Programación de móviles con MIT App Inventor

1. INTRODUCCIÓN

Los españoles pasamos una media de **6 horas al día en Internet**. Casi un **90%** disponemos de WhatsApp. Y es que, en nuestro país, hay más móviles que personas… en concreto, 54’34 millones de móviles, un 116’2 por ciento de la población.

Y no es de extrañar, ya que actualmente los smartphones nos permiten hacer todo tipo de acciones: Pedir comida a domicilio, comunicarnos con los seres queridos, ser nuestra agenda o calendario, conducirnos a cualquier lugar o simplemente, jugar para no aburrirnos…

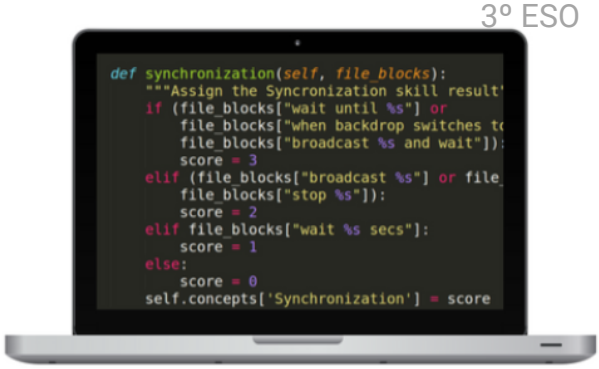
Conscientes de esto, las empresas cada vez apuestan más por crear sus propias APPs ya que es una forma de llegar a muchos clientes. (3,14 millones de APPs para Android y 2,09 millones de APPs para IOS).

Como hemos dicho, existen aplicaciones móviles (APPs) diseñadas para todo tipo de utilidades, pero ¿Sabes cómo han sido creadas esas aplicaciones?, ¿Crees que tu podrías crear una propia?

1.1. Lenguajes de programación de APPs para móviles

Para crear aplicaciones para teléfonos móviles se utilizan lenguajes de programación desde un ordenador, los más usados actualmente son los siguientes:

- **Java**: Es el principal lenguaje de programación de de aplicaciones para dispositivos móviles, por su rapidez, sencillez y versatilidad, por ejemplo, Android, Twitter, Netflix o Uber fueron creados con este lenguaje.
- **Kotlin**: Es uno de los lenguajes de programación para dispositivos móviles Android más empleados; dispone de un código muy intuitivo, sencillo y eficaz.
- **Python**: Las aplicaciones móviles con Python destacan por su código. Python facilita el trabajo a los programadores o ingenieros informáticos, ya que se utilizan menos líneas de código que en el caso de Java, por ejemplo.
- **JavaScript**: Si estás pensando en crear una app multiplataforma, JavaScript puede ser el lenguaje de programación para móvil que estés buscando. Es rápido, versátil, sencillo y destaca por su funcionalidad.
- **Swift**: Asimismo, un ejemplo de lenguaje de programación para Iphone es, sin duda, Swift. Este lenguaje creado por la gran manzana Apple ya no solo opera con apps para iOS, sino también sirve para Windows, Linux o macOS.



Lenguaje de programación gráfico (Por bloques)

Lenguaje de programación de tipo texto

Todos estos lenguajes de programación son **lenguajes de texto**, en los que hay que aprender una serie de palabras clave y una sintaxis. Su aprendizaje no es sencillo y requiere de mucho tiempo, por eso se enseña en estudios superiores de programación.

Existen otra clase de **lenguajes de programación gráficos** cuya curva de aprendizaje es más sencilla y por lo tanto son más adecuados para la enseñanza secundaria, como son:

- Tu-App
- Infinite Monkeys
- Mobapp Creator
- Upplicación
- Good Barber
- MIT App Inventor

En este curso vamos a emplear **Mit App** Inventor por ser un lenguaje por bloques, fácil de aprender y gratuito.

(Ya aprendimos el curso pasado la programación mediante bloques con Scratch y Micro:Bit).

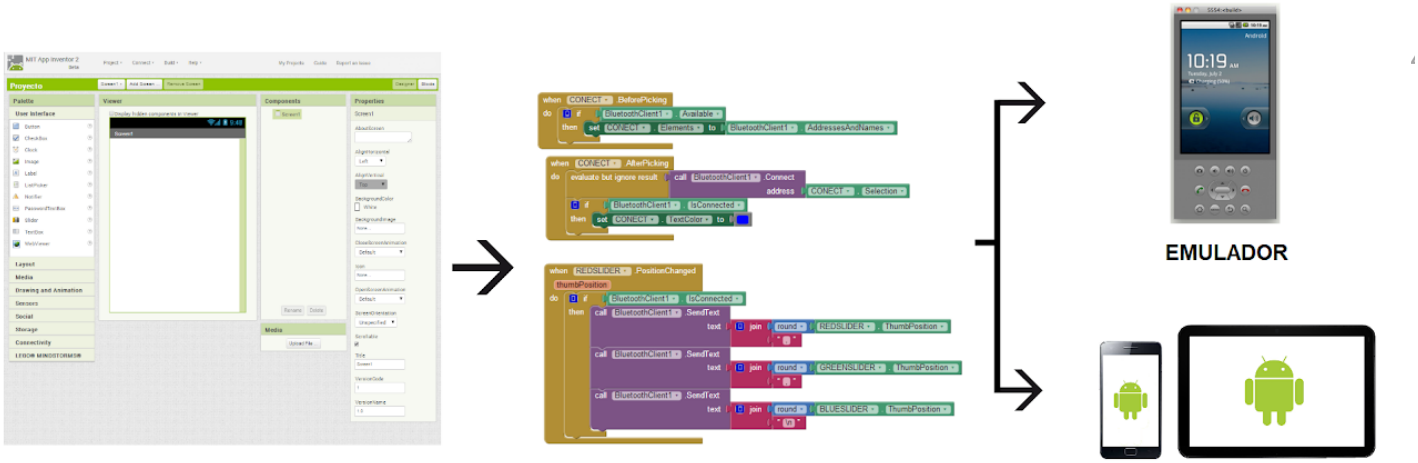
2. INTRODUCCIÓN A MIT APP INVENTOR

MIT App Inventor es una **herramienta web** de **programación gráfica** con la que se pueden crear APPs para dispositivos **Android**. Con App Inventor no es necesario tener unos extensos conocimientos sobre sintaxis de programación, ya que mediante la unión de bloques se pueden crear las instrucciones necesarias para crear las Apps, creando desde aplicaciones muy simples hasta aplicaciones mucho mas complejas.

App Inventor es un producto desarrollado por el **Instituto Tecnológico de Massachusetts** (MIT) junto con **Google**, de manera que como requisito se debe disponer de una cuenta de Google para acceder a la web de MIT App Inventor (Nos sirve la cuenta de correo del Instituto).

Para crear una APP con MIT App inventor hay que realizar tres pasos:

- El **diseño** «*estetico*» de la aplicación , en la que se seleccionan los componentes para su aplicación(botones, cajas de texto ,imágenes, etc).
- El **editor de bloques** , donde se escogen los bloques lógico que sean necesarios según la aplicación que tenga pensada de hacer.
- **Instalación** de la APP en un dispositivo o en un emulador, para probarla.

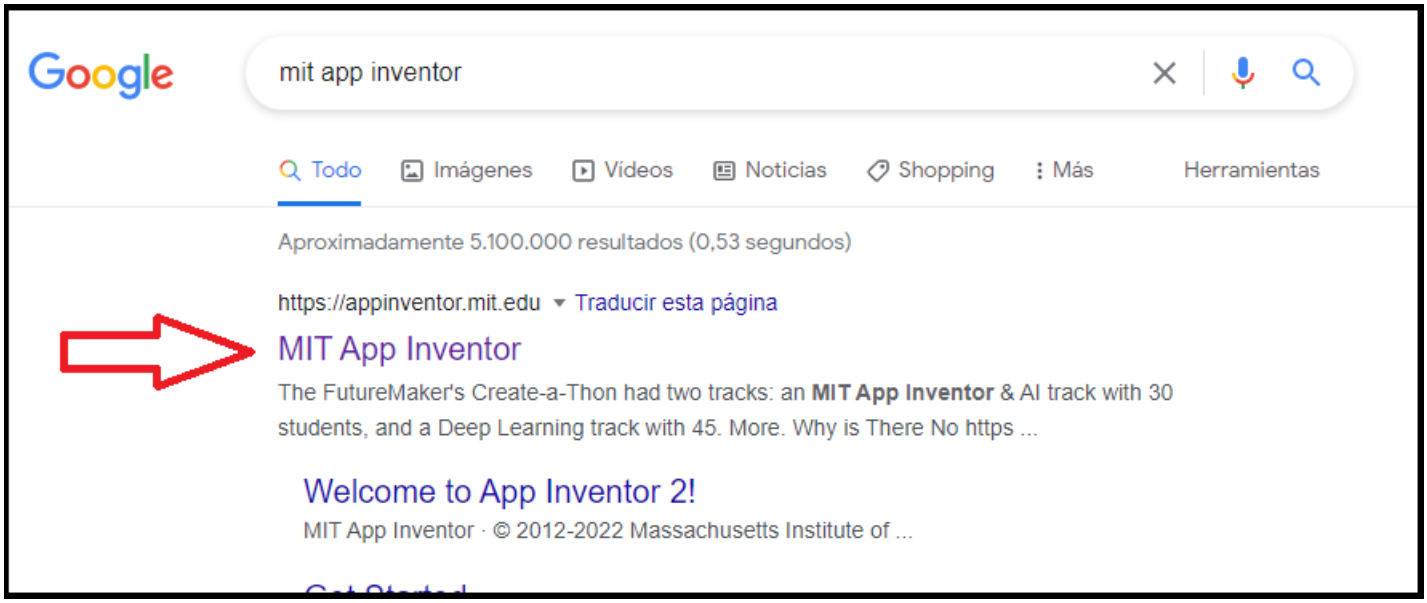


Si nos registramos en Google como **desarrollador**, podemos subir nuestras APPs a Google Play y obtener dinero por ellas (Pagando 25 € por cada APP que publiques).

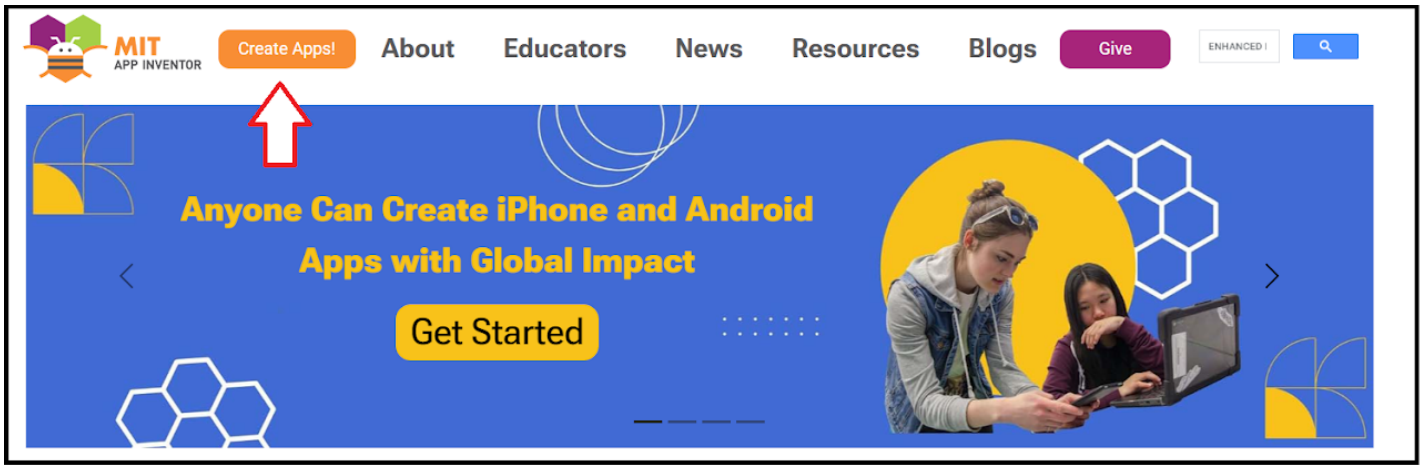
2.1. Primeros pasos en MIT App Inventor

Lo primero que tenemos que hacer es entrar en la página de **MIT App Inventor** para registrarnos con nuestro correo del instituto:

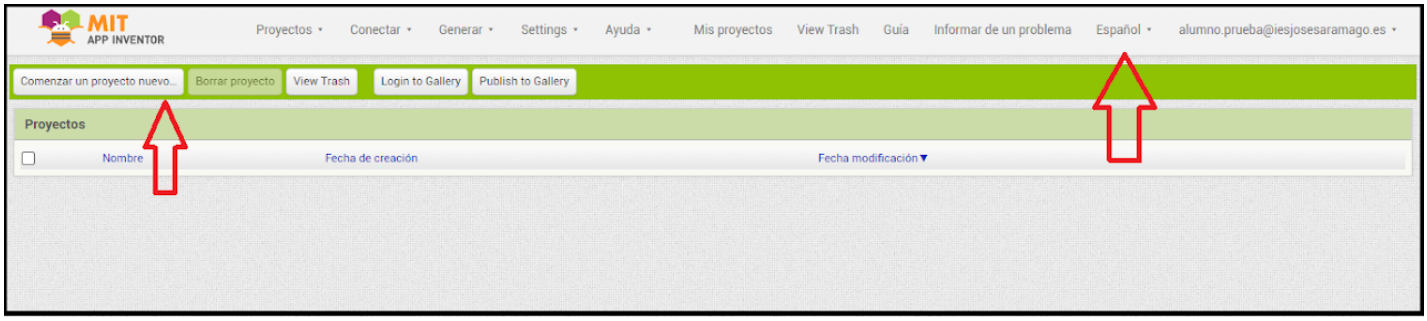
Para ello tecleamos "**mit app inventor**" en la barra de búsqueda de Google y elegimos la primera entrada:



Accedemos a la web de Mit App inventor y pulsamos en el botón naranja que pone "**Create Apps!**"



Aceptamos los **términos de servicio**, cerramos las **ventanas de bienvenida** y cambiamos el idioma a **Español** (No nos lo traduce todo).

