clic en "btnPresionar" (un botón), y aparecen los eventos:

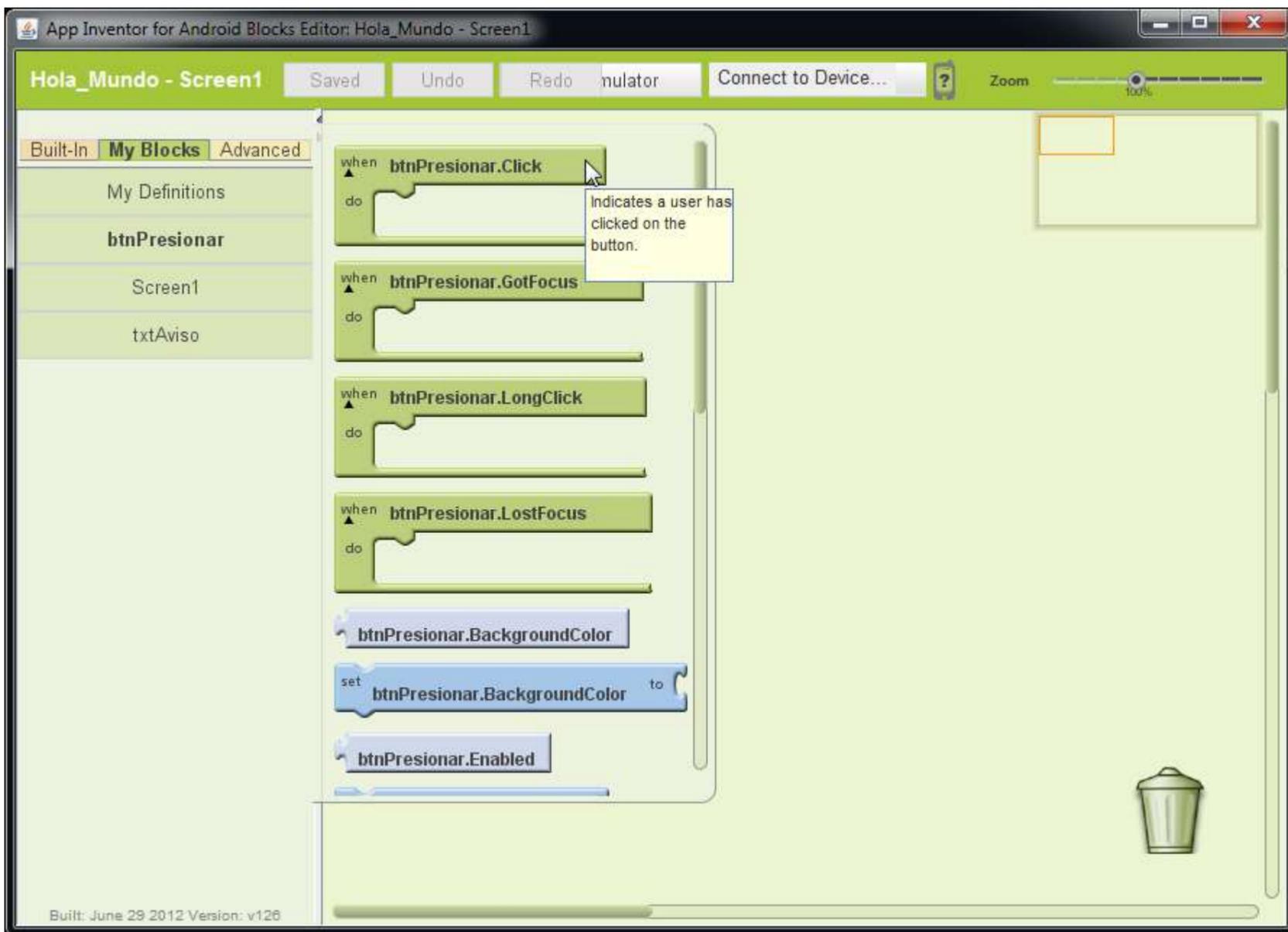


Imagen 45: La primera pieza es un evento que se dispara cuando se da clic en el botón

Arrastre esa pieza del rompecabezas que representa un evento a la pantalla:

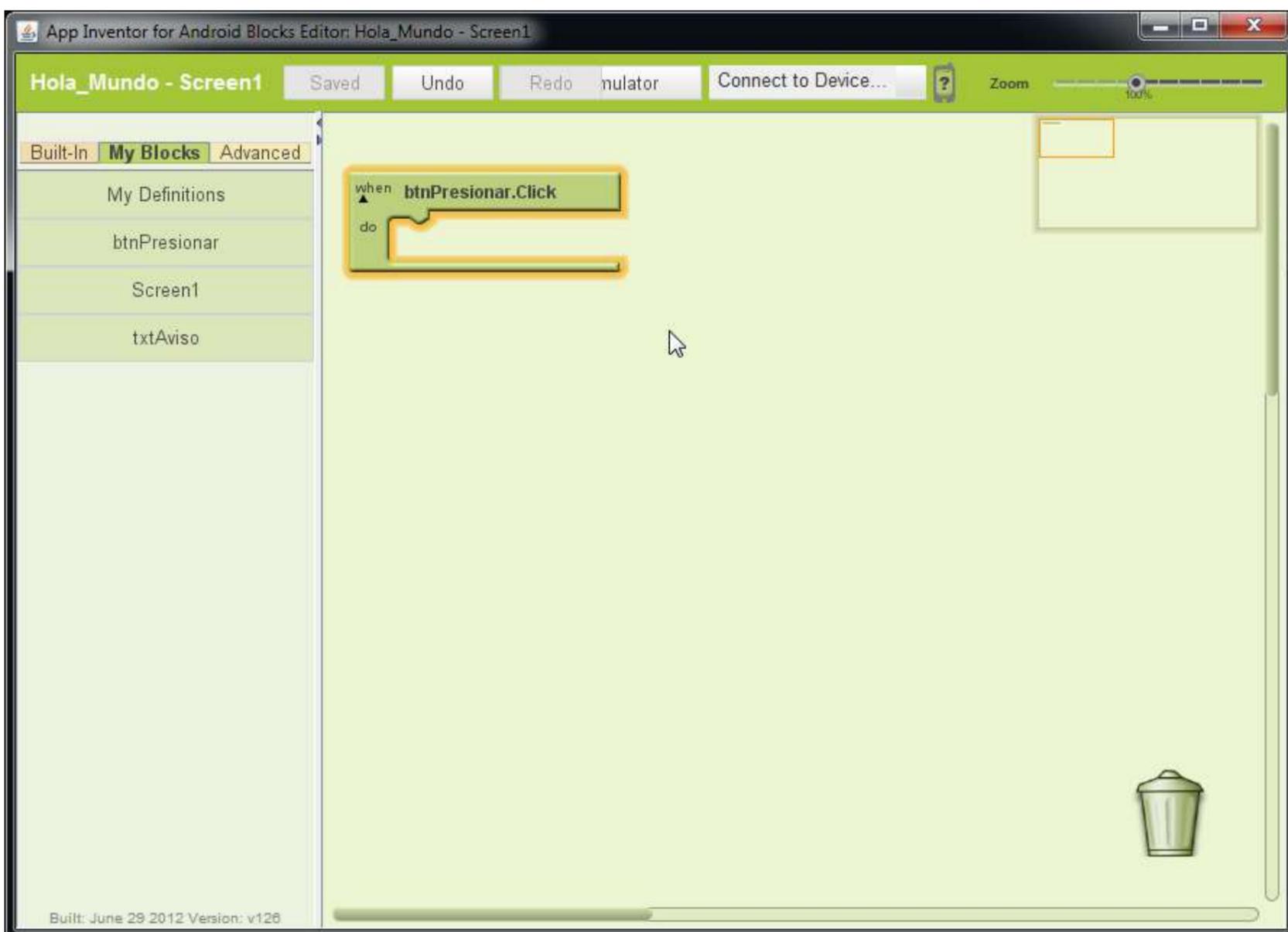


Imagen 46: Arrastrando la pieza que representa un evento a la pantalla

Ahora es el turno del objeto TextBox, al dar clic en este, nos muestra todos los eventos, acciones y cambio de valor de propiedades que pueden hacerse:

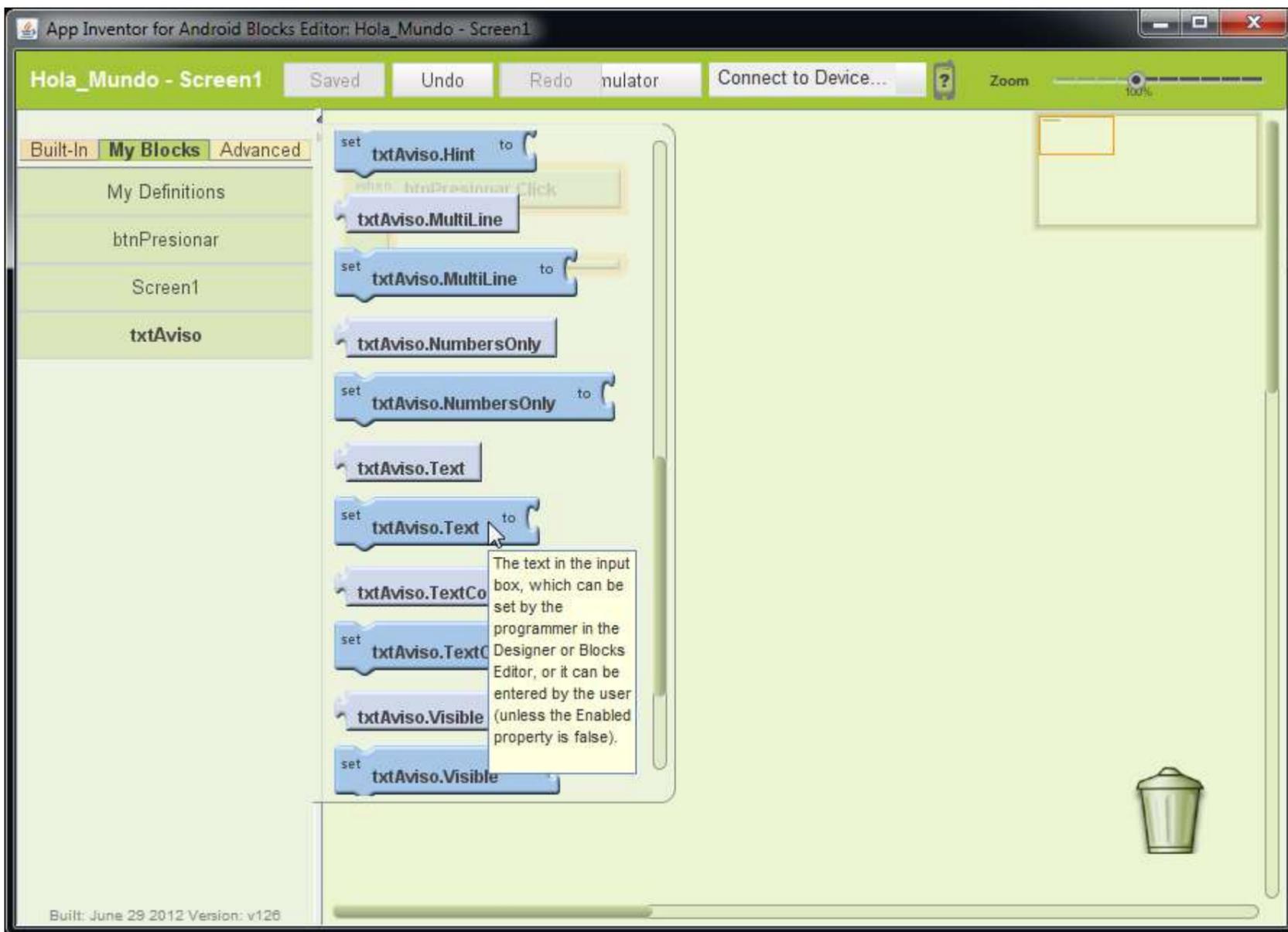


Imagen 47: Arrastre la pieza que dice: "set txtAviso.Text to" a la pantalla

Y observe que la pieza se acopla perfectamente al evento del botón

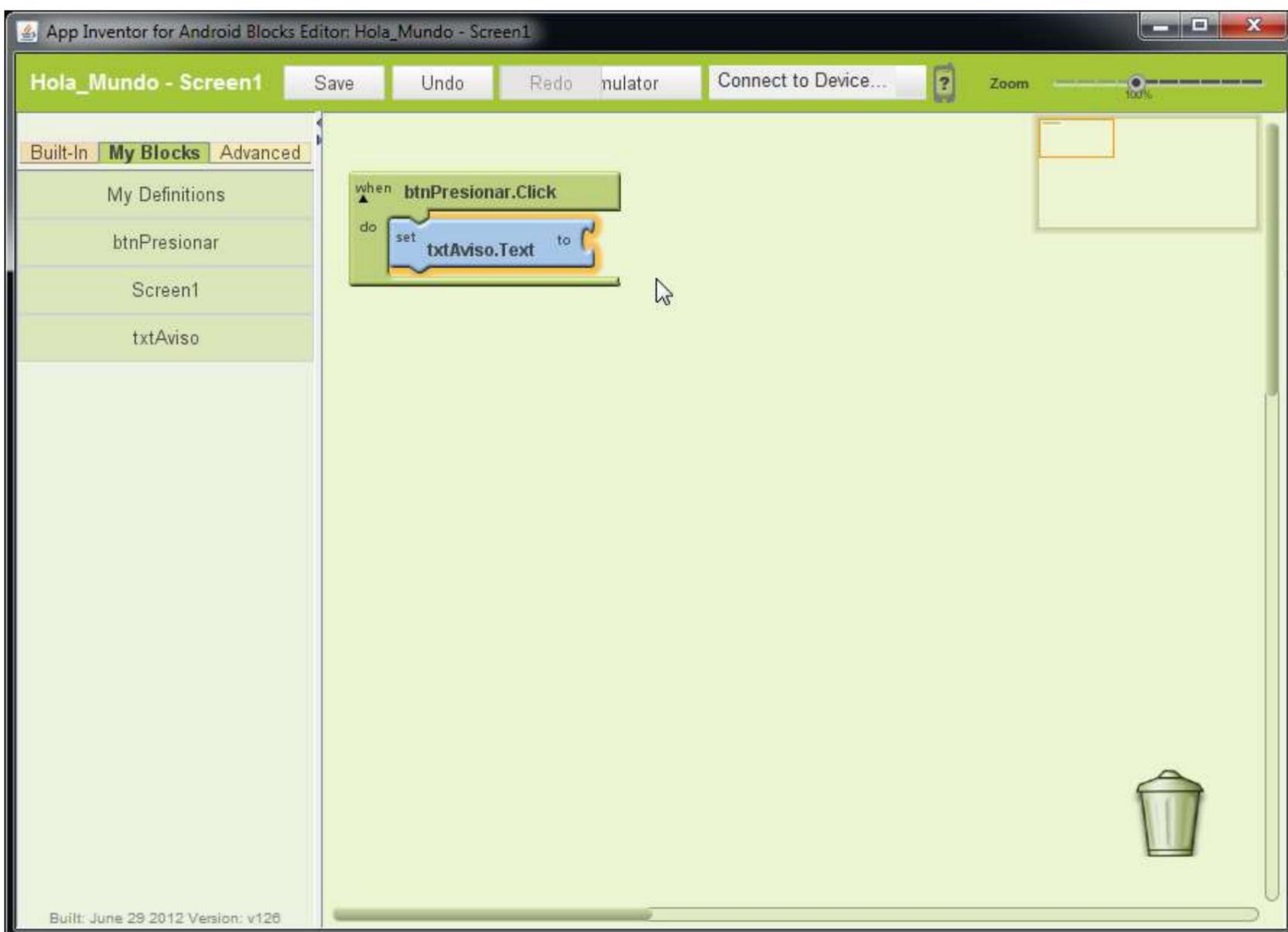


Imagen 48: Pieza se acopla al evento

Ahora es darle un valor al "TextBox", para ello en la sección de "Built-In" seleccione "Text" y de allí la pieza "text **text**", arrástrela hasta que se acople a la pieza de cambio de valor de propiedad del "TextBox"

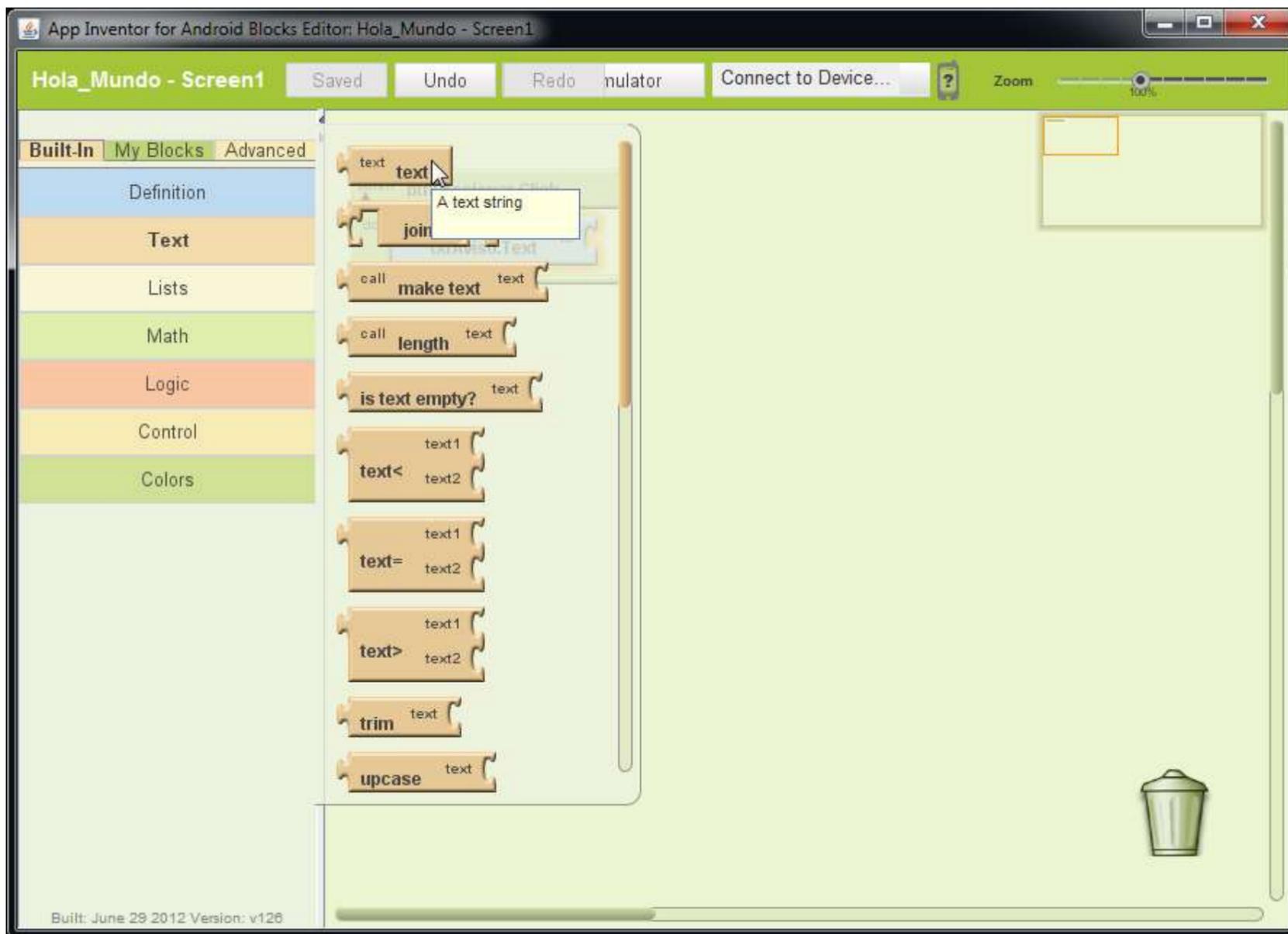


Imagen 49: Seleccione de "Built-In", en "Text", la pieza "text text"

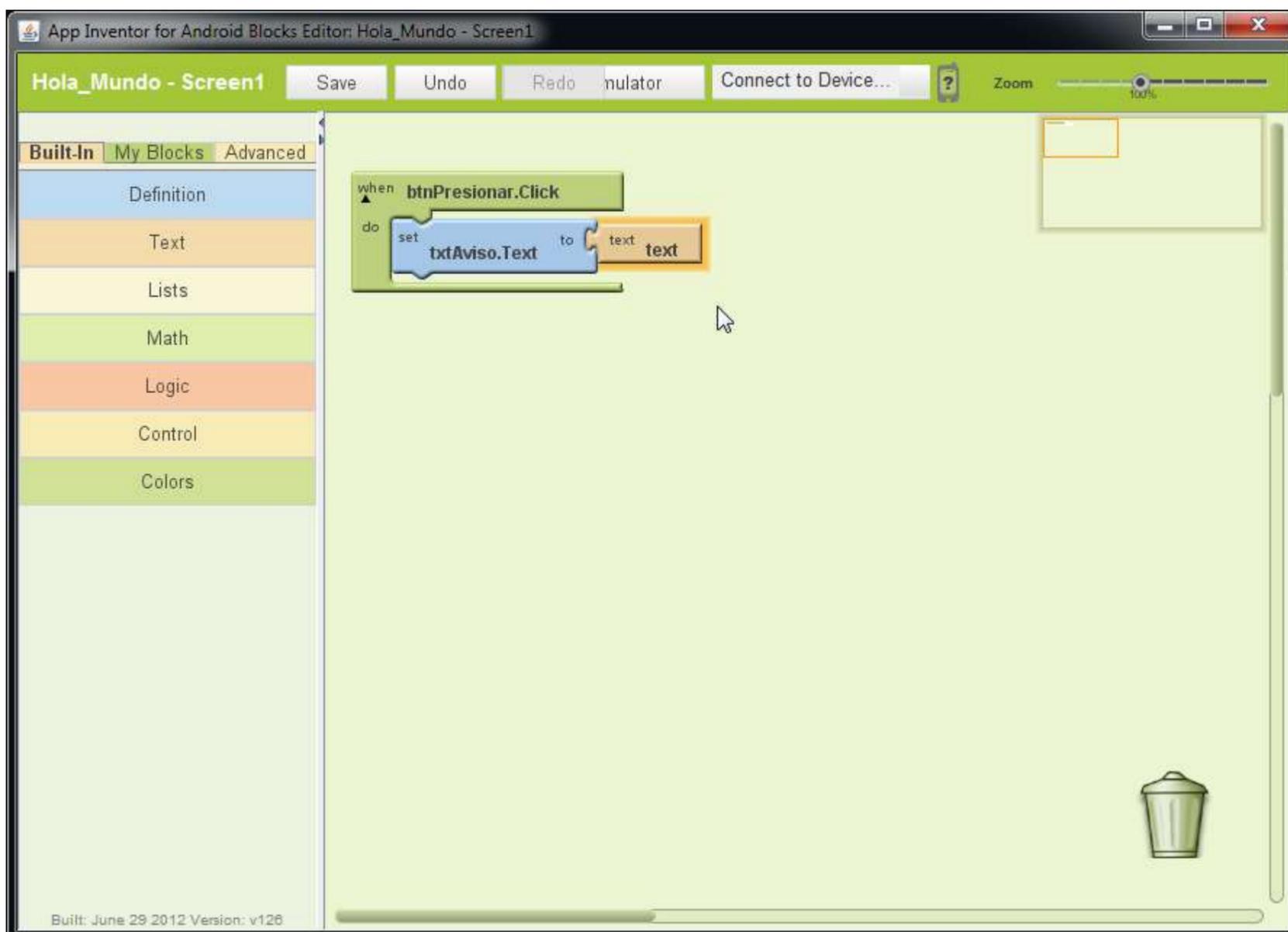


Imagen 50: Pieza acoplada

Escriba el texto "Hola Mundo" en la propiedad "text" como se ve en la imagen

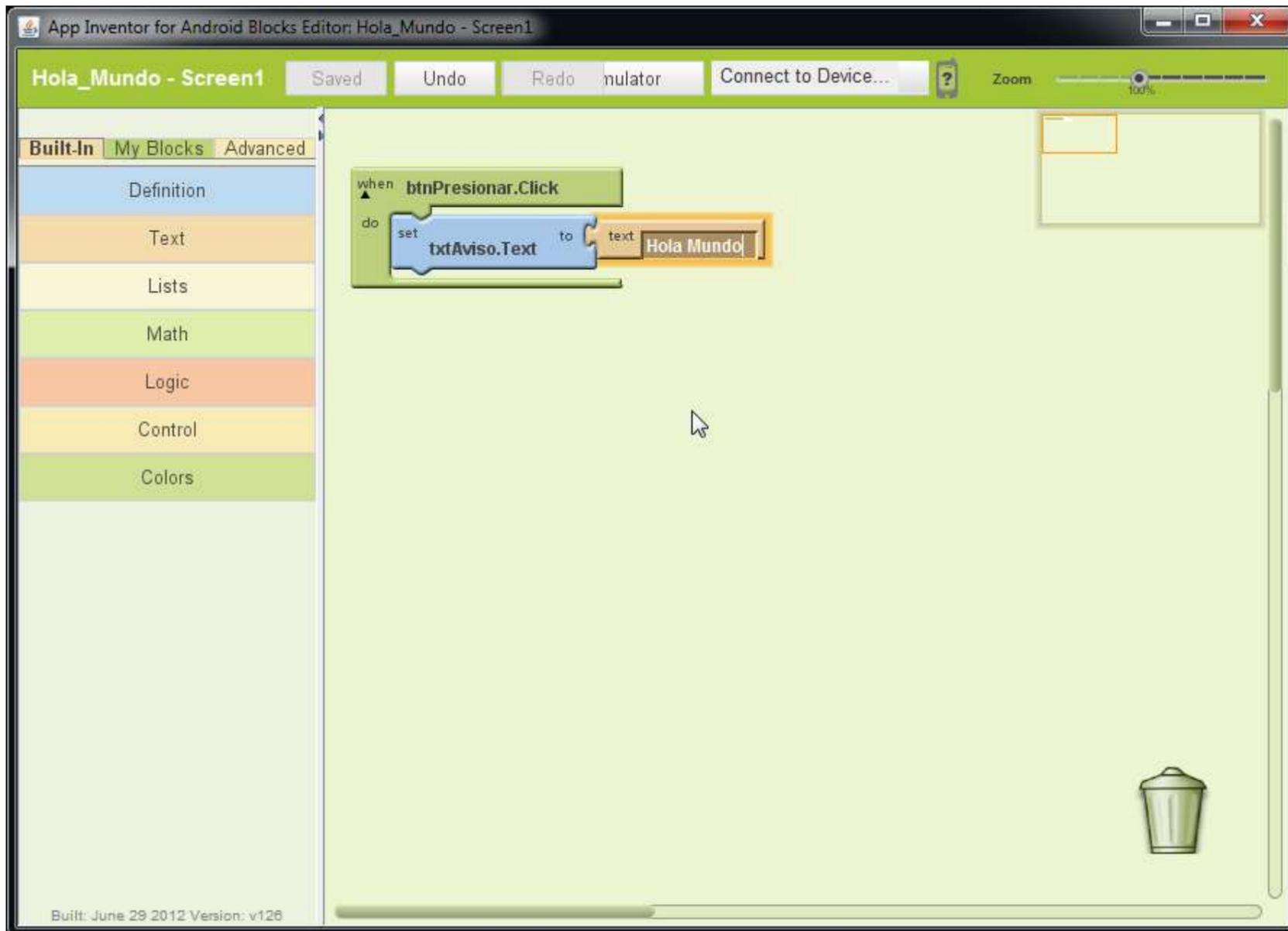


Imagen 51: Cambia el texto por el que se desea

Ya ha terminado, es momento de probar la aplicación en una simulación de Smartphone, de clic en "New Emulator" como se ve en la imagen:

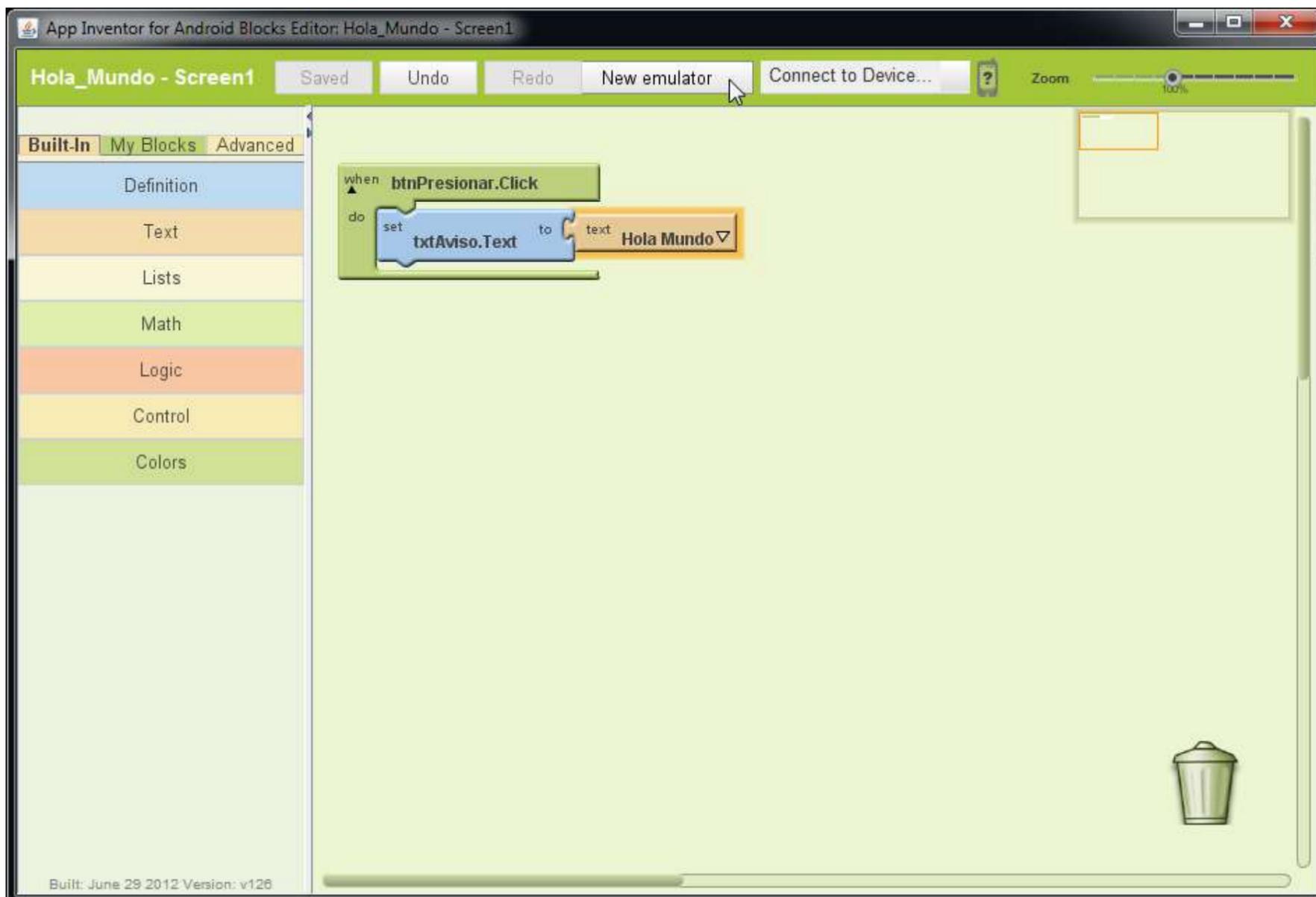


Imagen 52: Activa la simulación de SmartPhone

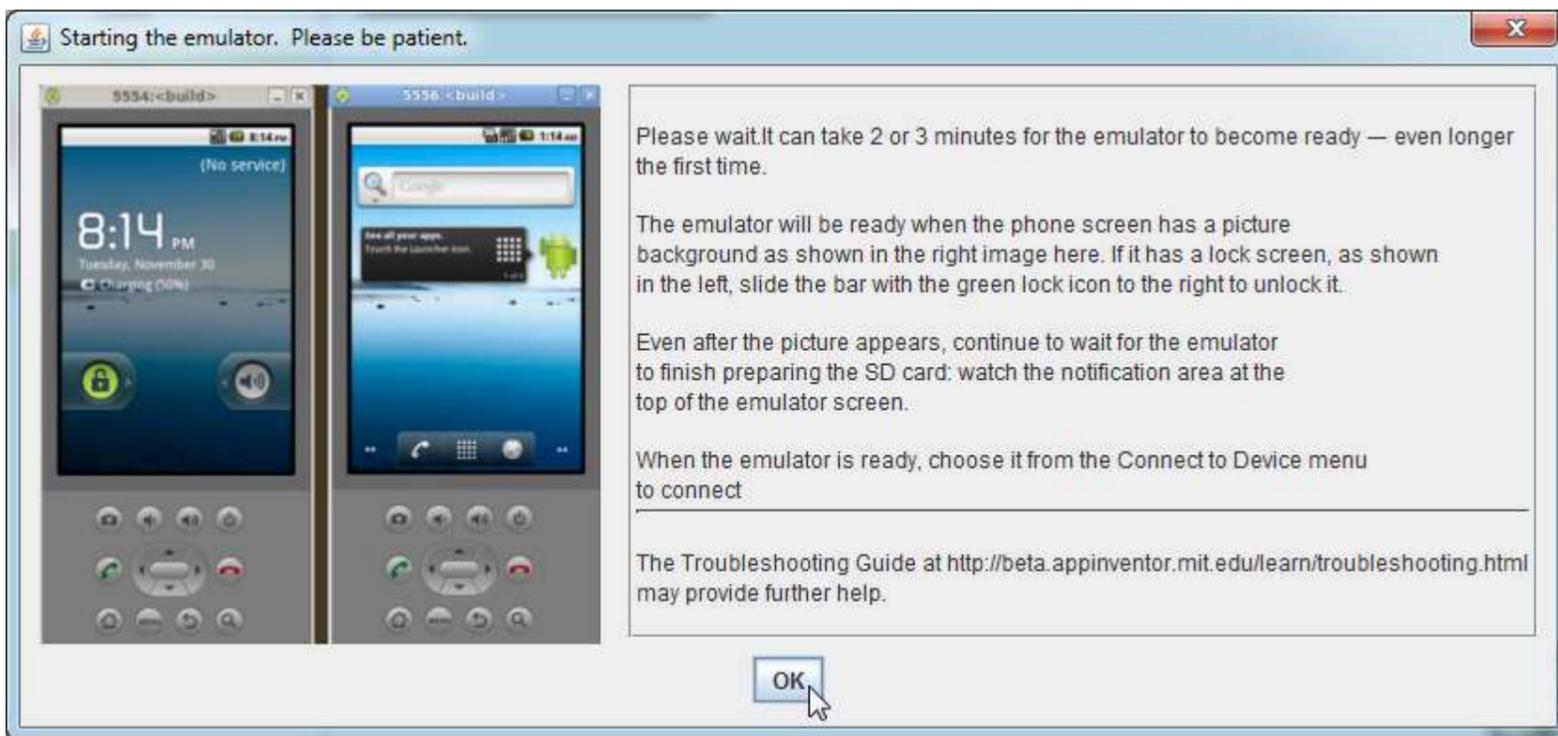


Imagen 53: Aviso que el emulador de SmartPhone inicia, toma de 2 a 3 minutos.



Imagen 54: El emulador ha iniciado

Una vez que el emulador ha iniciado, vuelva a la pantalla "Android Blocks Editor" y presione en "Connect to Device..." y seleccione "emulador-5554"

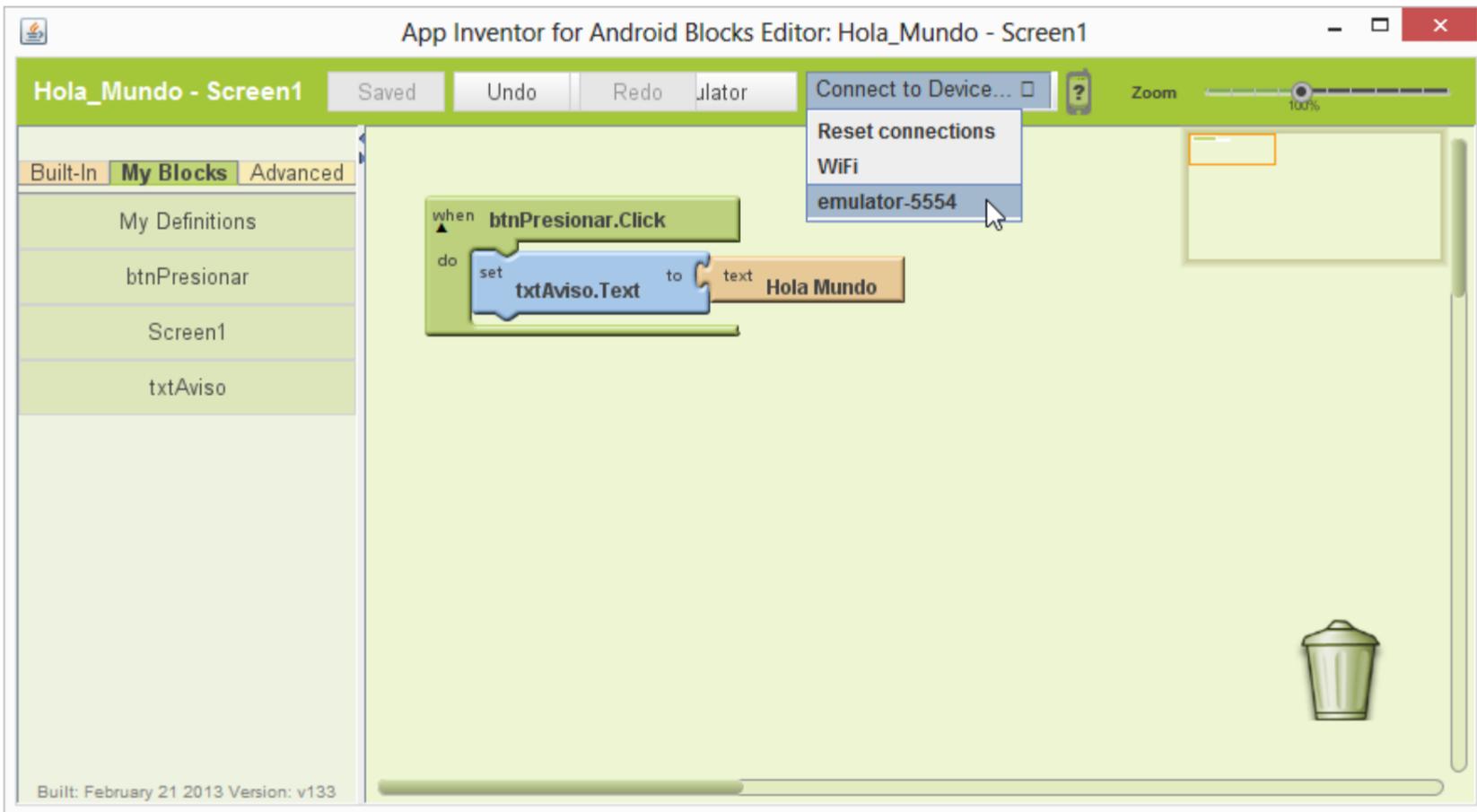


Imagen 55: Seleccione "Connect to Device..." y de allí a "emulator-5554"

Tomará tiempo, sea paciente

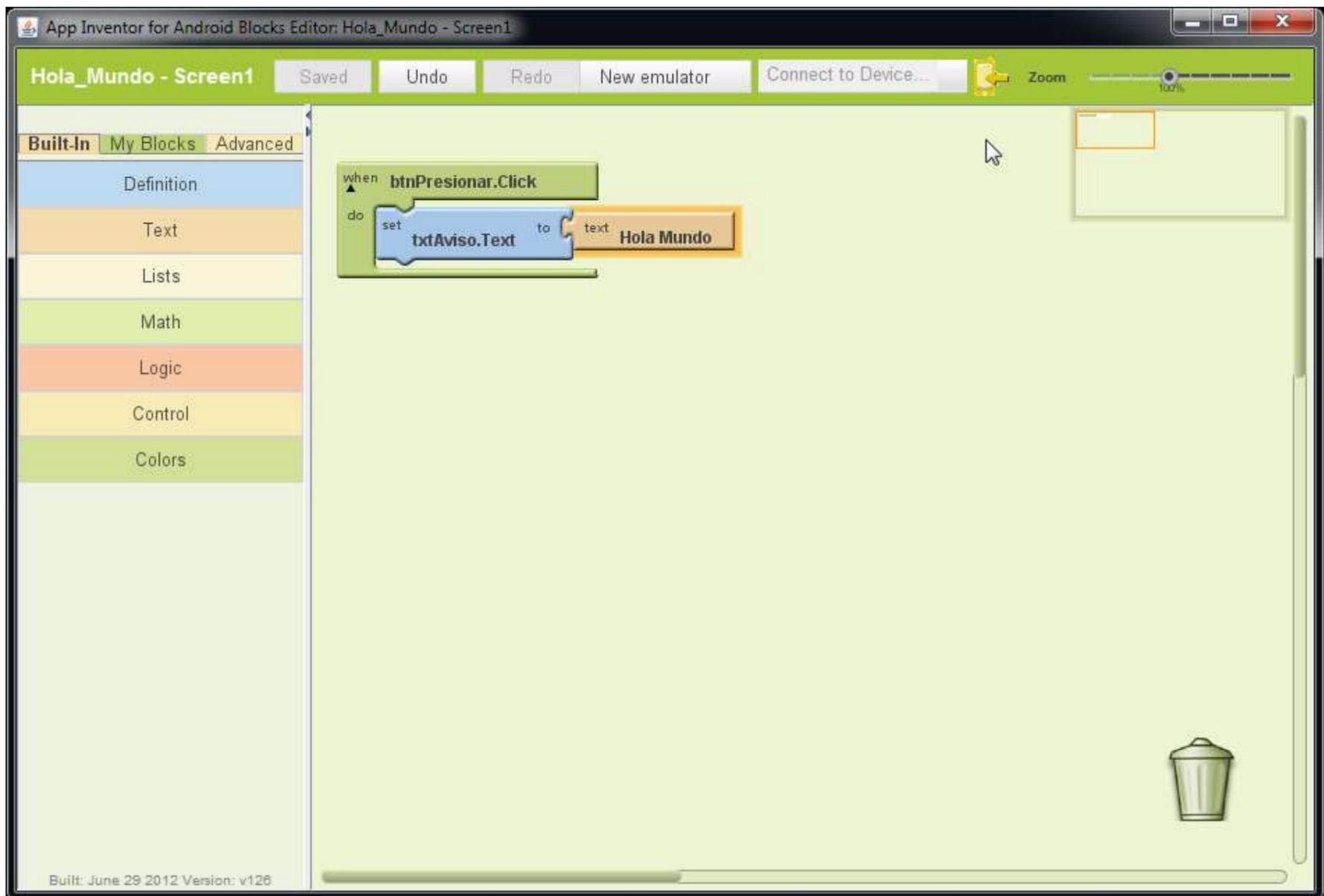


Imagen 56: Toma un tiempo.

