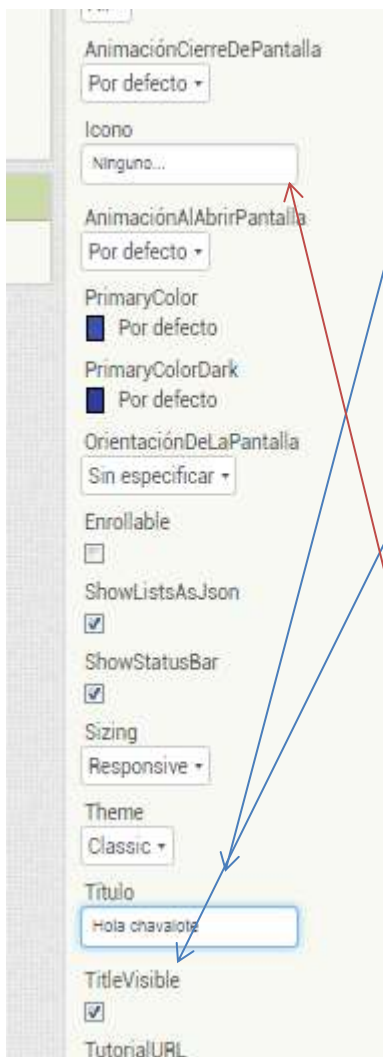


Guía para grabar el tutorial

Vamos a crear la primera aplicación. Como paso previo tenéis que tener los archivos que os di localizados y tener una cuenta en Gmail porque es ahí donde se guardarán vuestros proyectos de App Inventor.



Lo primero es crear un proyecto nuevo. El nombre del proyecto no puede tener espacios por lo que recomiendo separar el nombre por mayúsculas



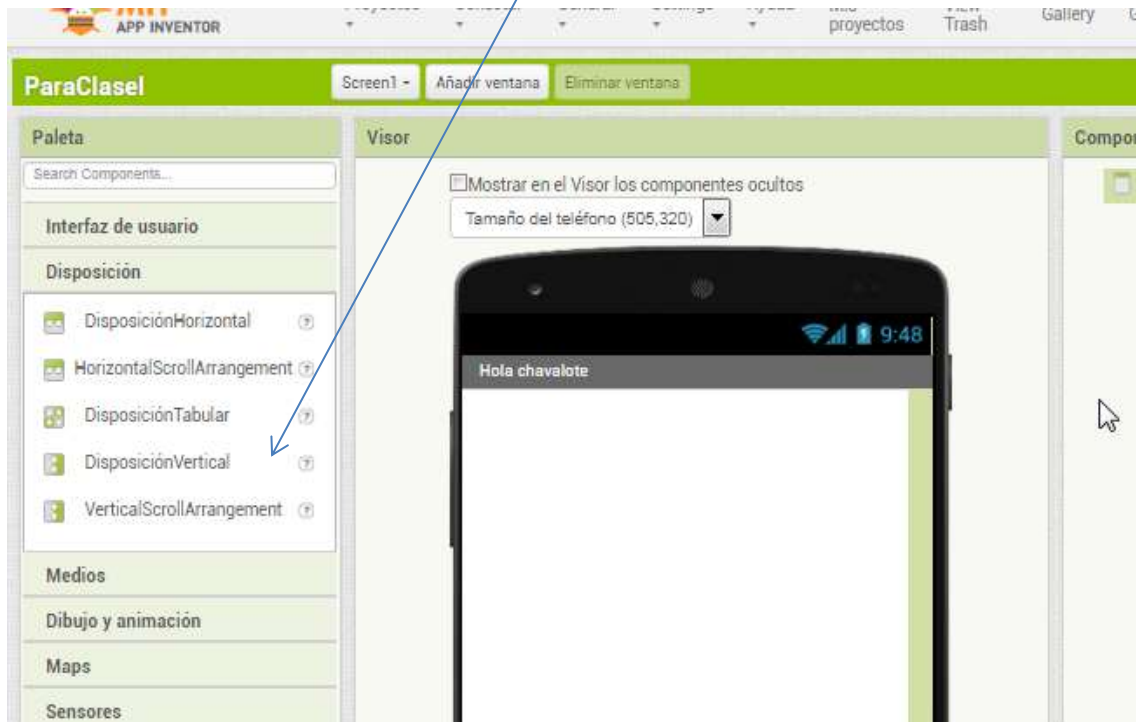
Lo primero es tocar un poco las propiedades. Se puede hacer o no según lo consideremos necesario. Yo cambiaré el título que siempre es muy necesario para localizar las cosas. Le pondré Hola majo.