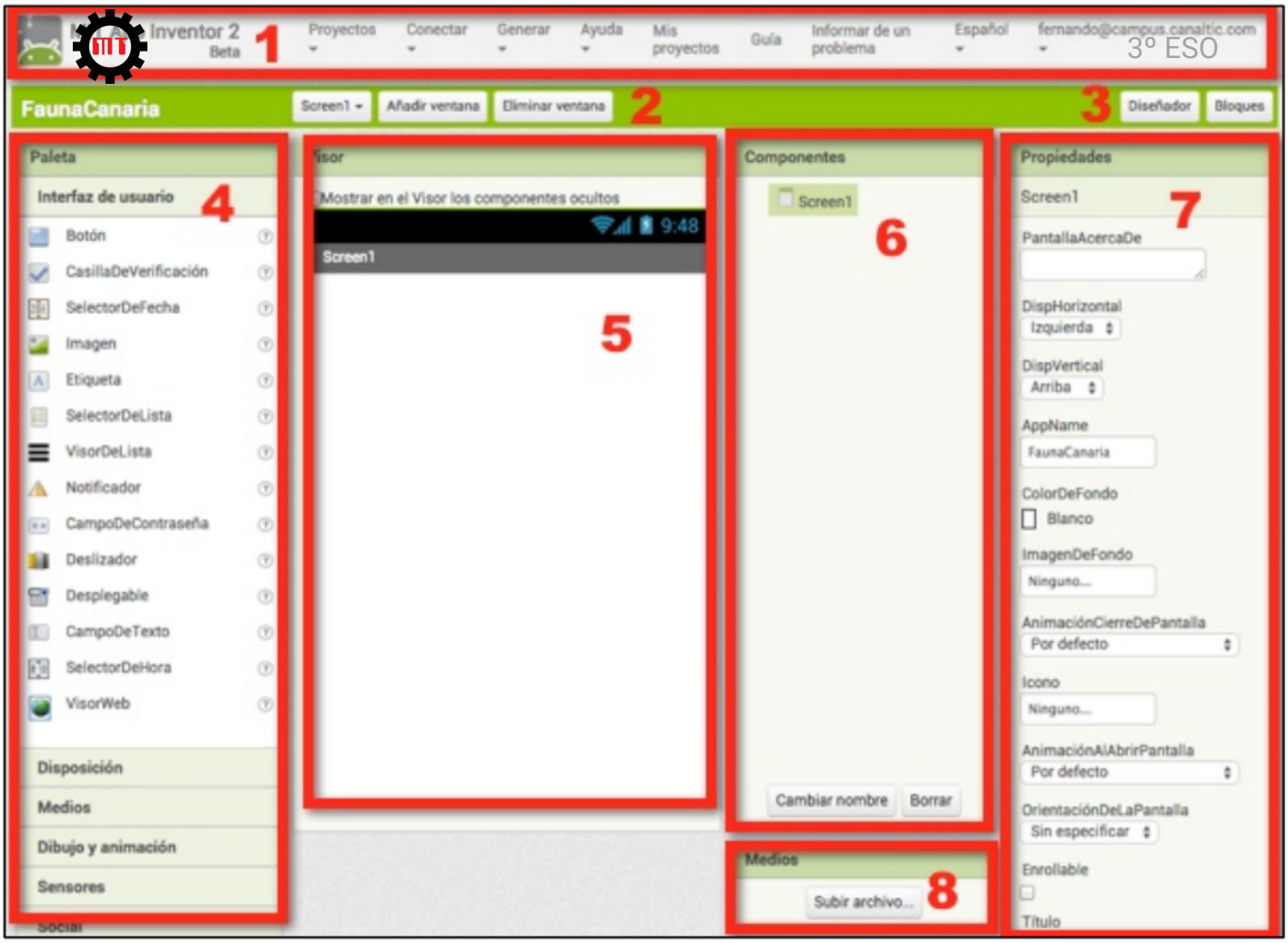


☝️

Esta es la pantalla donde tendremos todos los **proyectos** (APPs) que creemos (Al principio está vacía).

Para crear nuestro primer proyecto, pulsamos en el botón "**Comenzar un nuevo proyecto**".

Nos pide el nombre de nuestro nuevo proyecto (Por ejemplo "**Primer proyecto**") y pulsamos en "**Aceptar**" 👉


Inventor 2
Beta
1
Proyectos
Conectar
Generar
Ayuda
Mis proyectos
Guía
Informar de un problema
Español
fernando@campus.canaltic.com
FaunaCanaria
Screen1
Añadir ventana
Eliminar ventana
2
3
Diseñador
Bloques
Paleta
Interfaz de usuario
4
Botón
CasillaDeVerificación
SelectorDeFecha
Imagen
Etiqueta
SelectorDeLista
VisorDeLista
Notificador
CampoDeContraseña
Deslizador
Desplegable
CampoDeTexto
SelectorDeHora
VisorWeb
Disposición
Medios
Dibujo y animación
Sensores
Social
Visor
Mostrar en el Visor los componentes ocultos
9:48
Screen1
5
Componentes
Screen1
6
Cambiar nombre
Borrar
Medios
Subir archivo...
8
Propiedades
Screen1
7
PantallaAcercaDe
DispHorizontal
Izquierda
DispVertical
Arriba
AppName
FaunaCanaria
ColorDeFondo
Blanco
ImagenDeFondo
Ninguno...
AnimaciónCierreDePantalla
Por defecto
Icono
Ninguno...
AnimaciónAlAbrirPantalla
Por defecto
OrientaciónDeLaPantalla
Sin especificar
Enrollable
Título

Entramos en la pantalla de diseño de Mit App inventor, en la que podemos distinguir la siguientes zonas:

1. **Barra de menús**: Desde esta barra podemos acceder al gestor de proyectos, conectar con el emulador o dispositivo, generar el instalador APK, etc.
2. **Gestor de pantallas**: Con estos botones podemos crear o eliminar pantallas en caso de APPs con varias pantallas.
3. **Diseñador/Bloques**: Mediante estos botones podemos alternar entre los entornos del **Diseñador** y el **Editor de bloques**.
4. **Paleta de componentes**: Estos son los componentes ordenados por **categorías**, que se pueden añadir a la pantalla de la aplicación **arrastrándolas.** Se pueden añadir más componentes mediante extensiones.
5. **Visor**: Muestra una **vista previa** de la aplicación en la pantalla del dispositivo (Podemos elegir el tipo de dispositivo).
6. **Componentes**: Mediante una estructura de árbol cuyo nodo principal es el componente Screen (Pantalla) permite acceder a los distintos elementos que hemos situado en esa pantalla. Si pulsamos sobre cualquier componente, podremos configurarlo en el panel de propiedades.
7. **Propiedade**s: Permite definir los valores de los distintos parámetros del componente seleccionado en el panel de Componentes. Si seleccionamos el parámetro **Screen1**, podemos configurar aspectos generales de la APP como su nombre, icono, etc.
8. **Medios**: Desde este botón podemos subir archivos para nuestra APP, como imágenes, sonidos, etc.

MIT App Inventor.

1. **Barra de menús**: (izquierda): Nos permite crear o borrar proyectos, descargar la App y probarla, y también dispone de un menú de ayuda, donde podemos encontrar tutoriales, foros…
2. **Barra de menús** (derecha): Desde ella podemos ver todos nuestros proyectos, cambiar o cerrar nuestra sesión, ver nuestra galería…
3. **Gestor de pantalla**: Nos sirve para administrar nuestras ventanas y trabajar con ellas
4. **Diseñador/ Bloques**: Botones que nos permite cambiar a Diseñador o a Bloques según nuestras necesidades.
5. **Bloques**: Nos permite elegir las funciones que queremos usar. Nos podemos encontrar con bloques de: Control, Lógica, Matemáticas, Texto, Listas, Colores, Variables y Procedimientos. También nos encontramos con los componentes de la pestaña Diseñador, y esos componentes también tendrán sus propias funciones.
6. **Vista**: Aquí podemos ver nuestros bloques y unirlos según nuestro criterio.
7. **Mostrar avisos**: Sirve para saber cuando hay un error o incompatibilidad entre bloques, para poder cambiarlo.
8. **Mochila**: Nos permite guardar un conjunto de bloques para usarlo en algún otro proyecto.
9. **Aumentar/ Disminuir**: Con estos botones podemos aumentar o disminuir el tamaño con el que vemos los componentes.
10. **Papelera**: Permite borrar los bloques que no usemos.

2.1. Programación orientada a eventos

MIT App inventor funciona mediante la **programación orientada a eventos**. En la programación orientada a eventos, el dispositivo no lleva a cabo una serie de operaciones en orden, sino que está esperando a que se active un **evento**, en cuyo caso ejecuta la **respuesta** programada para ese evento, por ejemplo, si el usuario hace clic en un botón, la aplicación hace una foto con la cámara del dispositivo.

El trabajo del programador consiste en diseñar las respuestas a los distintos tipos de eventos para que la aplicación haga lo que queremos.

En el ejemplo anterior, el bloque marrón es el **controlador de eventos** y el bloque violeta es la **respuesta al evento**:

Cuando el **Botón1** sea pulsado -> Toma una foto con la **Cámara1**

Los eventos pueden ser divididos en 2 tipos diferentes: **automáticos** e **iniciados por el usuario**. Hacer clic en un botón , tocar o arrastrar en la pantalla, inclinar el teléfono son eventos iniciados por el usuario. Cuando en un juego choca una bola contra el borde, un temporizador, cuando llega un mensaje, son eventos no iniciados por el usuario.

Actividades (1)

Rellena el siguiente formulario sobre los contenidos vistos en la página:

https://docs.google.com/forms/d/e/1FAIpQLScR5BfV_83lqs5eX_K-Fz00ZtK4I_ASvsc66YhJ5YM2tC3XQA/viewform?usp=sf_link

3. PROGRAMACION CON MIT APP INVENTOR

Para comprender el funcionamiento de MIT App Inventor, vamos a realizar un ejercicio completo guiado con el que vamos a instalar nuestra primera App en nuestro dispositivo.

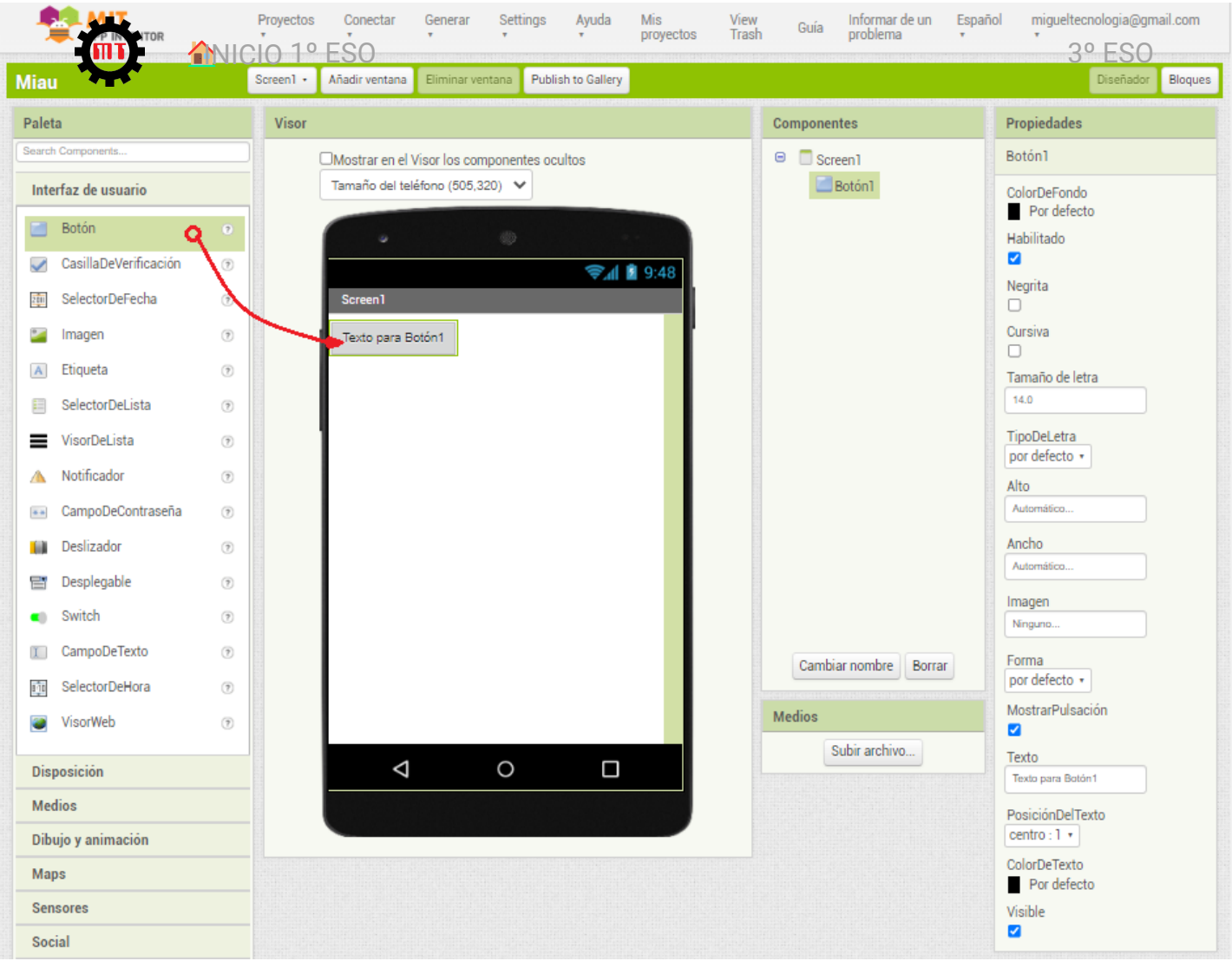
La App va a mostrar la foto de un gato en nuestra pantalla, que va a maullar cada vez que lo toquemos.