Ya está ejecutando en el emulador, observe que el icono se puso en verde.

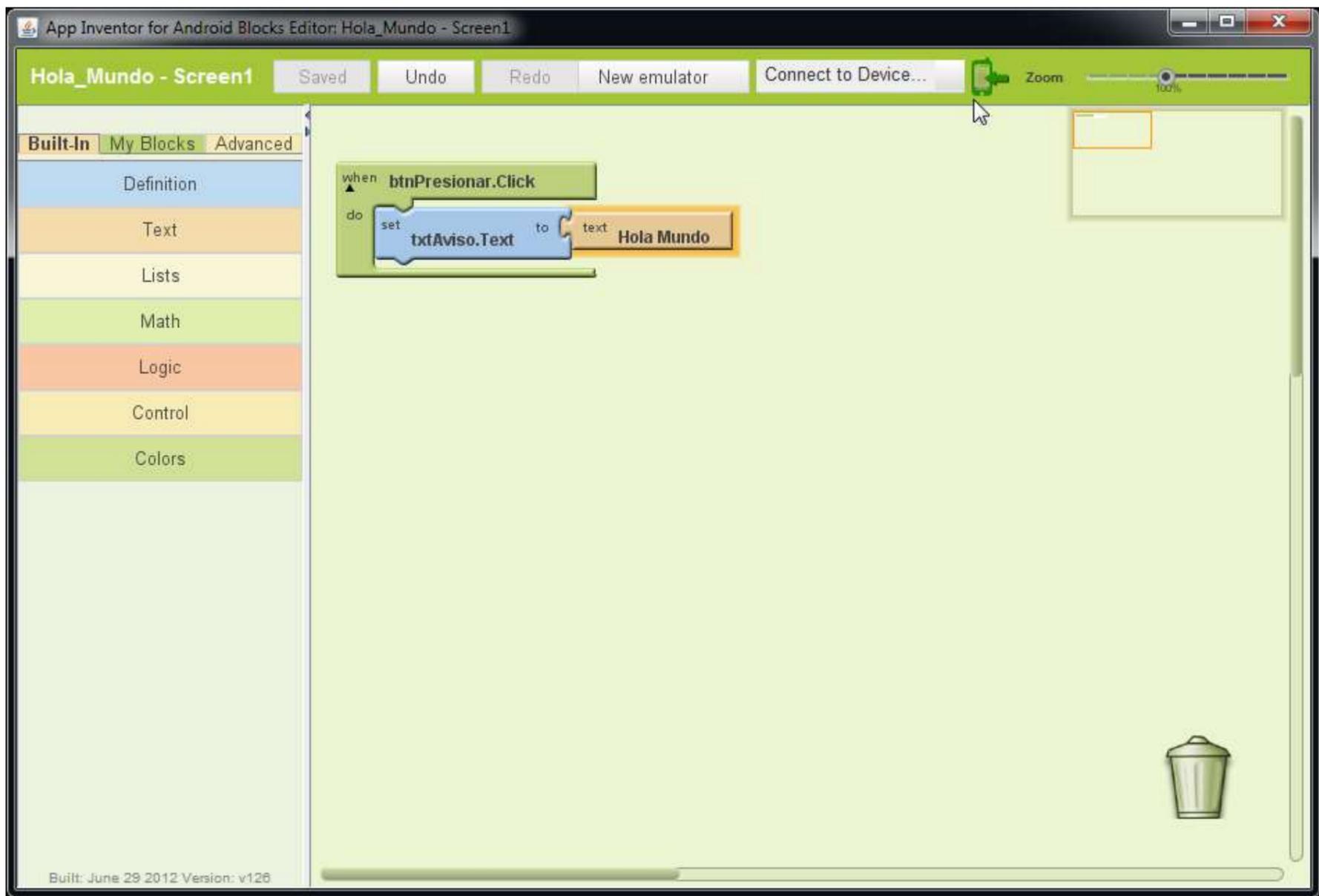


Imagen 57: Icono en verde señala que el programa está corriendo en el emulador

Desbloquee el SmartPhone deslizando la barra

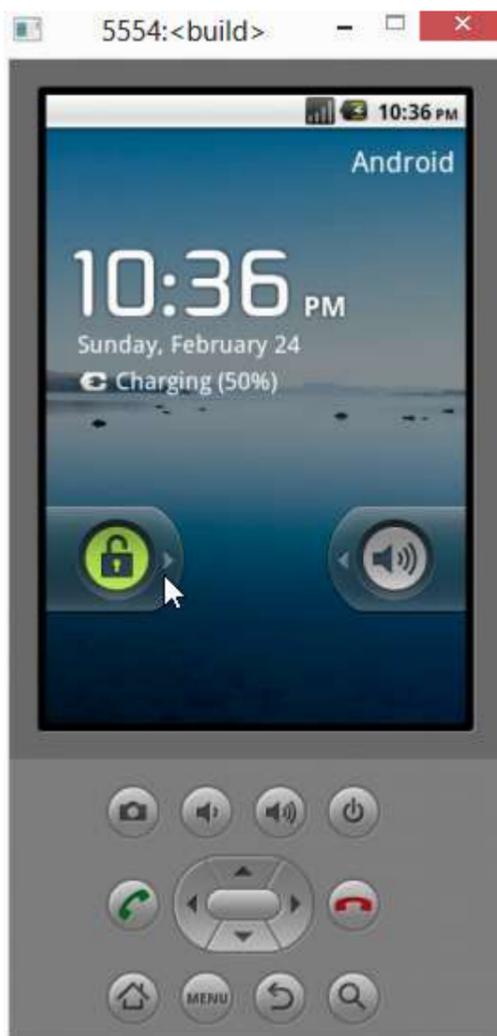


Imagen 58: Desbloqueando el SmartPhone

Y la aplicación correrá

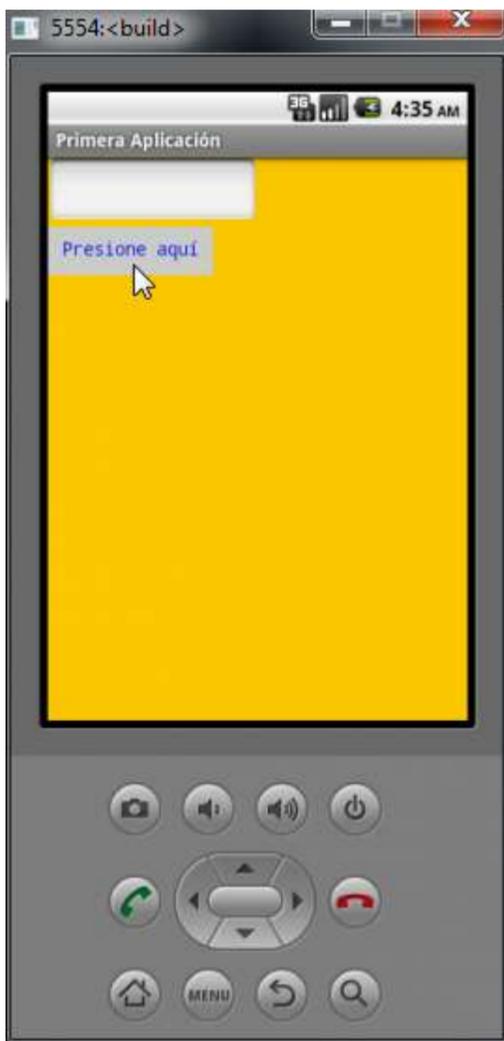


Imagen 59: Aplicación ejecutándose en un SmartPhone (emulación)

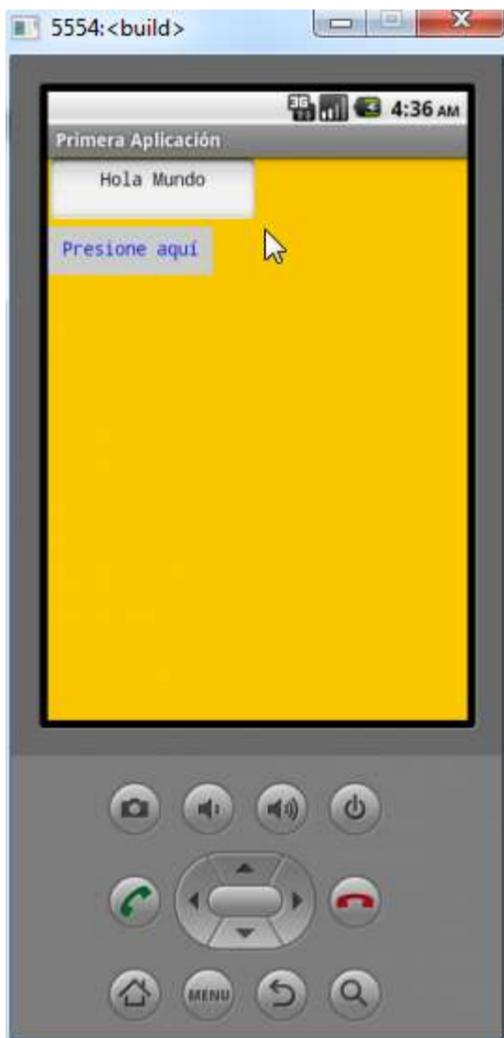


Imagen 60: Demostración que funciona la aplicación

Paso 6. Leer un par de números y luego sumarlos

Aquí aprenderá cómo es la captura de datos y una operación matemática.

Esta es la pantalla, tiene los siguientes objetos gráficos: etiquetas (Label), cajas de texto (TextBox) y un botón (Button). Ya los objetos fueron nombrados con el estándar.

Objeto que usará	Prefijo y forma de nombrar	Ejemplos
Botón (Button)	btn + Letra mayúscula + resto de nombre	btnAceptar, btnPresionar, btnProcesar
Caja de texto (TextBox)	txt + Letra mayúscula + resto de nombre	txtAviso, txtNombre, txtApellido
Etiqueta (Label)	lbl + Letra mayúscula + resto de nombre	lblNumeroA, lblResultado

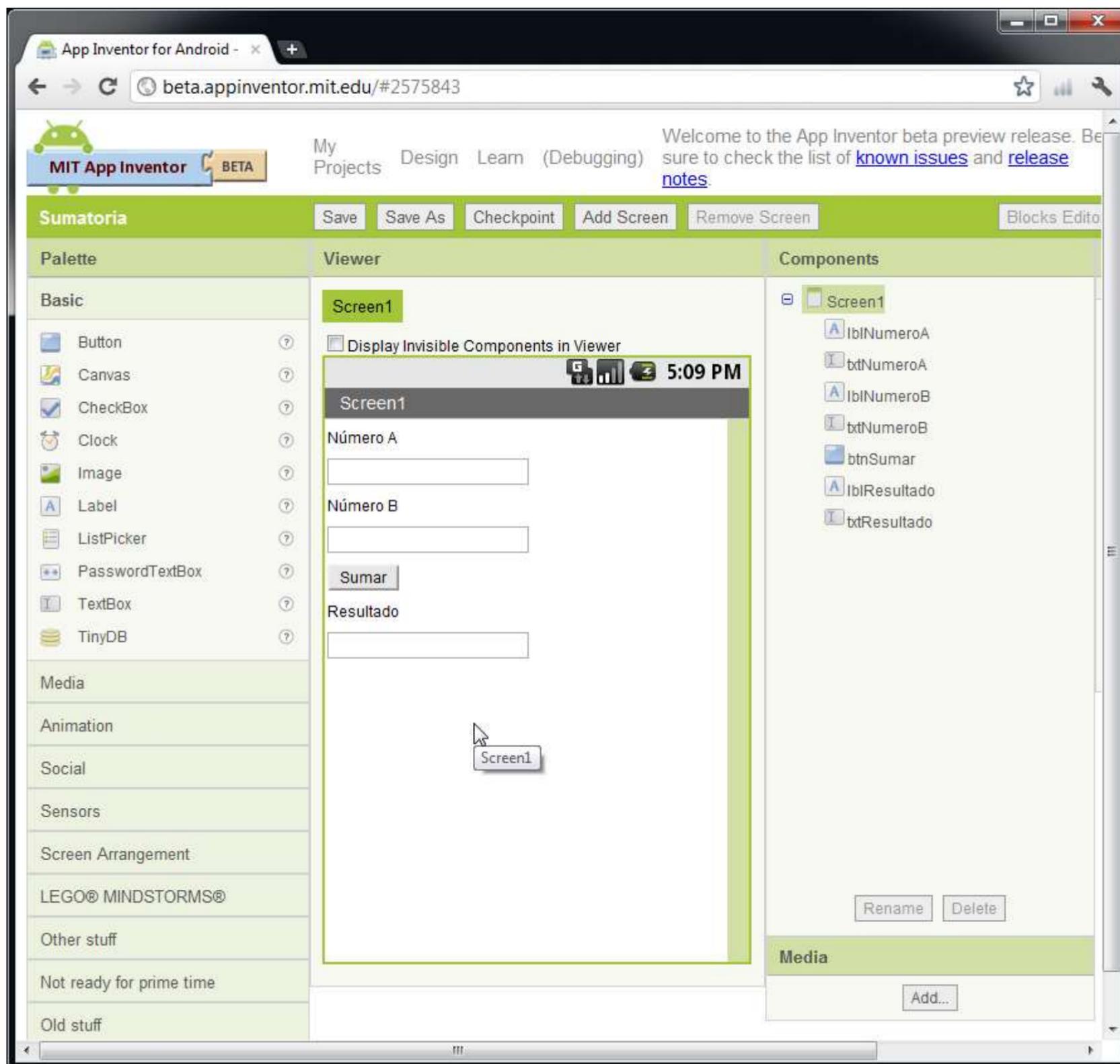


Imagen 61: Diseño de página para sumar dos números y presentar el resultado

En el "Blocks Editor", se puede llamar a piezas de código dando clic con el botón izquierdo del mouse en el fondo, saldría este menú:



Imagen 62: En "Blocks Editor" es dar clic en la zona libre y aparece ese menú

Primero, debe definir las variables, luego es presionar el botón "Defin..." y del menú contextual seleccionar "variable"

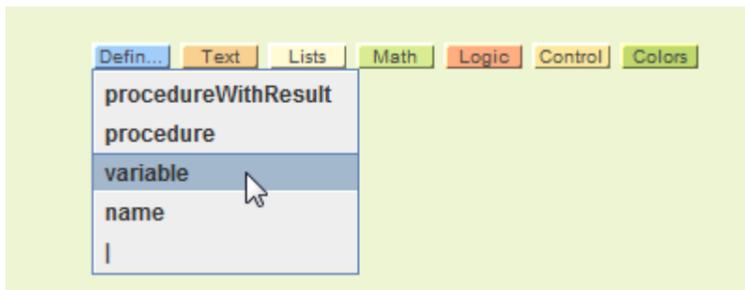


Imagen 63: Menú de "Definition"

Aparece esto



Imagen 64: Definiendo una variable

Le da un nombre a la variable y decide qué tipo de variable es: numérica o texto, para eso vuelva a dar clic (con el botón izquierdo del ratón) en la pantalla, selecciona Math y luego "123"

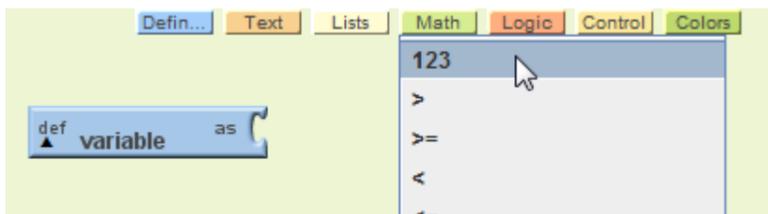


Imagen 65: Seleccionando en "Math" el valor numérico

Y acopla la pieza, luego le puede cambiar el valor

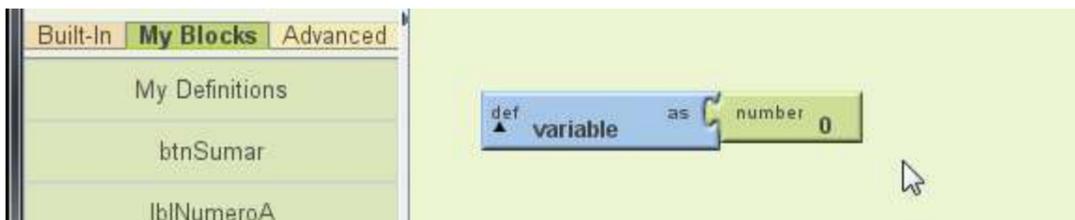


Imagen 66: La variable ha quedado definida como numérica

Por último varía el nombre de la variable

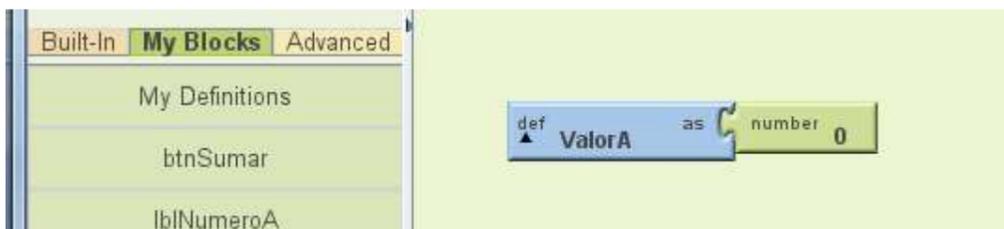


Imagen 67: Puede cambiarle el nombre a la variable

La captura de datos y la suma posterior se programa en el evento del botón



Imagen 68: Preparándose para la captura de datos

Para capturar datos se hace el siguiente procedimiento: Da clic en "My Definitions", allí aparece las variables que ha definido:



Imagen 69: En "My Blocks" y luego en "My Definitions" aparecen las variables definidas

Seleccione "set global Valor A to" y lo acopla



Imagen 70: En el evento del botón, la variable cambiará su valor

Y luego, lee el valor que el usuario ha digitado en "txtNumeroA", observe que en App Inventor, no se requiere un "cast" para pasar de texto a número.

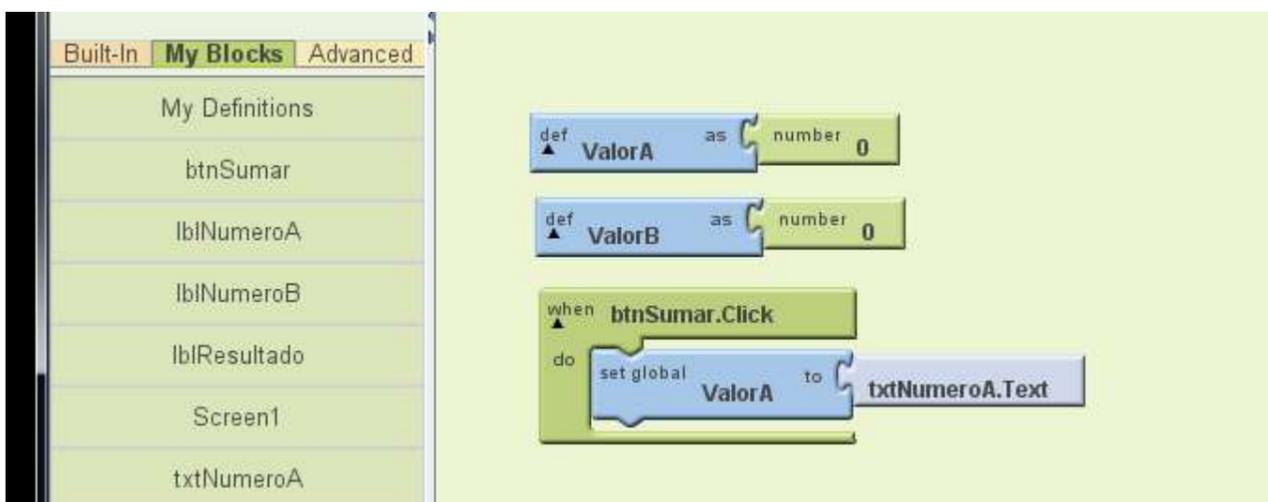


Imagen 71: Captura el valor del TextBox, no es necesaria la conversión

Llega el momento de sumar, se requiere una tercera variable donde se guarda el resultado y una pieza que nos permita sumar, esa pieza se encuentra en "Built-In" en "Math"

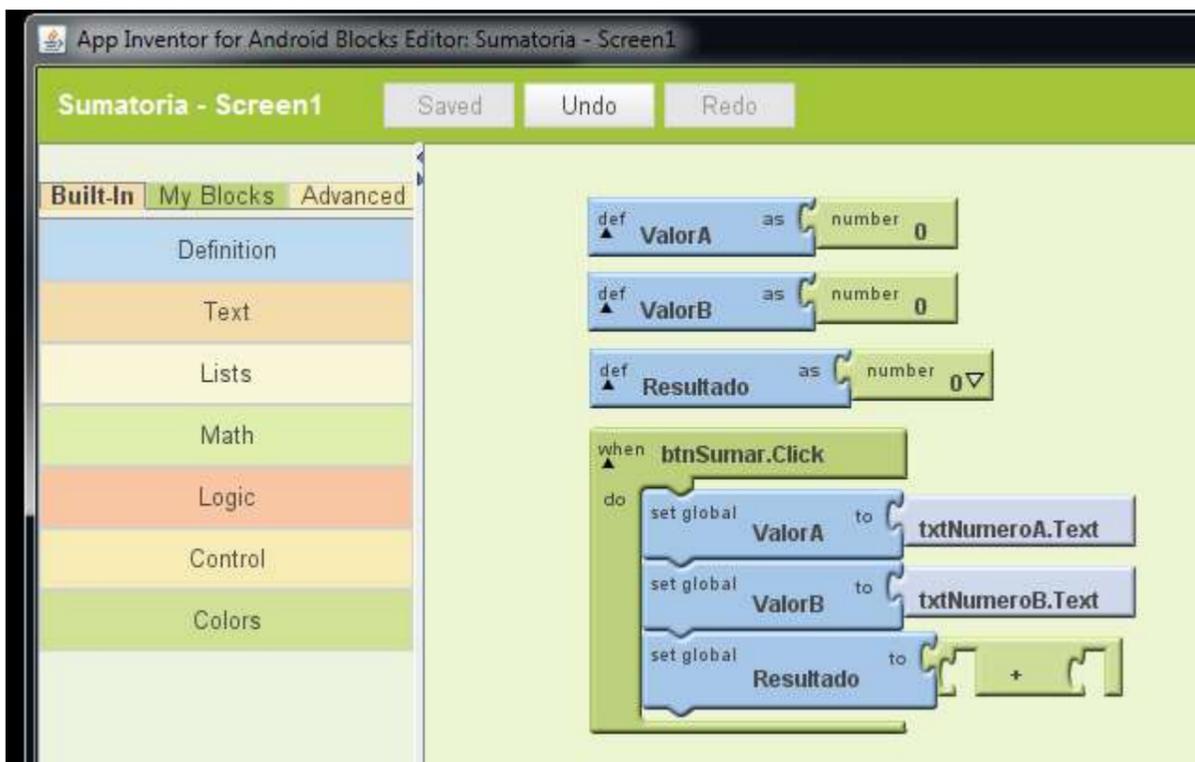


Imagen 72: Preparándose para sumar, se requiere la pieza que suma, se encuentra en "Built-In" y allí "Math"

Ahora es agregar las dos variables para que sumen

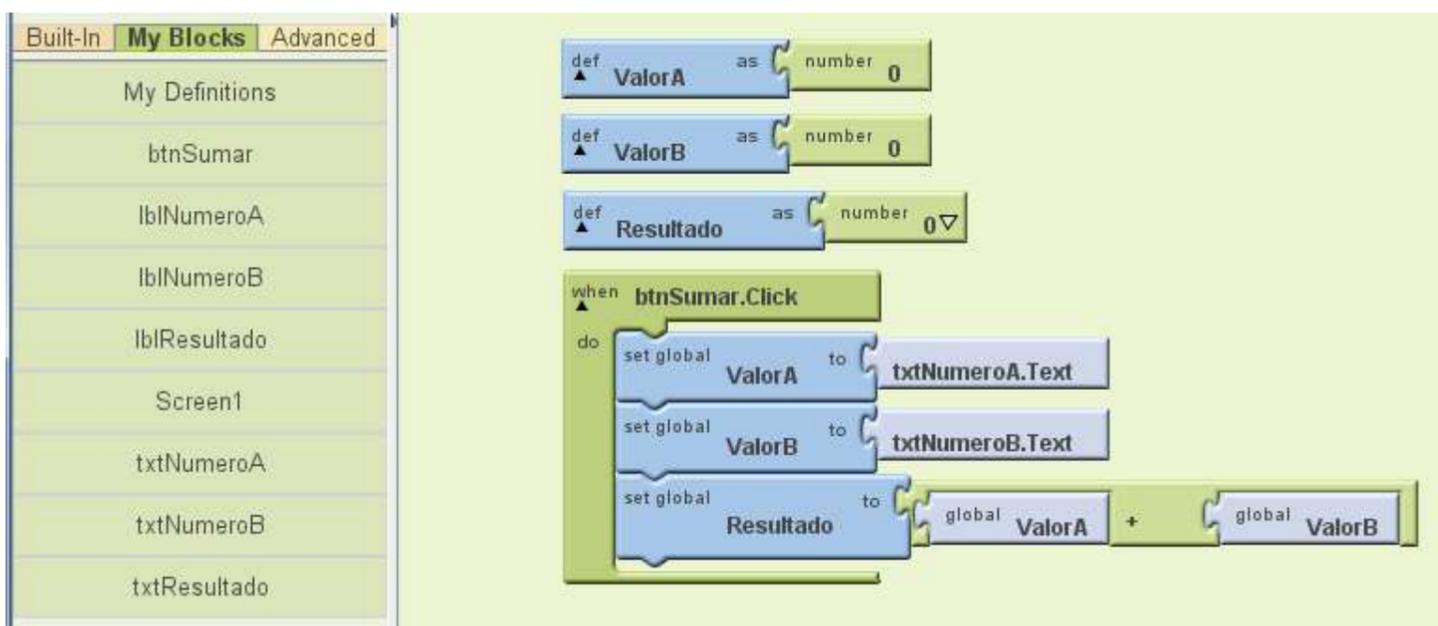


Imagen 73: Operación suma realizada, en "My Definitions" encuentra las piezas "global ValorA" y "global ValorB"

Y presentar el resultado

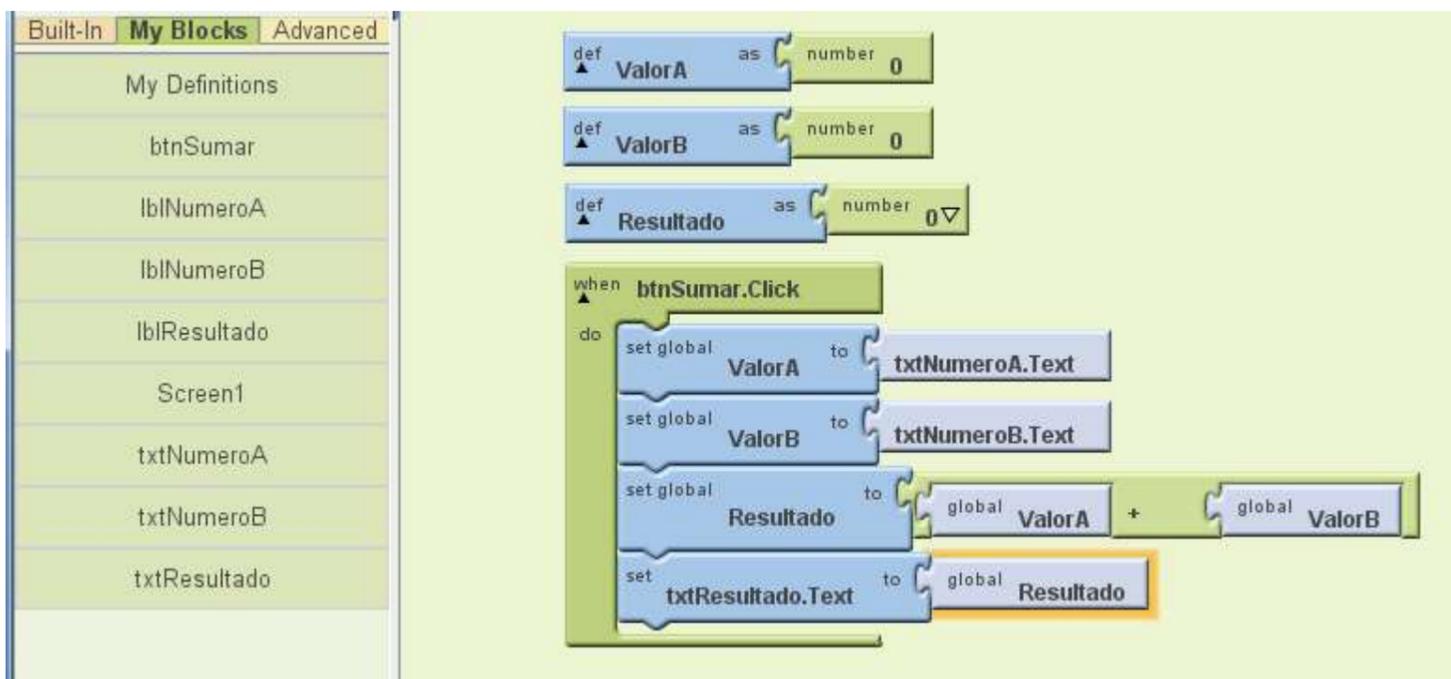


Imagen 74: Mostrando el resultado en "txtResultado" y finalizando el algoritmo.

Ejecute la aplicación en el emulador



Imagen 75: El programa al iniciar el emulador



Imagen 76: Ejecutando la sumatoria

Paso 7. Haciendo operaciones matemáticas más complejas

Tenemos una expresión algebraica como esta:

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

¿Cómo se puede implementar en App Inventor?

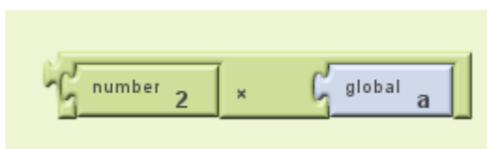
Para iniciar, debemos cambiar para que se vean explícitamente las operaciones matemáticas

$$x = \frac{-b + \sqrt{b * b - 4 * a * c}}{2 * a}$$

Luego lo pasamos a formato horizontal y quitamos ese menos inicial agregando un cero, la expresión sigue siendo equivalente

$$x = (0 - b + \text{raiz}(b * b - 4 * a * c)) / (2 * a)$$

Tenemos la parte $(2 * a)$



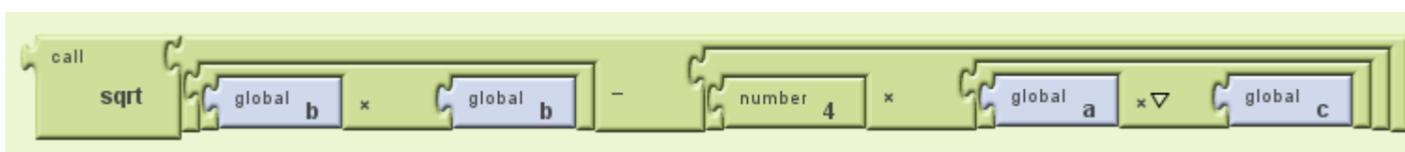
Tenemos la parte $4 * a * c$ (se requieren dos piezas de multiplicación)



Tenemos la parte de $b * b - 4 * a * c$



Tenemos la parte de $\text{raiz}(b * b - 4 * a * c)$



Tenemos la parte de $(0 - b + \text{raiz}(b * b - 4 * a * c))$



Ahora todo junto en un proyecto, observe como se implementa

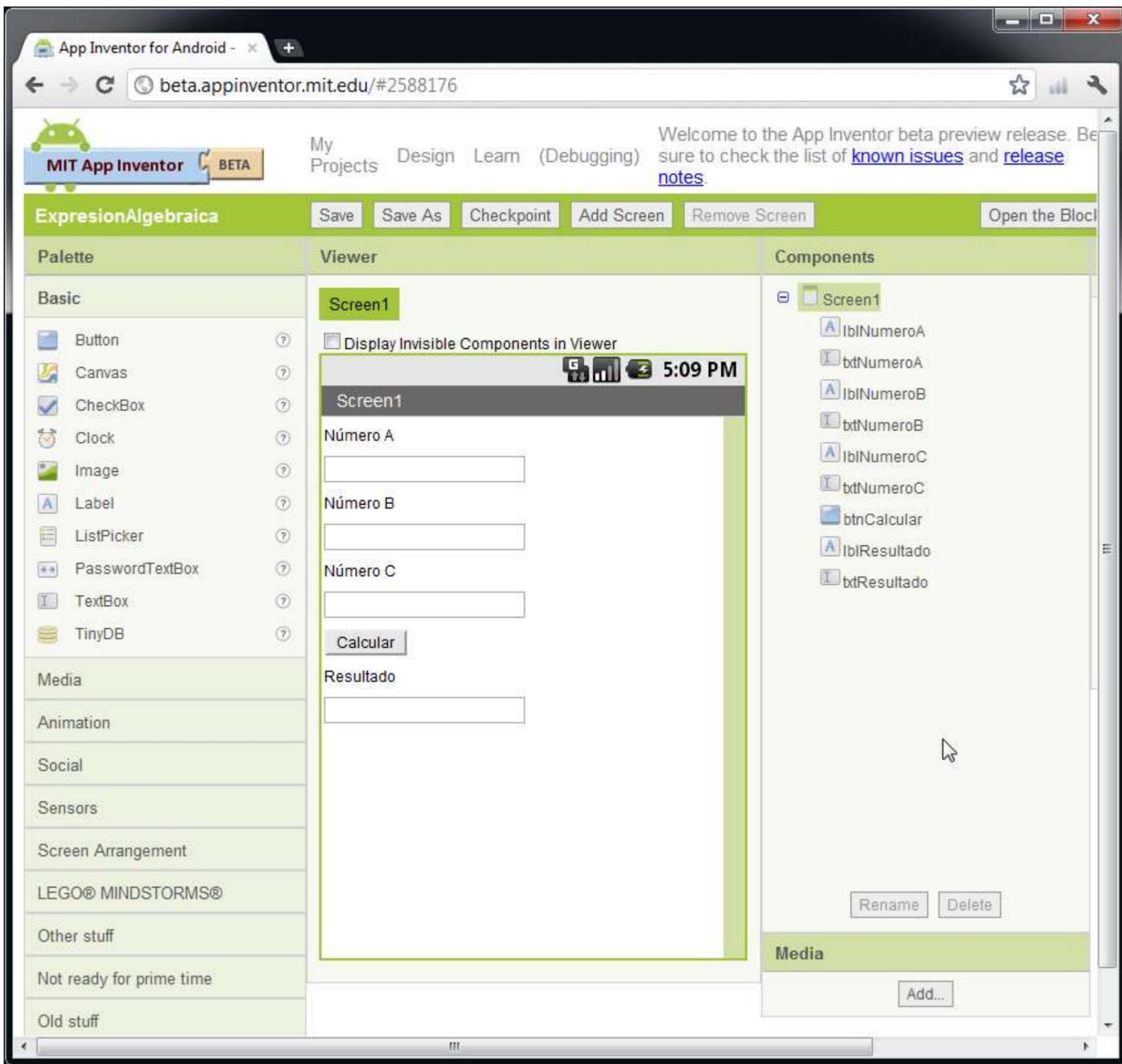
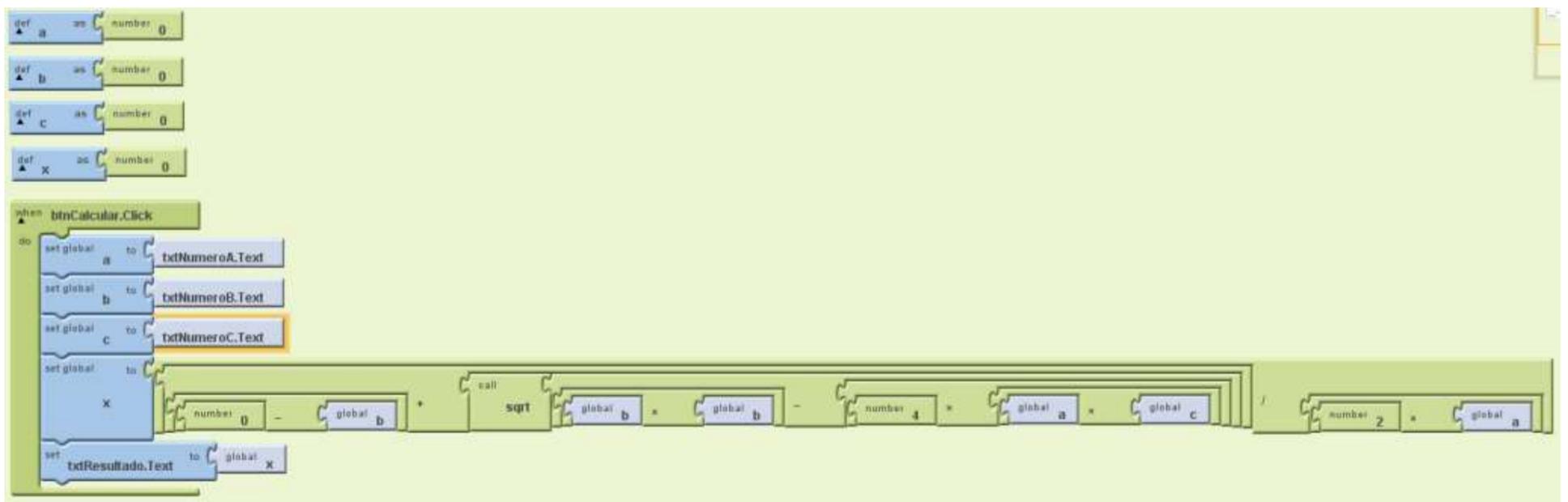


Imagen 77: Diseño de la aplicación



Esta es su ejecución



Imagen 78: Ejecución de la aplicación en el emulador

Y probado con Microsoft Excel

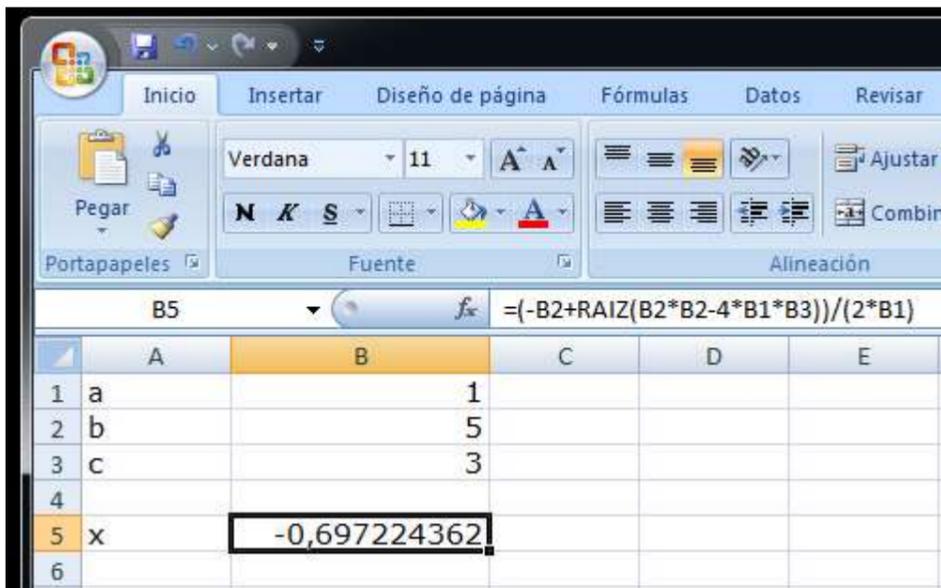


Imagen 79: Prueba de la operación en Excel

Es un poco complejo, ¿alguna forma de hacer más sencilla esa operación matemática? Si, una forma es crear variables e ir dividiendo la expresión en partes más pequeñas.