Esto es muy útil cuando la aplicación tiene varias pantallas. Así estamos más organizados y se encuentran mejor las cosas.

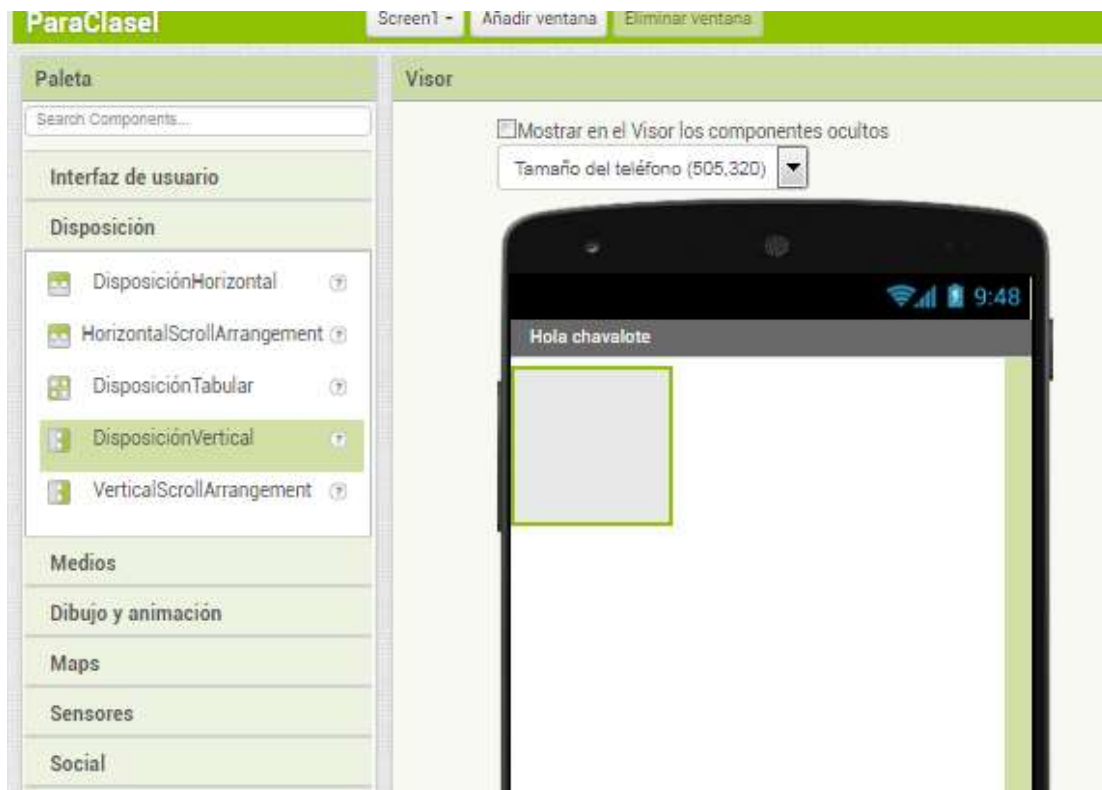
Se puede elegir también que el título esté visible o no. Yo lo voy a quitar.

También se le podría poner aquí un icono para nuestra App.

Nuestra aplicación constará de una imagen y un botón como elementos visibles. Para colocarlo lo primero que vamos a poner es una disposición vertical.