1º Antes de nada, descarga en el ordenador estos dos archivos que vamos a necesitas:

- **Kitty.png**
- **Miau.mp3**

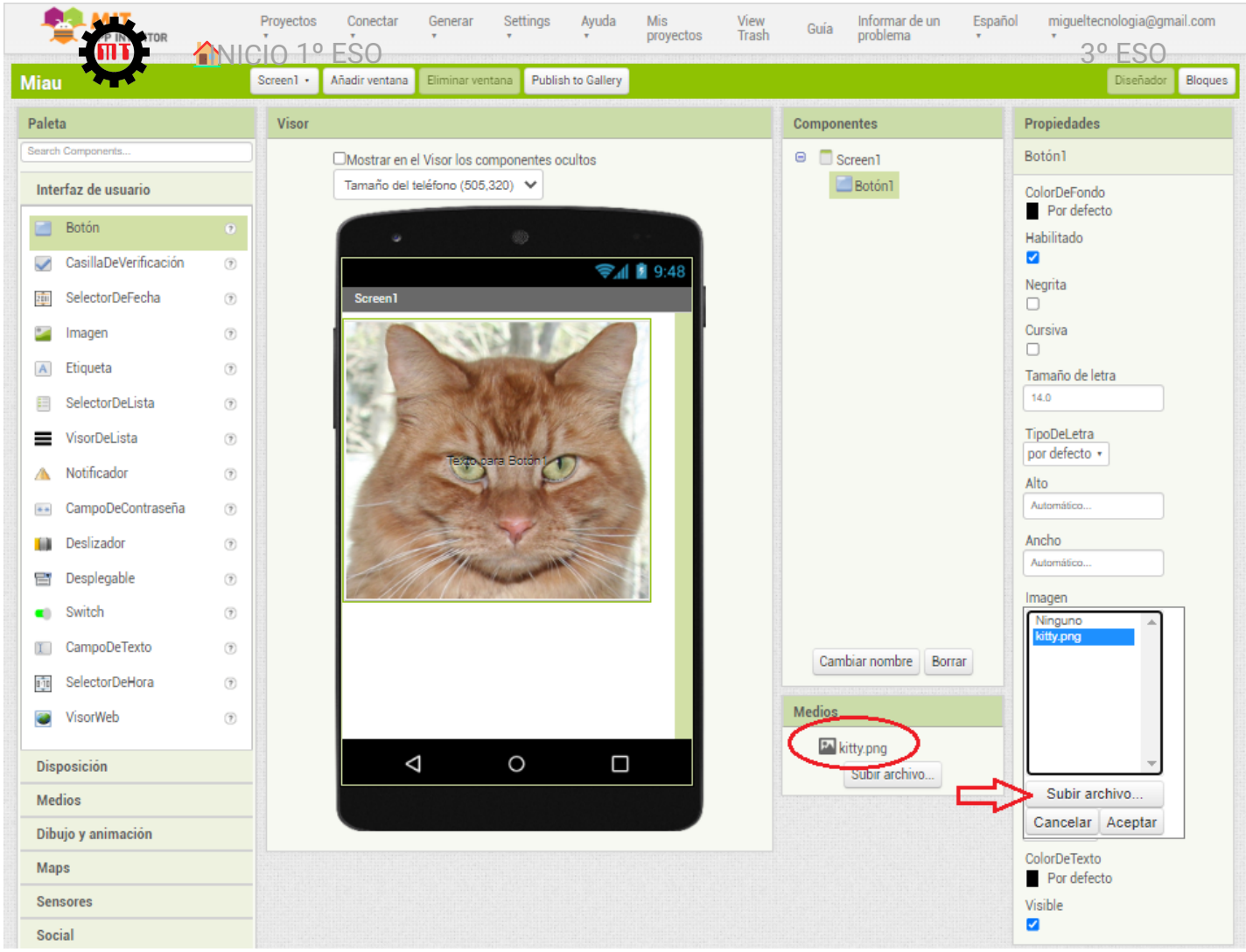
2º Entra en la página MIT App Inventor, inicia sesión con tu correo y pon el idioma en Español si no lo has hecho antes y crea un nuevo proyecto llamado Miau.

3º Arrastra un botón al visor del dispositivo:



En este momento, se han creado varios controladores de eventos en el área de bloques, que utilizaremos más tarde.

4º Cambia el aspecto del botón por la imagen del gato descargada antes, para ello, en el panel de propiedades del botón, pulsa en la opción **Imagen** y selecciona **Subir archivo**. Ahora busca el archivo "Kitty,png" descargado antes y selecciónalo:



Observa cómo la imagen del botón gris plano ha cambiado por la imagen del gato y en el panel de **Medios** aparece el archivo "kitty.png" subido a nuestro proyecto.

Para quitar el texto “Texto para el Botón1” que aparece por debajo del gato hay que borrar el valor de una propiedad **Texto del botón**, en el panel de propiedades.

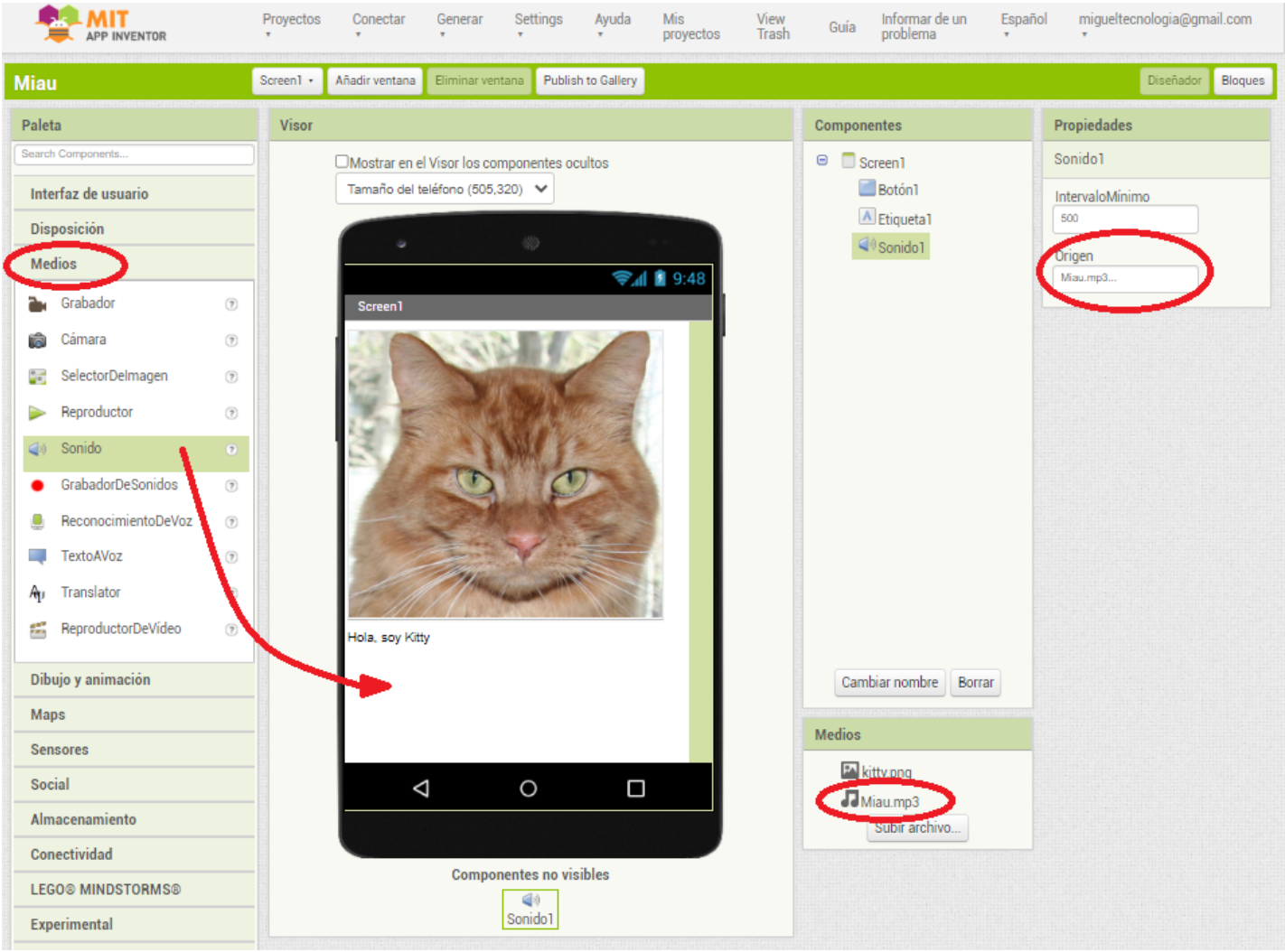
Si no vemos la cara del gato entera en la pantalla del dispositivo deberemos cambiar los valores de las propiedades **Ancho** y **Alto** del botón por “**Ajustar al contenedor**”, para que se ajusten al tamaño máximo disponible en la pantalla del dispositivo.

Para incluir una etiqueta debajo del gato que ponga “Hola, soy Kitty” arrastramos un componente **Etiqueta** hasta el visor, y la soltamos debajo del gato.

Cambiar el texto “**Texto para Etiqueta1**” por “**Hola, soy Kitty**” en el panel de propiedades de la etiqueta.

5º Ahora añadiremos un sonido a nuestra aplicación, arrastrando hasta el visor el icono **Sonido**, que está dentro del grupo **Medios** de la **Paleta**. Ojo, este objeto no se verá en el móvil o en el emulador, porque no es una imagen, ni un botón, ni una etiqueta. Por eso aparece debajo del visor, en el apartado **Componentes no visibles** .

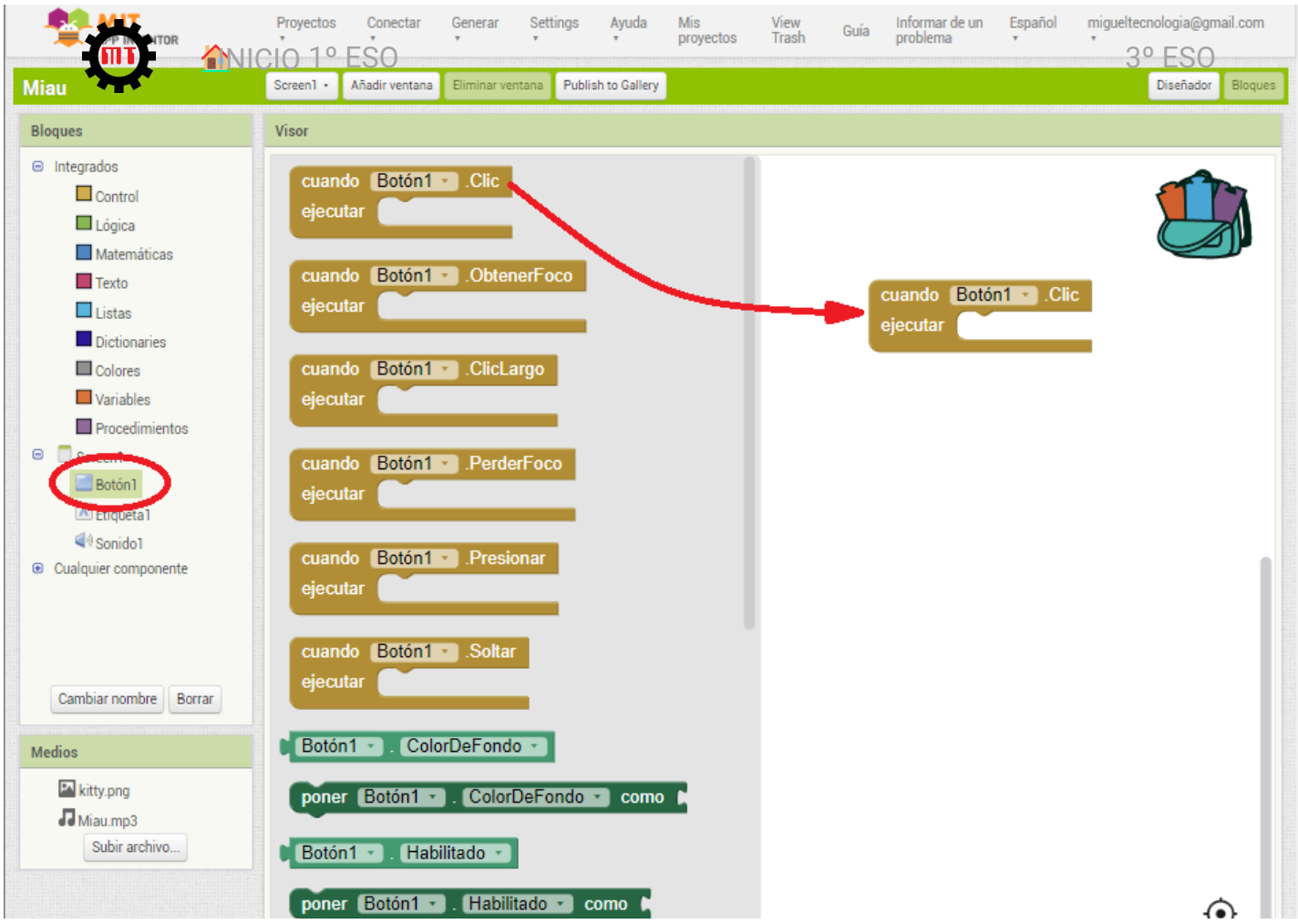
Vamos a asociar a este objeto que hemos creado el sonido "**Miau.pm3**" que hemos descargado. De nuevo hay que usar el panel de propiedades para este componente, haremos clic sobre el valor de la propiedad **Origen** del componente Sonido1 y subiremos el archivo descargado.