Paso 8. El SI condicional

Un algoritmo sencillo de si condicional

Si Edad \geq 18 entonces
 Imprima "Es mayor de edad"
De lo contrario
 Imprima "Es menor de edad"

Implementándolo en App Inventor:

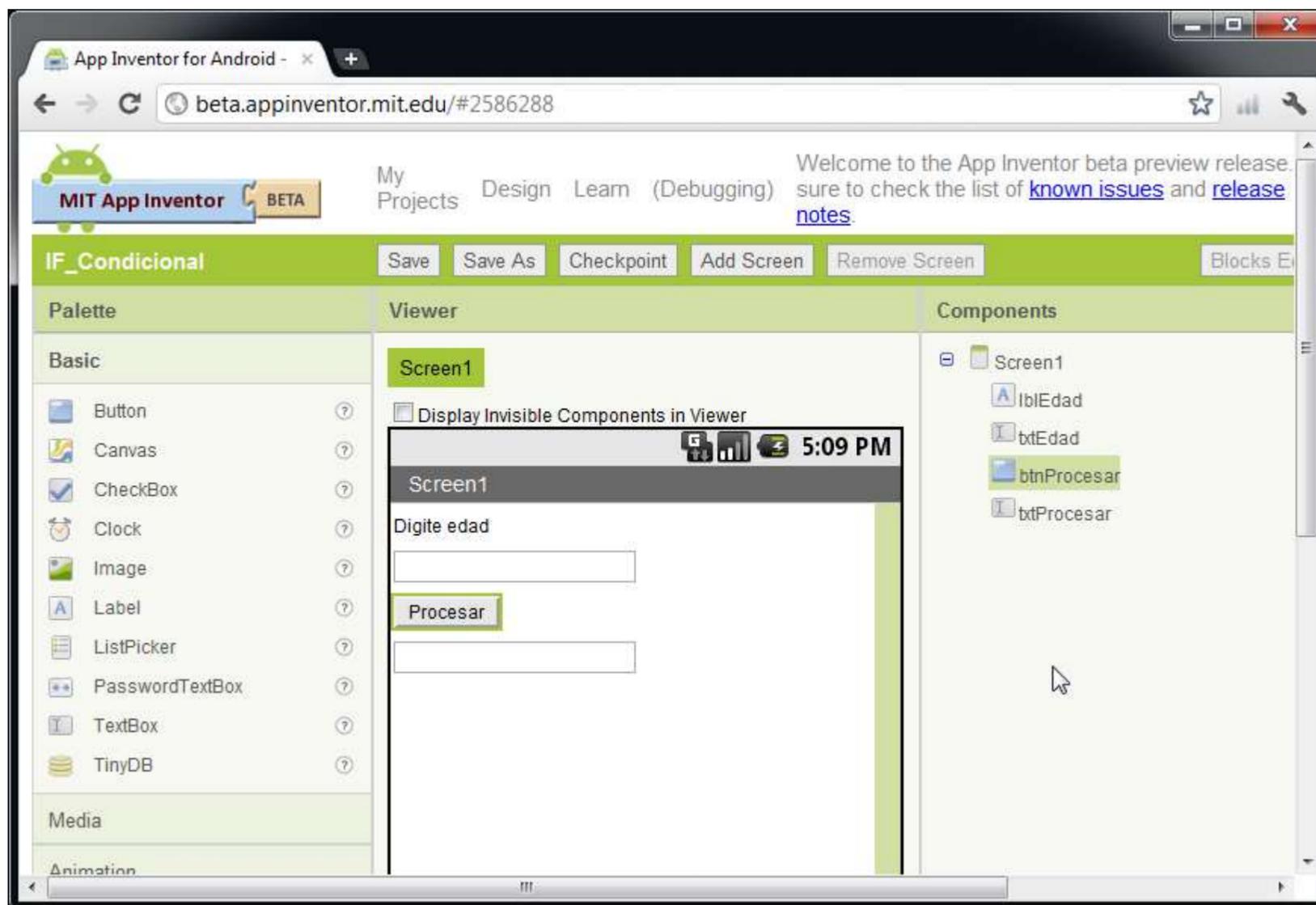


Imagen 80: Diseño de pantalla que pide una edad en la primera caja de texto y luego muestra si es menor o mayor de edad

Este es el algoritmo en el formato de App Inventor

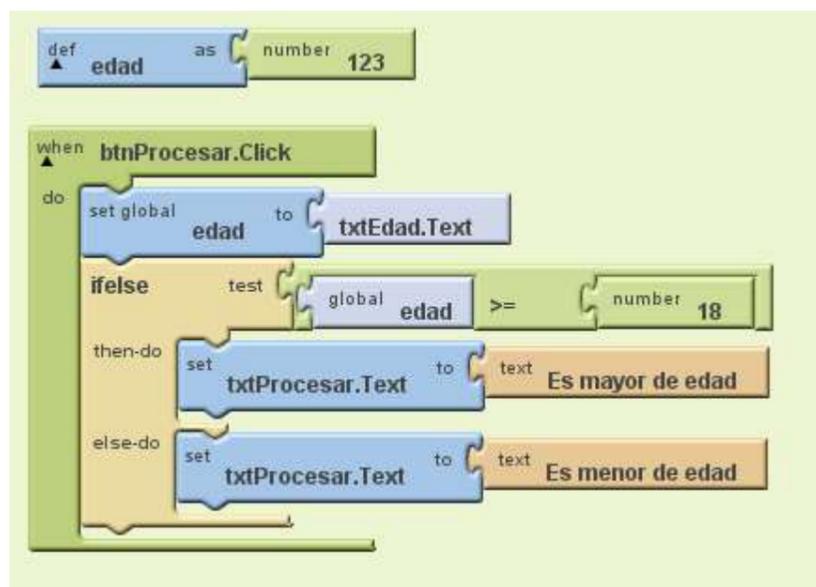


Imagen 81: Algoritmo en App Inventor que implementa el si condicional

La pieza "ifelse" se encuentra en "Built-In" y "Control"

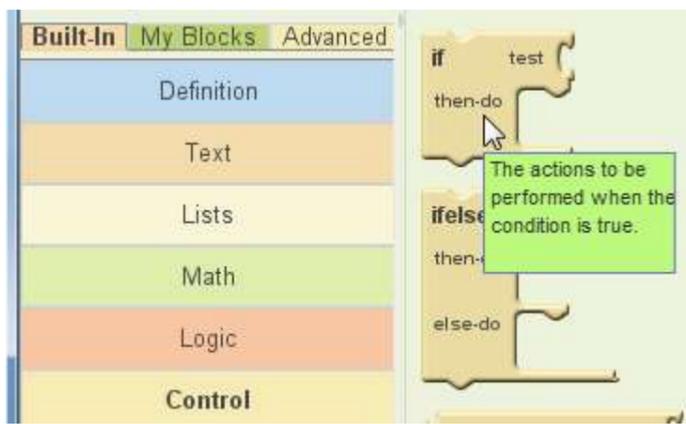


Imagen 82: La pieza del si condicional se encuentra en Built-In

Las piezas para comparar se encuentran en "Built-In" y "Math"

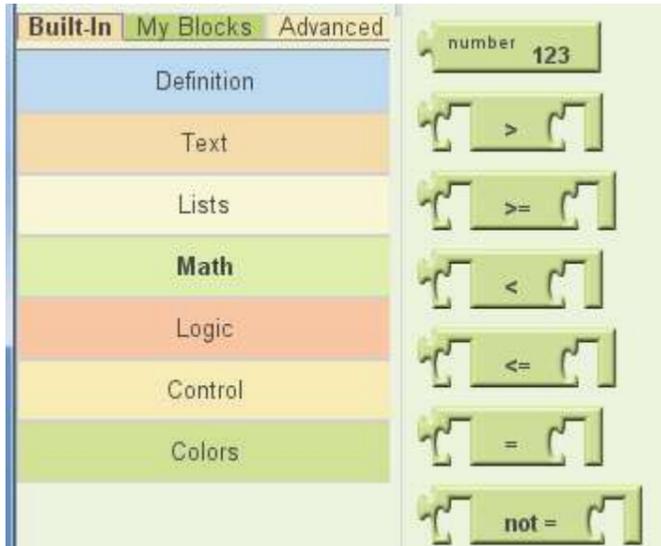


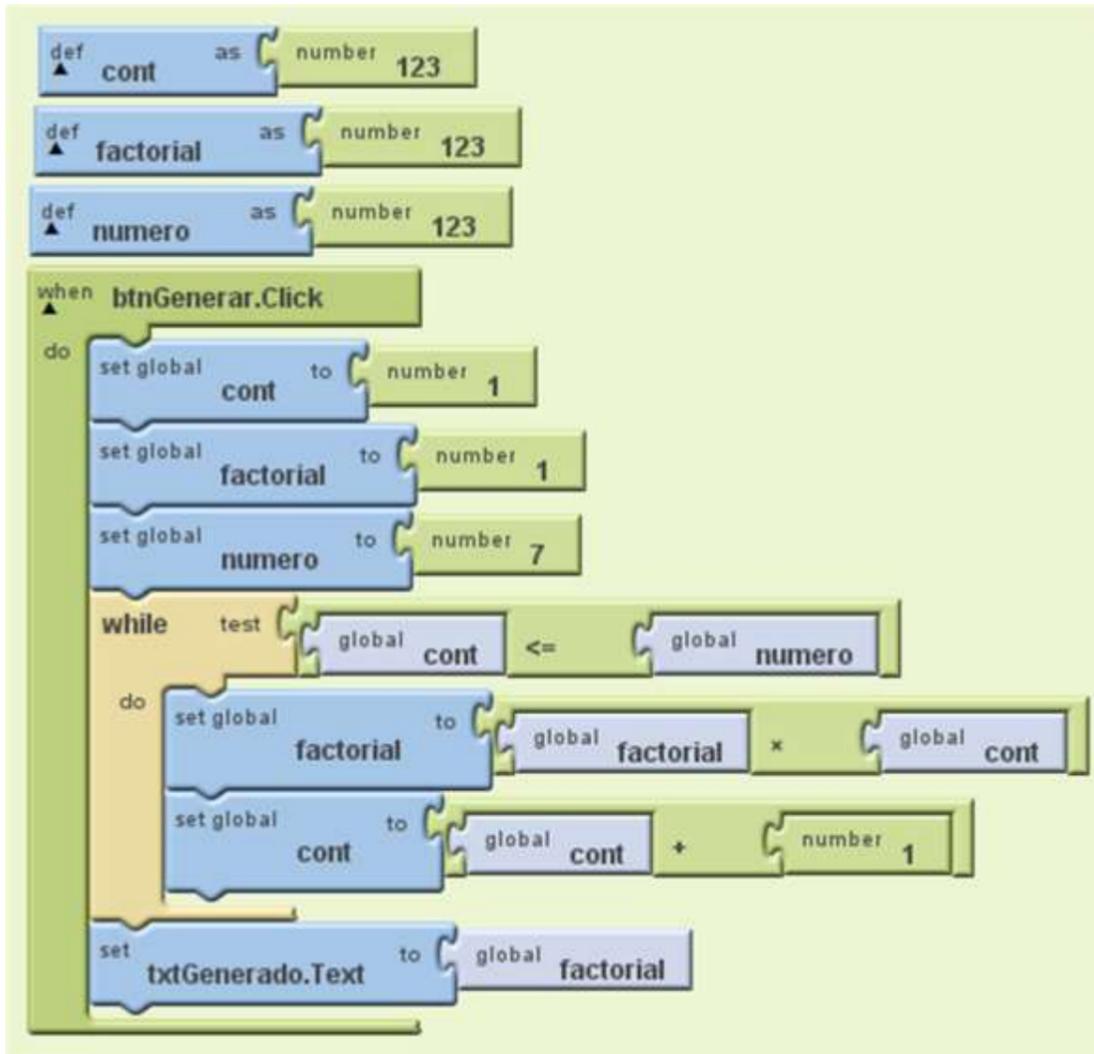
Imagen 83: La pieza para hacer comparativos dentro del si condicional se encuentra en Built-In y luego Math

Paso 9. Ciclos

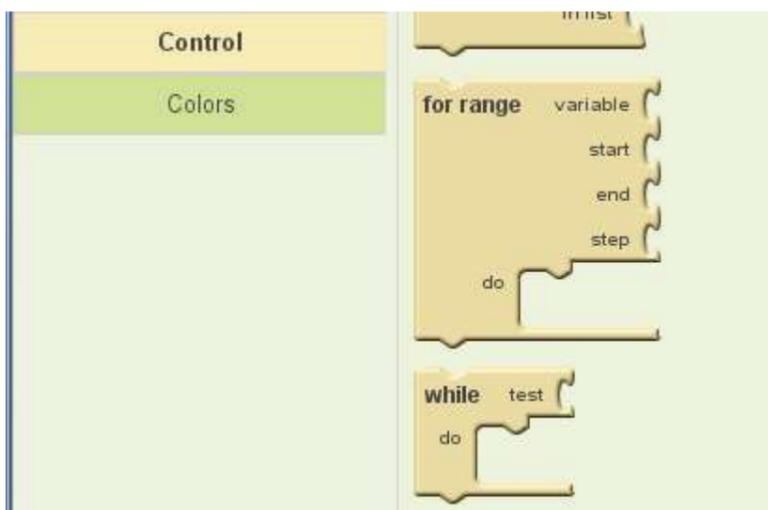
Un algoritmo que calcula el factorial de un número es así:

```
cont=1
factorial = 1
numero = 7
Mientras cont <= numero hacer
    factorial = factorial * cont
    cont = cont + 1
Fin Mientras
Imprima factorial
```

En App Inventor se implementa así:



La pieza "while" se encuentra en "Built-In" y allí en "Control":



Paso 10. Operaciones con cadenas (Básico)

A. Longitud de cadenas

Determinar el tamaño de una cadena incluyendo los espacios

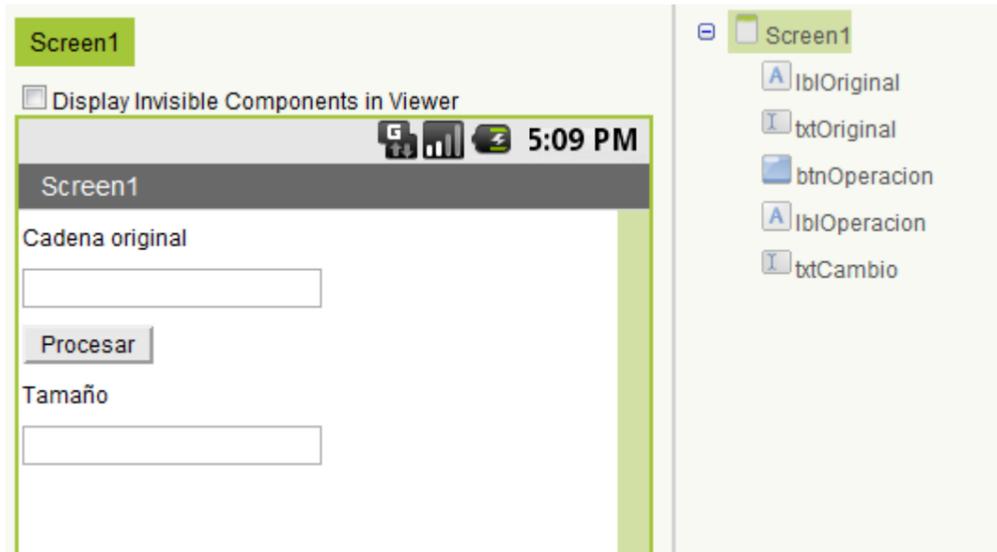


Imagen 84: Diseño de la pantalla para mostrar la funcionalidad tamaño de texto

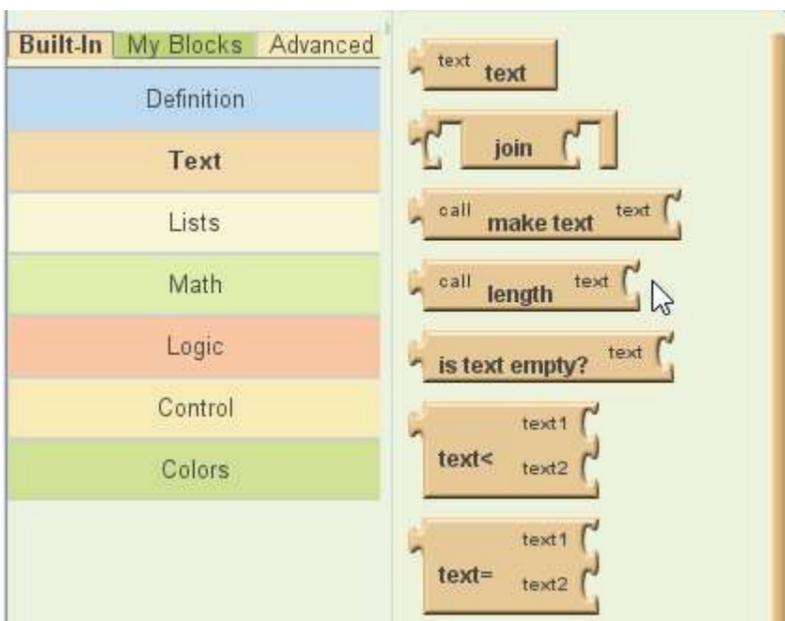


Imagen 85: Se utiliza el bloque "call length text"

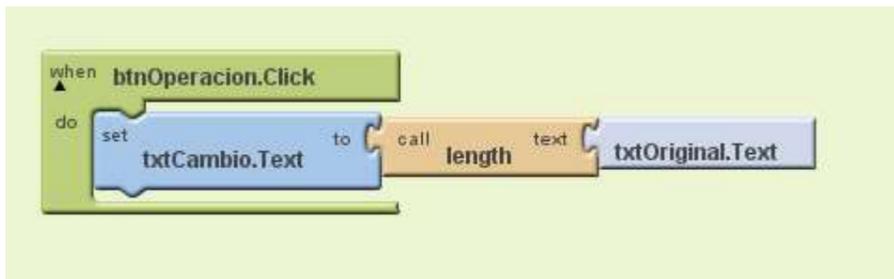


Imagen 86: Uso de la pieza "call length text"



Imagen 87: Ejecución del programa

B. Chequear si una cadena está vacía

Chequear si una cadena no tiene ni un solo caracter

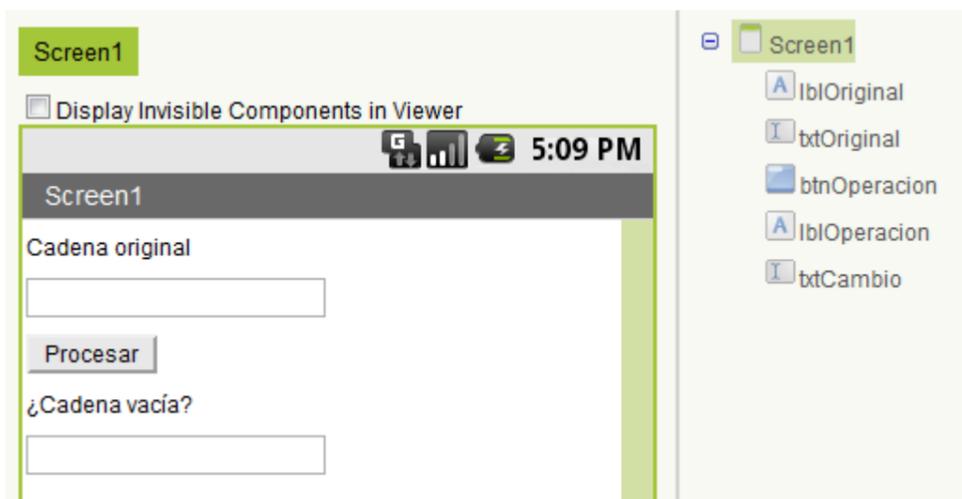


Imagen 88: Diseño de la pantalla para mostrar la funcionalidad chequear cadena vacía

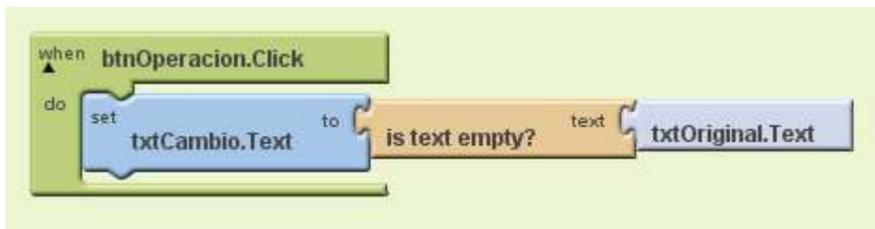


Imagen 89: Algoritmo para probar si la cadena es vacía



Imagen 90: Observe que se retorna un valor booleano

C. Concatenación de cadenas

Suma varias cadenas y muestra el resultado

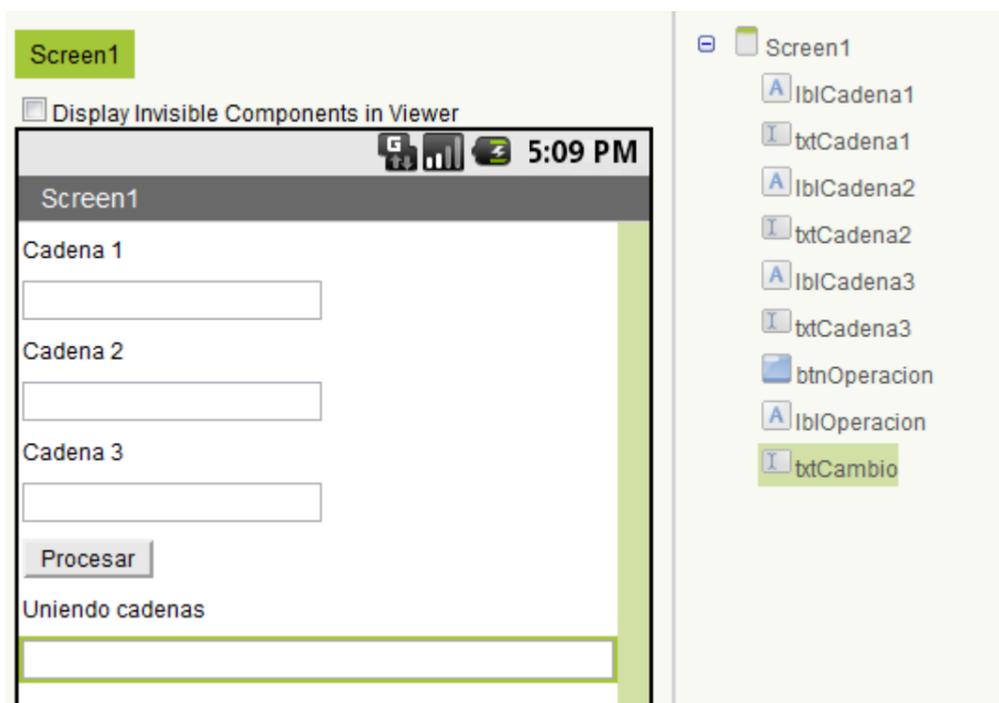


Imagen 91: Diseño para probar la funcionalidad de concatenar cadenas

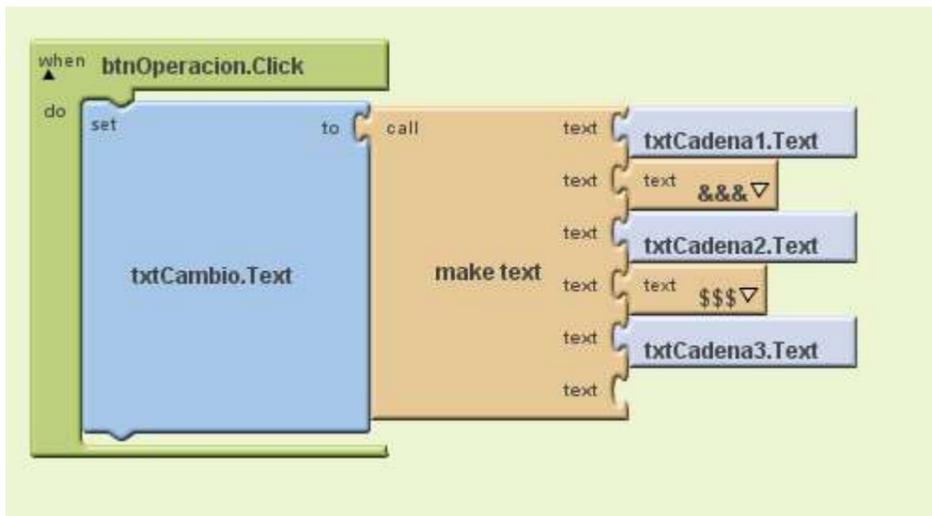


Imagen 92: Algoritmo para conectar cadenas



Imagen 93: Ejecución del algoritmo de conectar cadenas

D. Comparación de cadenas

Compara los caracteres de cada cadena y retorna verdadero(true) si son iguales, falso(false) en caso contrario

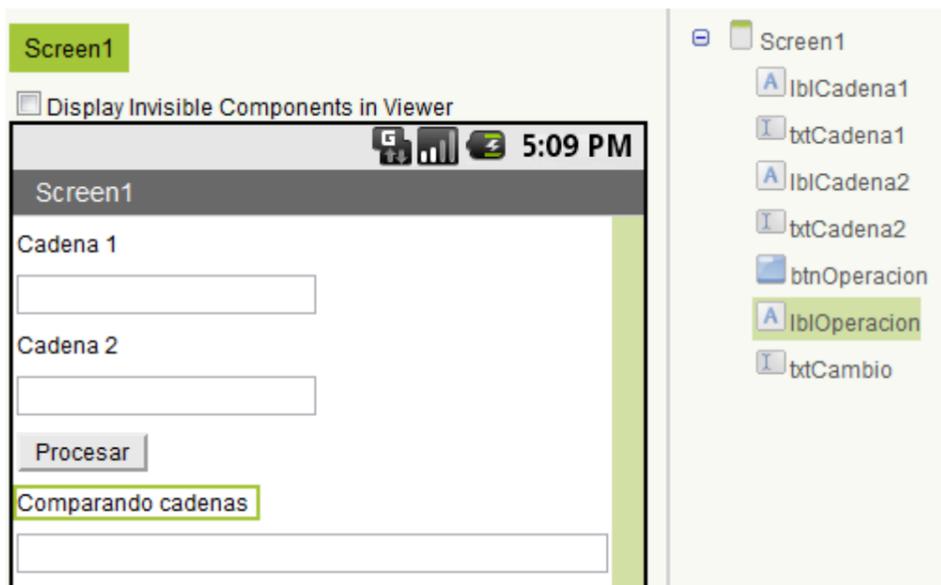


Imagen 94: Diseño de pantalla para probar la funcionalidad de comparación de cadenas

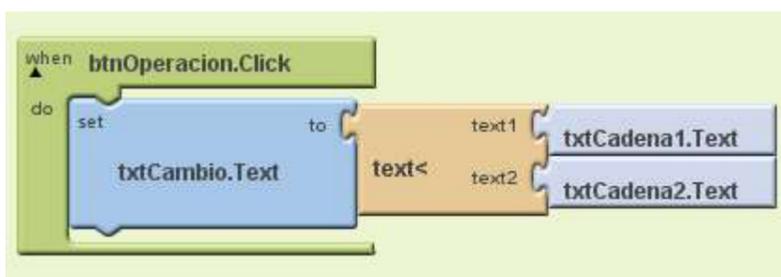


Imagen 95: Algoritmo de comparación de cadenas



Imagen 96: Ejemplo de ejecución de comparación de cadenas. Resultado booleano.

E. Quitar espacios al inicio y al final

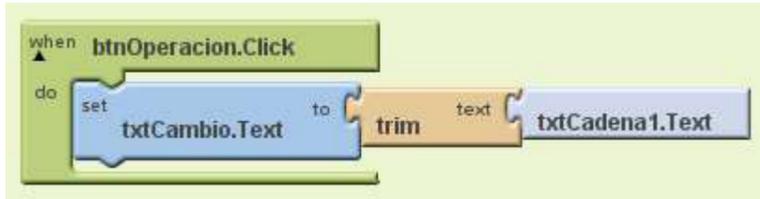


Imagen 97: El bloque "trim" quita los espacios al inicio y final

F. Convertir a mayúsculas

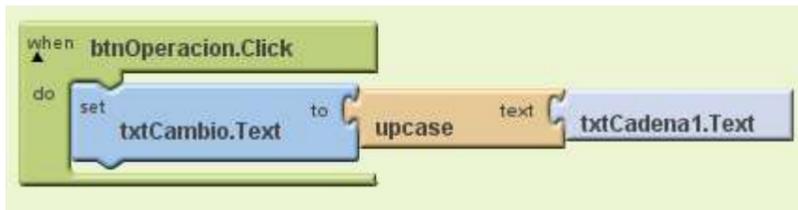


Imagen 98: El bloque "upcase" convierte a mayúsculas

G. Convertir a minúsculas

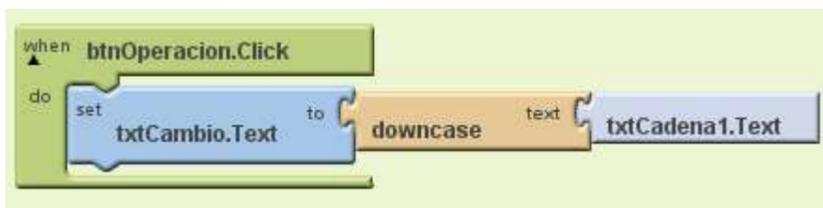


Imagen 99: El bloque "downcase" convierte a minúsculas

H. Señala en qué posición se encuentra una determinada subcadena en la cadena

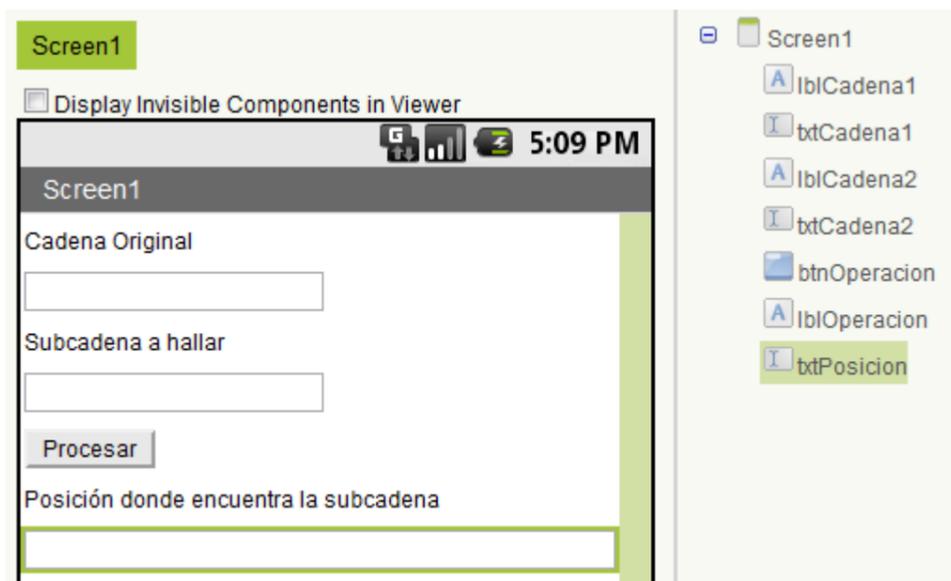


Imagen 100: Diseño GUI de la pantalla para probar la búsqueda de subcadenas

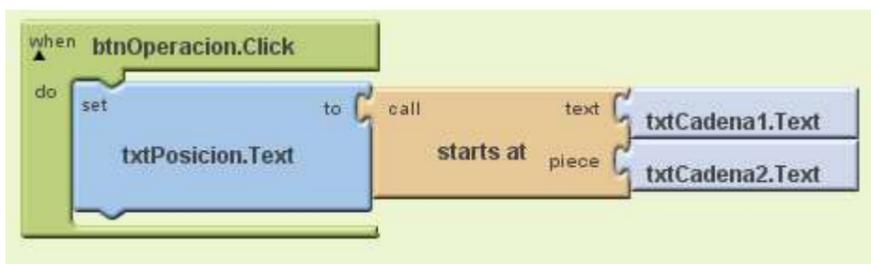


Imagen 101: El bloque "start at" busca una cadena dentro de otra y retorna la posición de encuentro.