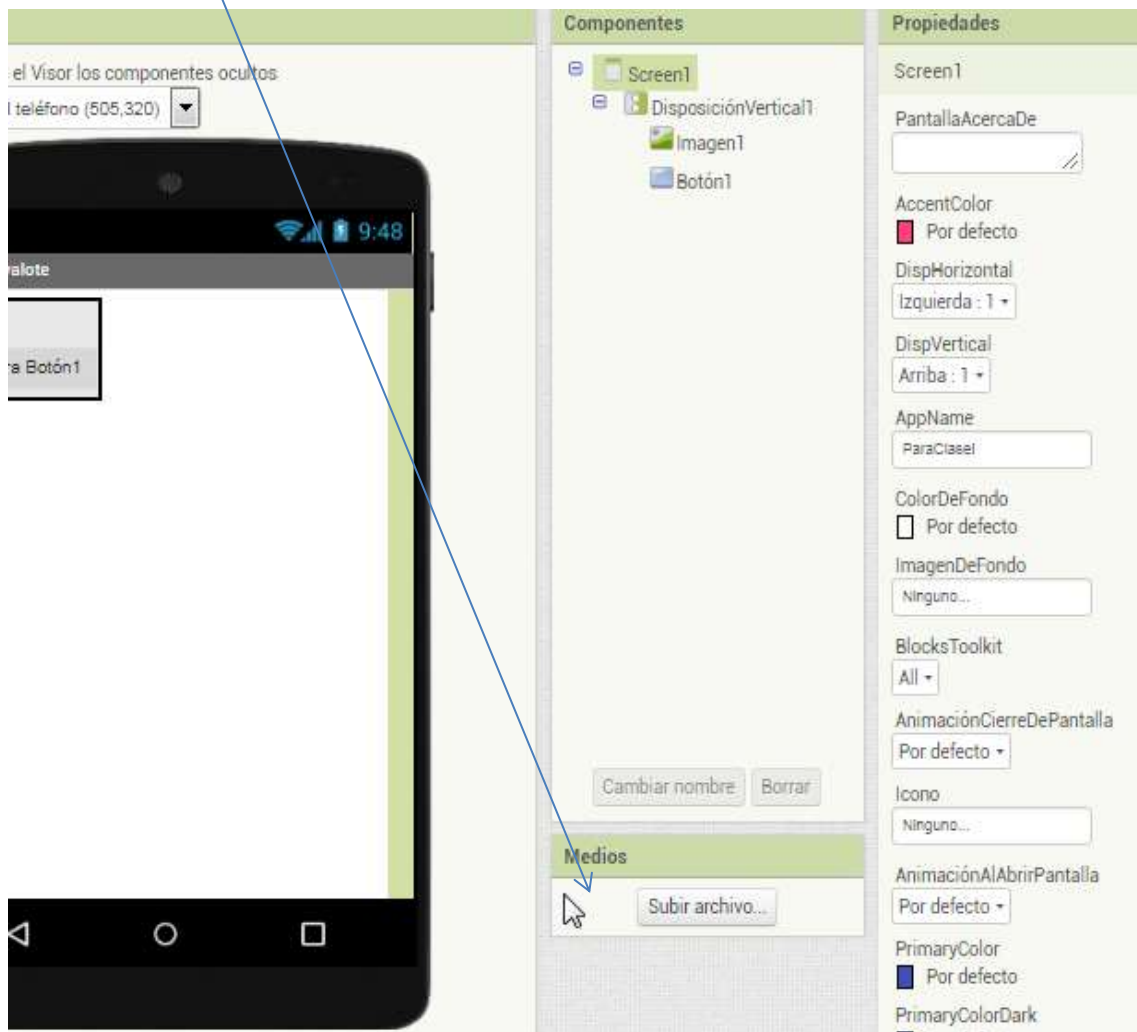
Nos quedará así:



Dentro de ella pondremos una imagen y un botón. Esto se encuentra dentro del *interfaz de usuario*.



En la parte de medios habrá que subir los archivos que os he mandado.