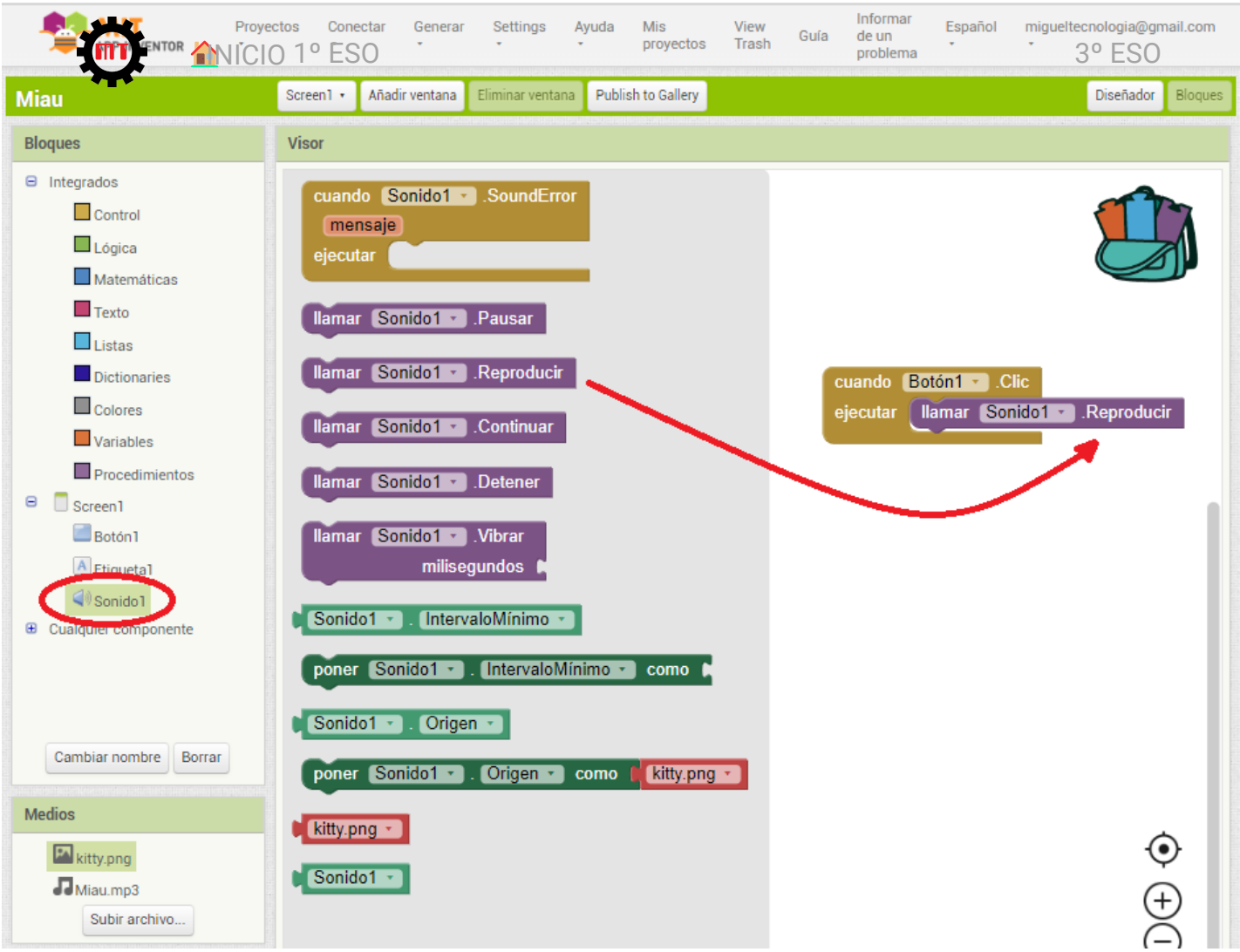
Observa cómo el archivo de sonido "Miau.mp3ª también se ha añadido al panel de **Medios** de nuestro proyecto.

Ya hemos terminado el diseño, ahora vamos a programar cómo queremos que se comporte nuestra App en el entorno de programación de **Bloques**.

6º. Pulsa el botón **Bloques** y accede al entorno de programación por bloques. Pulsa sobre el botón **Botón 1** del menú de **Bloques** y arrastra el bloque generador de eventos de la siguiente imagen:



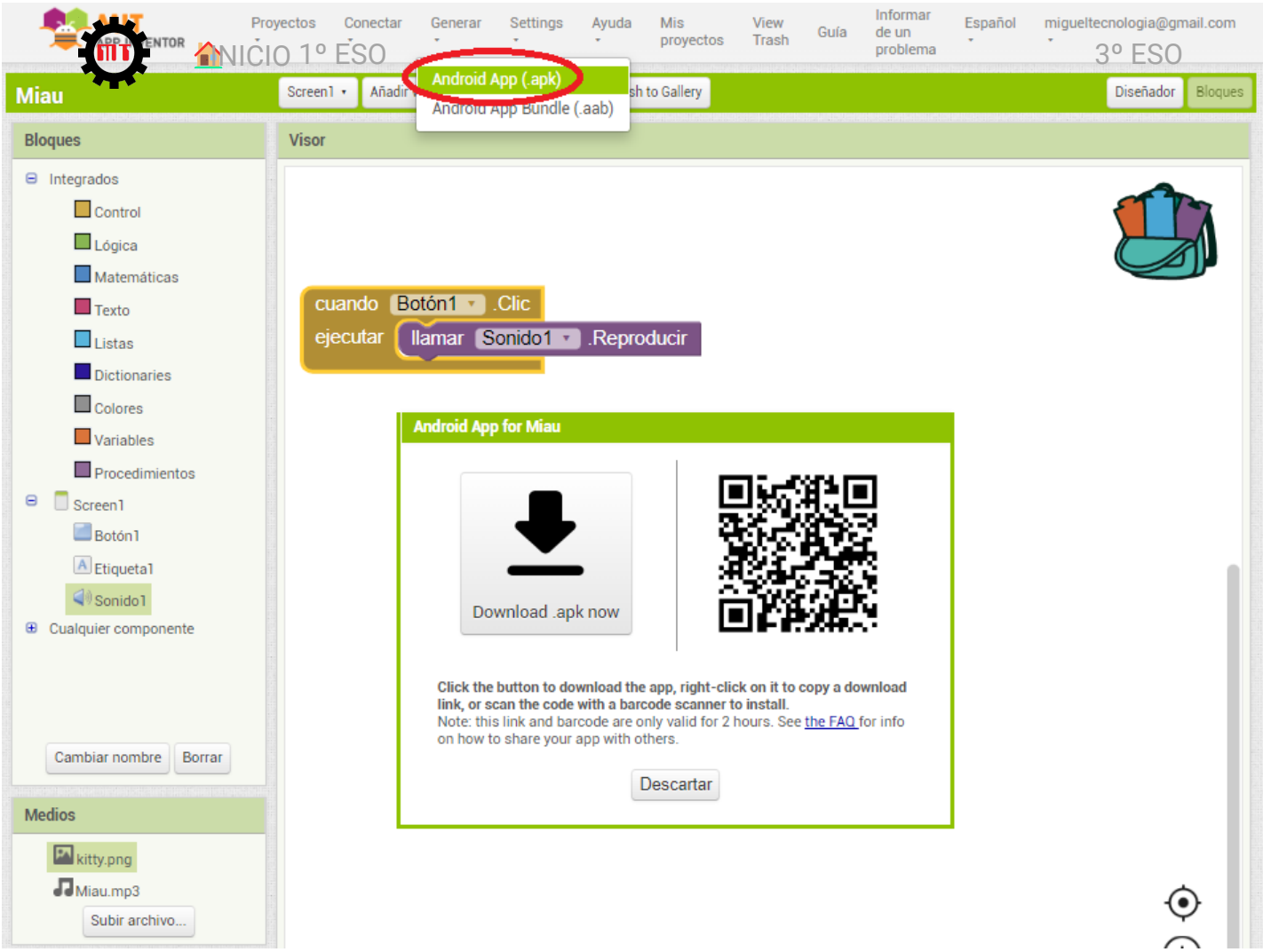
7º Pulsa sobre el botón **Sonido1** del menú de **Bloques** y arrastra el bloque que se indica en la siguiente la siguiente imagen, encajándolo dentro del bloque de evento:



¡Enhorabuena por la primera aplicación!

Para instalarla en el móvil permanentemente, como cualquier otra aplicación, podemos generar un código QR.

Para ello hacemos clic en **Generar** y elegimos la opción **Android App (.apk)**. Después de un rato, se mostrará un código QR que nos llevará a la página de descarga de nuestro archivo de instalación **Miau.apk.**



Si escaneamos el código QR generado con nuestro dispositivo, descargamos el archivo de instalación, que instalará la App, creándonos un icono con el nombre Miau que nos permite ejecutar nuestra aplicación.

Es posible que tengamos que tengamos que otorgar permisos para poder instalar aplicaciones de origen desconocido.

Actividades (2)

Añade los elementos necesarios al proyecto **Miau** para que el gato maúlle también al agitar el dispositivo.

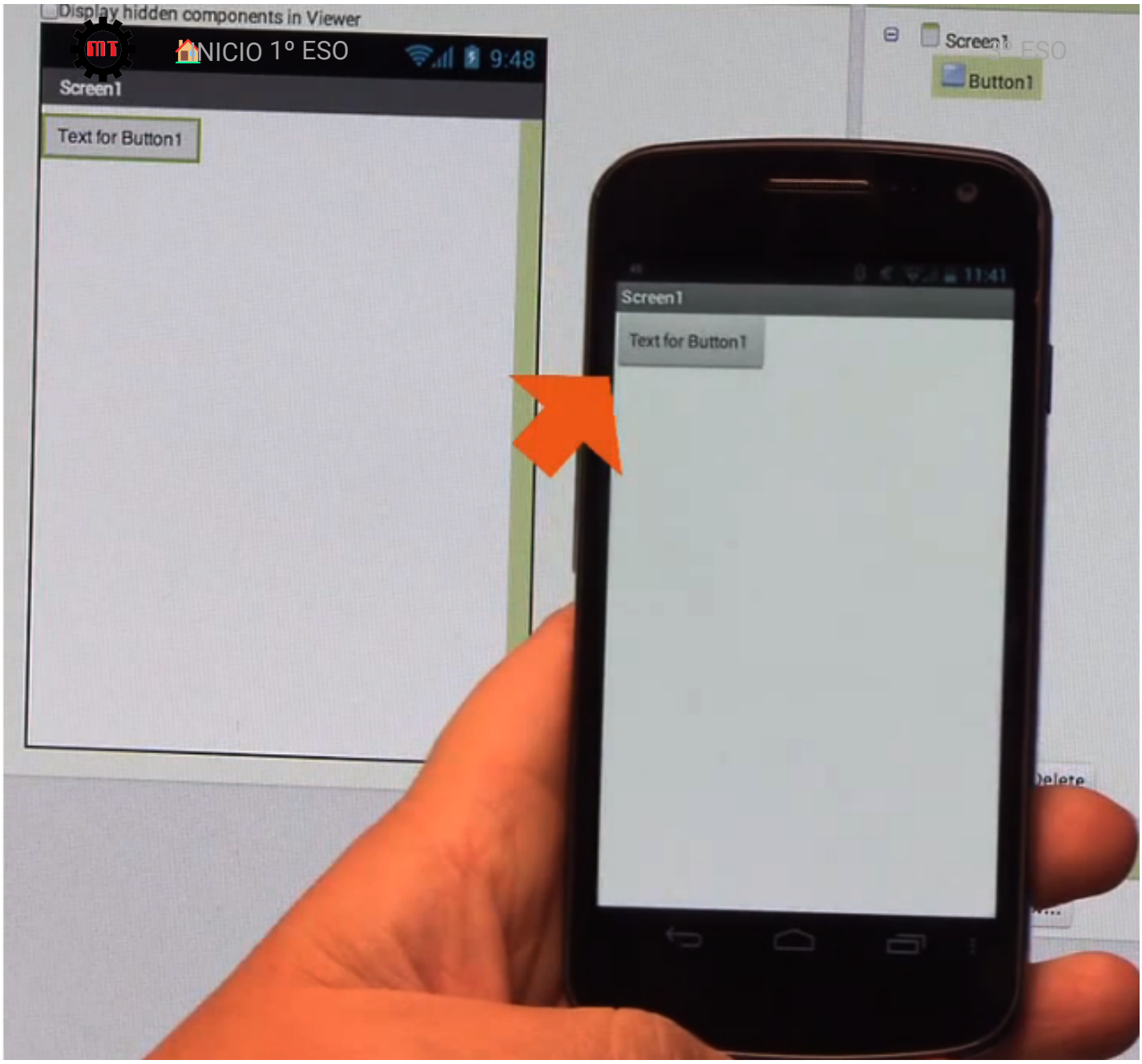
Cuando la termines, genera el archivo .apk e instálala en tu dispositivo.

(PISTA: Busca y utiliza el componente **Acelerómetro** dentro de la categoría **Sensores** de la **Paleta**.)

En el ejemplo anterior, una vez hecha la programación, hemos generado el código QR que nos permite instalar la aplicación en nuestro dispositivo de forma definitiva.

Este método es un poco engorroso en el caso de que estemos desarrollando una App y tengamos que probarla constantemente hasta conseguir el producto final.

Para evitar esto, MIT App Inventor dispone de un método con el que podemos ver en nuestro dispositivo los cambios que hacemos en la programación en **tiempo real**.