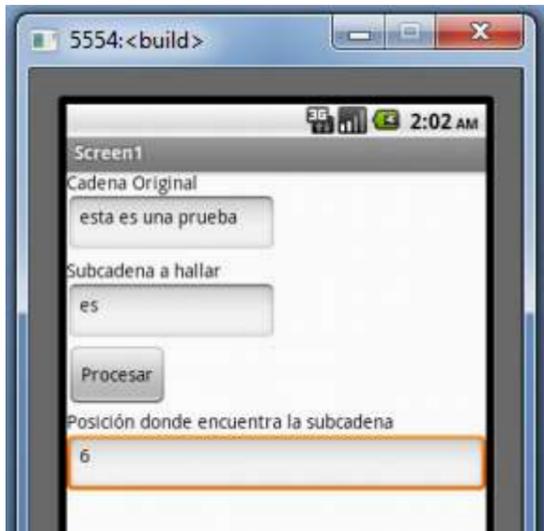


Imagen 102: Retorno de la posición de la subcadena en una cadena



Imagen 103: Retorno de la posición de la subcadena en una cadena

I. Retorna verdadero si encuentra la subcadena dentro de la cadena

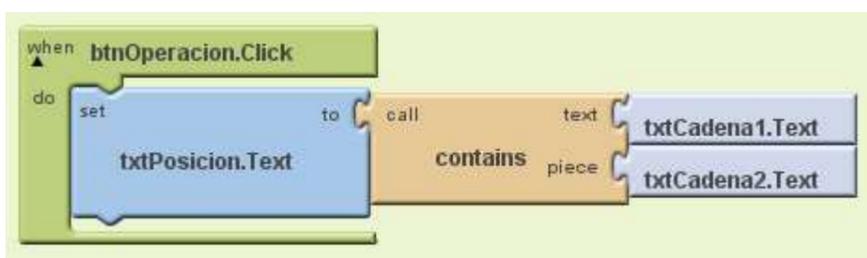


Imagen 104: El bloque "contains" retorna verdadero si encuentra una subcadena dentro de una cadena





J. Extraer una subcadena

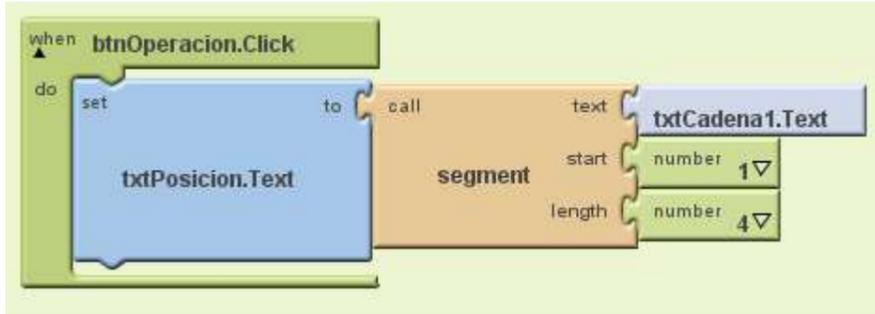
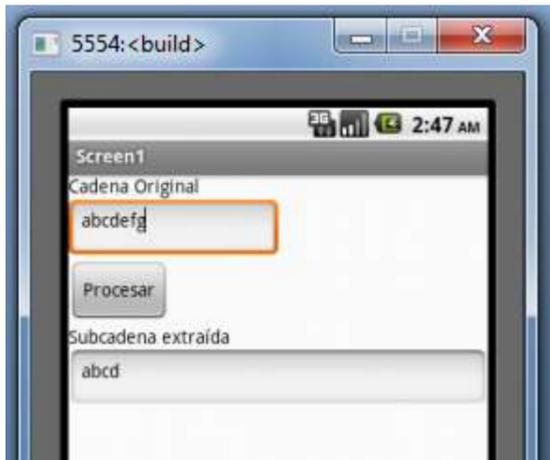
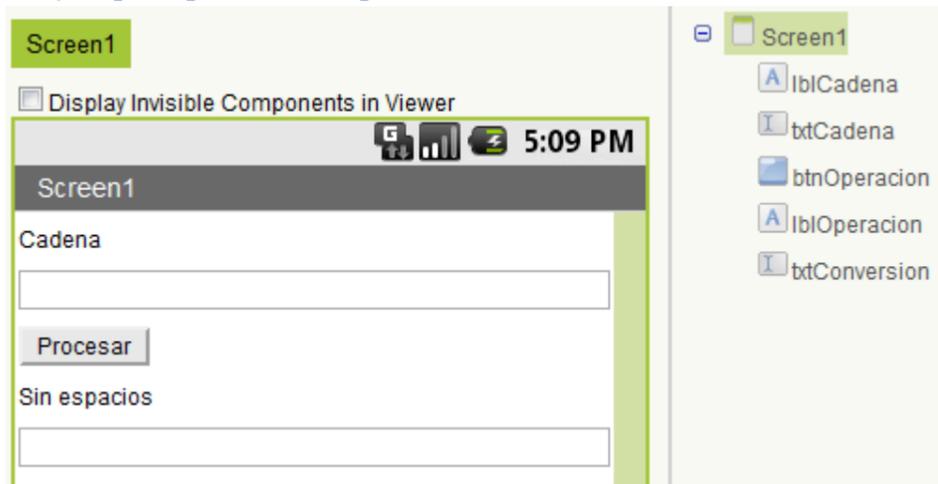


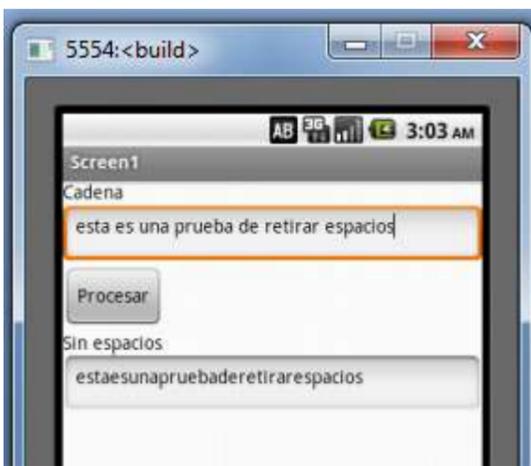
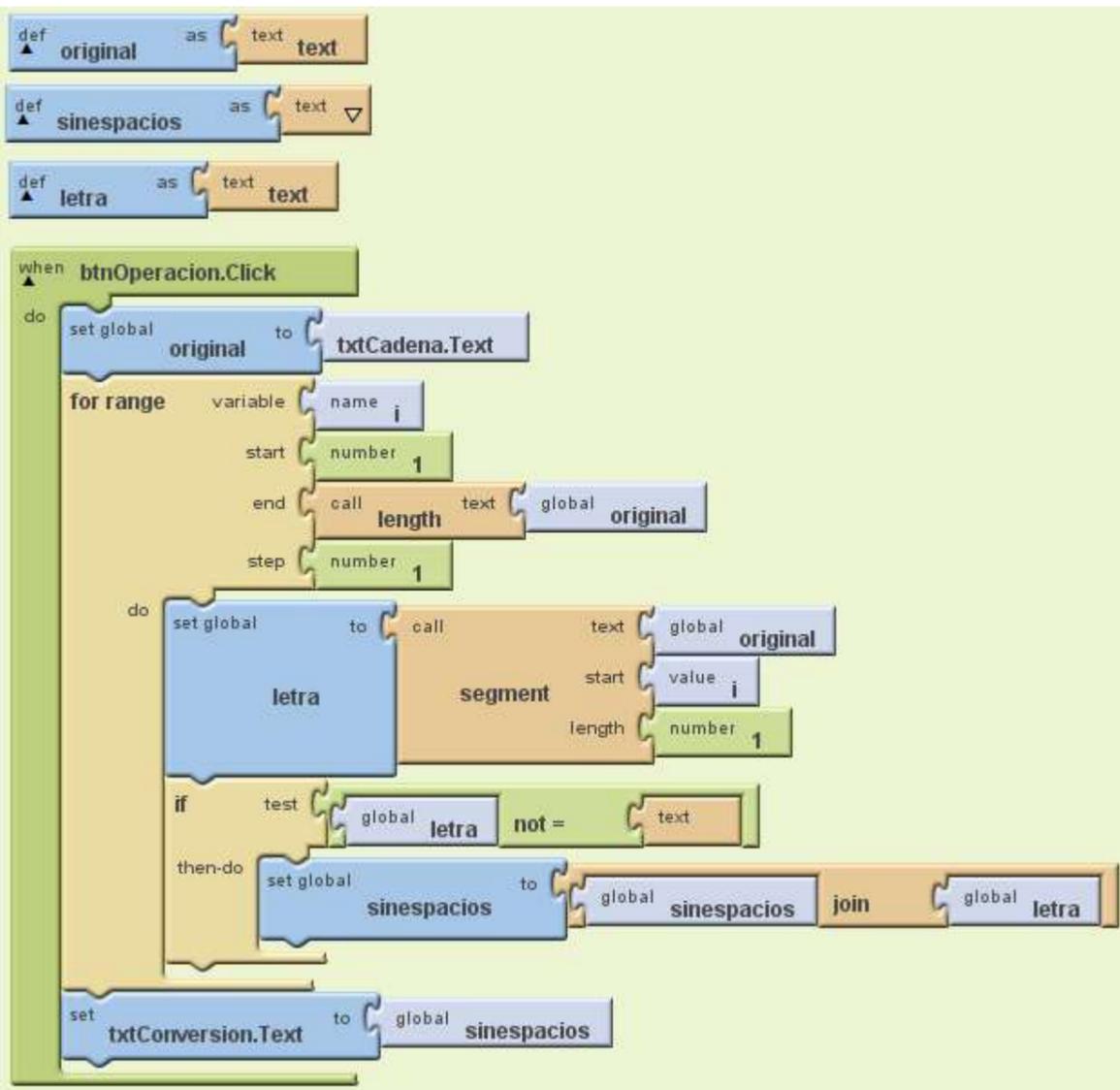
Imagen 105: Extrae una subcadena de la cadena original desde la primera letra y toma cuatro(4) caracteres



K. Ejemplo: quitar los espacios de una cadena



iOJO! Tenga cuidado con el carácter espacio, es una pieza "text" pero tiene un espacio, el problema es que no se ve y puede confundirse con la pieza "text" vacía (ambas lucen igual).



L. Reemplazar una subcadena por otra

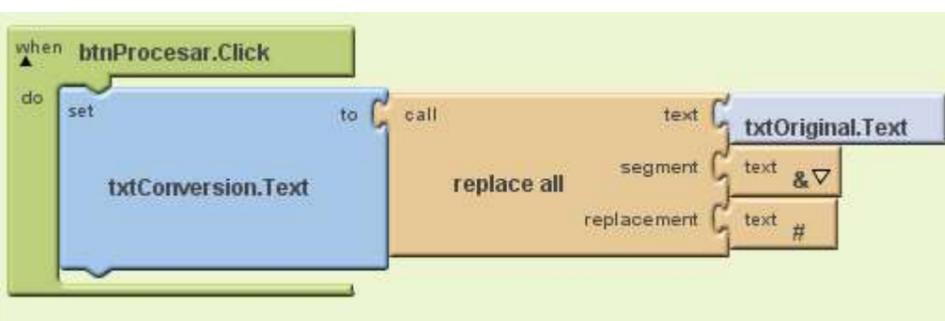
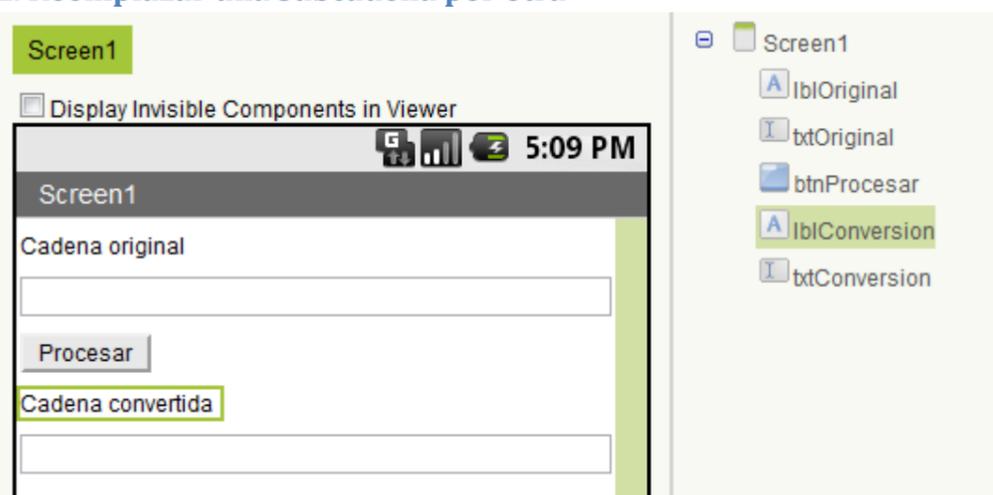


Imagen 106: Reemplaza el carácter & por el carácter #



Paso 11. Operaciones con listas

Una lista es una colección de ítems

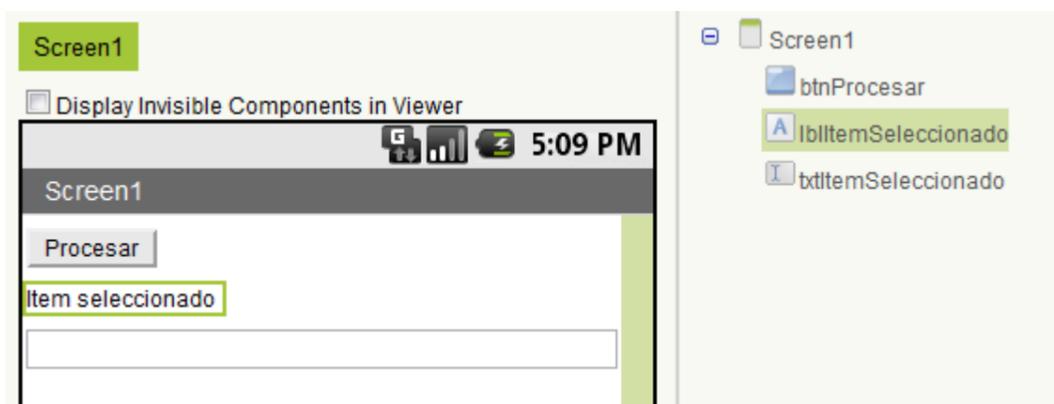


Imagen 107: Diseño de pantalla para el trabajo con listas

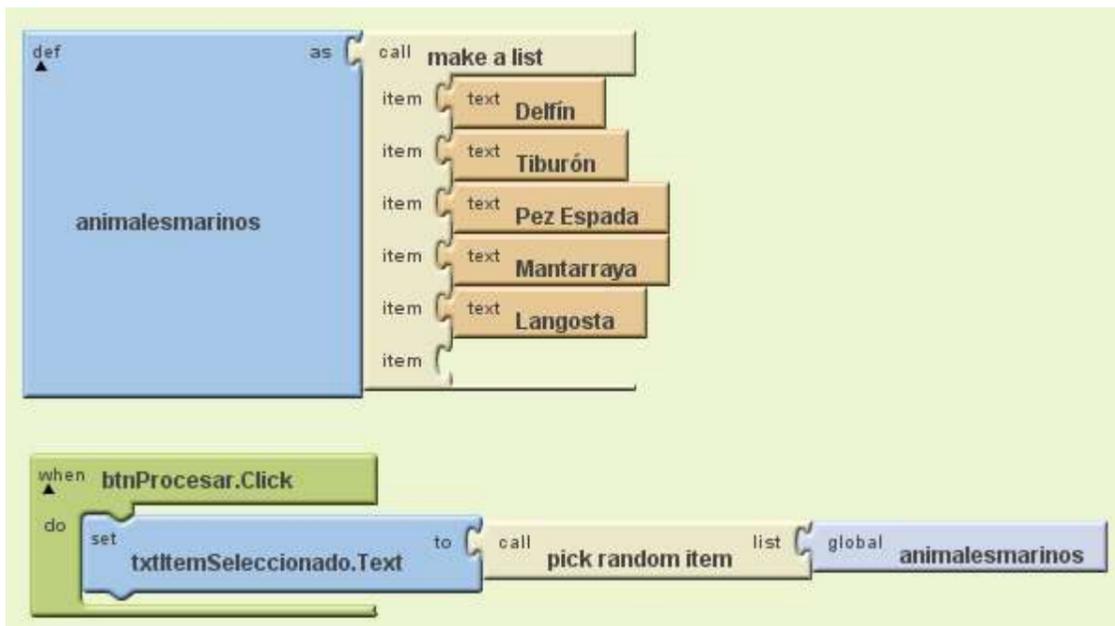


Imagen 108: Primero se define una lista y cuando se oprime el botón se selecciona al azar un ítem de esa lista



Imagen 109: Ejecución del programa. Se selecciona al azar un ítem de la lista.



Imagen 110: Ejecución del programa. Se selecciona otro ítem de la lista.

A. Longitud de una lista

Muestra cuantos ítems tiene una lista

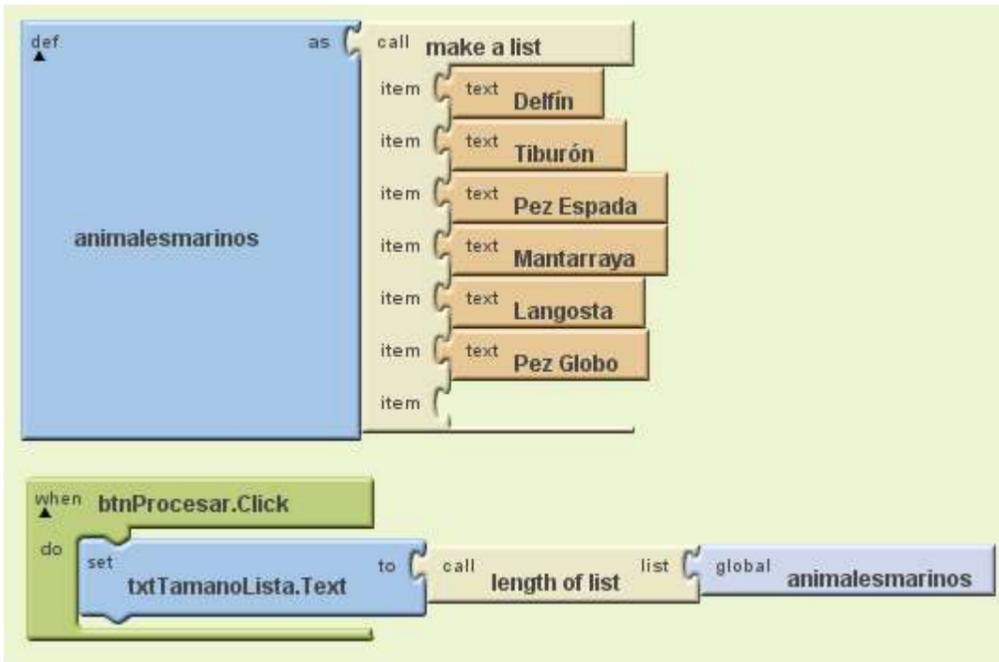


Imagen 111: Tamaño de la lista



Imagen 112: Ejecución tamaño de lista

B. Seleccionar un ítem de una lista

The diagram shows Scratch code blocks. A blue block defines a list named 'animalesmarinos' with six items: Delfin, Tiburón, Pez Espada, Mantarraya, Langosta, and Pez Globo. A green flag block is followed by a 'when btnProcesar.Click' block. Inside the 'do' loop, there are three blocks: 'set global seleccionar to txtIndice.Text', 'set txtTamanoLista.Text to', and 'call select list item'. The 'select list item' block has three inputs: 'list' pointing to the 'animalesmarinos' global list, 'index' pointing to the 'seleccionar' global variable, and 'global' pointing to the 'animalesmarinos' global variable. Below the code, a screenshot of the application shows the 'Procesar' button and a text field 'Escriba un índice de lista' containing the number '3'. Below it, a text field 'Ítem seleccionado' contains 'Pez Espada'. A virtual keyboard is visible at the bottom of the screen.

C. Mostrar una lista en un "textbox multiline"

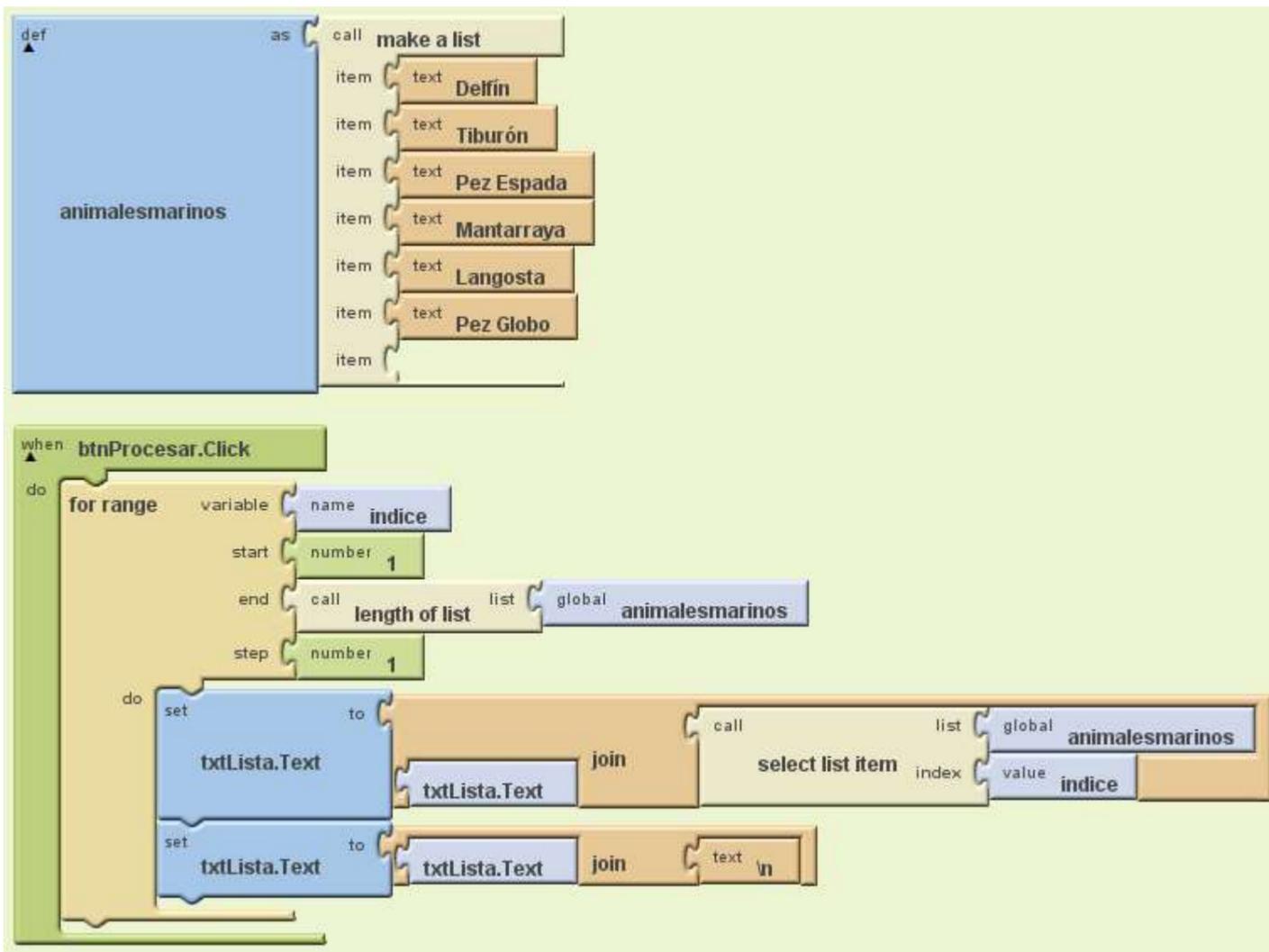


Imagen 113: Volcar una lista a un "textbox multiline"



Imagen 114: Muestra el listado

D. Reemplazar un ítem de una lista

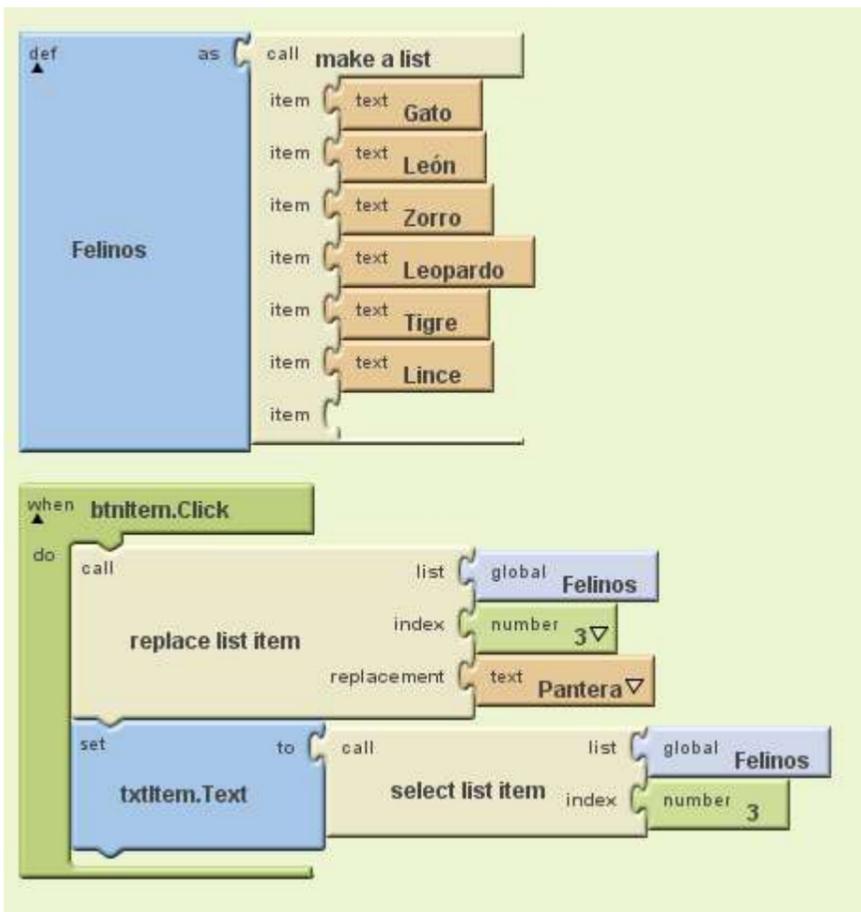


Imagen 115: Reemplaza el tercer ítem de la lista definida

E. Verifica si hay un ítem en una lista

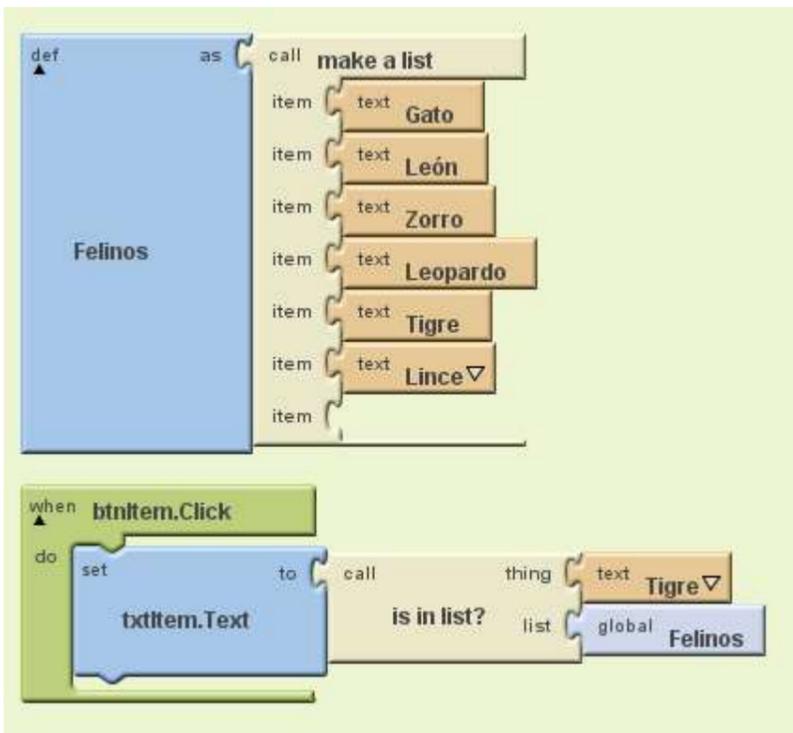


Imagen 116: Chequea si el ítem "Tigre" está en la lista retornando "true" o "false". Se considera mayúscula y minúscula

F. Retorna en qué posición se encuentra un determinado ítem en la lista

Si el ítem no se encuentra en la lista retorna cero (0).

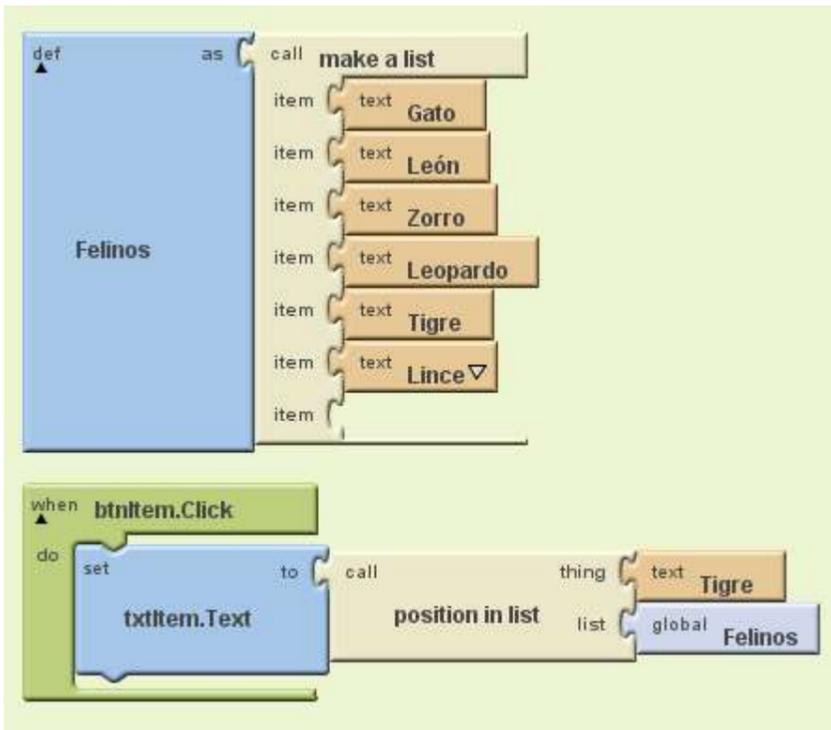


Imagen 117: Retorna que "Tigre" está en la posición 5.

G. Retorna si la lista está llena o vacía

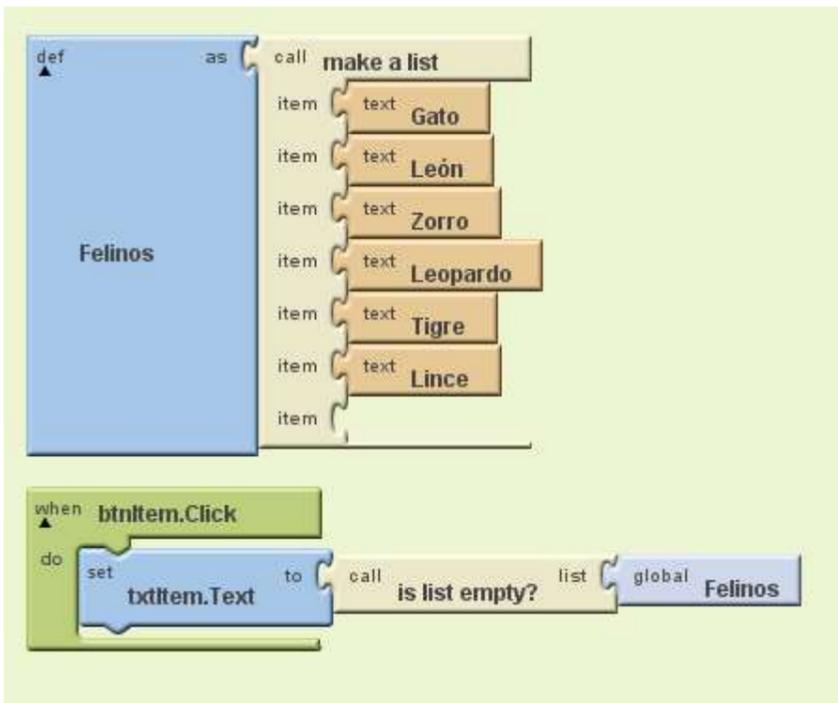


Imagen 118: Retorna "false" porque la lista tiene ítems.

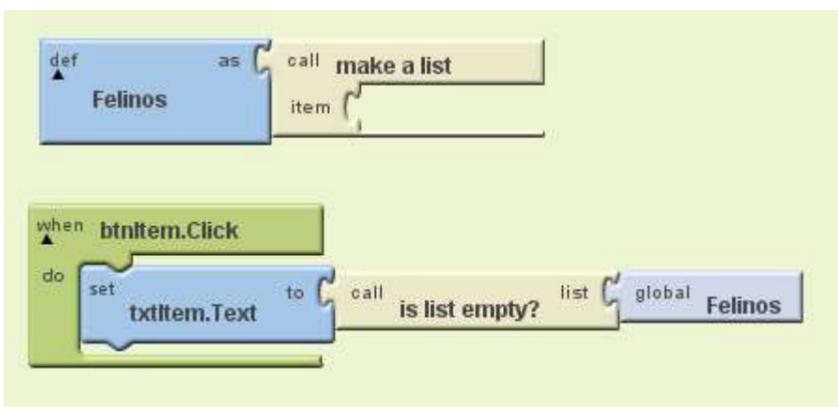


Imagen 119: Retorna "true" porque la lista está vacía.

H. Inserta un ítem en determinada posición de la lista

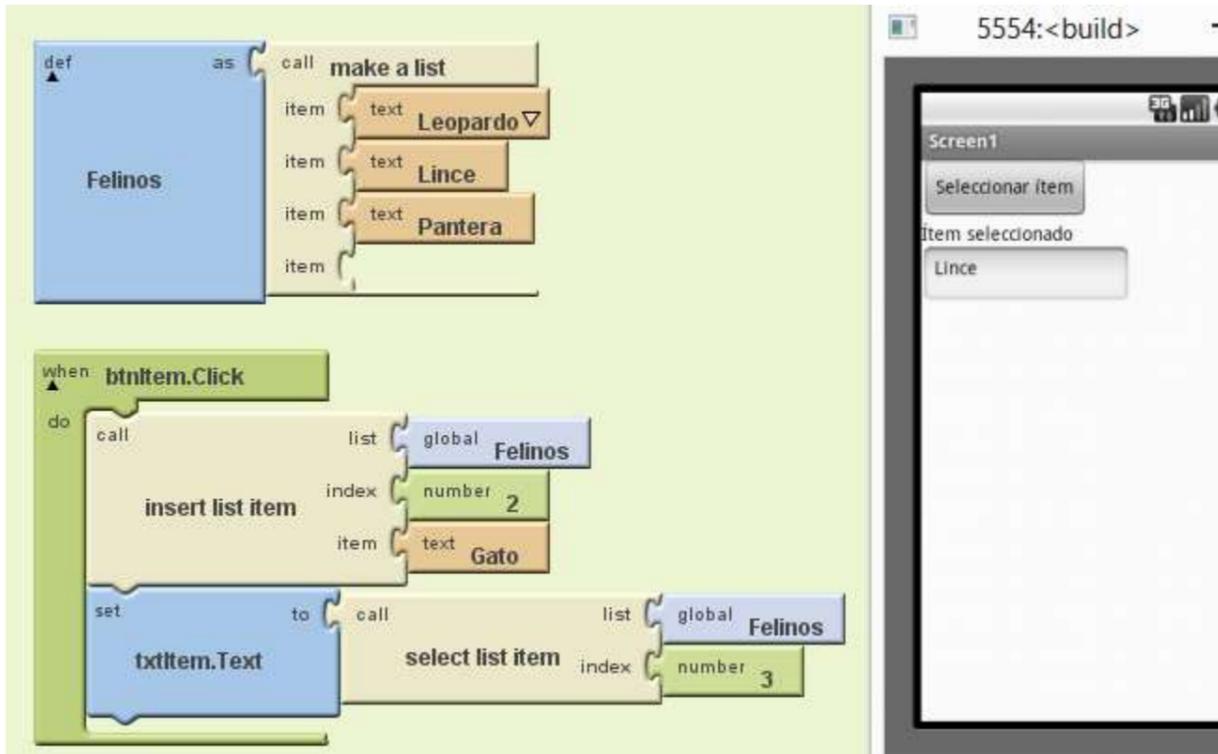


Imagen 120: En la posición 2 se inserta "Gato" corriendo a "Lince" a la tercera posición

I. Quita un elemento de la lista

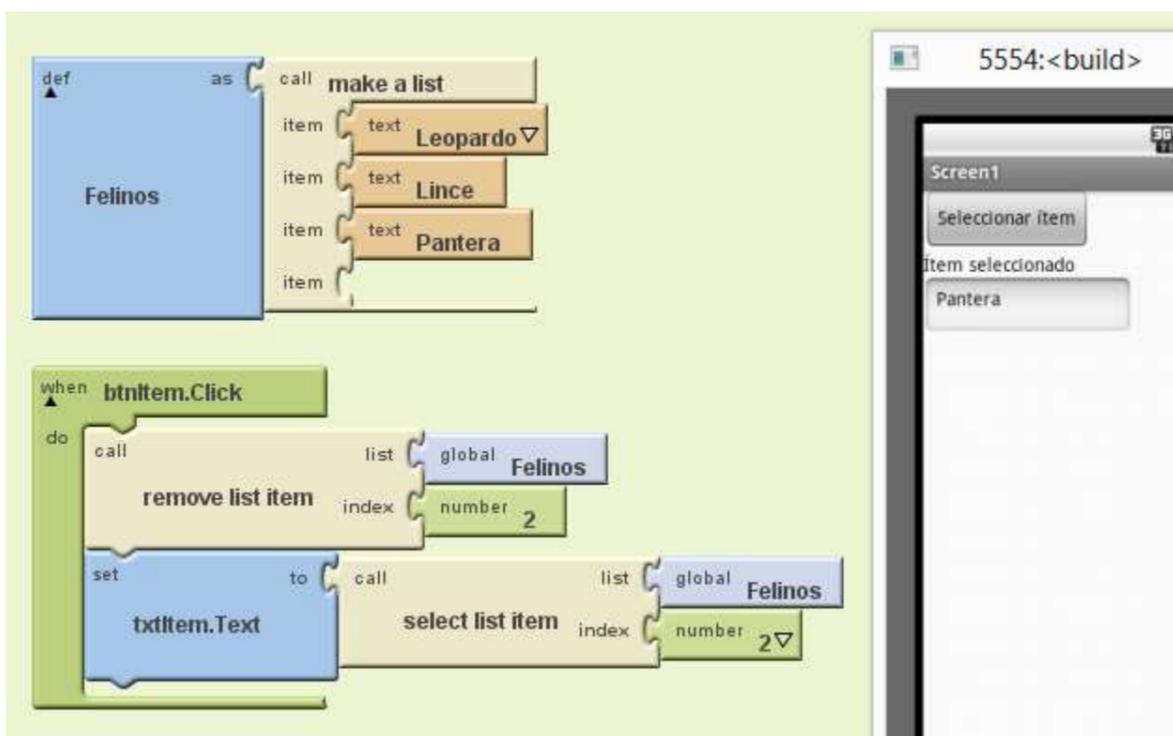


Imagen 121: Quita el segundo elemento de la lista, luego el tercero ocupa su lugar

En el ejemplo, si se presiona de nuevo el botón se borrará otro ítem y en una siguiente presión el programa colapsa por no poder mostrar el ítem señalado.

J. Copia una lista en otra

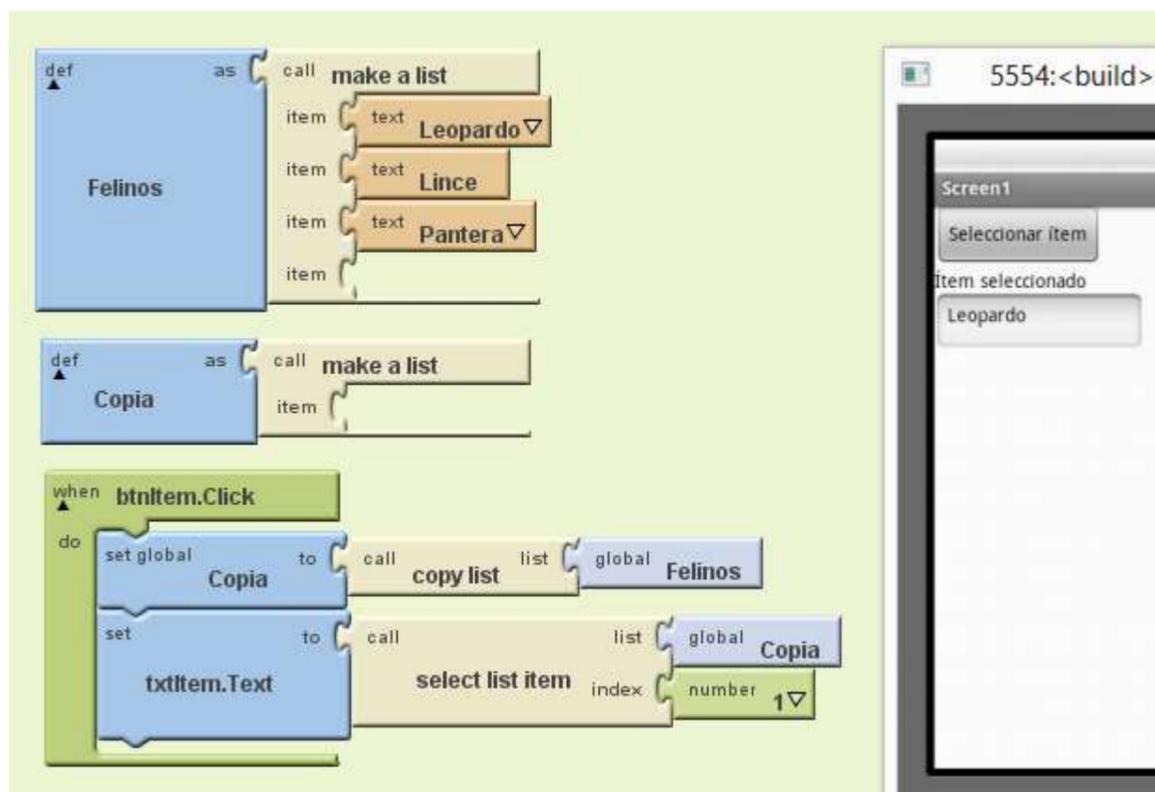


Imagen 122: Copia una lista en otra

K. Adiciona una lista a otra lista

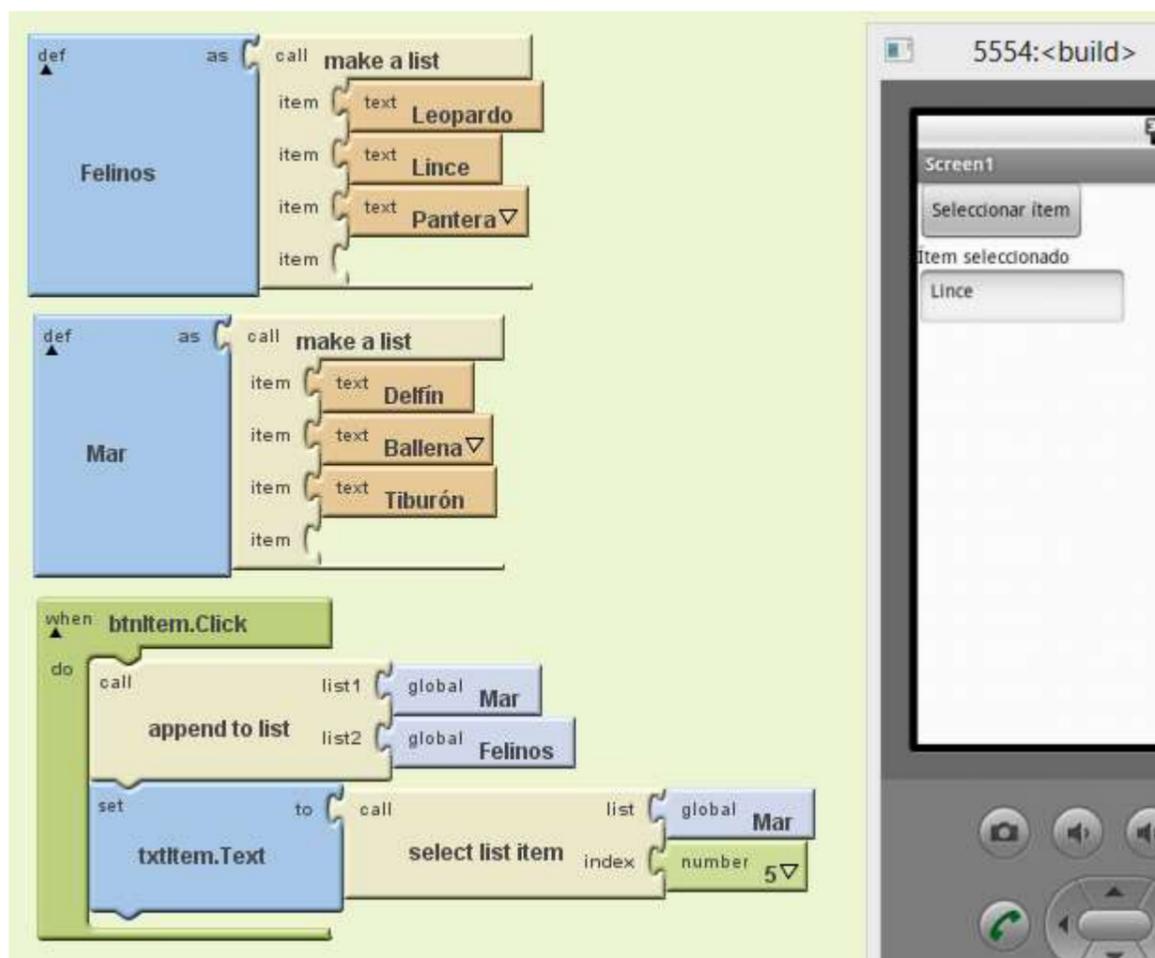


Imagen 123: La lista "Mar" recibe los ítems de la lista "Felinos"

L. ¿Es determinado objeto una lista?

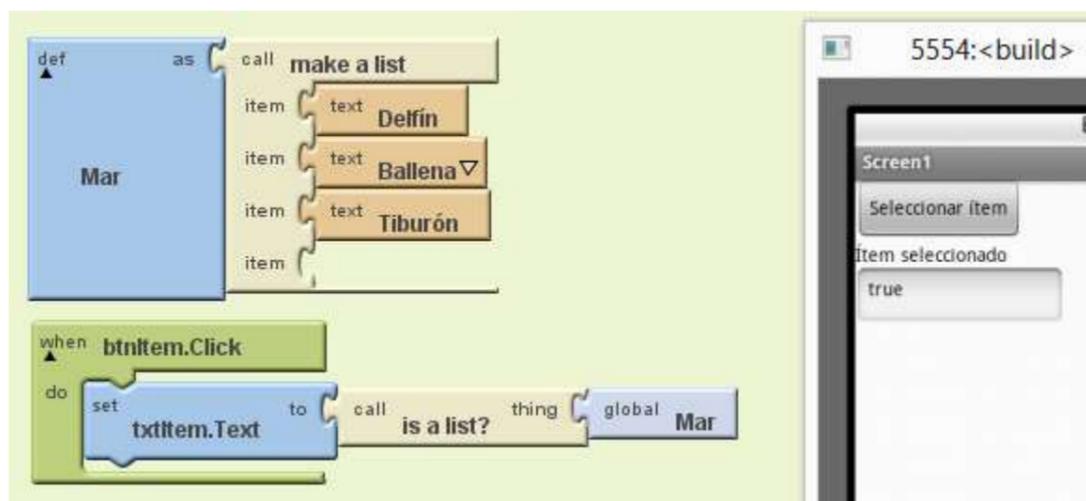


Imagen 124: ¿Es determinado objeto una lista? "true" si es una lista, "false" si no.

Paso 12. Procedimientos y Funciones

MIT App inventor permite hacer uso de procedimientos y funciones

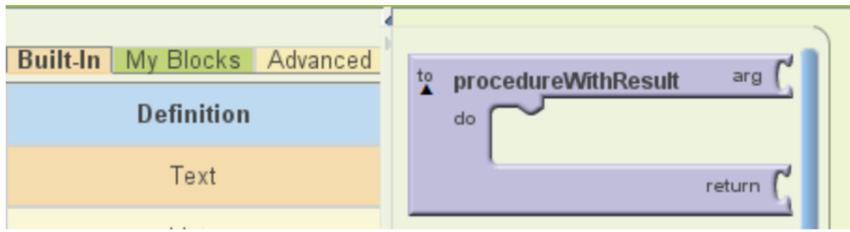


Imagen 125: En "Definition" encuentra el bloque de procedimientos

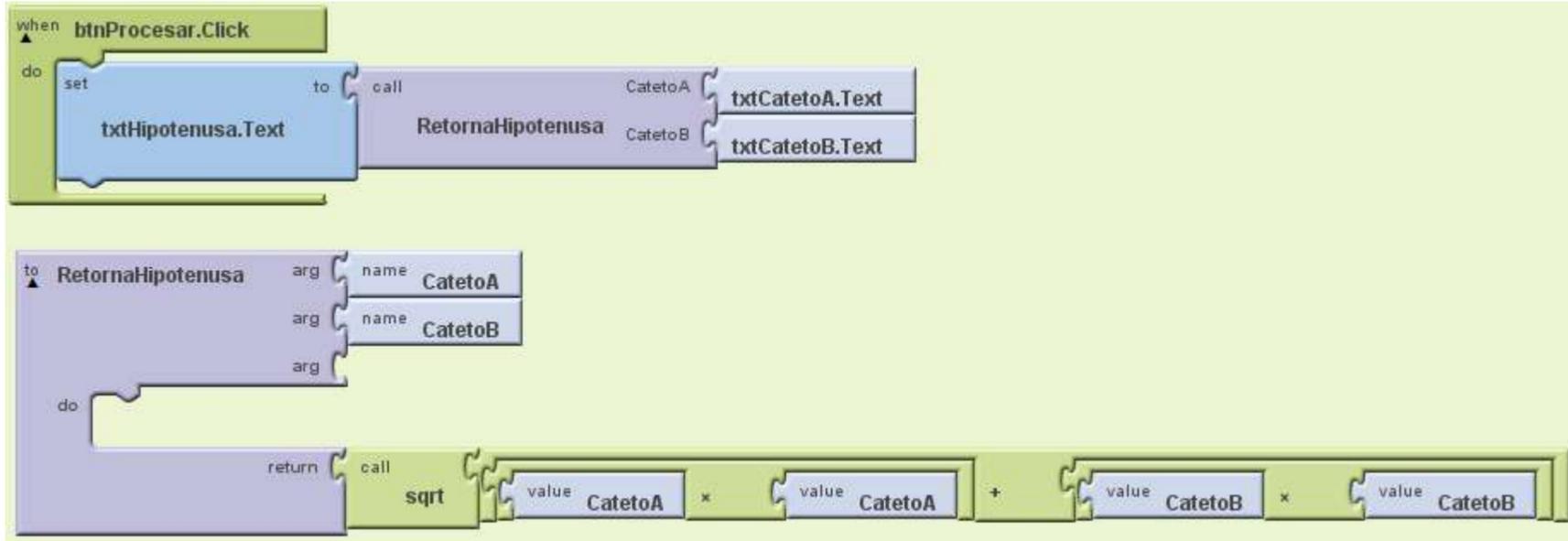


Imagen 126: Uso de una función con argumentos de entrada y el valor de salida



Imagen 127: Resultado de utilizar una función

Paso 13. Gráficos

Para hacer gráficos (líneas, puntos, círculos) se requiere un objeto llamado Canvas, el cual se arrastra a la ventana de diseño



Imagen 128: Se requiere un objeto llamado "Canvas"

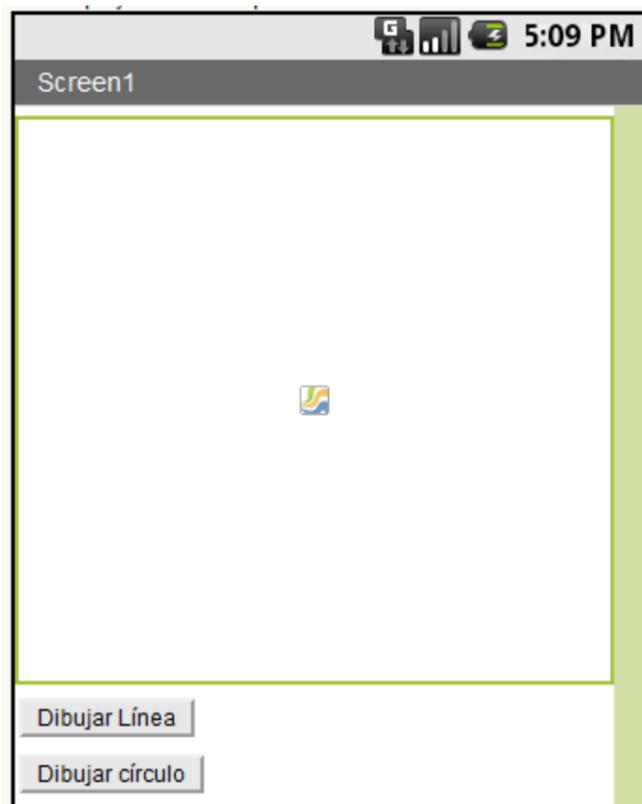


Imagen 129: Objeto "Canvas" junto con dos botones



Imagen 130: El objeto "Canvas" se le ha puesto el nombre de Lienzo

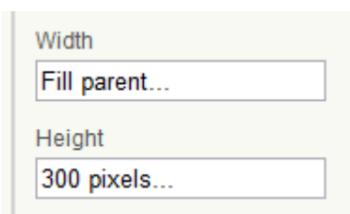


Imagen 131: Se ajusta el ancho y alto de las propiedades del objeto "Canvas"

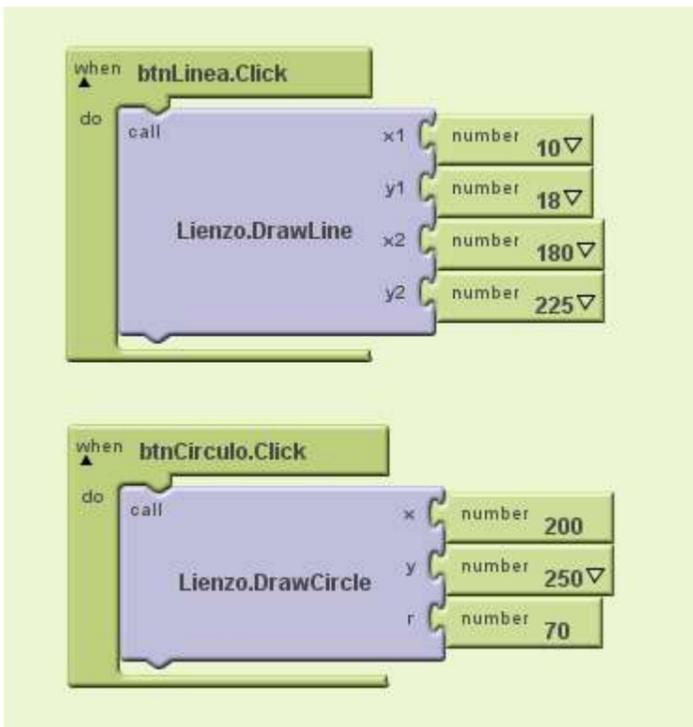


Imagen 132: Cuando se presionan los botones, desde el objeto "Canvas" llamado Lienzo, se hace llamado a las funciones de dibujado

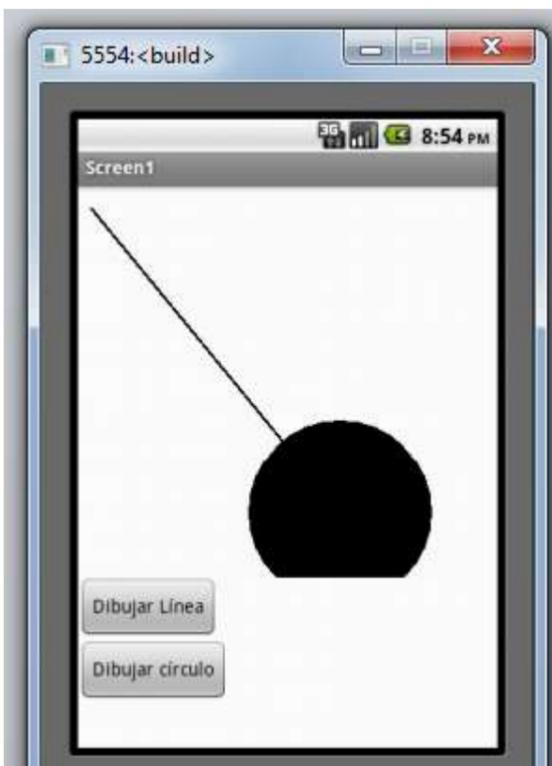


Imagen 133: Así se dibuja en el objeto "Canvas"

A. Dibujar puntos, líneas y círculos

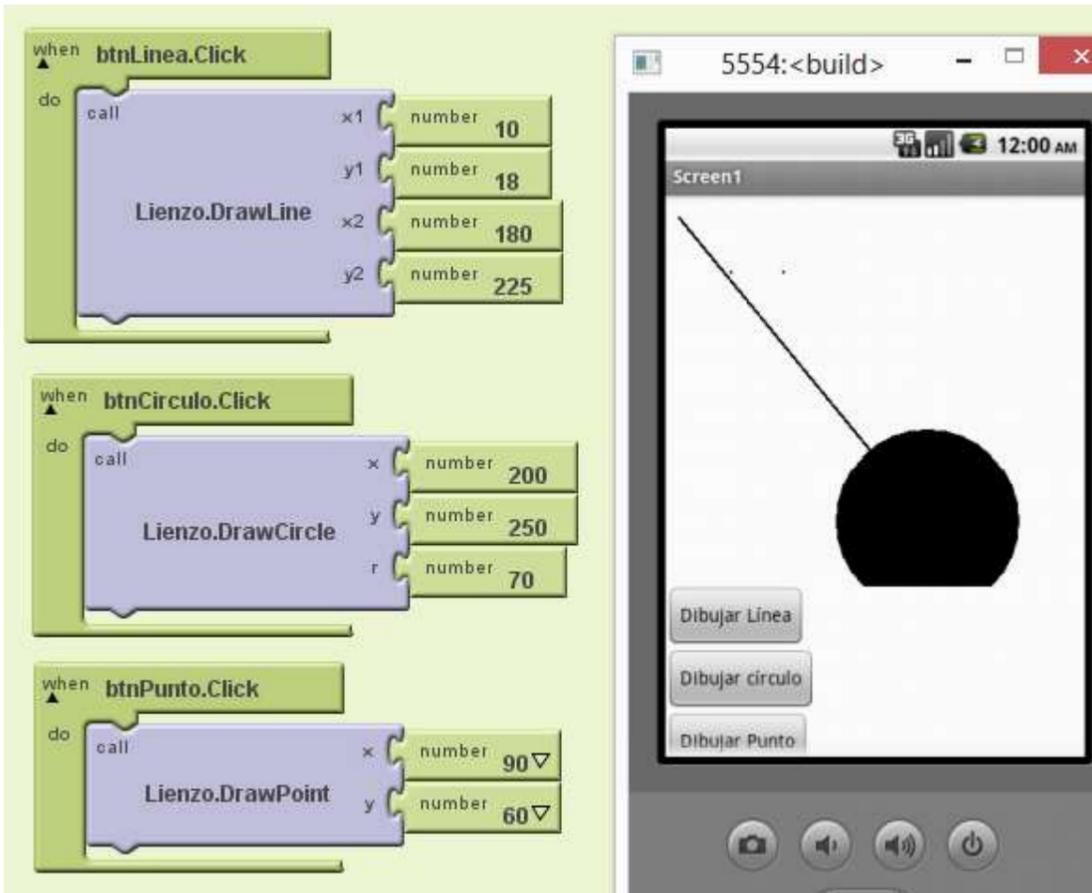


Imagen 134: La función "DrawPoint" permite dibujar puntos en determinada posición

B. Dibujar textos

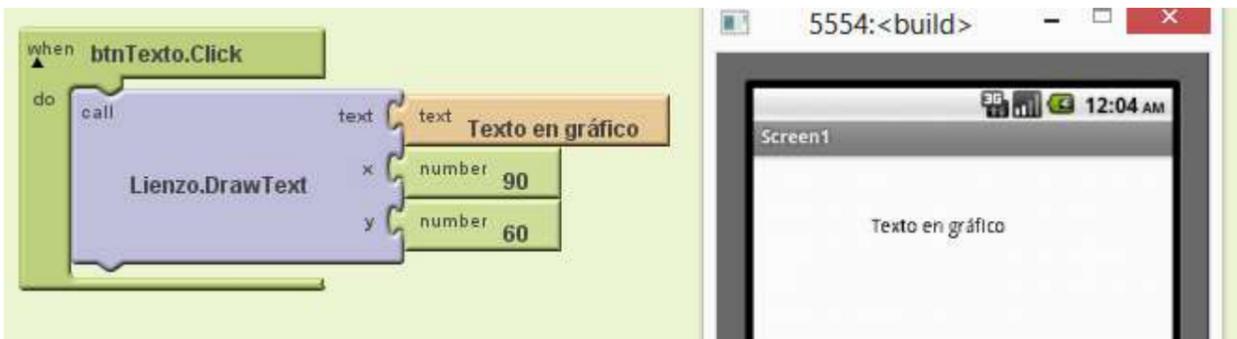


Imagen 135: La función "DrawText" permite dibujar textos en determinada posición

C. Dibujar texto girado en un determinado ángulo

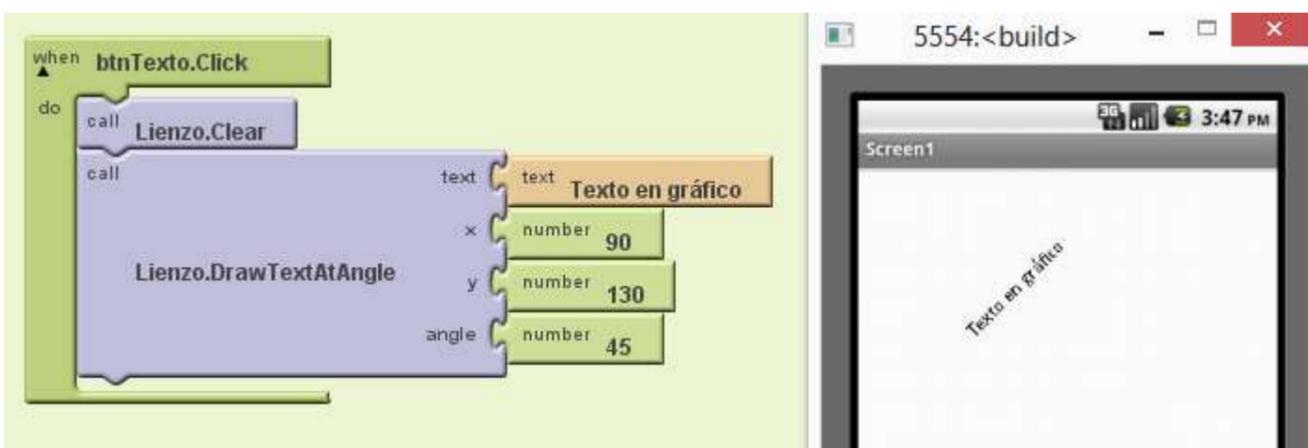


Imagen 136: Con la instrucción "clear" borra el lienzo, con la instrucción "DrawTextAtAngle" dibuja el texto en un ángulo

D. Tamaño y color del texto

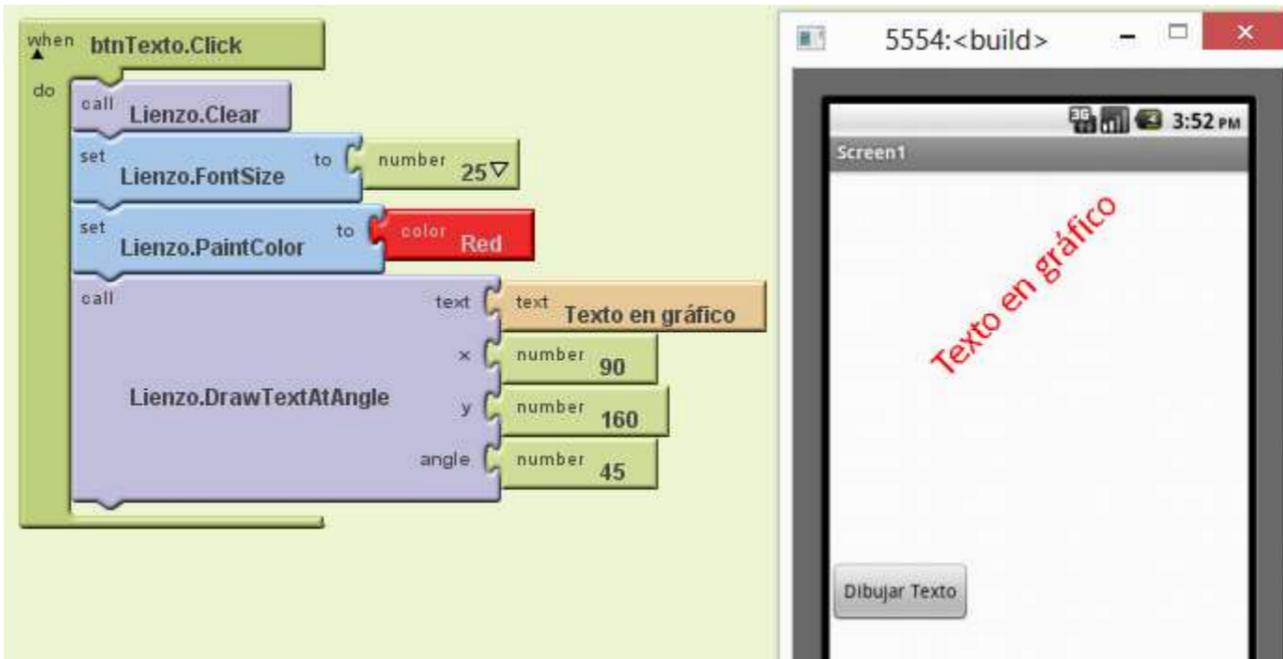


Imagen 137: Con "FontSize" cambia el tamaño de la fuente y con "PaintColor" cambia el color de lo que se dibuja

E. Fondo del lienzo en un color determinado

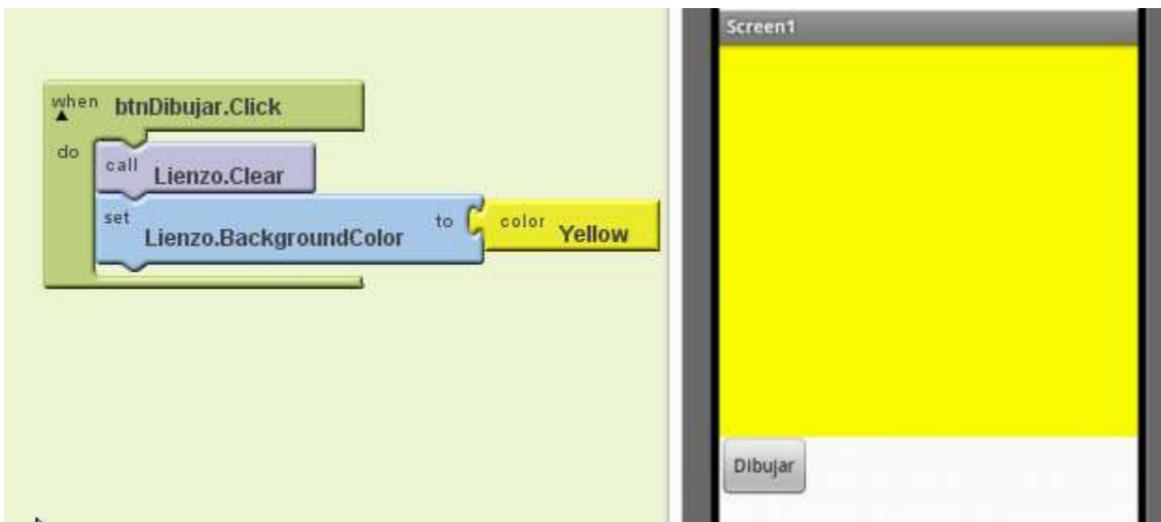


Imagen 138: Con "BackgroundColor" se puede cambiar el color de fondo del lienzo

F. Dibujando una línea con determinado ancho

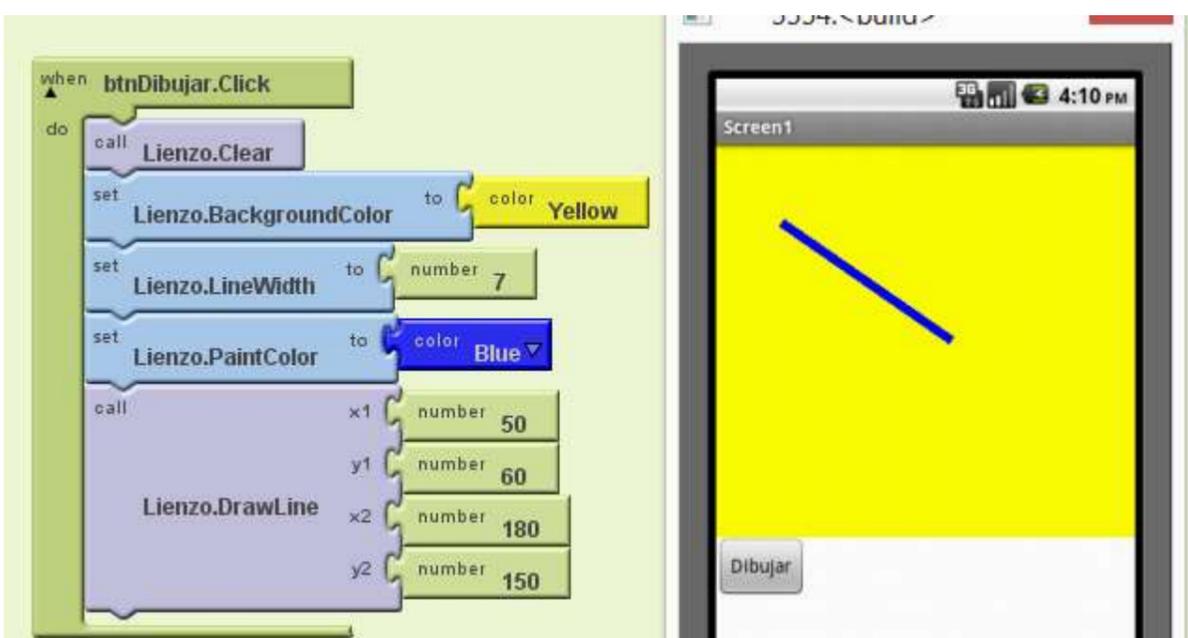


Imagen 139: Dibuja de color azul la línea con determinada anchura "LineWidth"

G. Imagen de fondo

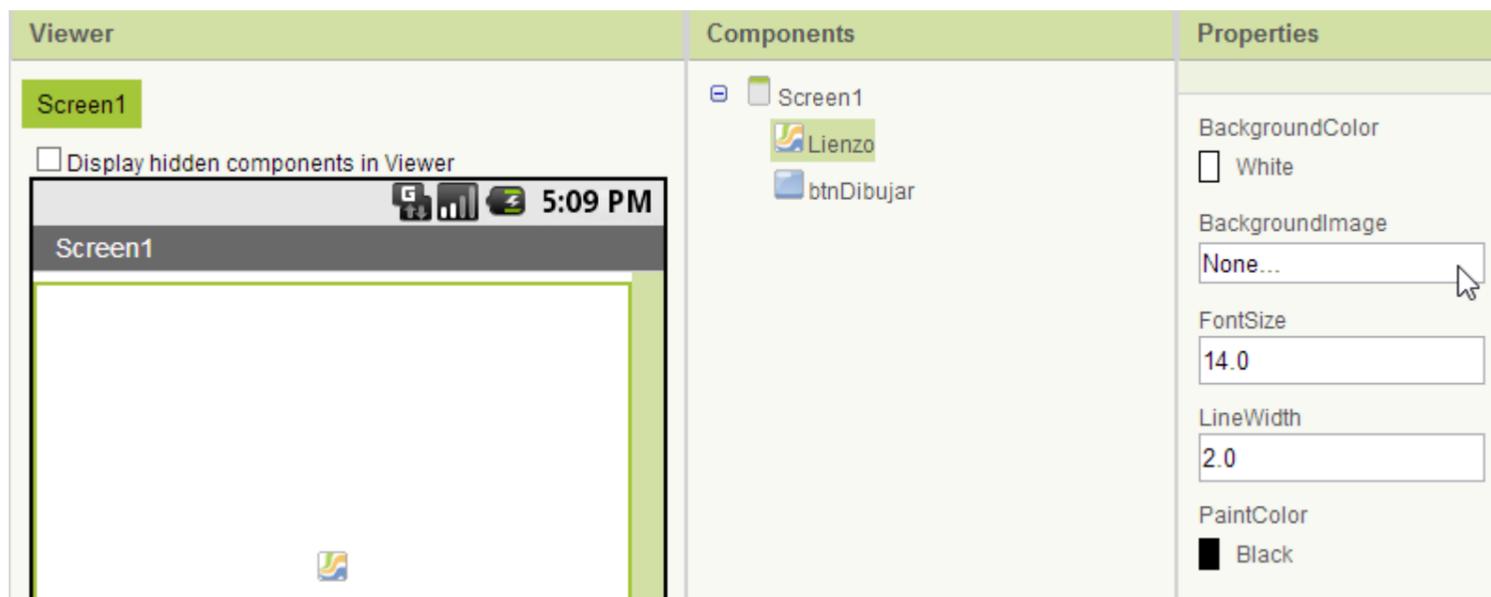


Imagen 140: Puede poner una imagen de fondo al lienzo, de clic en BackgroundImage

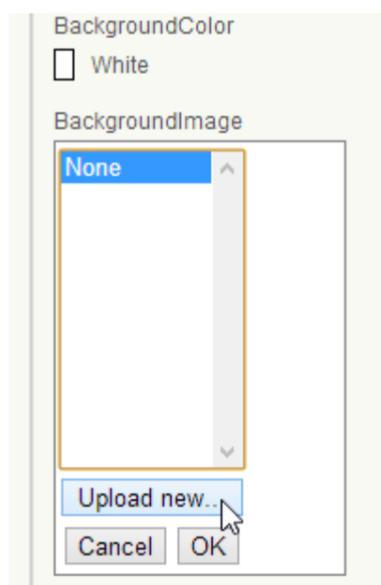


Imagen 141: De clic en "Upload new..."

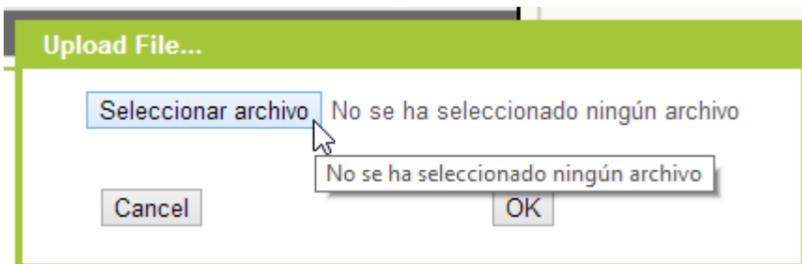


Imagen 142: De clic en "Seleccionar archivo"

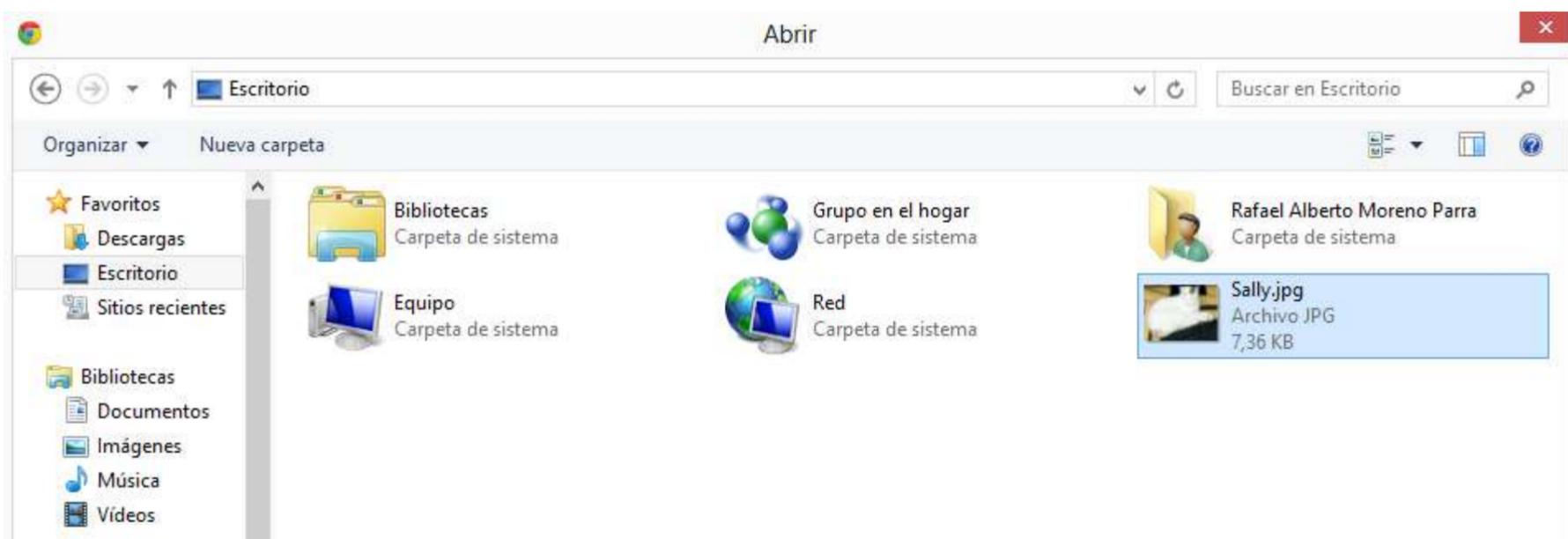


Imagen 143: Seleccione una imagen. Recomendado: De tamaño pequeño

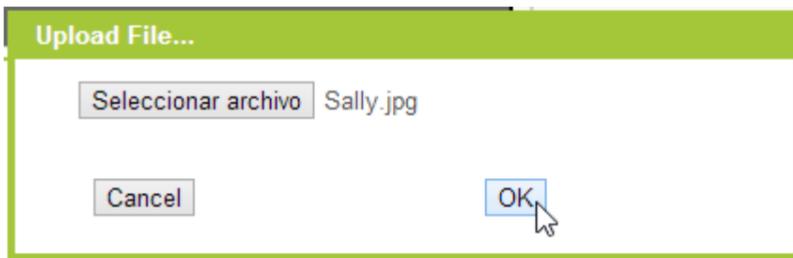


Imagen 144: Imagen cargada. De clic en "OK"

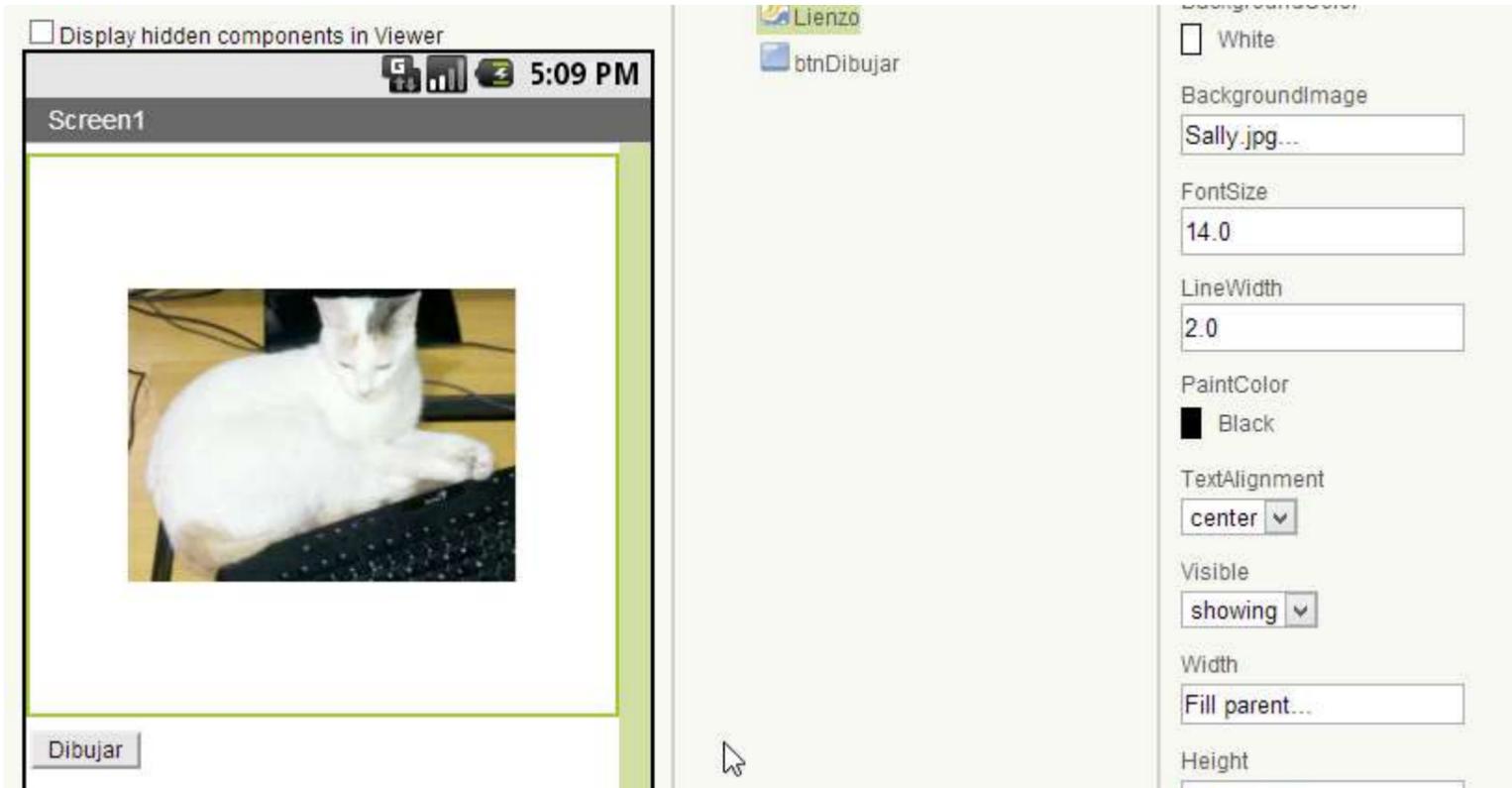


Imagen 145: Ya hay una imagen de fondo



Imagen 146: Ejecución con imagen de fondo

H. Dibujar círculos con clic de ratón (inicio de captura de eventos)

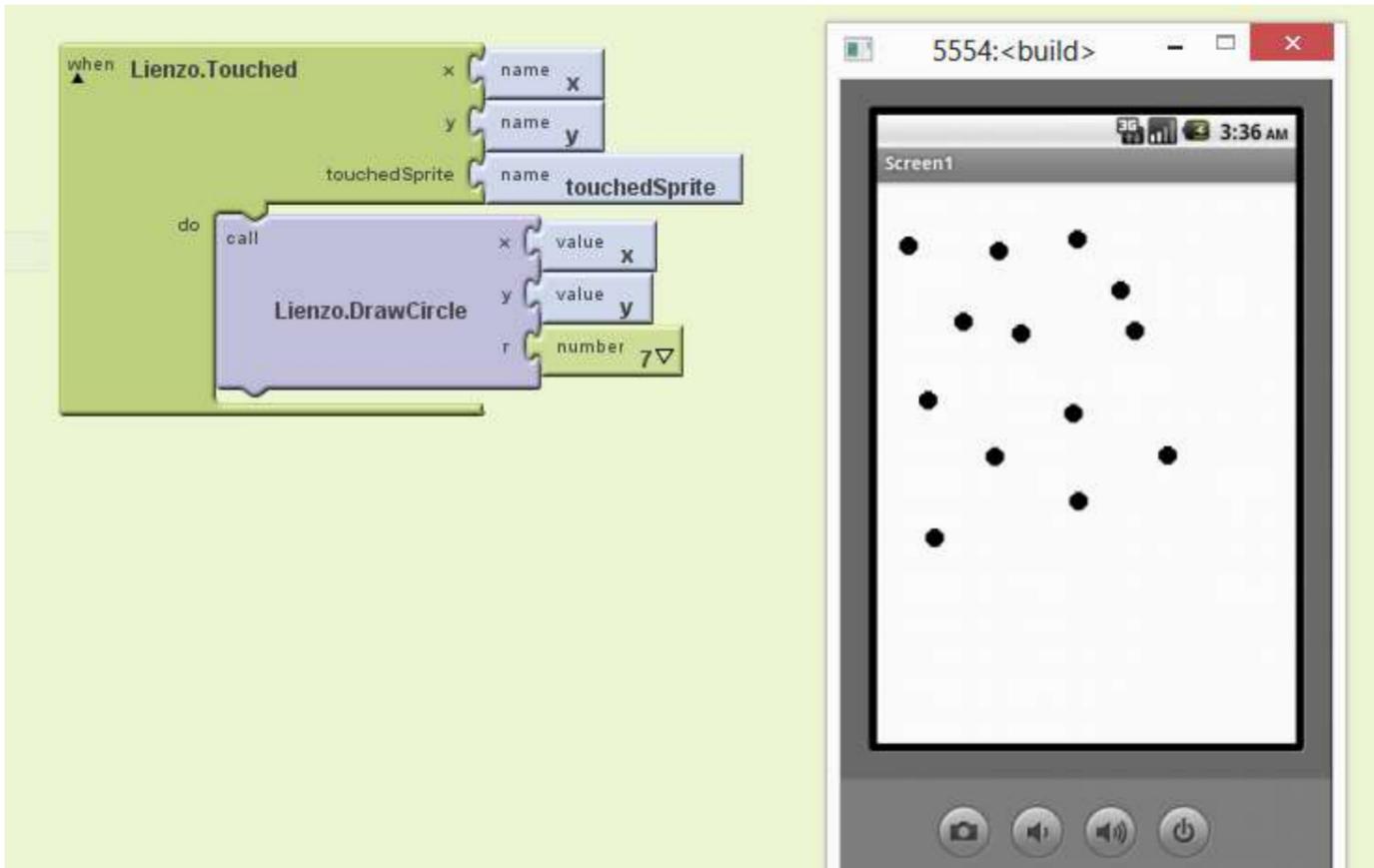


Imagen 147: Captura de eventos

El evento "Touched" define las variables "x" y "y" que son las coordenadas del plano donde el usuario da clic. Luego la sentencia que se ejecuta al hacer "Touched", que en el ejemplo es "DrawCircle" puede hacer uso de esas variables. El usuario al hacer clic en algún punto del canvas, va dibujando círculos rellenos.