Display hidden components in Viewer
INICIO 1º ESO
9:48
Screen1
Text for Button1
Screen1
Button1
11:41
Screen1
Text for Button1

Para poder operar de esta forma, primero tenemos que instalar en nuestro positivo la siguiente aplicación disponible en Google Play y App Store:

(ATENCIÓN: Las Apps generadas solo se pueden ejecutar en dispositivos Android)

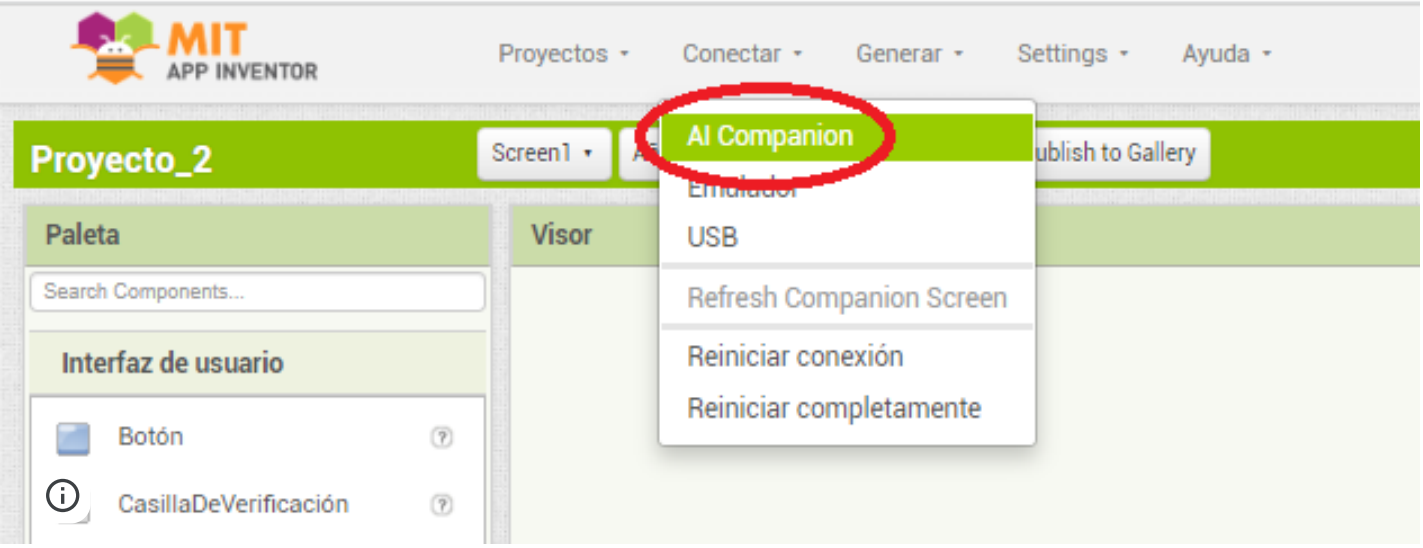
MIT AI2 Companion

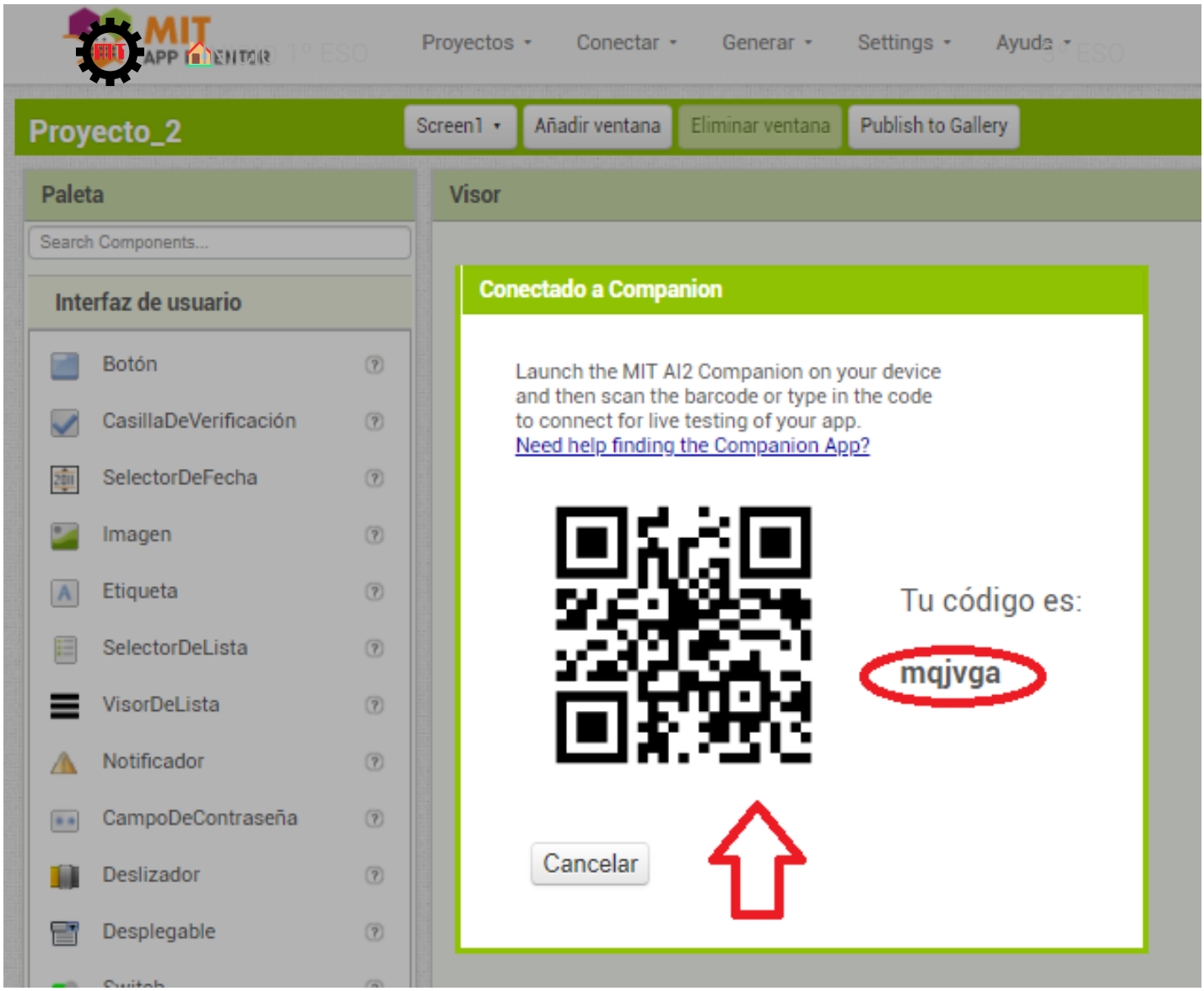
Una vez instalada la App de MIT AI2 Companion, entra en MIT App Inventor y elige:

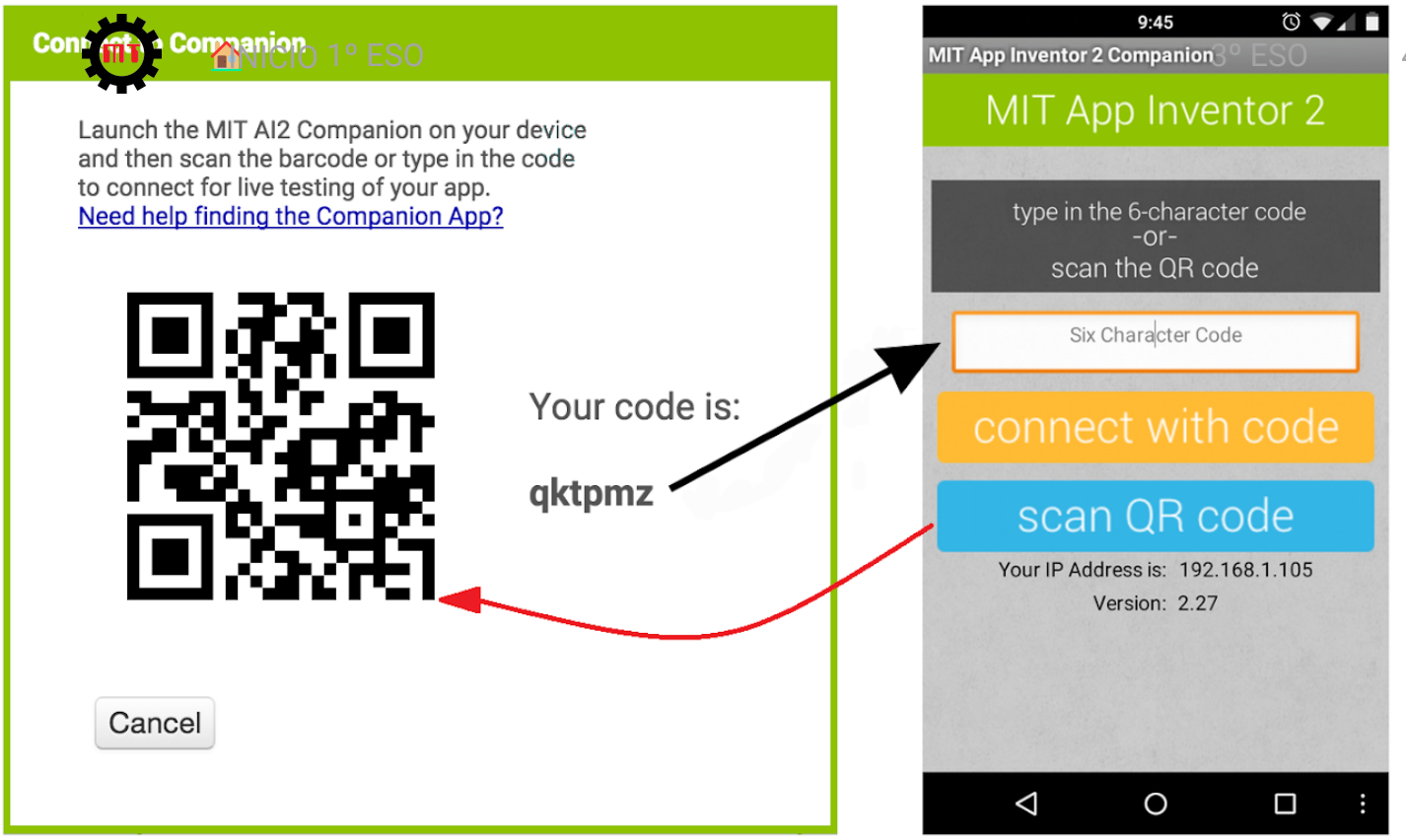
Conectar -> AI Companion 👉

Al cabo de un tiempo, nos aparece una ventana como la de abajo, con la que podemos conectar nuestro dispositivo a la página de MIT App Inventor del ordenador. 👇

(ATENCIÓN: Tanto el ordenador como el dispositivo tienen que estar conectados a la misma red wifi o Ethernet)




Proyectos
Conectar
Generar
Settings
Ayuda
Proyecto_2
Screen1
Añadir ventana
Eliminar ventana
Publish to Gallery
Paleta
Search Components...
Interfaz de usuario
Botón
CasillaDeVerificación
SelectorDeFecha
Imagen
Etiqueta
SelectorDeLista
VisorDeLista
Notificador
CampoDeContraseña
Deslizador
Desplegable
Visor
Conectado a Companion
Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app.
Need help finding the Companion App?
Tu código es:
mqjvga
Cancelar



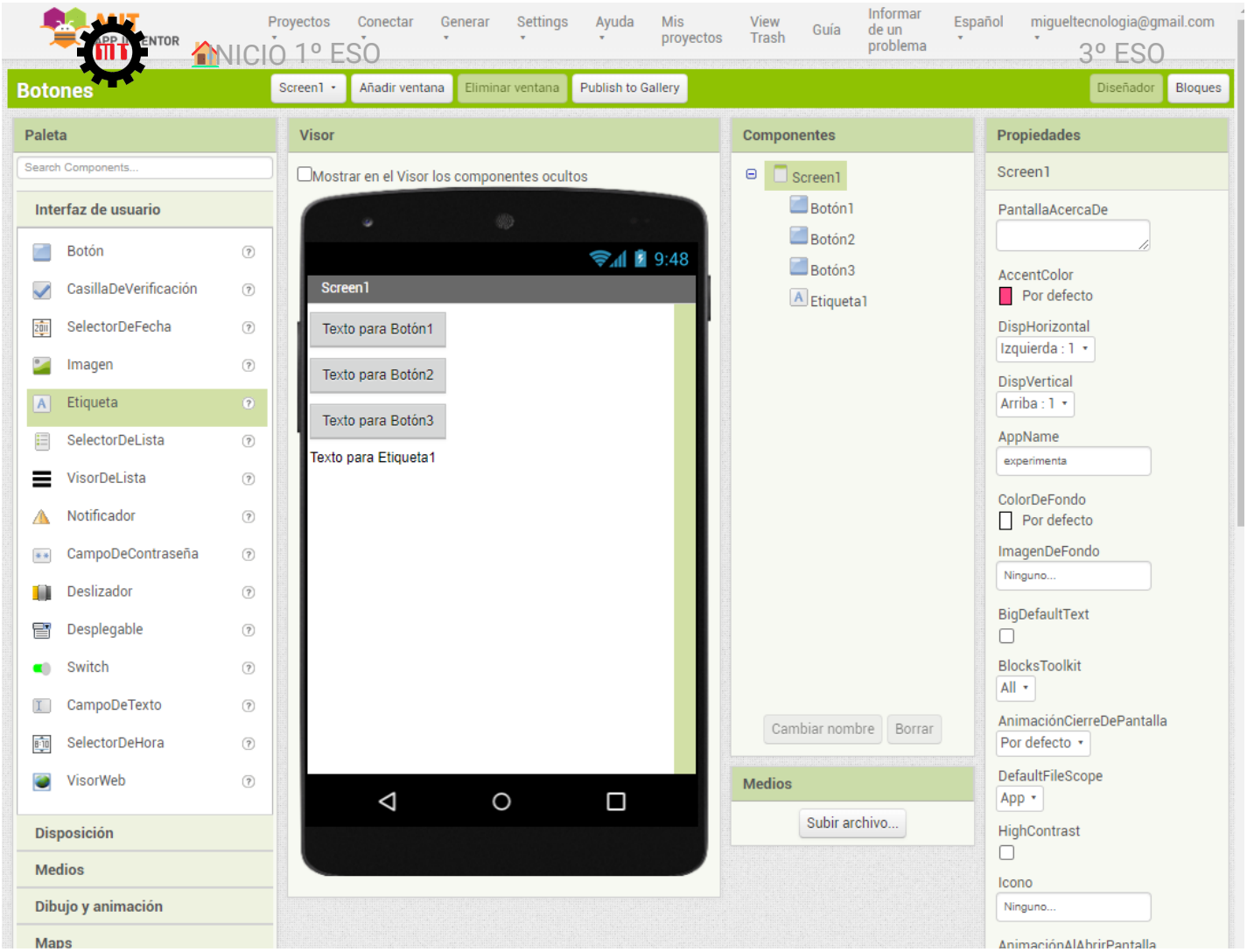
Si abrimos MIT AI2 Companion en el dispositivo, tenemos dos formas de conectarnos:

- Escribiendo el código de 6 caracteres que nos ofrece MIT App Inventor y pulsando el botón amarillo "**connect with code**"
- Escaneando el código QR tras pulsar el botón azul "**scan QR code**"

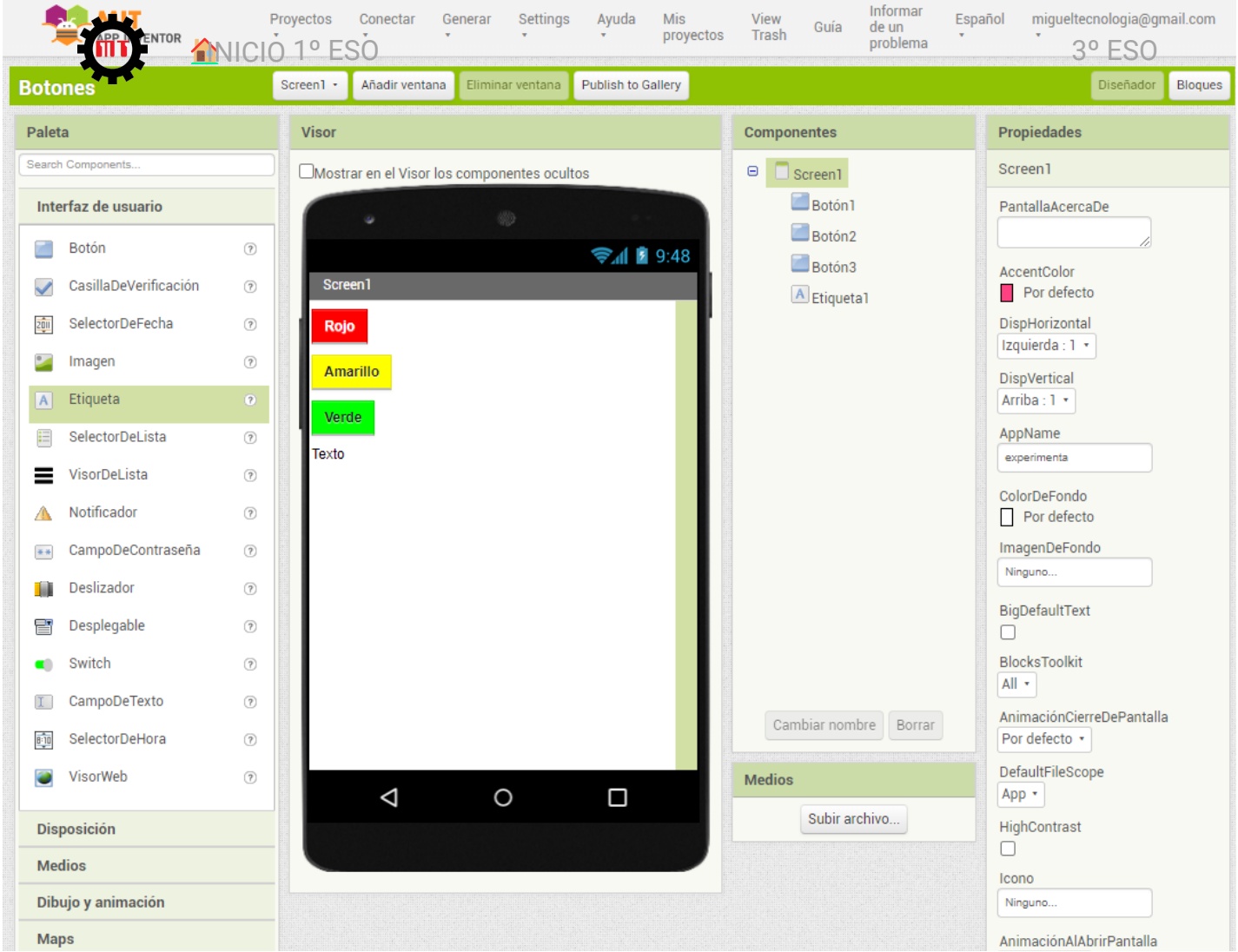
Hagamos un ejercicio guiado para comprender el uso de botones y etiquetas:

1º Entra en MIT App Inventor en el ordenador, ejecuta la App de MIT AI2 Companion en el dispositivo.

Arrastra tres botones y una etiqueta de texto dentro del visor del dispositivo:



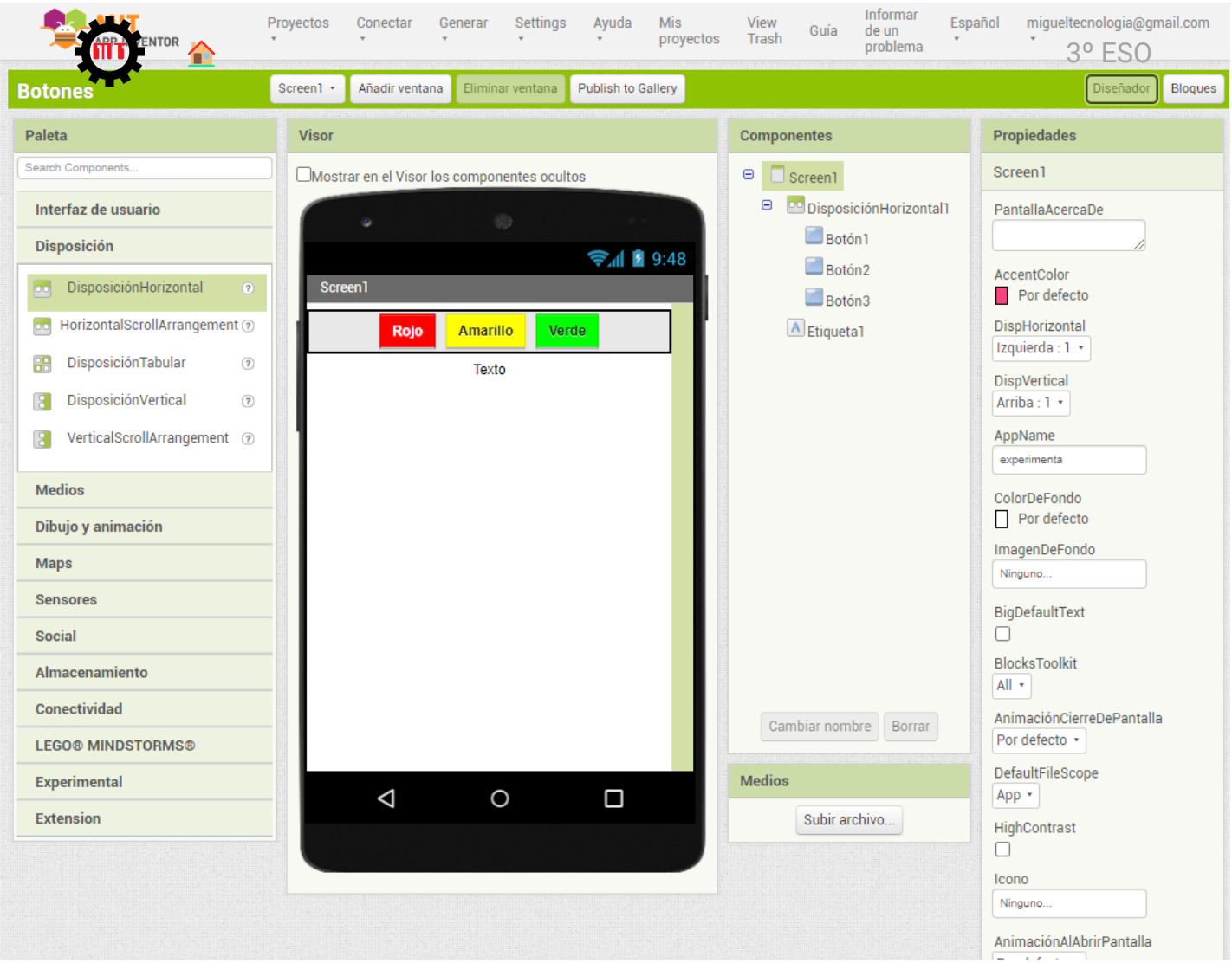
2º Cambia los nombres de los botones y su color de fondo. y cambia el texto de la etiqueta desde su panel de propiedades correspondiente:



3º Añade al visor una **DisposiciónHorizontal** de la categoría **Disposición** y arrastra dentro los tres botones (Verás cómo se alinean horizontalmente)

Las disposiciones nos permite distribuir los elementos dentro de la pantalla de nuestra App de la forma que queramos.

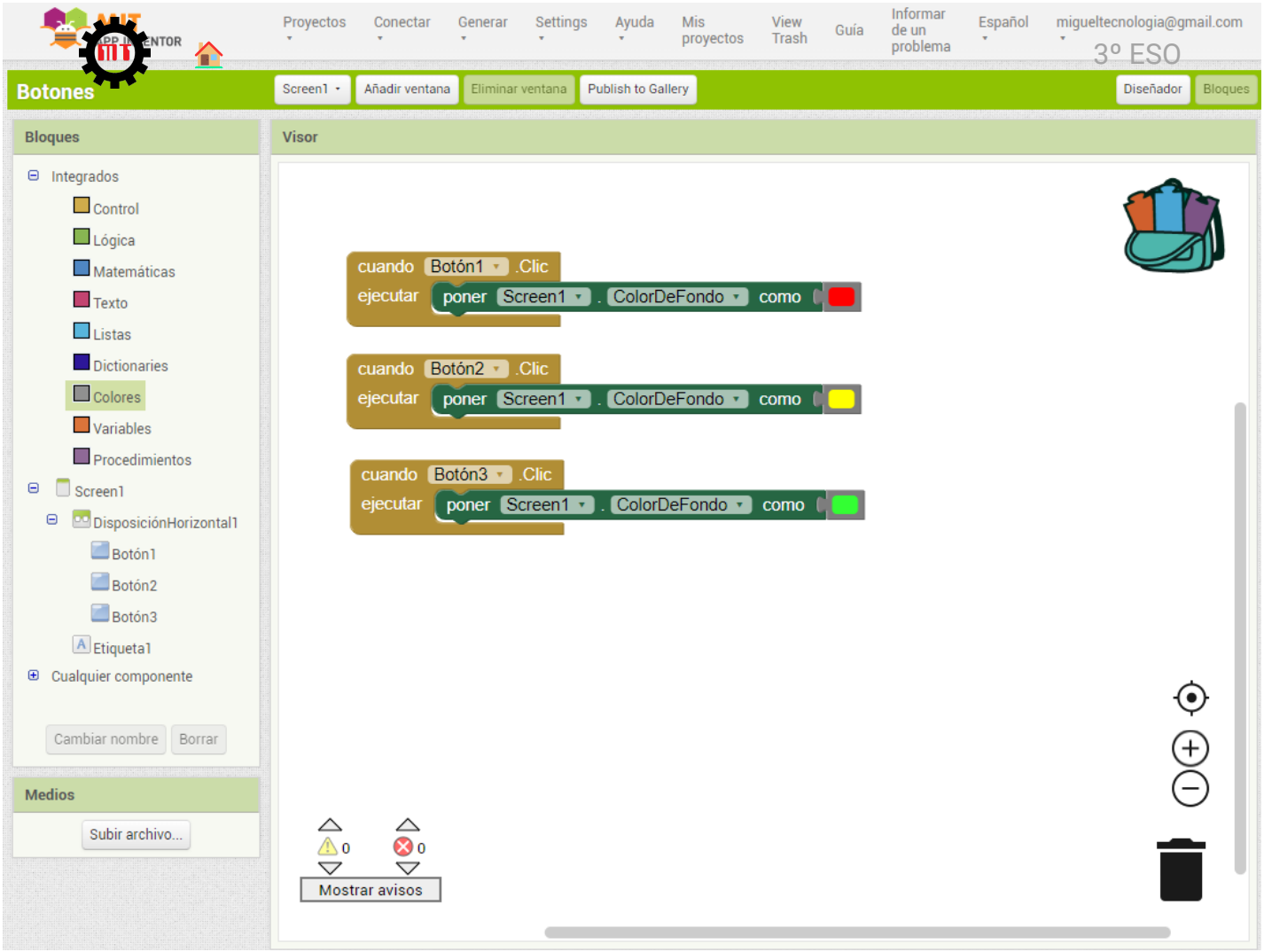
Tanto en la **DisposiciónHorizontal** como la Etiqueta, hemos seleccionado como **Ancho -> Ajustar al contenedor**, para que ocupe todo el ancho de la pantalla y hemos seleccionado la disposición **Centrada**.



4º Queremos conseguir que cambie el color de fondo de **Screen1** cada vez que pulsemos uno de los tres botones, para ello crea el siguiente programa añadiendo los bloques correspondientes. (Intenta comprender lo que hacen cada uno de los bloques que has empleado)

Los botones son los generadores de eventos, luego de ellos tenemos que seleccionar los bloques marrones.

Prueba el funcionamiento del programa realizado.



5º Vamos a añadir otros bloques para que nos muestre un mensaje en la **Etiqueta1** cada vez que pulsamos un botón:

Cambia el programa anterior poniendo en cada botón el nombre de un animal. Al pulsar cada botón, se reproduce el sonido que hace dicho animal.