I. Hacer un sencillo programa de dibujo (captura de eventos)

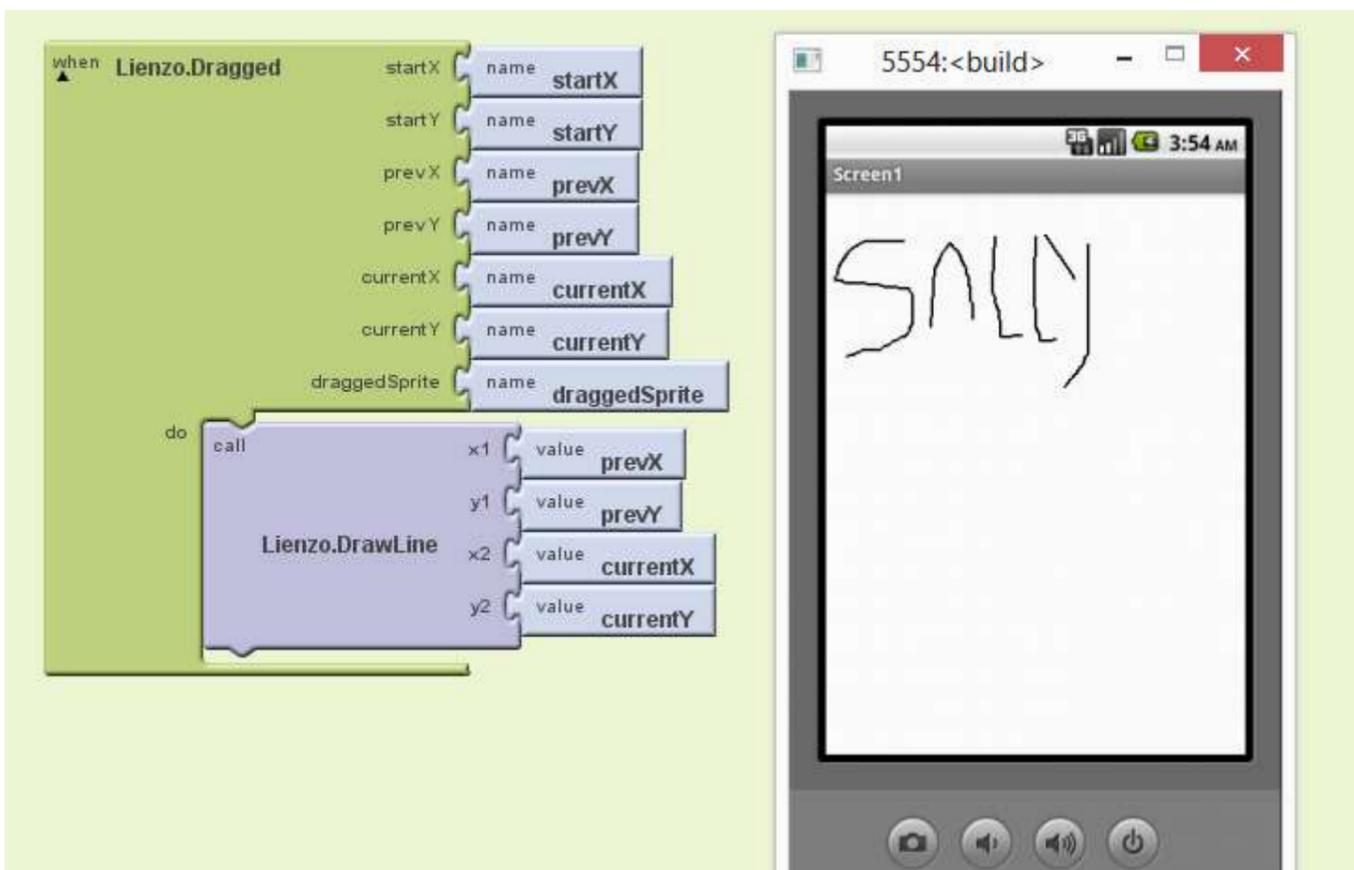


Imagen 148: Al mantener presionado el botón izquierdo del ratón, se dibuja lo que quiere.

Paso 14. Animación

Hay varias formas de hacer animación, se muestra usando un componente de tiempo y otro usando las propiedades de un objeto gráfico

A. Usando un componente de tiempo



Imagen 149: Objeto "Clock" para manejar la animación



Imagen 150: Se utiliza el objeto "Ball" que será lo que se va a animar

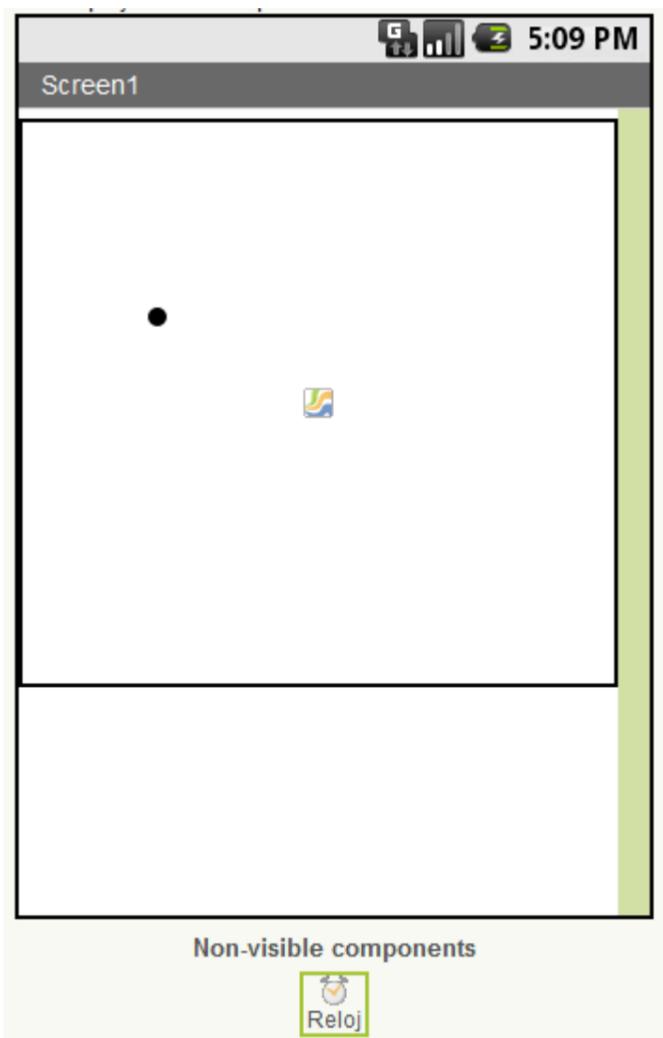


Imagen 151: El proyecto es un objeto "Canvas", un objeto reloj y un objeto "Ball"

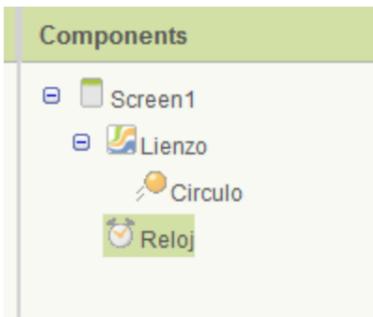


Imagen 152: Los respectivos nombres a cada objeto

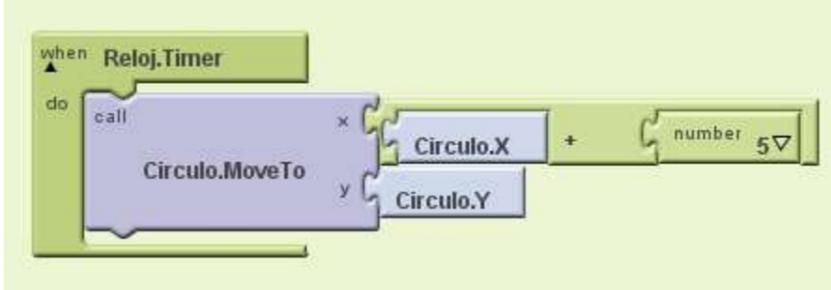


Imagen 153: El algoritmo de animación. Mover el objeto "Ball" horizontalmente.

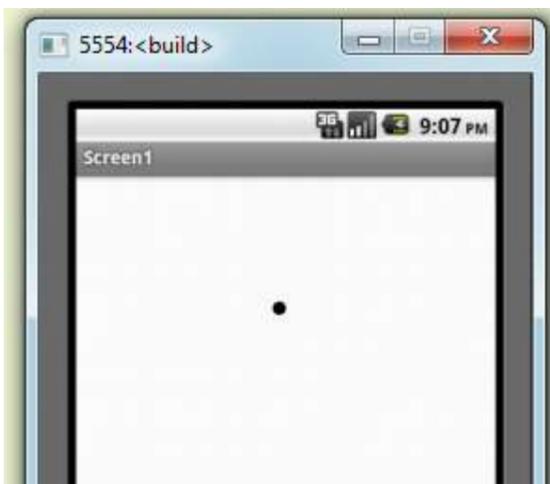


Imagen 154: Un "frame" de la animación

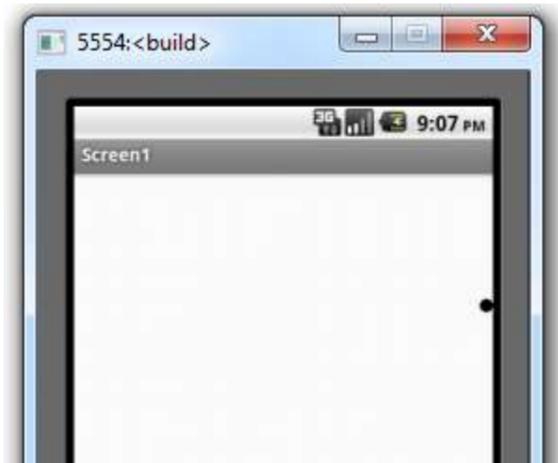


Imagen 155: Otro "frame" de la animación

B. Hacer que una pelota rebote en los bordes (sin requerir componente de tiempo)

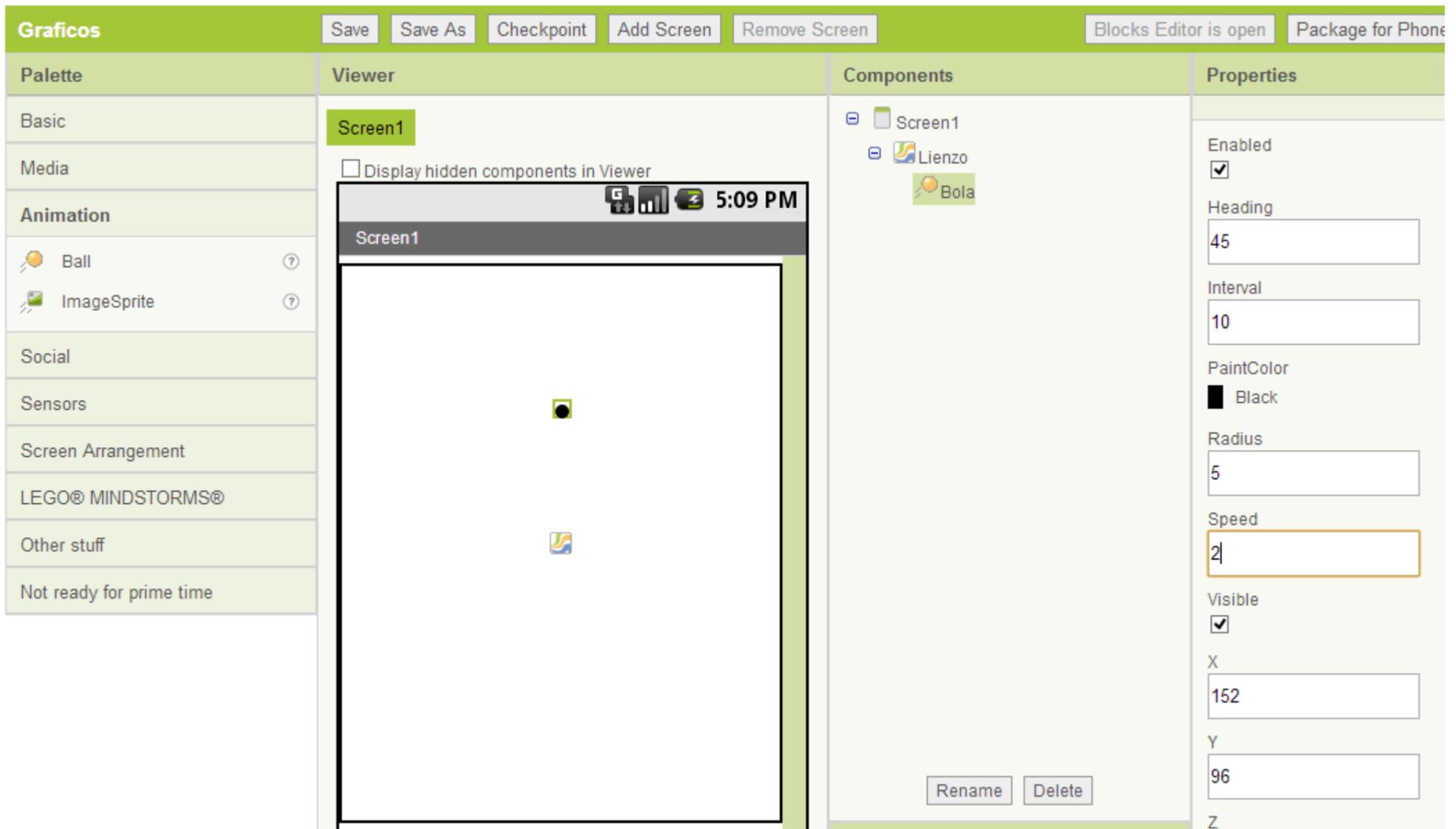


Imagen 156: Se pone un objeto Balln al interior del lienzo

En el lienzo, ubique un componente Ball en su interior, renómbrelo a "Bola" y dele valores a los siguientes atributos:

Enabled: Debe estar habilitado

Heading: Es la dirección hacia donde se dirige en grados. 0 grados es hacia la derecha horizontal (ir al Oriente), 90 grados es hacia arriba vertical (ir al Norte), luego 45 grados sería como ir en dirección Nororiental, 180 grados es ir hacia la izquierda horizontal (ir al Occidente), 270 grados es ir hacia abajo vertical (ir al Sur). Se puede seleccionar entre 0 y 360 grados. En el ejemplo se escogió 45.

Interval: Cada cuantos milisegundos cambia de posición la bola.

Radius: Radio de la bola

Speed: Cuantos pixeles salta por cada cambio de posición de la bola.

Visible: Debe estar habilitado

X: Posición inicial en **x** de la bola

Y: Posición inicial en **y** de la bola

Este es el código que debe programarse

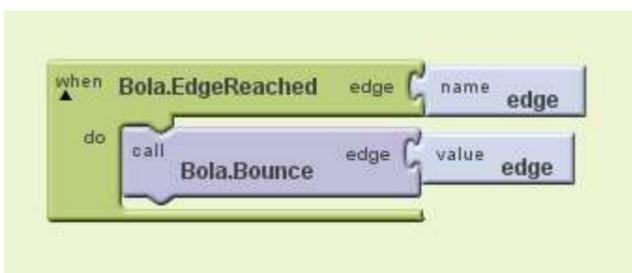


Imagen 157: Código para que la bola rebote en el borde del lienzo



Imagen 158: Captura de la animación de rebote

Paso 15. Bases de datos

Si quiere hacer sencillos sistemas para almacenar y recuperar datos, se tiene acceso a un componente de base de datos llamado "TinyDB"

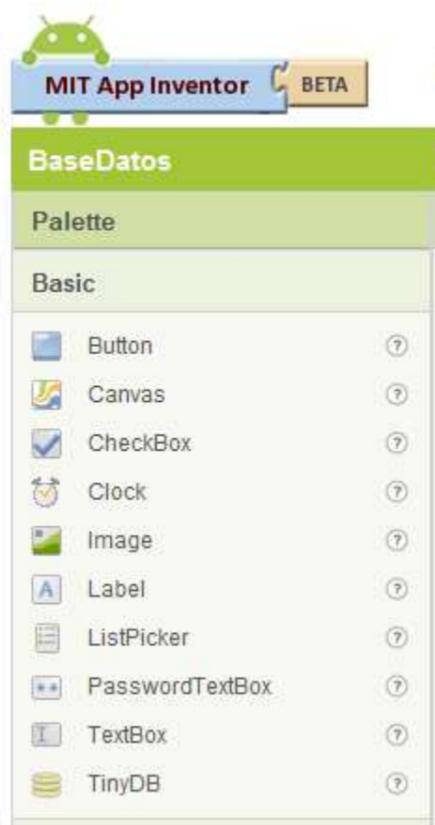


Imagen 159: Se requiere el objeto TinyDB

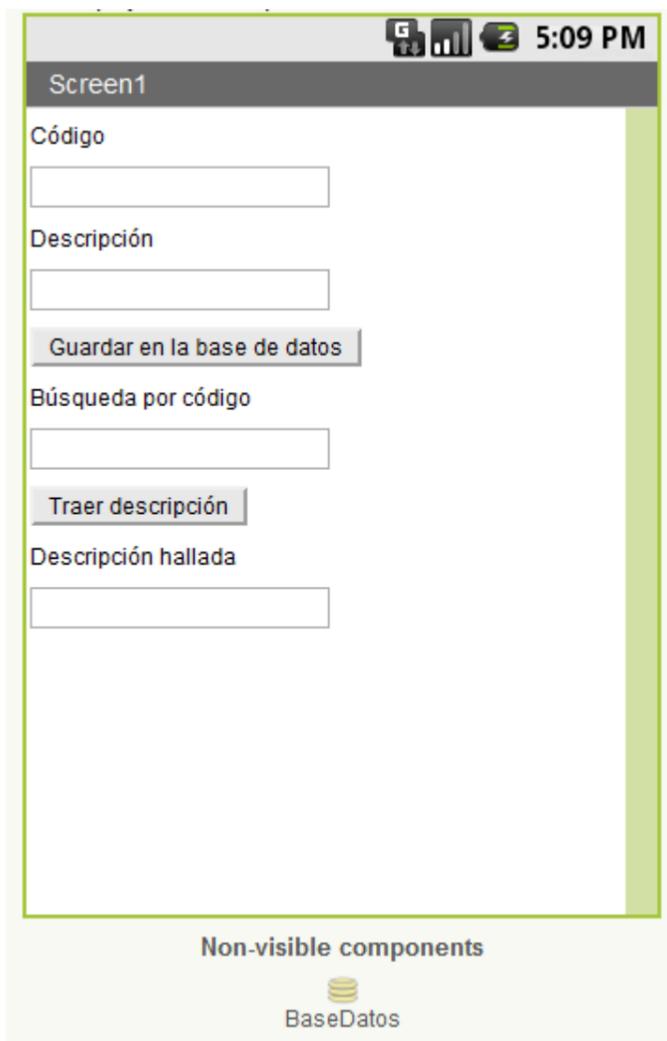


Imagen 160: Formulario para guardar y recuperar datos

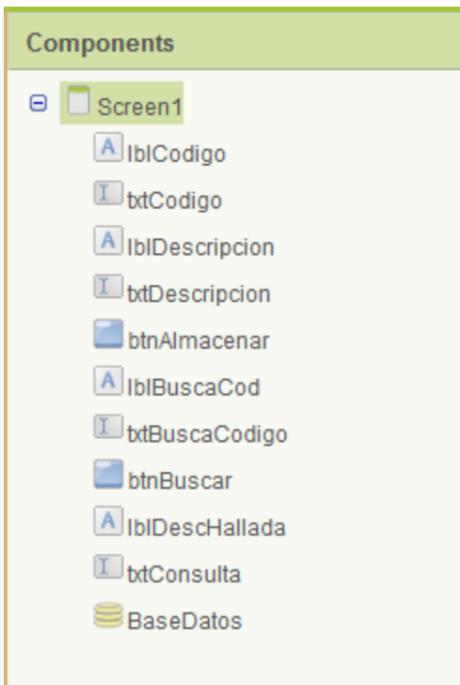


Imagen 161: La lista de componentes u objetos

Para almacenar en la base de datos, se usa la función StoreValue la cual tiene dos argumentos: un tag que sería una llave primaria y un valueToStore que sería el valor a guardar



Imagen 162: Guardando registros en la base de datos

Para consultar, se usa la función GetValue que tiene como argumento la llave primaria y retorna el valor almacenado.



Imagen 163: Restaura el valor almacenado consultando a través de la llave primaria

Paso 16. La capacidad inalámbrica de MIT App Inventor

¿Quiere ejecutar las aplicaciones directamente en un Smartphone Android y de manera sencilla? Eso es posible con la nueva funcionalidad wifi de MIT App Inventor. Sólo es seguir los siguientes pasos:

- Instalar la aplicación MIT AI Companion de la Play Store en el Smartphone
- Ejecutar ese programa en el Smartphone
- En MIT App Inventor decir que se conecta vía Wifi
- MIT App Inventor genera un código que se digita en el Smartphone
- Eso es todo, la aplicación corre directamente en el Smartphone

A continuación imágenes de este proceso



Imagen 164: Buscar en la "Play Store" a MIT AICompanion

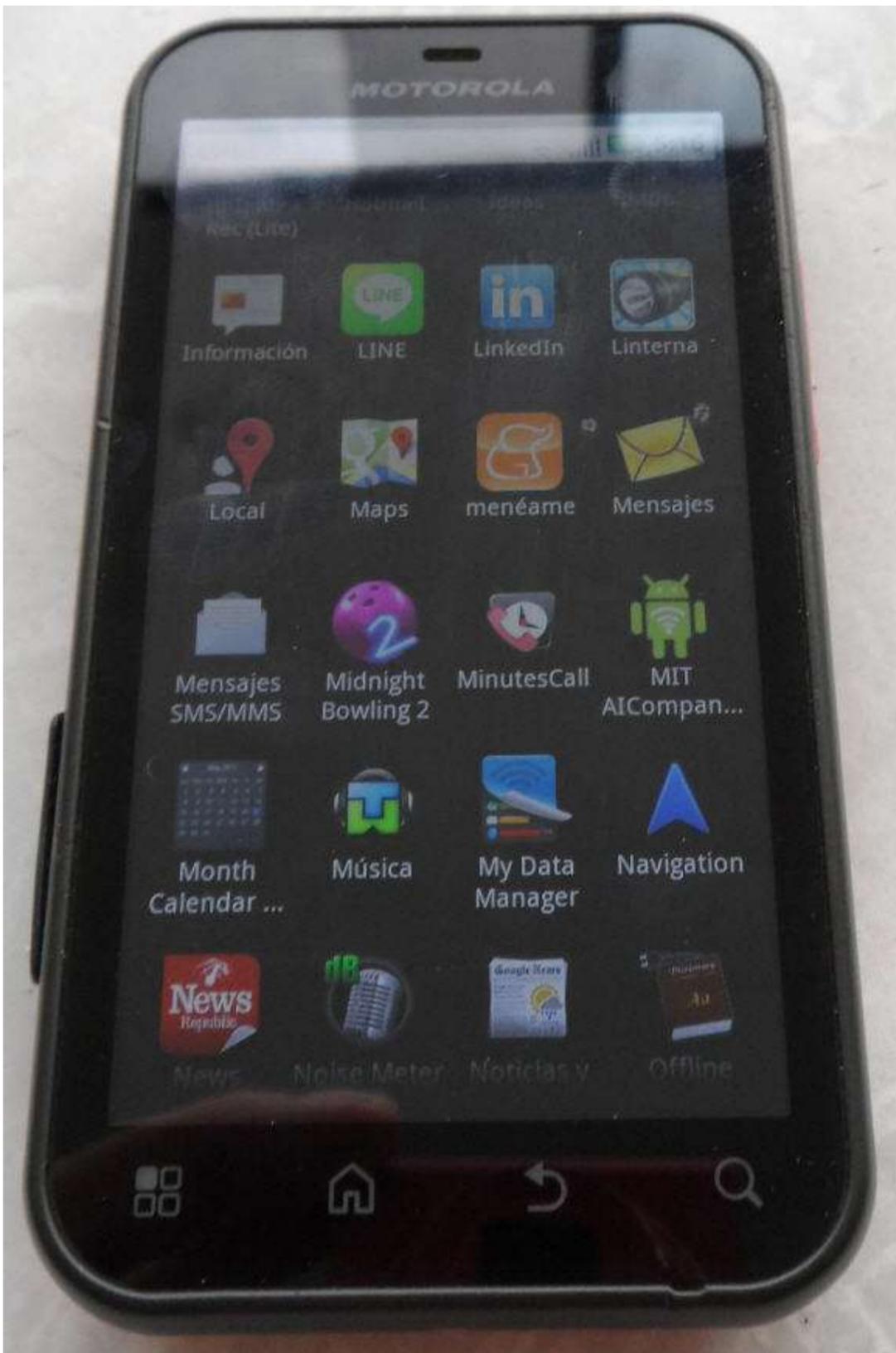


Imagen 165: Se instala la aplicación en el Smartphone

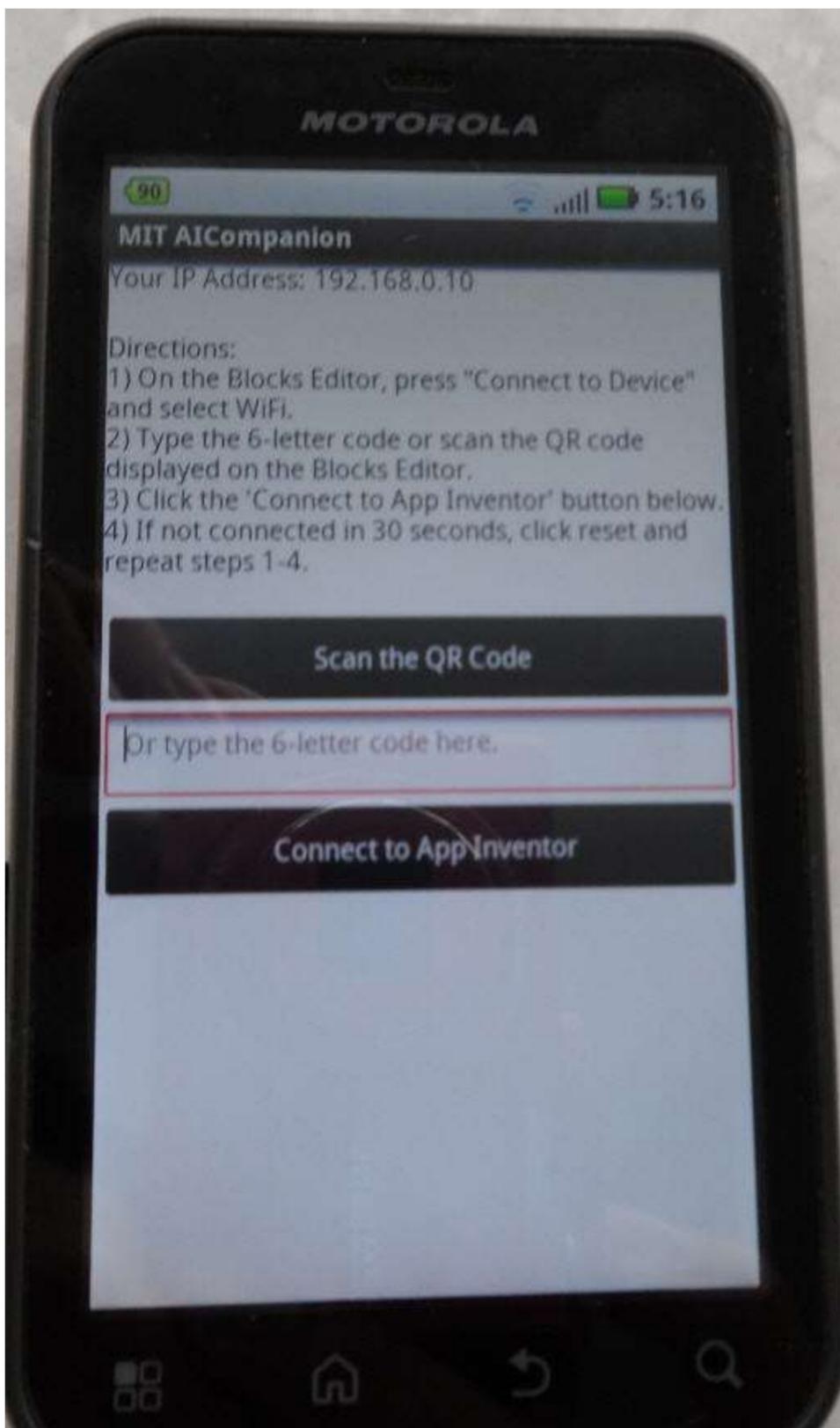


Imagen 166: Se ejecuta la aplicación en el smartphone

En el MIT App Inventor se selecciona Wifi

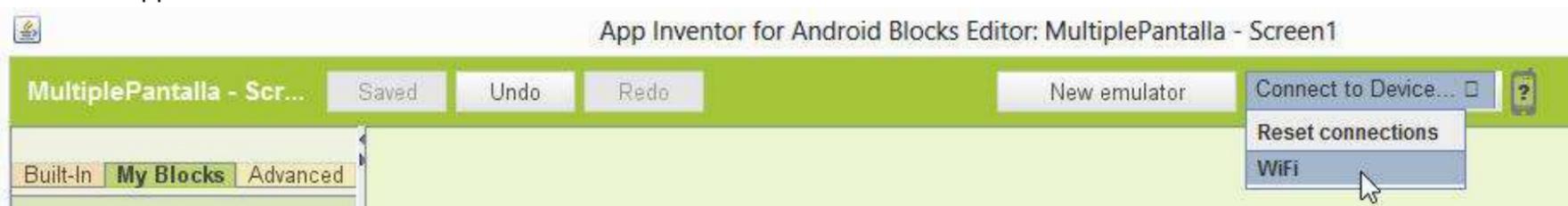


Imagen 167: Se selecciona WiFi en el App Inventor Blocks Editor

Sale un aviso en el que se digita un código o se escanea un código QR



Imagen 168: En el smartphone se digita el código mostrado o se escanea el código QR

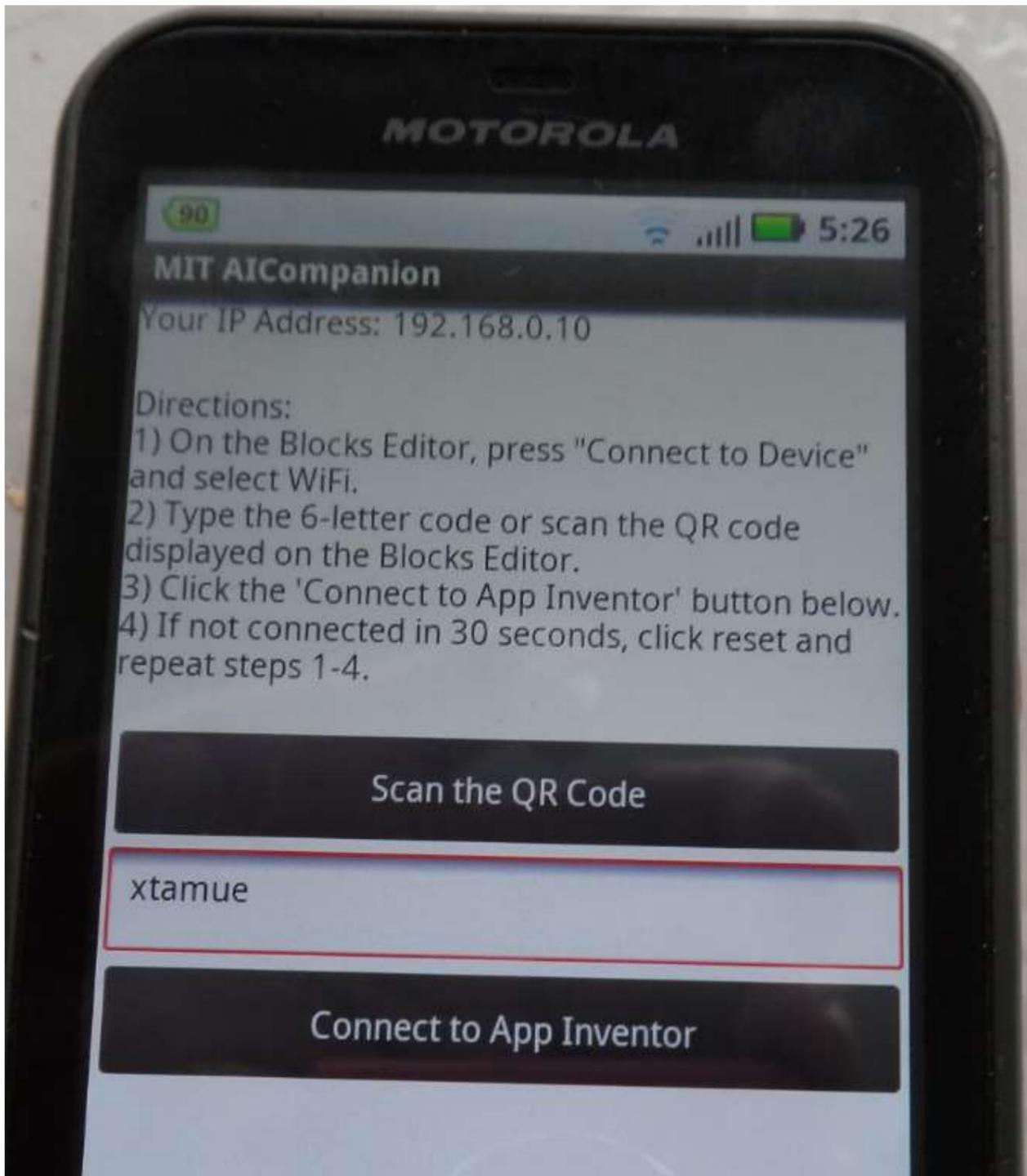


Imagen 169: Se digita el código en el smartphone

Paso 17. Tomando vídeos

Aquí es necesario hacer uso de la capacidad Wifi de MIT App Inventor para probar la capacidad de grabación directa de un Smartphone Android.

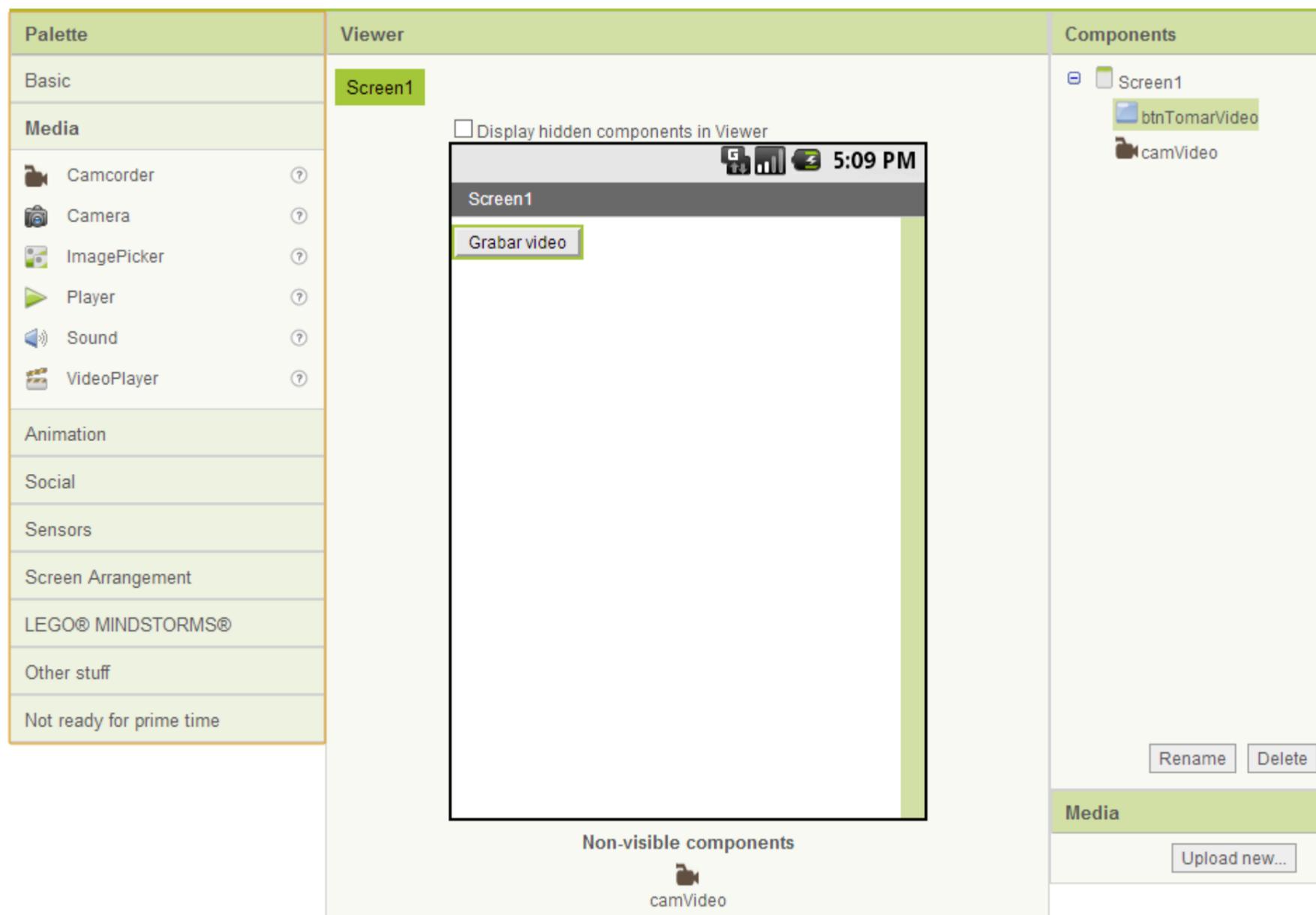


Imagen 170: Se pone un botón y un componente Media llamado "Camcorder"

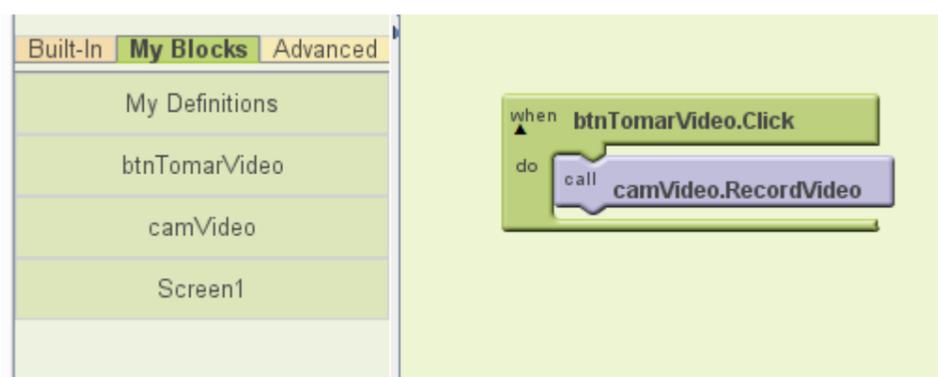


Imagen 171: Ese es el evento a programar

Se prueba directamente en el SmartPhone

Paso 18. Tomando fotos

Aquí es necesario hacer uso de la capacidad Wifi de MIT App Inventor para probar la capacidad de tomar fotos de un Smartphone Android.