Instala la app creada en tu dispositivo con Mit AI2 Companion.

4. EJERCICIOS NO GUIÁDOS

A partir de ahora, haremos ejercicios con MIT App Inventor y crearemos la App en nuestro móvil a través de MIT AI2 Companion, pero de forma no guiada, ya que se supone que conocemos el funcionamiento del programa.

De todas formas, se ofrece la programación de cada ejercicio.

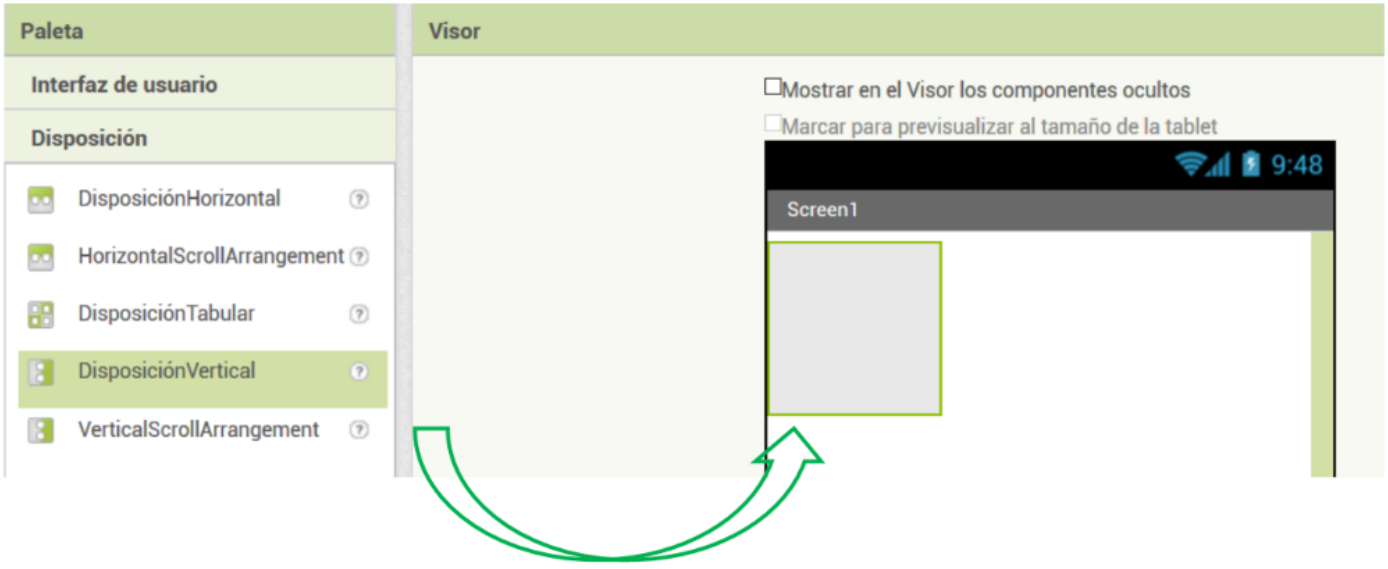
4.1. Calculadora sumadora simple

Crea un nuevo proyecto llamado "Calculadora " y realiza la interfaz como la siguiente en el visor de componentes, insertando tres etiquetas, tres campos de texto y un botón.

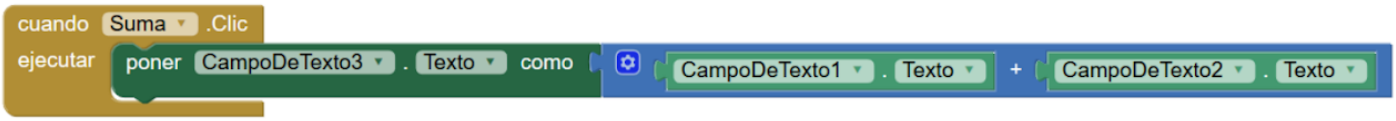
Inserta **antes** un cuadro de "DisposiciónVertical", que te permitirá que todos los componentes se organicen verticalmente.

El funcionamiento del programa es simple, se introducen dos números en los dos primeros campos de texto y se muestra la suma de ambos números en el tercer campo de texto, después de haber pulsado el botón "Suma".

Con los campos de texto podemos pedir datos al usuario del programa.



La programación de los bloques es la siguiente. Intenta comprender cómo funciona el programa.



Actividades (4)

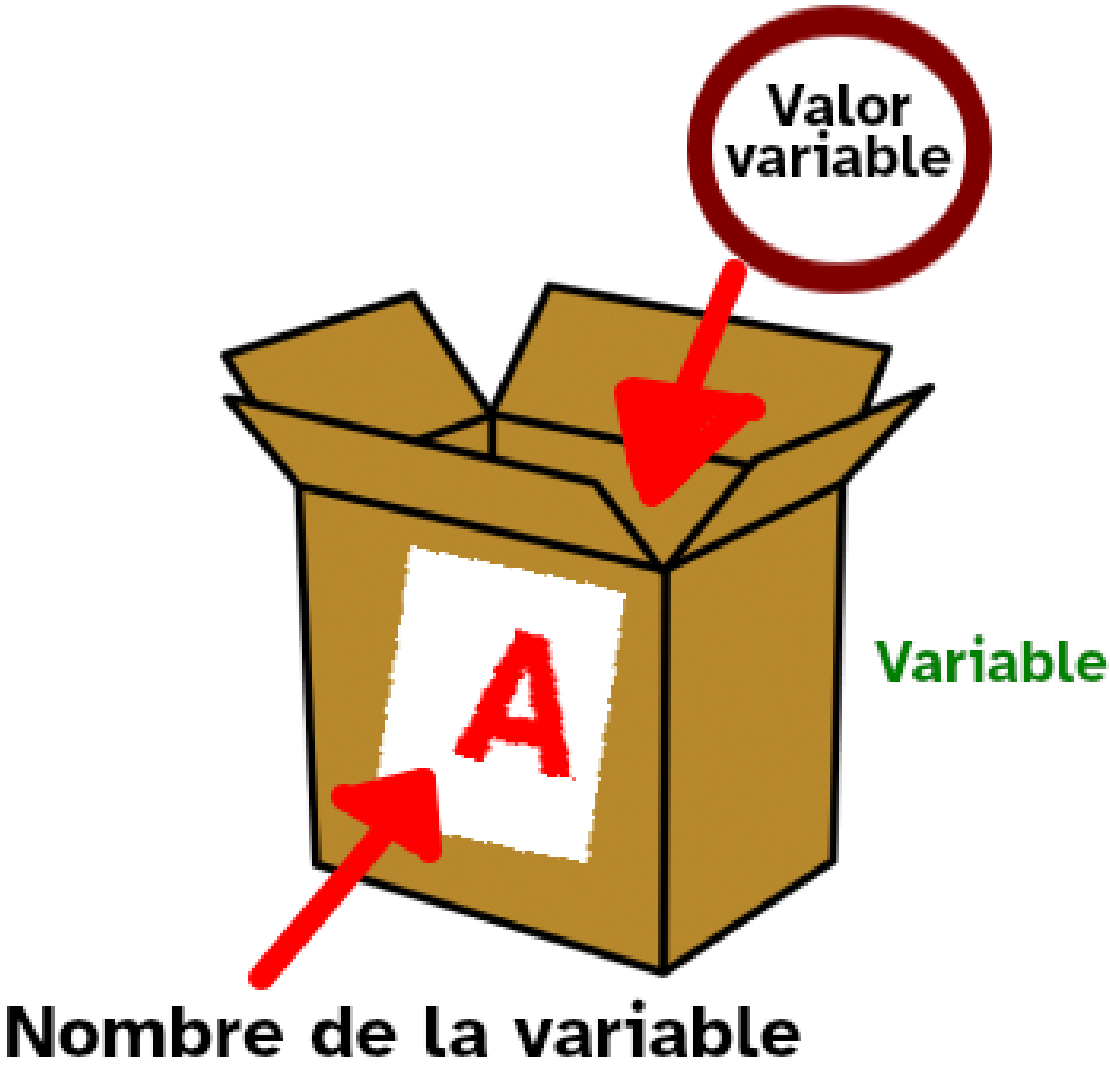
Modifica el programa anterior para que en lugar de sumar los dos números, los multiplique (Cambia el texto del botón por "Multiplicación").

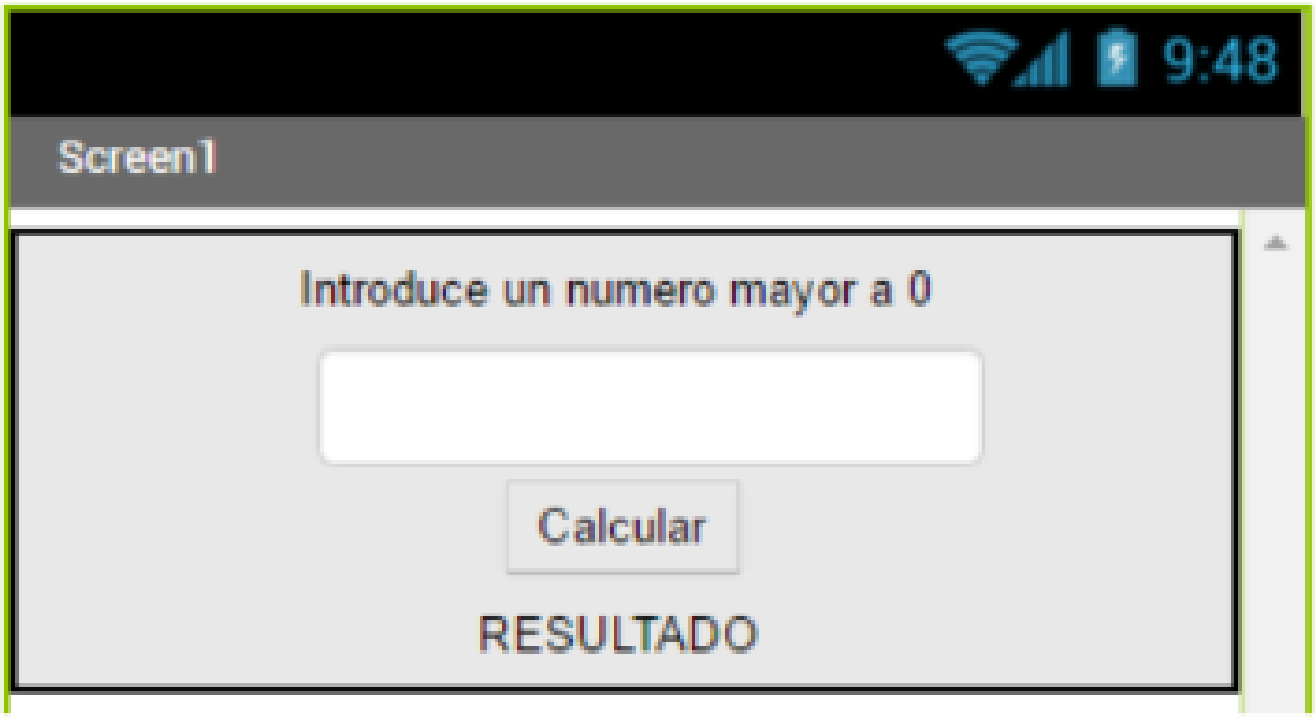
Crea un nuevo proyecto llamado "Par_impar" y realiza la interfaz como la siguiente en el visor de componentes, insertando dos etiquetas, un campo de texto y un botón.

Inserta **antes** un cuadro de "DisposiciónVertical", que te permitirá que todos los componentes se organicen verticalmente.

El funcionamiento del programa es simple, se introduce un número y se pulsa el botón "Calcular", entonces la etiqueta "RESULTADO" nos dice si el número introducido es par o impar.

En este programa utilizaremos **variables**. Una variable es una especie de caja con nombre que nos guarda un valor durante el funcionamiento del programa. Podemos escribir, leer o cambiar el valor de la variable en dodo momento


Valor variable
A
Variable
Nombre de la variable



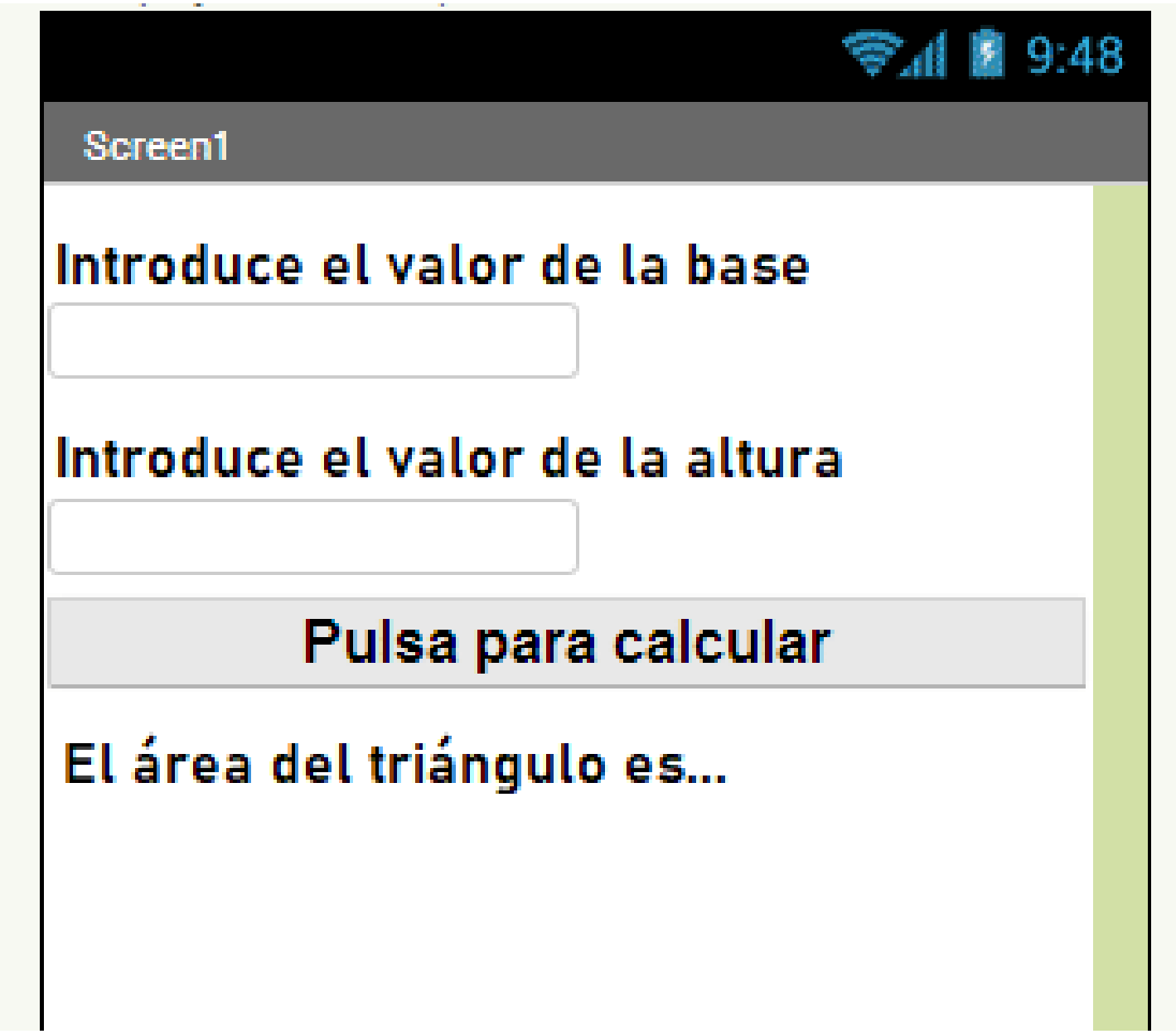
La programación de los bloques es la siguiente. Intenta comprender cómo funciona el programa. ¿Cuáles son las variables creadas y para qué sirven?

Crea un nuevo proyecto llamado "Triangulos" y realiza la interfaz como la siguiente en el visor de componentes, insertando tres etiquetas, dos campos de texto y un botón.

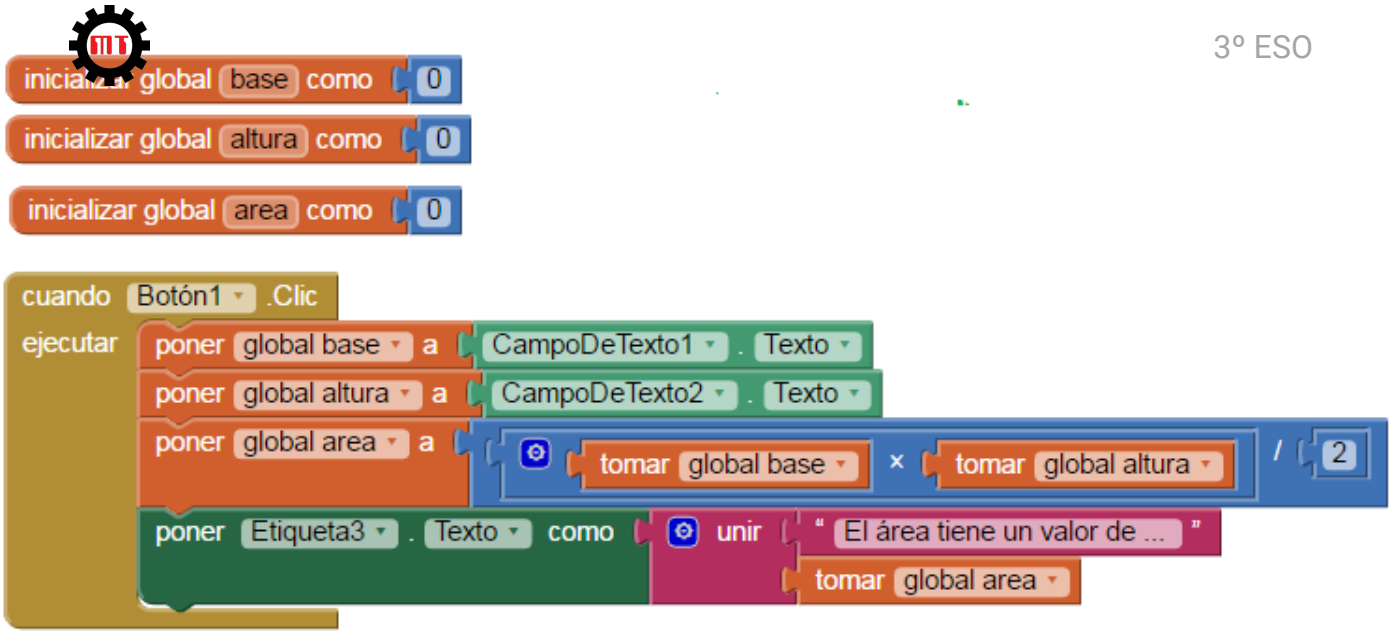
Inserta **antes** un cuadro de "DisposiciónVertical", que te permitirá que todos los componentes se organicen verticalmente.

El funcionamiento del programa es el siguiente: se introducen los valores de la base y la altura de un triángulo y se muestra el valor del área pulsar el botón.

En este programa también utilizaremos **variables**.



La programación de los bloques es la siguiente. Intenta comprender cómo funciona el programa.

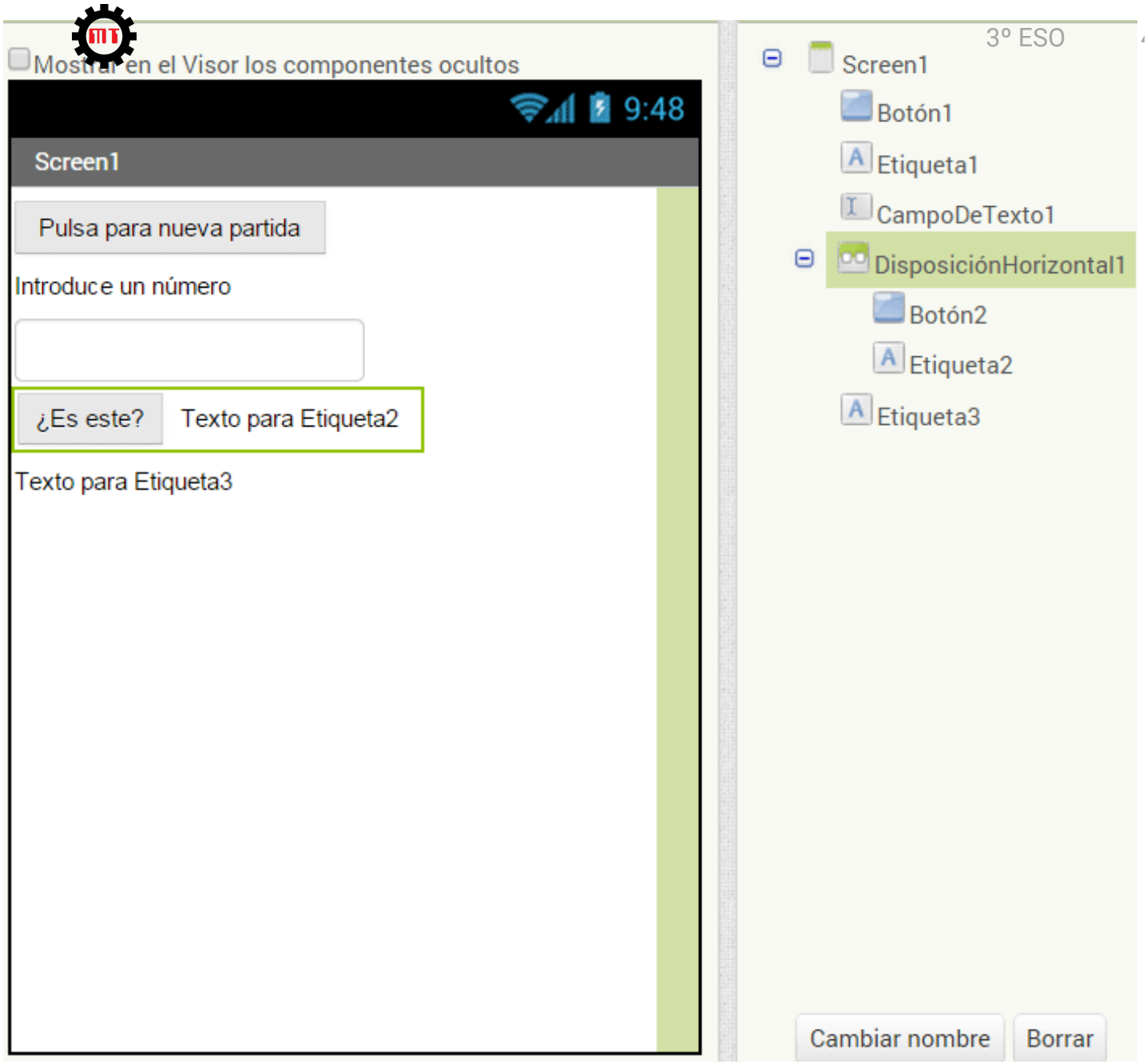


4.4. Adivina

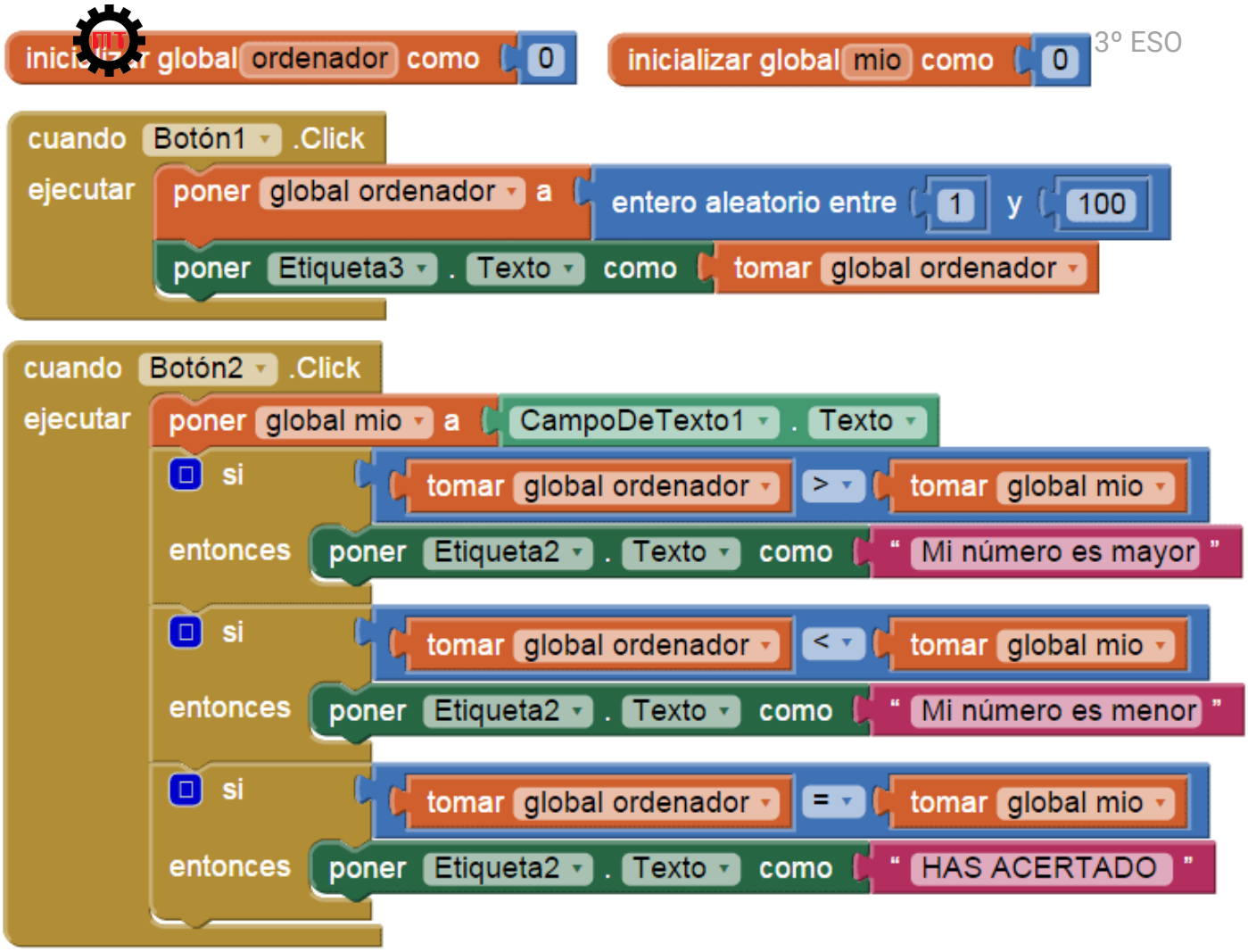
Crea un nuevo proyecto llamado "Adivina" y realiza la interfaz como la siguiente en el visor de componentes, insertando tres etiquetas, un campo de texto y dos botones.

Inserta un bloque de "DisposiciónHorizontal" para poder colocar el botón 2 y la etiqueta 2 en la misma línea

El funcionamiento del programa es el siguiente: Se pulsa el botón superior para comenzar una partida. Se trata de adivinar el número aleatorio entre 1 y 100 que ha creado el programa.



La programación de los bloques es la siguiente. Intenta comprender cómo funciona el programa.



5. EJERCICIOS GUIÁDOS POR VÍDEO

Vamos a realizar unos ejercicios un poco más complejos, siguiendo las intrucciones de los siguientes vídeos.

Sigue las instrucciones de este vídeo para realizar una App que nos permita traducir al Inglés y al Francés.

Sigue las instrucciones de este vídeo para realizar una App que nos permita dibujar.

Sigue las instrucciones de este vídeo para realizar un sencillo juego de adivinar números.