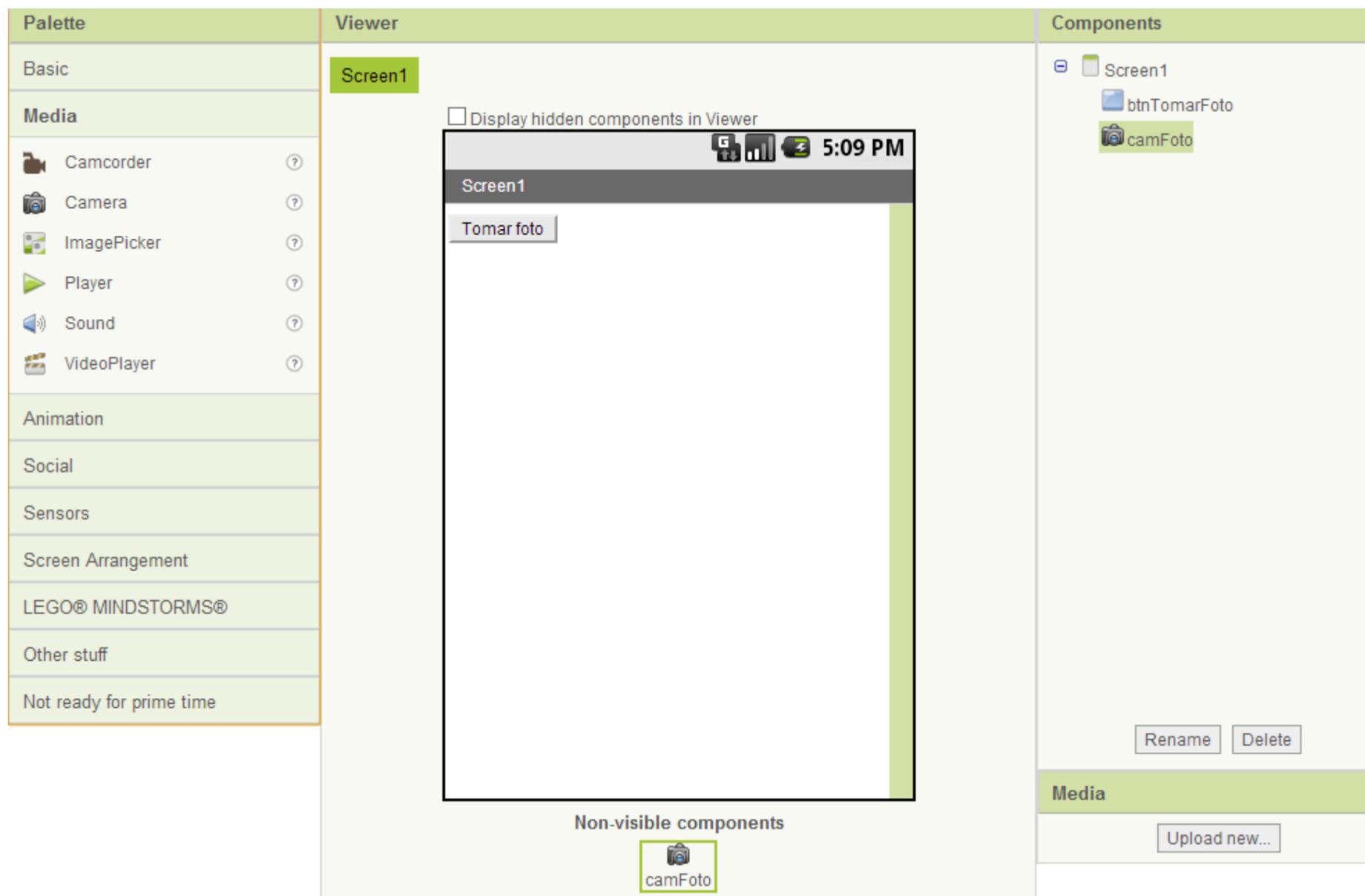


Imagen 172: Se pone un botón y un componente Media llamado "Camera"

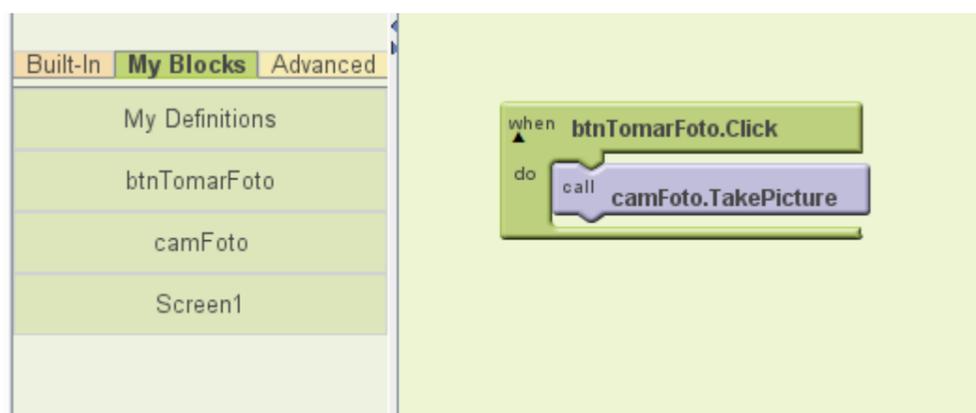


Imagen 173: Ese es el evento a programar

Paso 19. Ver galería de imágenes

Aquí es necesario hacer uso de la capacidad Wifi de MIT App Inventor para probar la capacidad de mostrar imágenes de un Smartphone Android.

Se toma el control "ImagePicker" que es similar a un botón, al dar clic sobre este, se accede a la galería de imágenes que tiene el Smartphone.



Imagen 174: ImagePicker es el control a usar, en el ejemplo se llamará "imgGaleria"

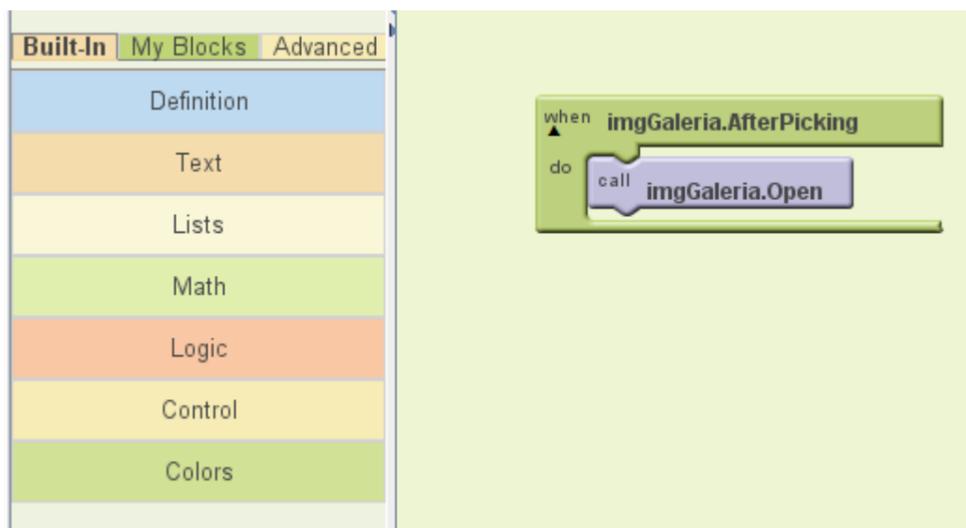


Imagen 175: Al presionar a "imgGaleria" se llama a imgGaleria.Open que muestra la galería de imágenes

Paso 20. Usando el acelerómetro

En el ejemplo, se va a mover una bolita según como mueva el Smartphone

The screenshot displays the MIT App Inventor web interface for an Android application. The interface is titled "Acelerometro" and shows a project named "Screen1". The "Palette" on the left includes "Sensors" with "AccelerometerSensor", "LocationSensor", and "OrientationSensor". The "Viewer" shows a mobile screen with a black dot and a small icon. The "Components" panel shows "Screen1", "Canvas1", "Balon", and "acelera". The "Properties" panel shows settings for the selected component, including "BackgroundColor" (White), "FontSize" (14.0), "TextAlignment" (center), and "Height" (300 pixels). The "Non-visible components" section at the bottom shows "acelera".

Se tiene un componente Canvas y en el interior un componente Ball (grupo Animation) que se llamará "Balon", y se agrega el componente AccelerometerSensor que se llamara "acelera".

En el editor de bloques, entonces se hacen la siguiente programación

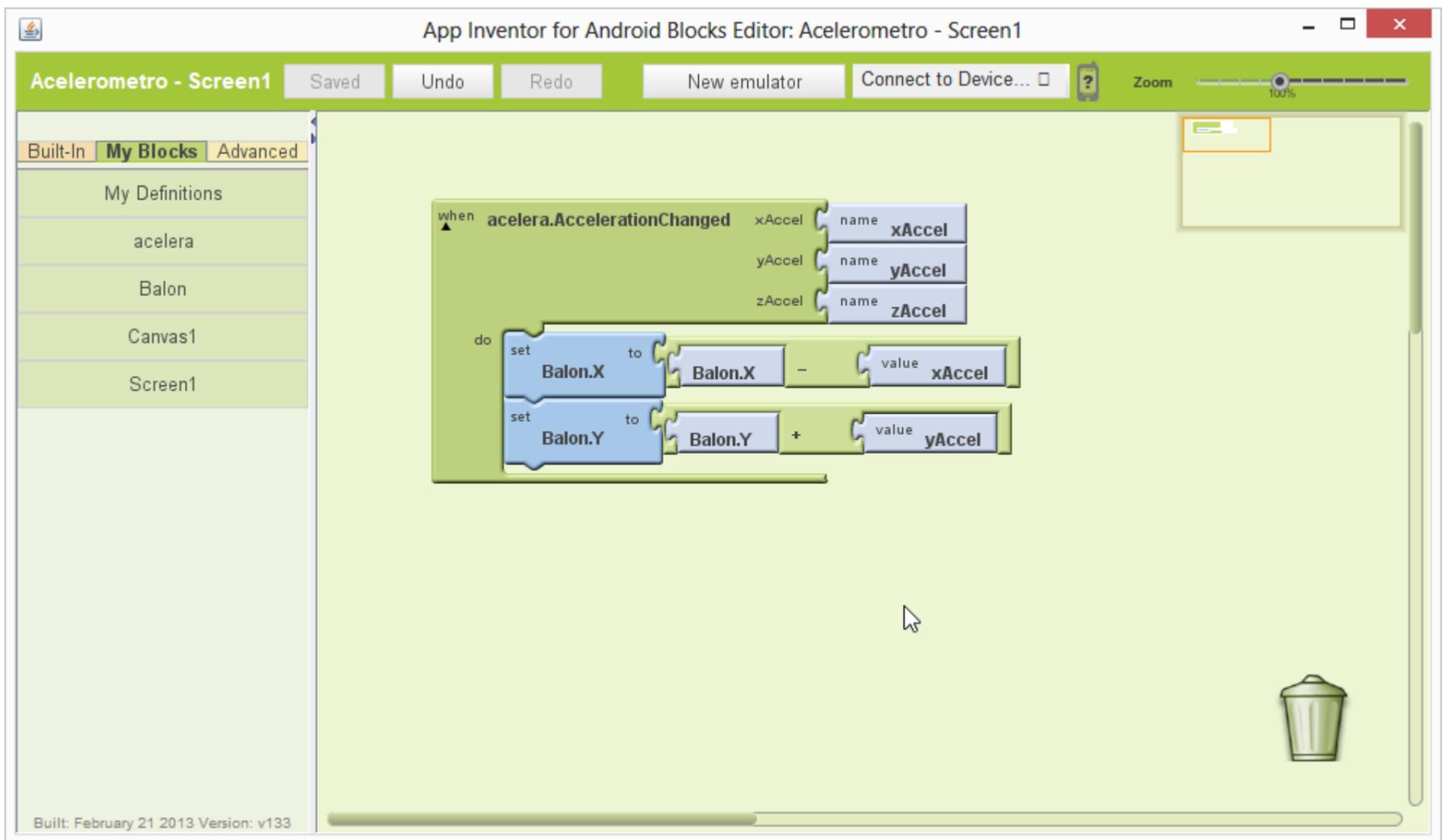


Imagen 176: Usando el acelerómetro para mover una figura animada.

Paso 21. Organización en pantalla

Los componentes visuales pueden ordenarse en la pantalla de tres formas: Horizontal, en forma de tabla y vertical. Para eso se hace uso de los componentes HorizontalArrangement, TableArrangement, VerticalArrangement

The screenshot displays the MIT App Inventor interface. At the top, there is a navigation bar with 'MIT App Inventor BETA' and 'My Projects Design Learn (Debugging)'. A welcome message for the beta release is visible, along with a 'Motd Update: 2/21/2015' notification. The main workspace is divided into four panels: 'Pantallas' (Screens), 'Viewer', 'Components', and 'Properties'. The 'Viewer' panel shows a mobile screen with a status bar at 5:09 PM and a grid of 12 buttons labeled 'Botón 1' through 'Botón 12'. The 'Components' panel shows a tree view with 'Screen1' containing 'HorizontalArrangement1' (with buttons 1-3), 'VerticalArrangement1' (with buttons 4-6), and 'TableArrangement1' (with buttons 7-12). The 'Properties' panel on the right shows settings for the selected 'TableArrangement1' component, including 'Columns' set to 2 and 'Rows' set to 3.

Imagen 177: Uso de las tres formas de ordenar en pantalla

Especial mención en TableArrangement que se configura cuantas columnas y filas tendrá.

Paso 22. Múltiples pantallas

Este es una técnica que sirve para tener varias pantallas dentro de nuestro Smartphone Android. En el ejemplo se utilizan dos componentes canvas y dos botones, en cada canvas hay una imagen de fondo.

The screenshot shows the MIT App Inventor web interface. At the top, there is a navigation bar with 'MIT App Inventor BETA' and 'My Projects Design Learn (Debugging)'. A welcome message for the beta release is visible, along with a 'Motd Update: 2/21/2011' and a note about using MIT 2.05 for WiFi Development. The main workspace is titled 'MultiplePantalla' and includes buttons for 'Save', 'Save As', 'Checkpoint', 'Add Screen', and 'Remove Screen'. The interface is divided into four main sections: 'Palette', 'Viewer', 'Components', and 'Properties'. The 'Palette' on the left lists various components like Button, Canvas, CheckBox, Clock, Image, Label, ListPicker, PasswordTextBox, TextBox, and TinyDB. The 'Viewer' in the center shows a preview of the app with two canvases, each containing a different image of a cat. The 'Components' panel on the right shows a tree view with 'Screen1' containing 'PantallaA', 'PantallaB', 'btnPantalla1', and 'btnPantalla2'. The 'Properties' panel on the far right shows settings for the selected component, including BackgroundColor (White), BackgroundImage (Sally.jpg...), FontSize (14.0), LineWidth (2.0), PaintColor (Black), TextAlignment (center), Visible (showing), Width (Fill parent...), and Height (300 pixels...). A 'Media' section at the bottom right lists 'Sally.jpg' and 'Tobit.jpg'.

Imagen 178: Dos canvas, los dos con imágenes de fondo

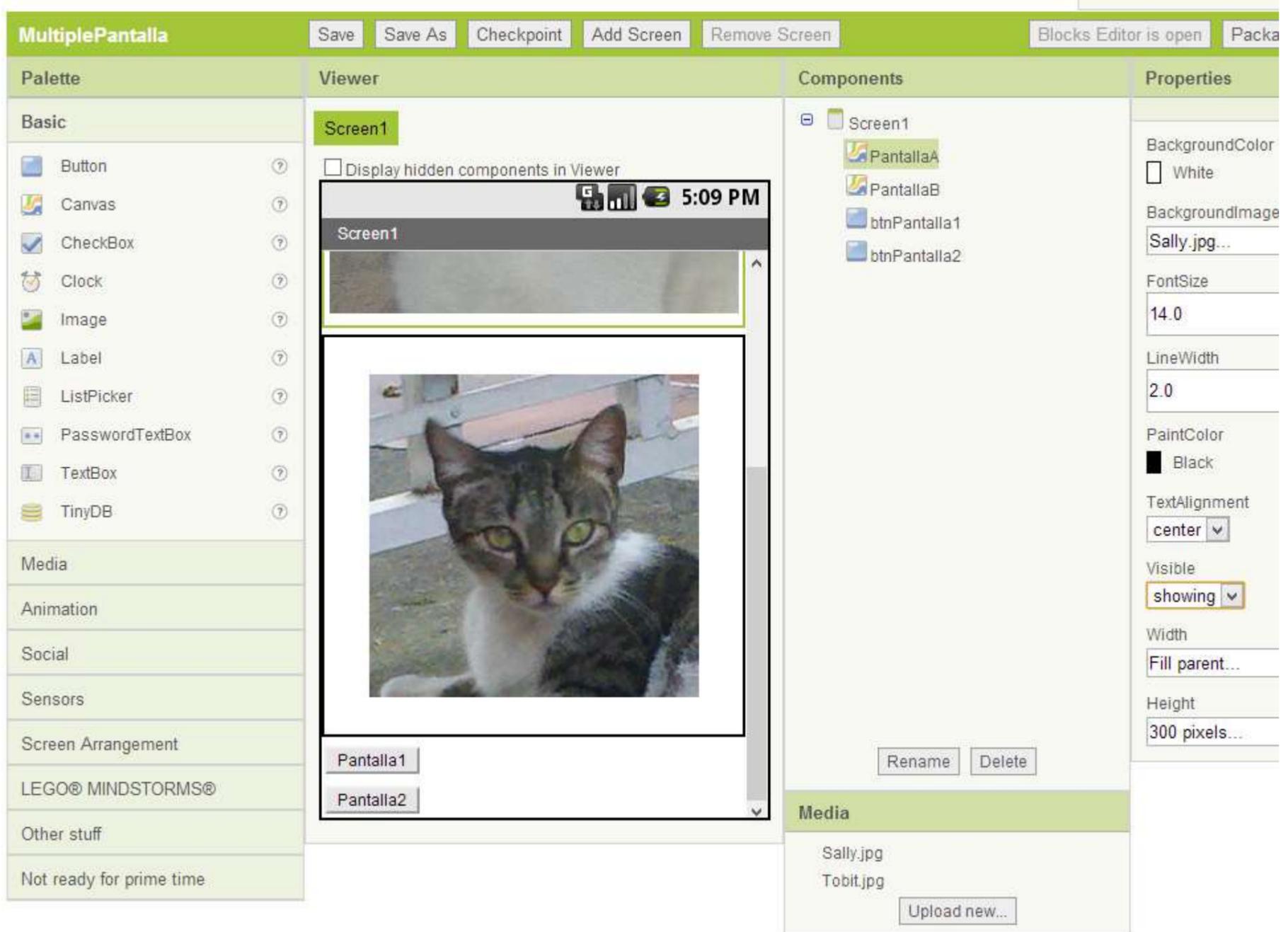


Imagen 179: Al deslizar abajo, se ve el segundo canvas y los dos botones



Imagen 180: Se pone la propiedad Visible del primer canvas en "false"

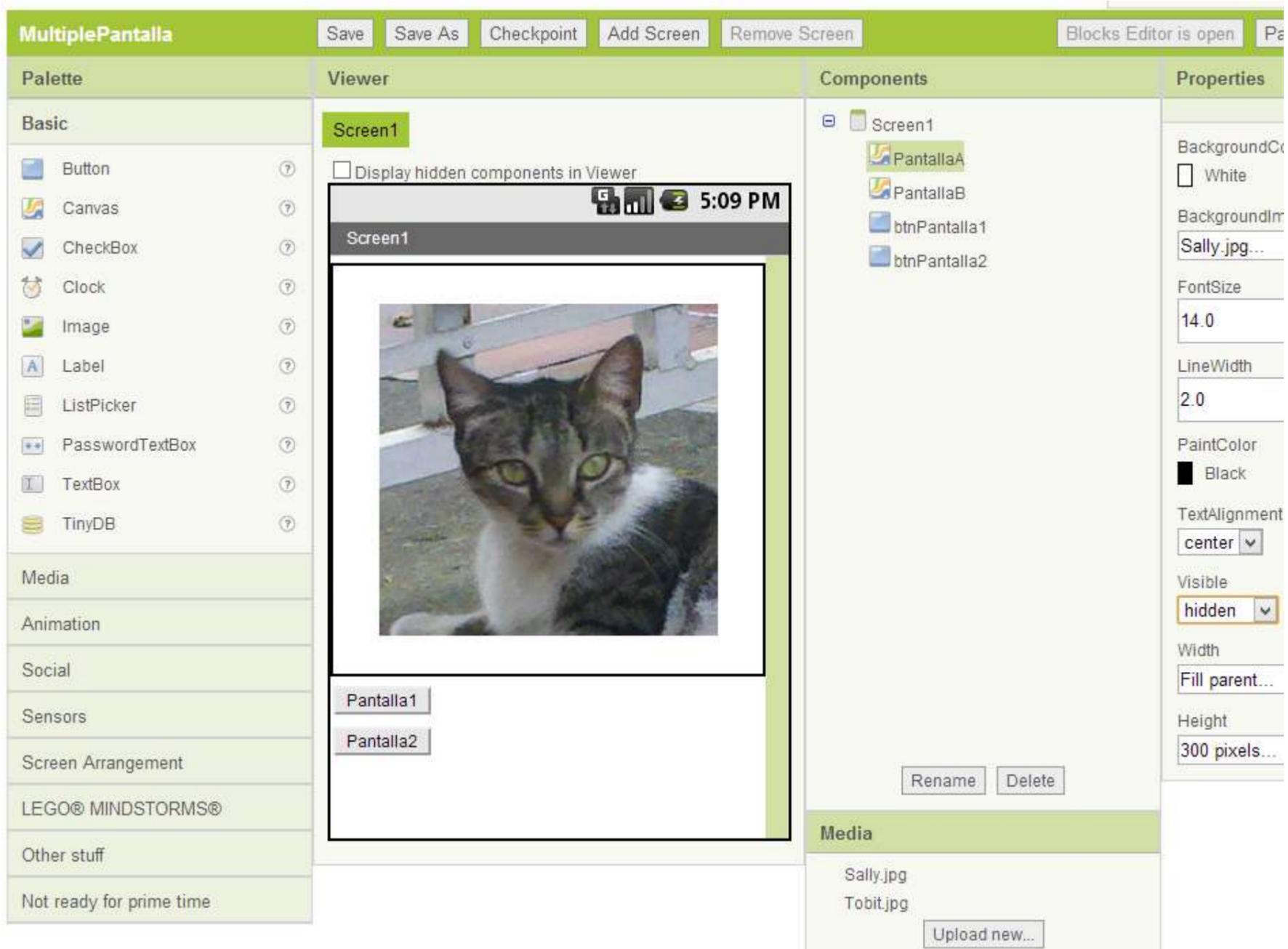


Imagen 181: El primer canvas ya no se observa, esta invisible

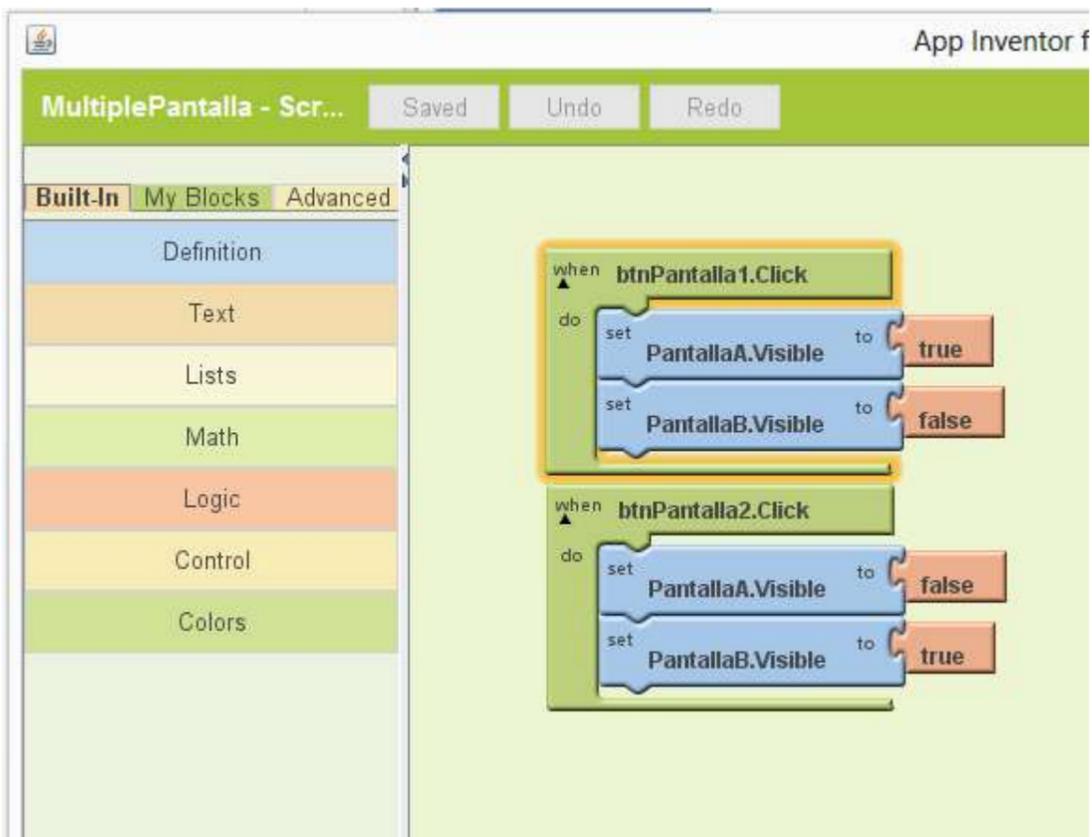
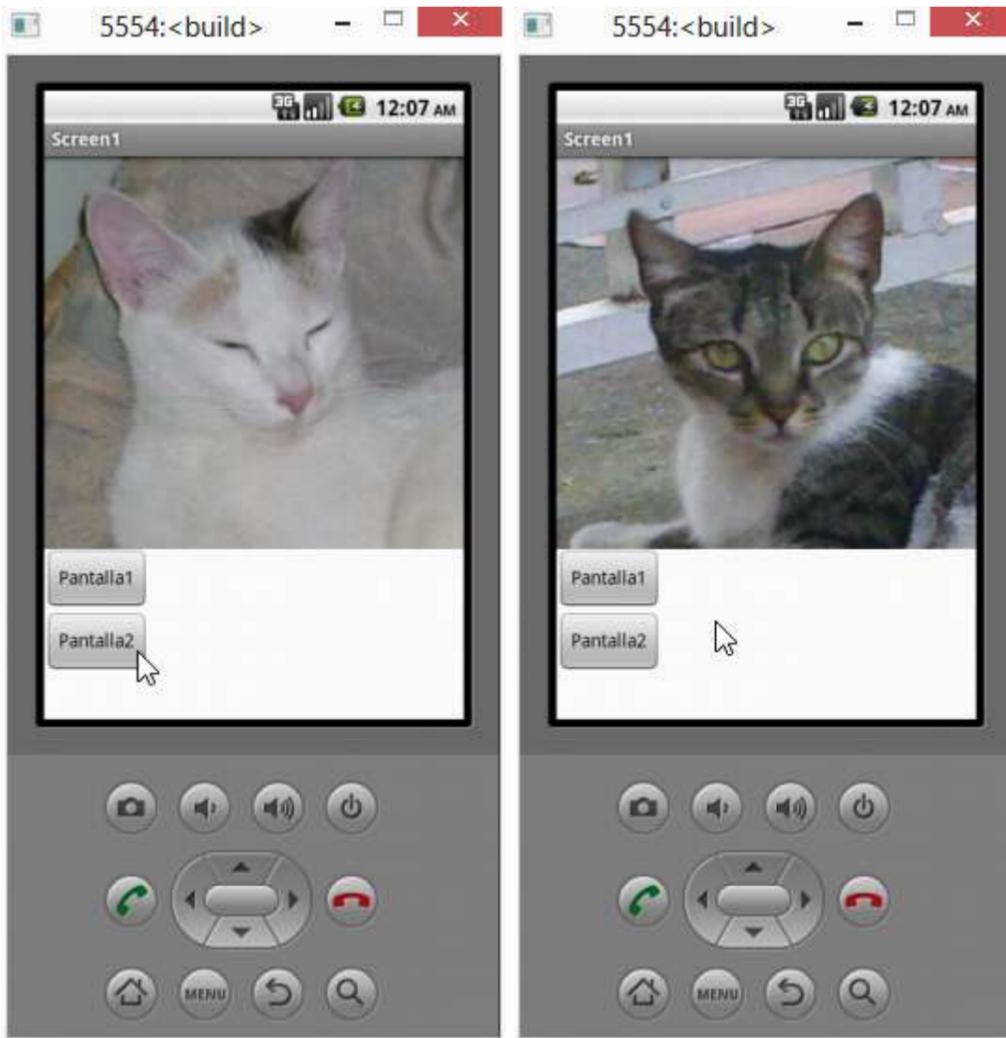


Imagen 182: La programación de los botones es como se muestra

Al ejecutar, se alterna entre un canvas y otro al presionar los botones



Anexo 1. Trabajando fuera de línea

¿Qué hacer en caso que no haya conexión a Internet? Originalmente, MIT App Inventor sólo funciona conectado permanentemente a Internet pero puede llegar el caso que eso no sea posible. En la siguiente dirección, el autor provee una forma de trabajar sin necesidad de estar conectados a Internet.

<http://sourceforge.net/projects/ai4a-configs/>

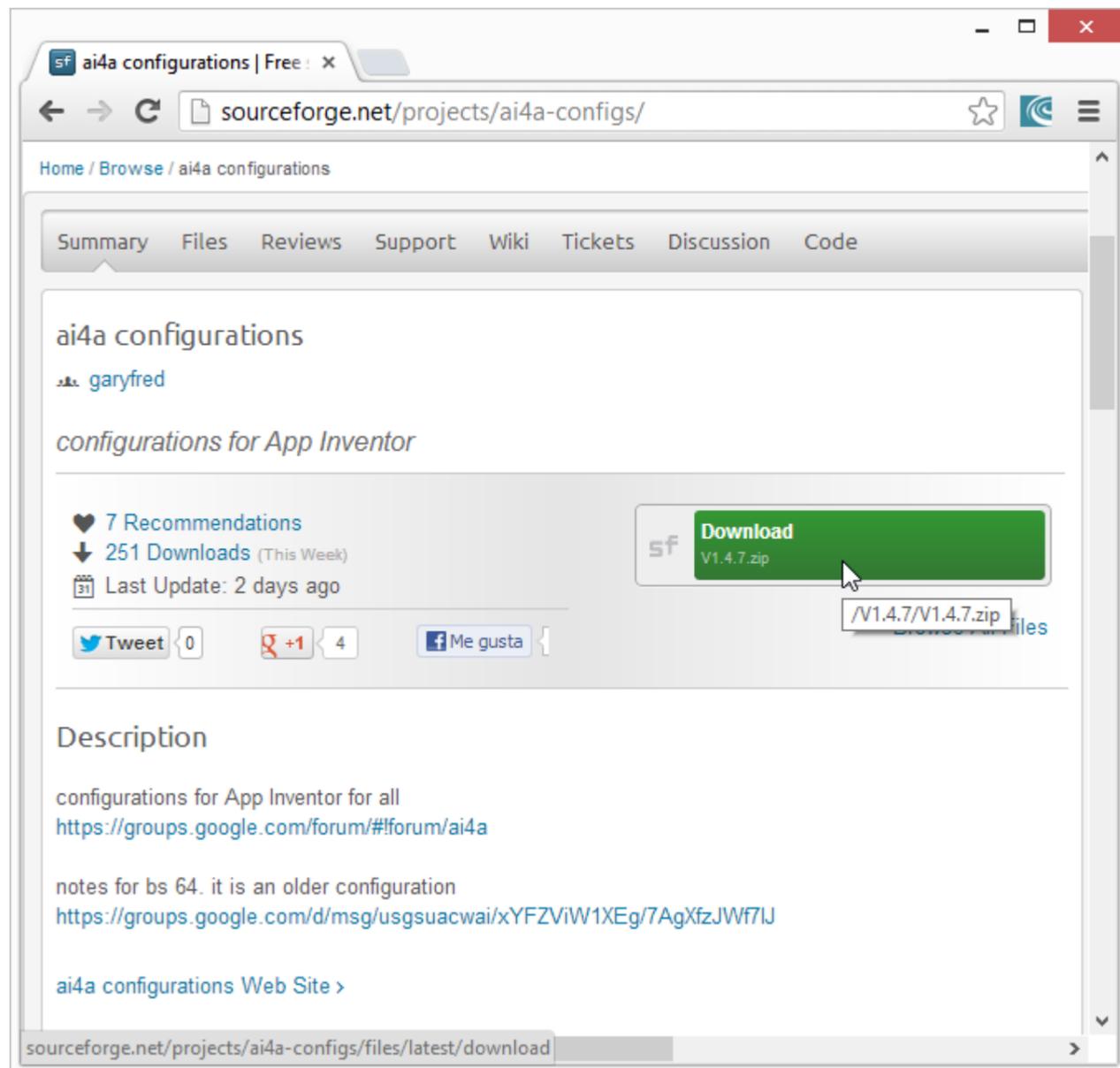


Imagen 183: Paquete requerido para trabajar fuera de línea con MIT App Inventor

Se descarga el paquete y se descomprime

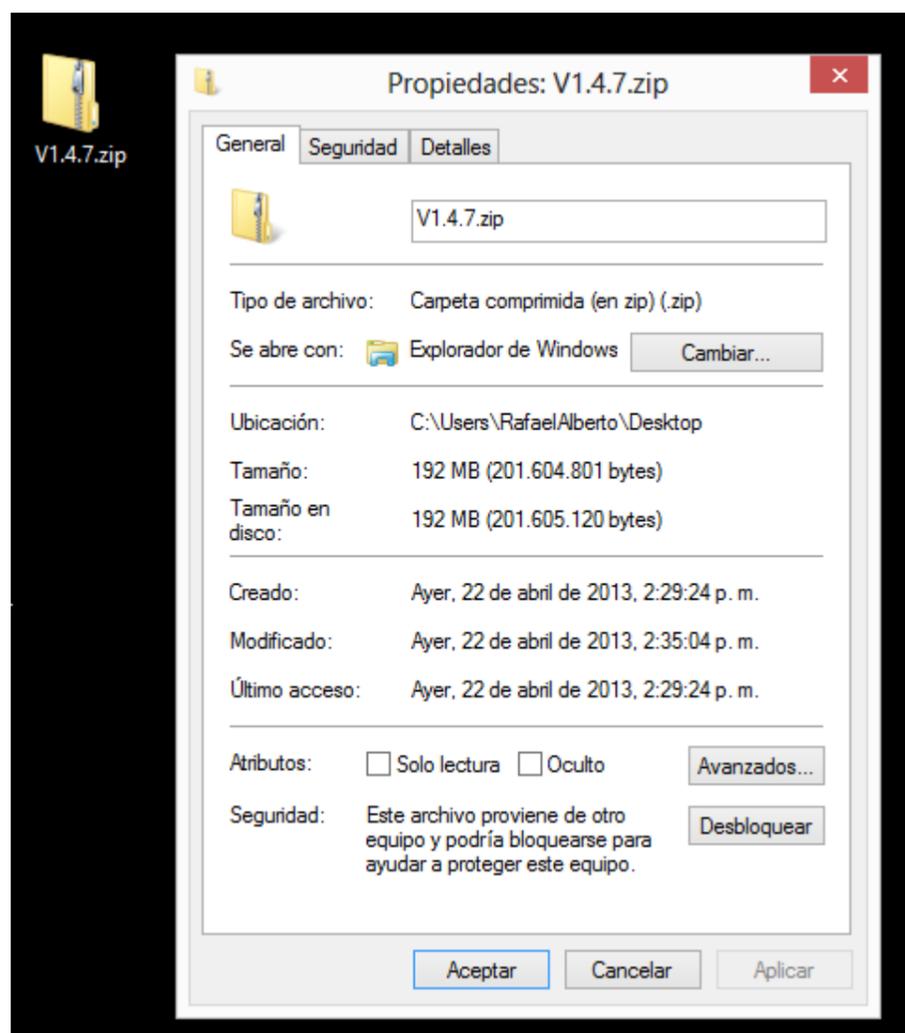


Imagen 184: Paquete descargado

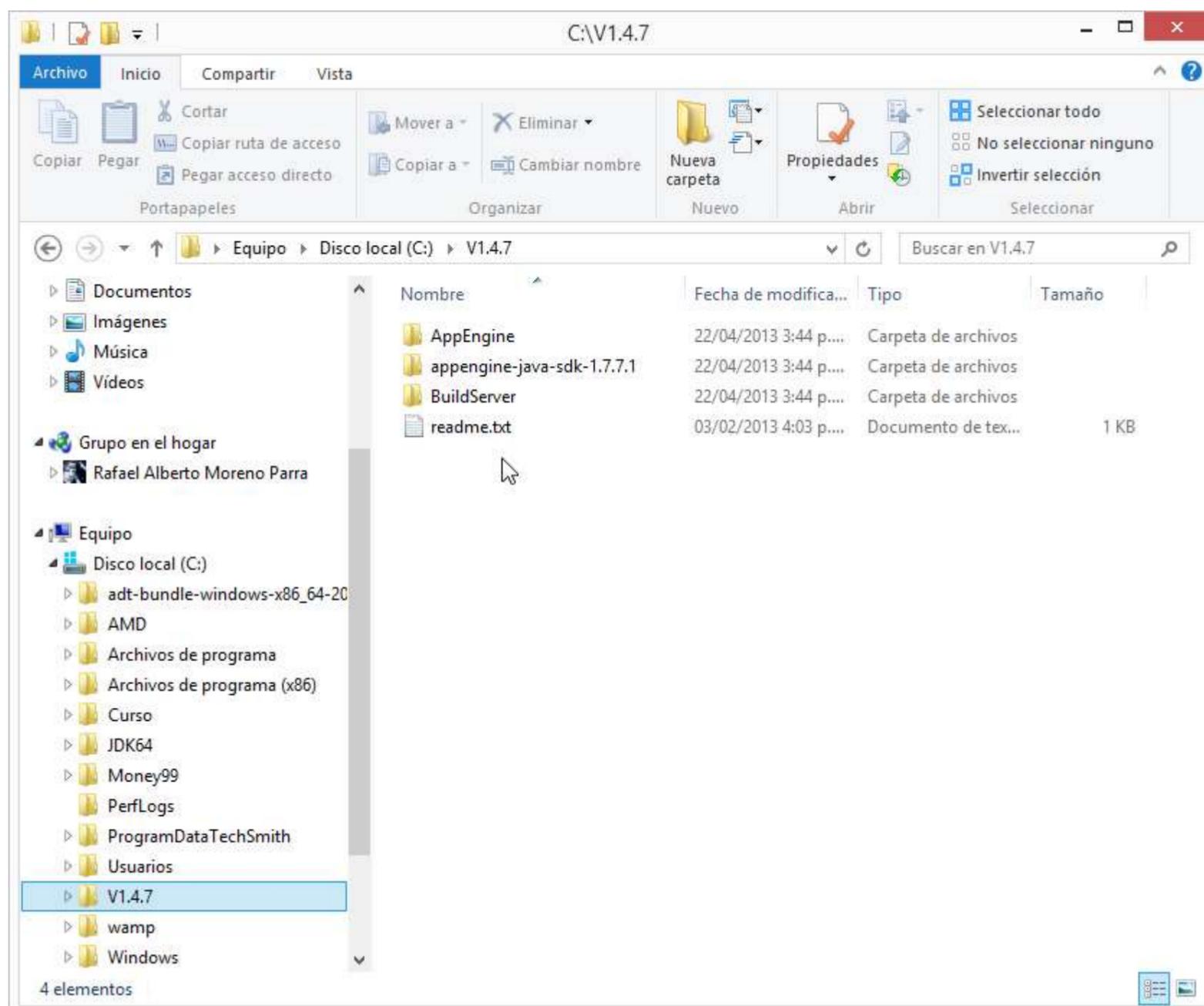


Imagen 185: Paquete descomprimido en C:\

El siguiente paso es ejecutar

C:\V1.4.7\AppEngine\startAI.cmd

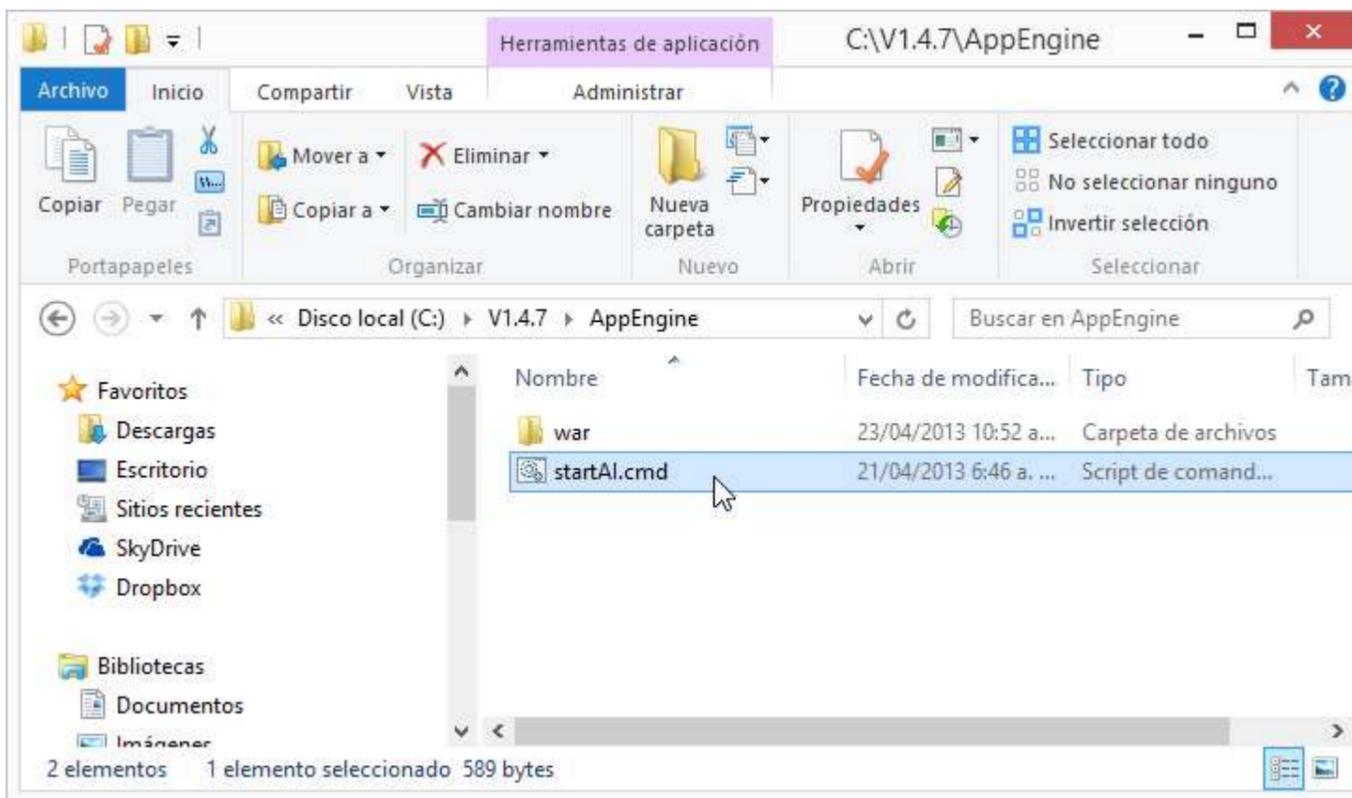


Imagen 186: Ejecutando startAI.cmd



Imagen 187: El "Dev App Server" está ejecutando

Y luego C:\V1.4.7\BuildServer\launch-buildserver.cmd (en una máquina de 64bits) y darle permiso en el FireWall

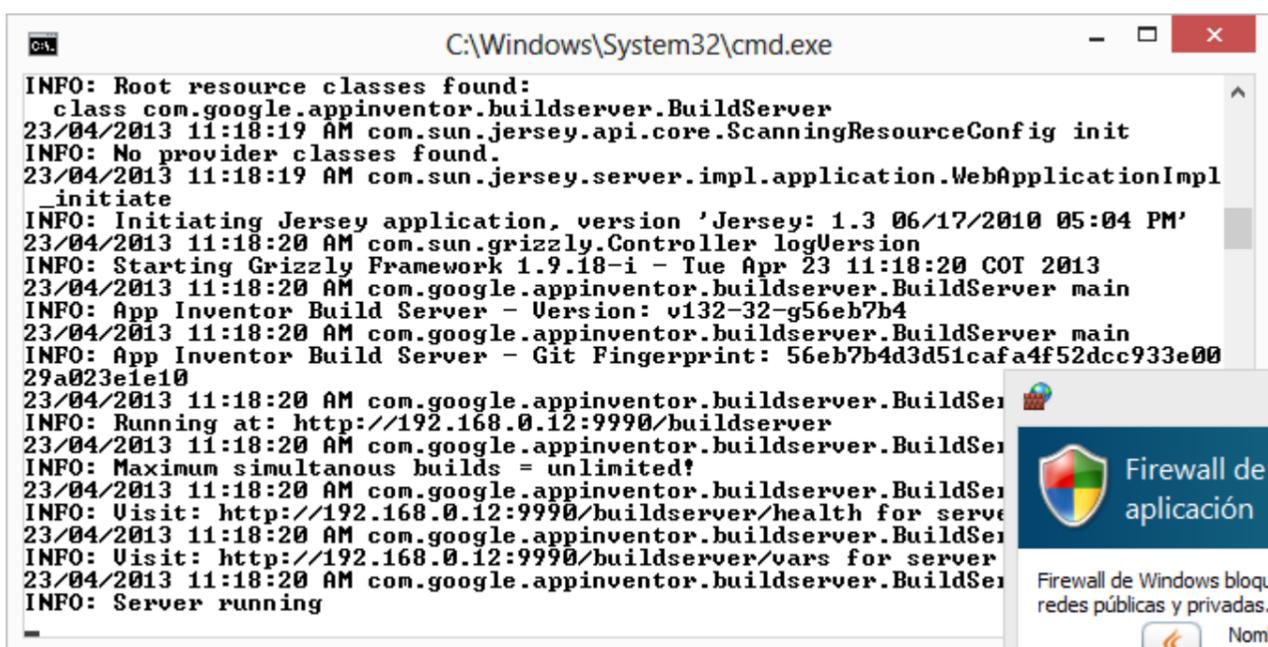


Imagen 188: Ejecución del launch-buildserver.cmd

Con esos dos paquetes corriendo, entonces se abre el navegador en <http://localhost:8888>

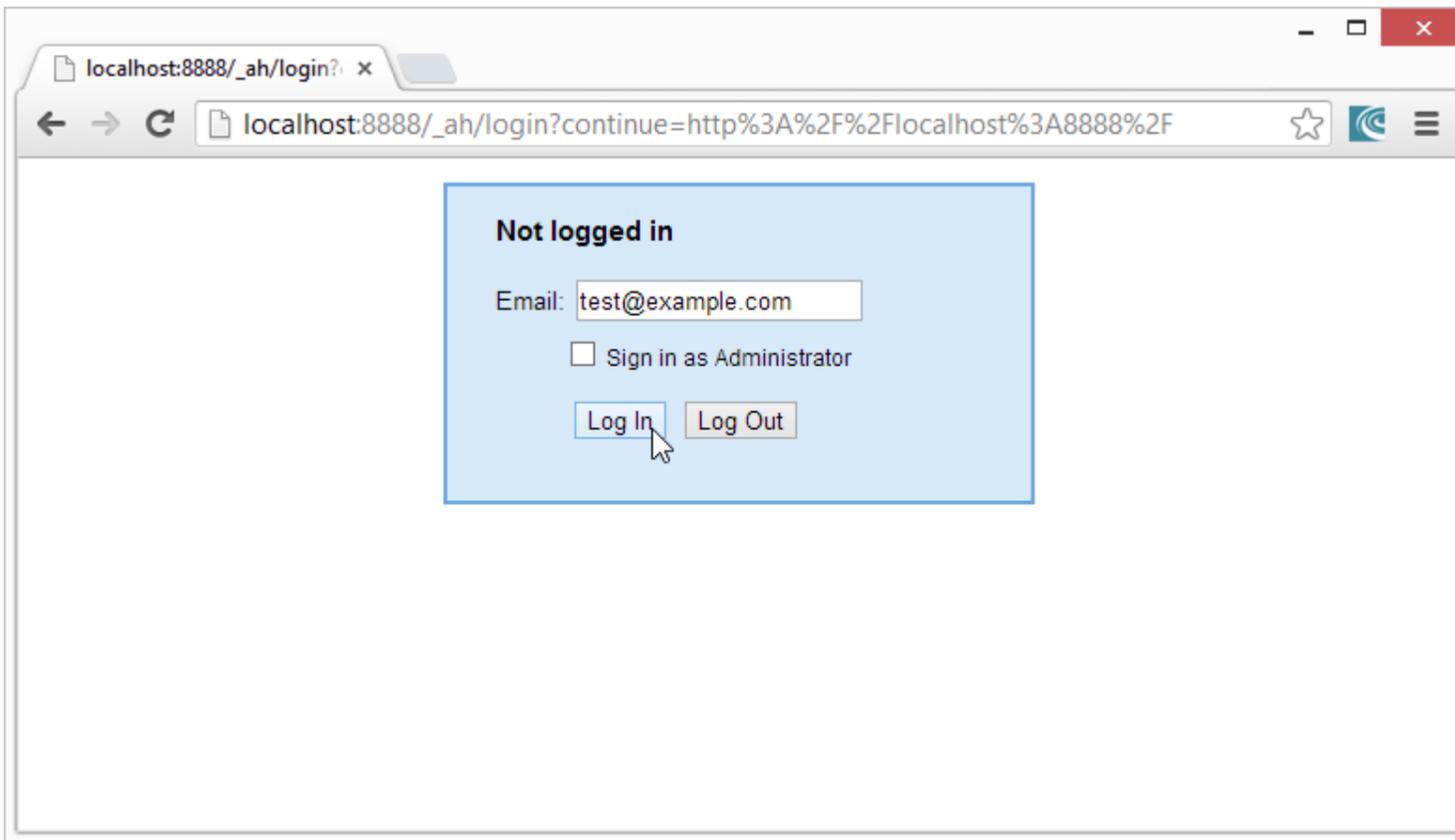


Imagen 189: Entra por el navegador a la dirección <http://localhost:8888>

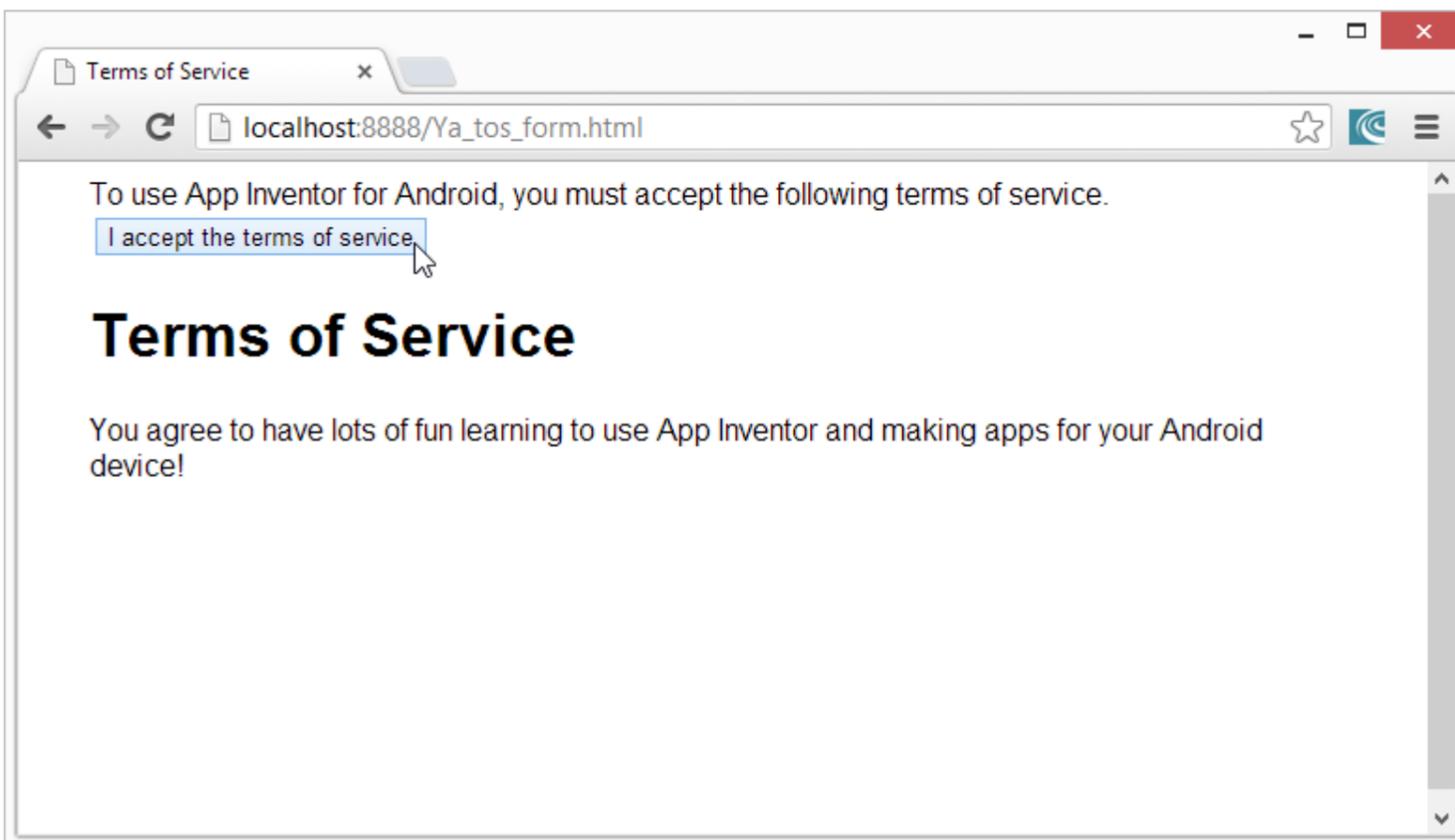


Imagen 190: Acepta los términos del servicio

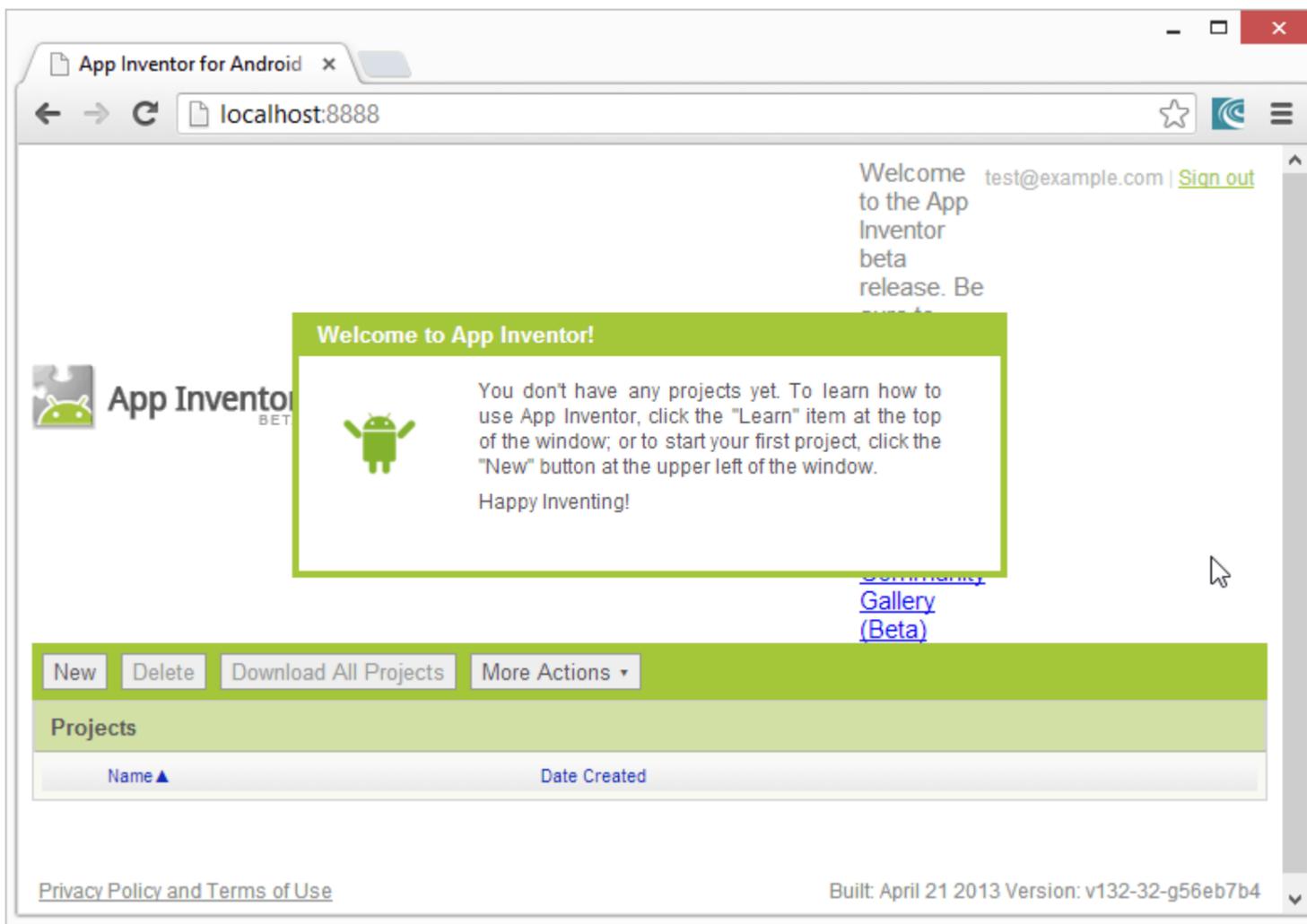


Imagen 191: Bienvenida al MIT App Inventor local

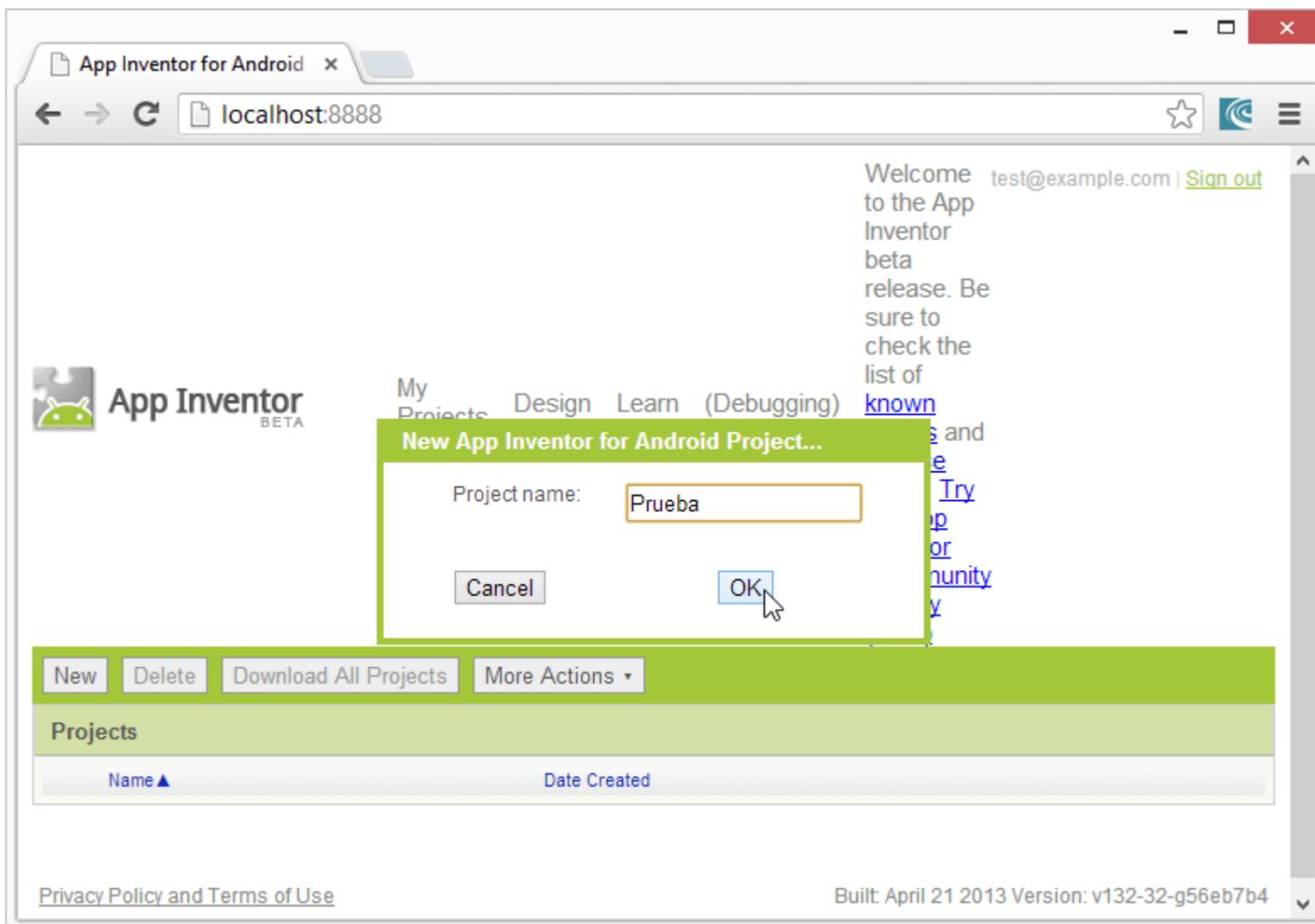


Imagen 192: Se crea un proyecto de prueba

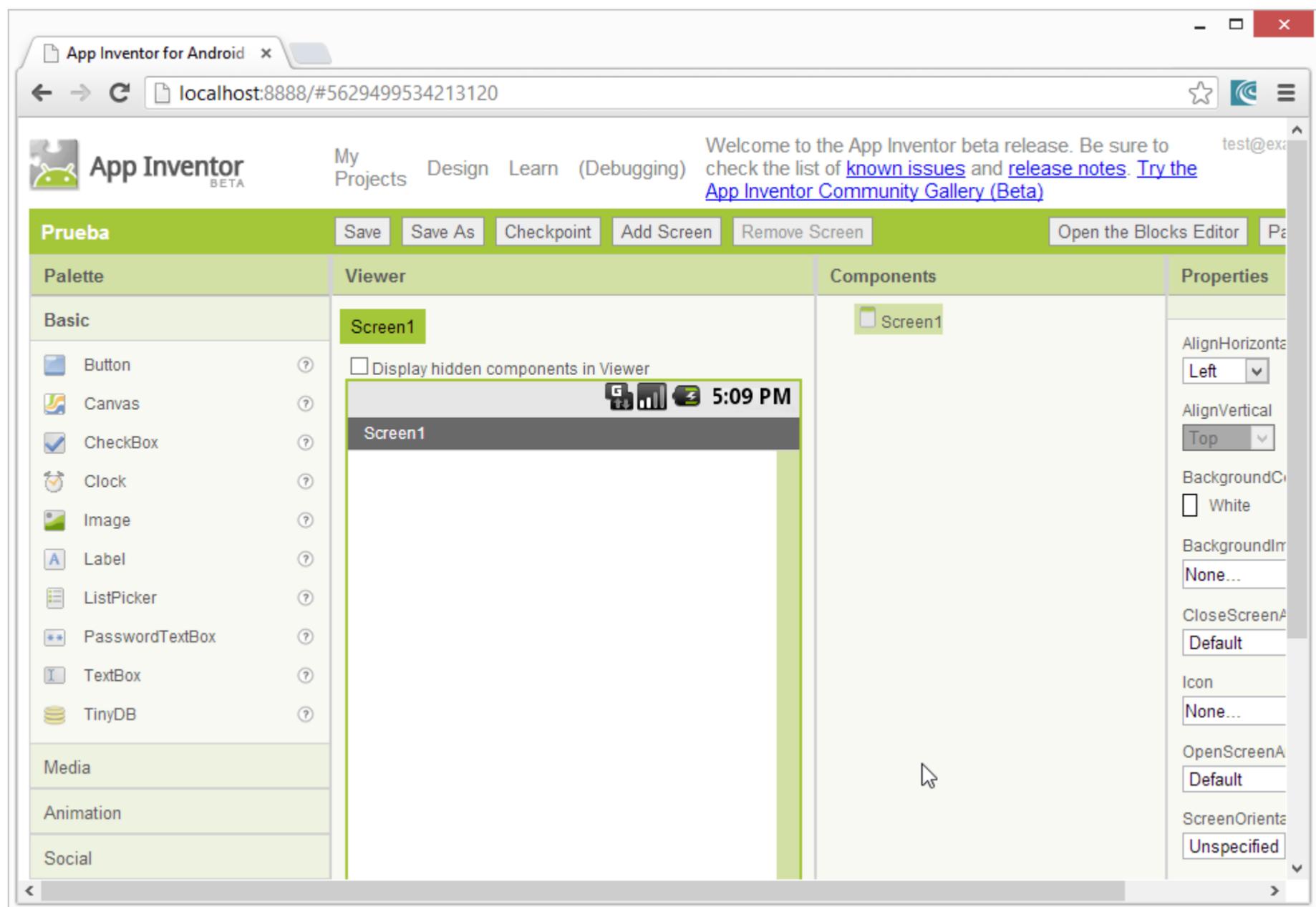


Imagen 193: Entorno gráfico de desarrollo local

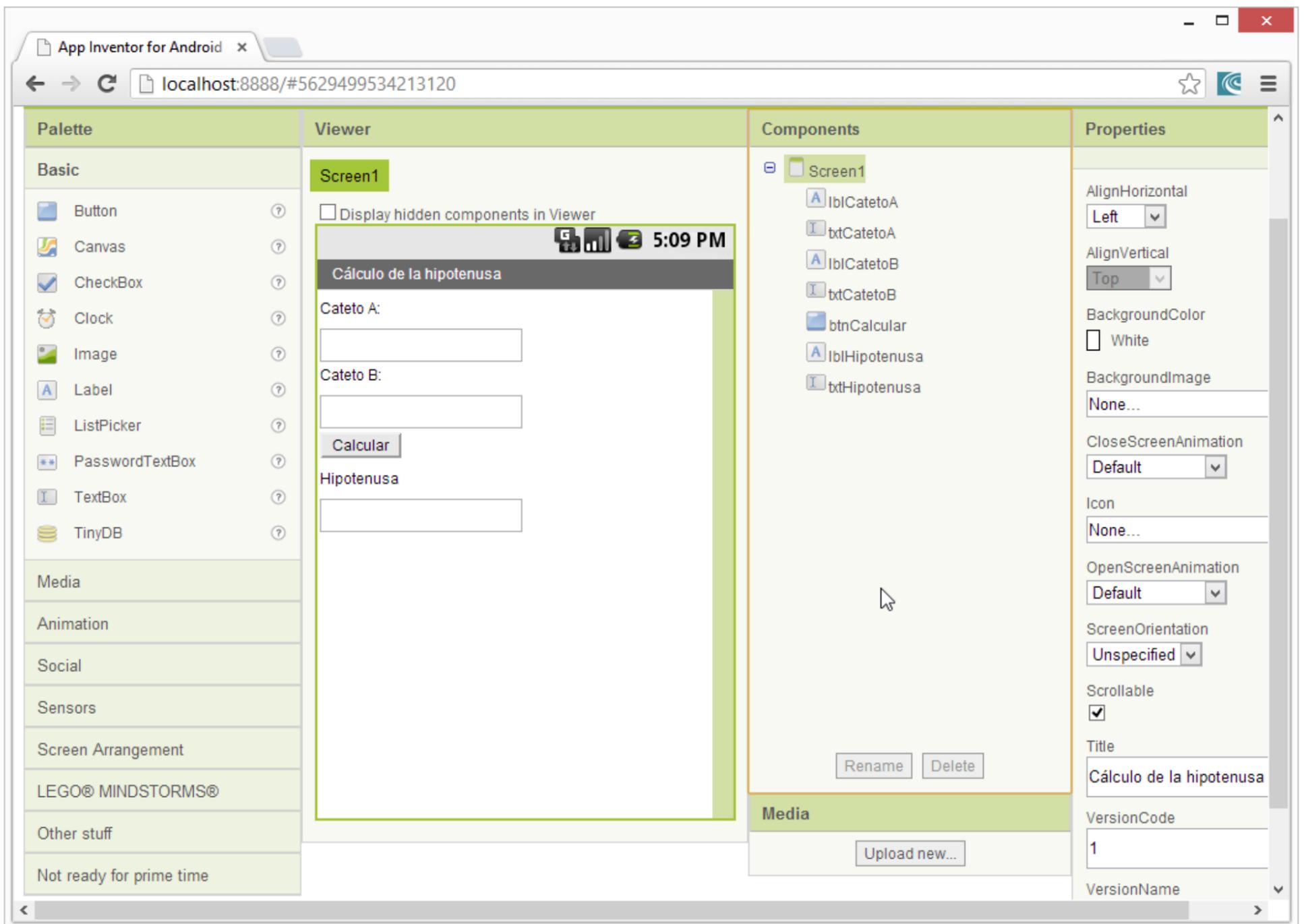


Imagen 194: Genera una pantalla

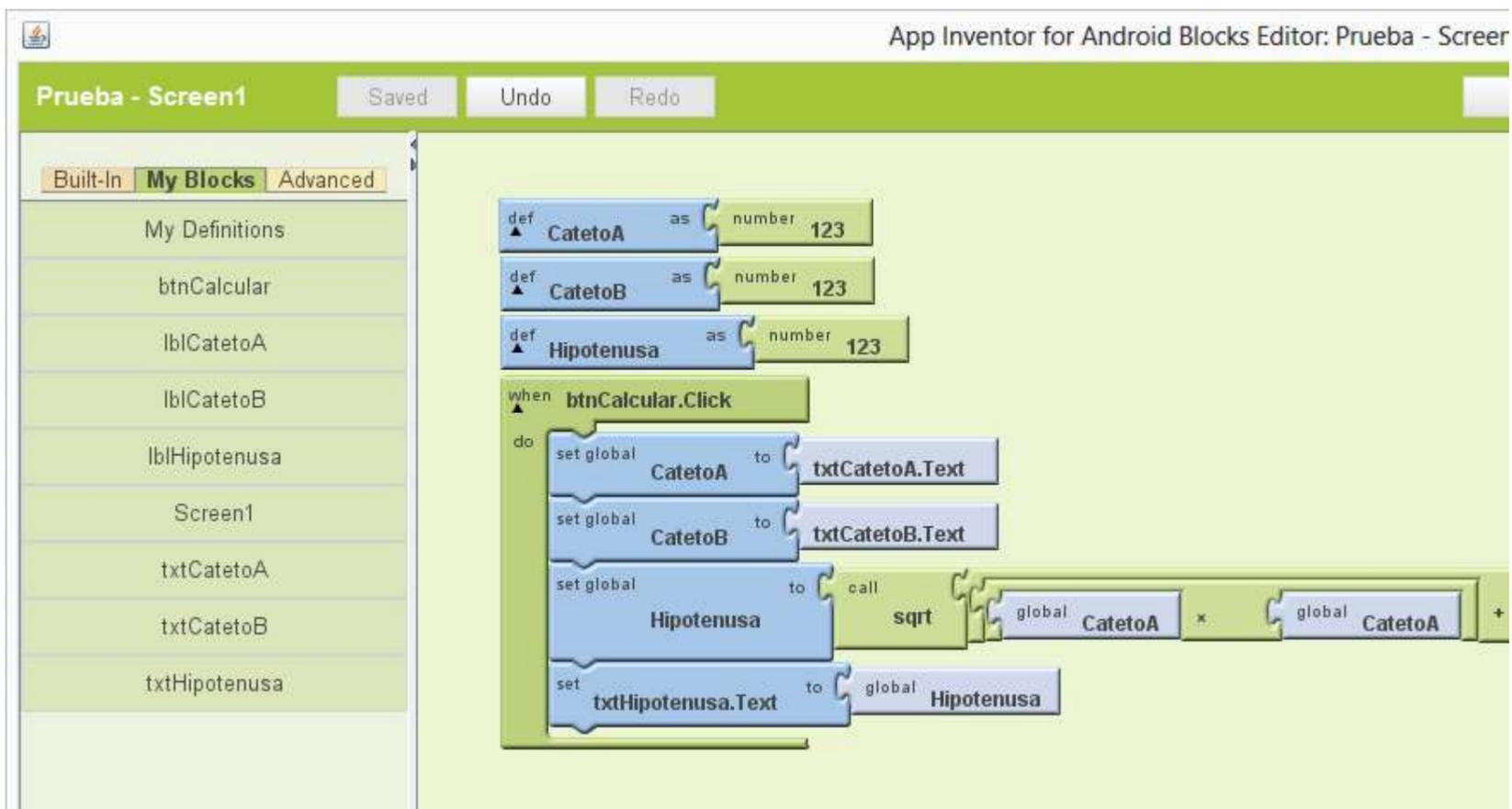


Imagen 195: Ingresa el código por el "Blocks Editor"

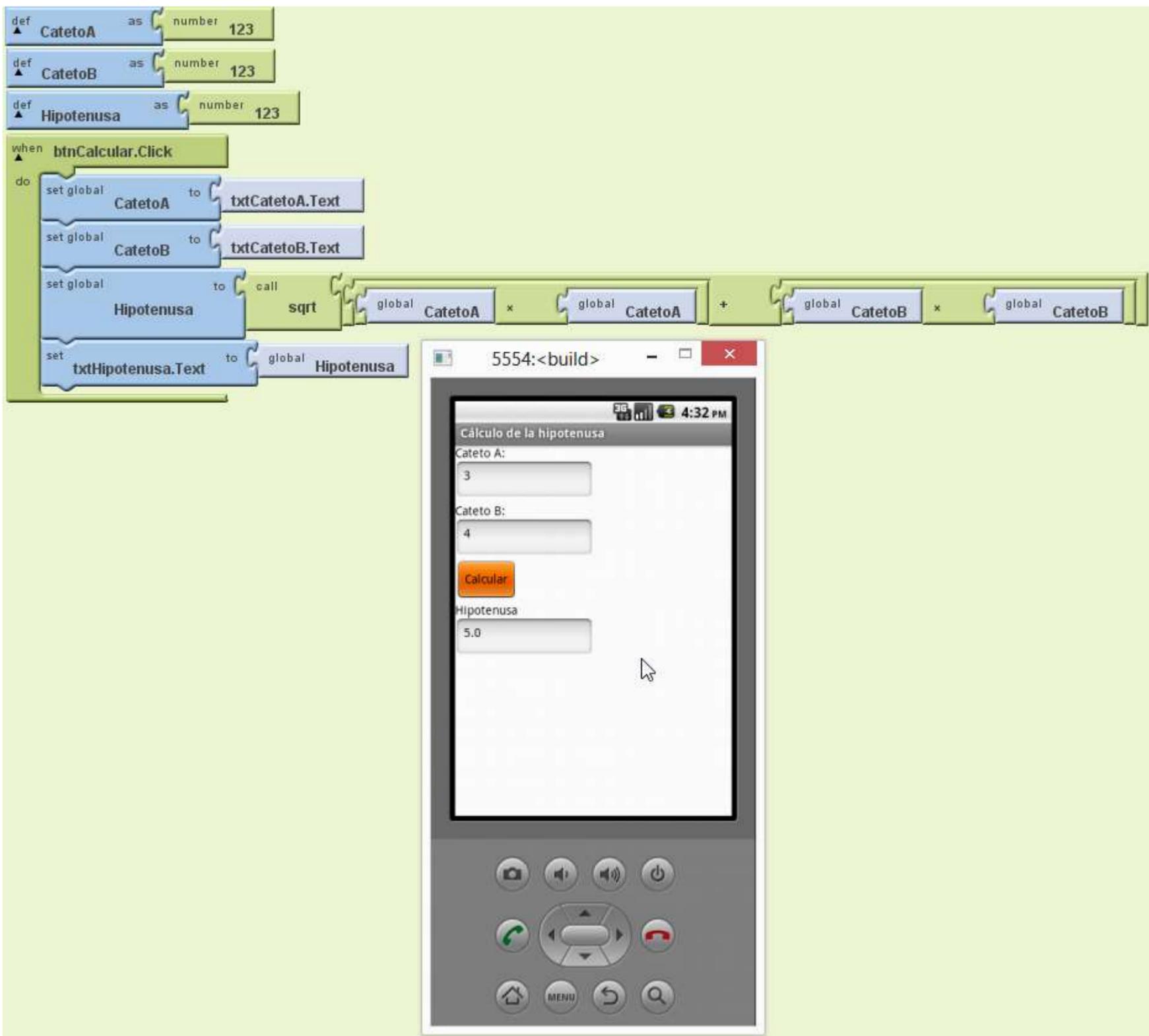


Imagen 196: Ejecuta la aplicación emulada en un smartphone android virtual

Importante sobre la versión fuera de línea

1. Esta versión **NO** funciona si tiene instalado el JDK 7 update 21 (32 bits o 64 bits) en el computador. Se genera un mensaje de error que dice: "Cannot find cached resource for URL" cuando trate de activar el "Blocks Editor"
2. Esta versión funciona correctamente con JDK 6 update 45
3. Debe tener configuradas las variables de sistema JAVA_HOME y CLASSPATH

Información de soporte sobre esta versión fuera de línea en:

<https://groups.google.com/forum/#!topic/ai4a/pqy4NsIPI0E>

<https://groups.google.com/forum/#!forum/ai4a>