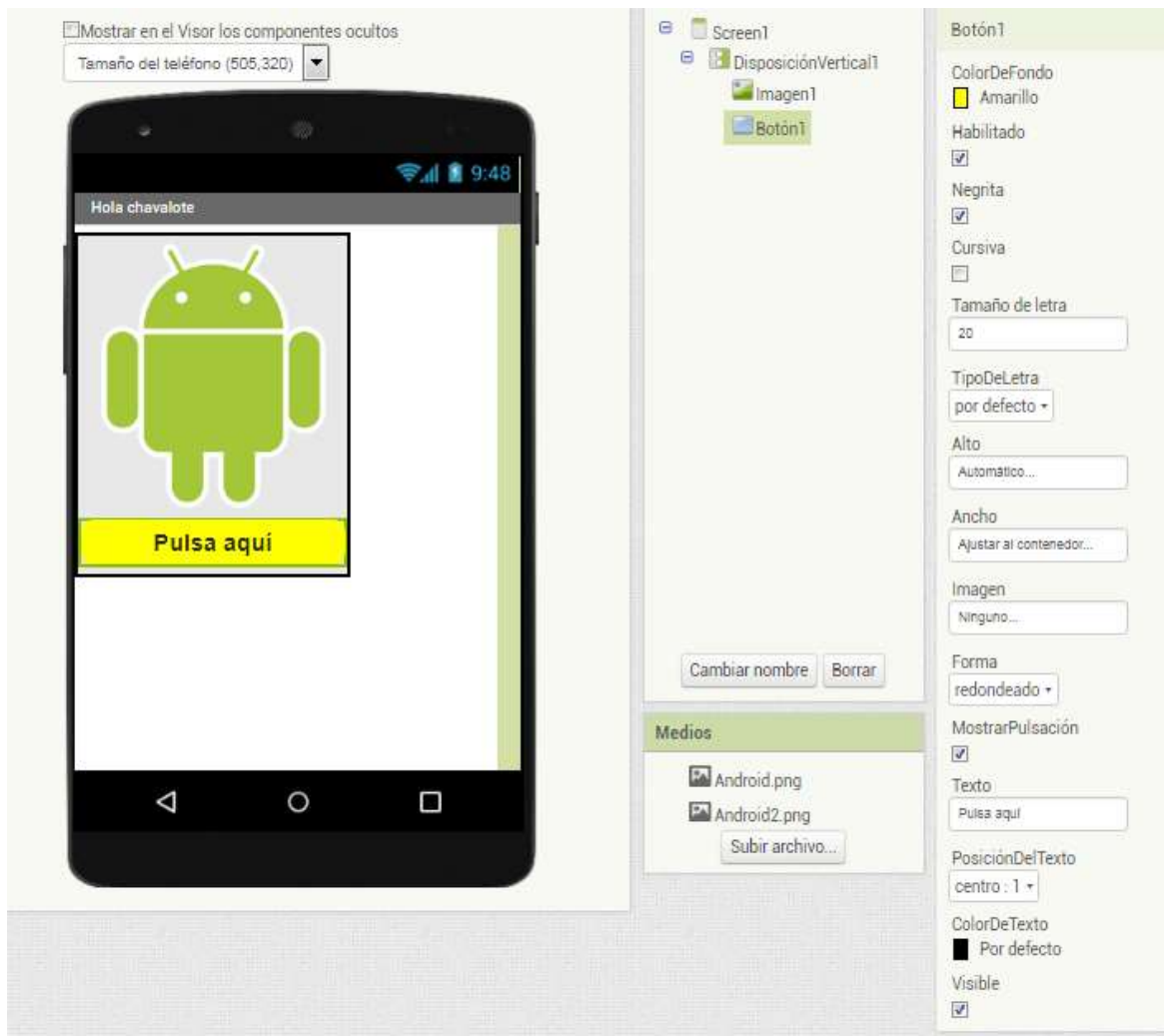
Si habéis hecho esta operación con éxito el panel de medios os debiera aparecer así:



En la imagen1 pondremos como imagen el *Android* y como queda muy alto pondré 200 píxeles de alto y de ancho.

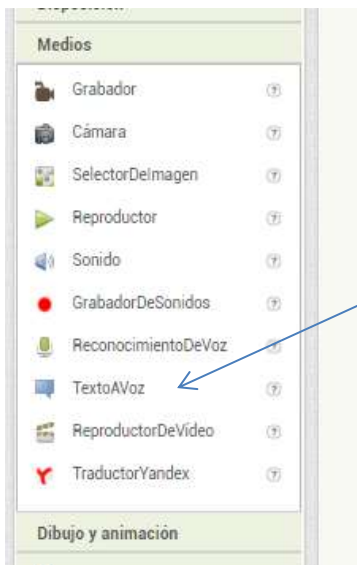


A continuación toco algo las propiedades del botón. Color de fondo amarillo, por ejemplo, el texto en negrita, el tamaño de la letra algo más grande, por ejemplo 20, el ancho lo voy a ajustar al contenedor, el texto lo cambio y pongo *pulsa aquí*, y, por último, cambio el botón a redondeado.

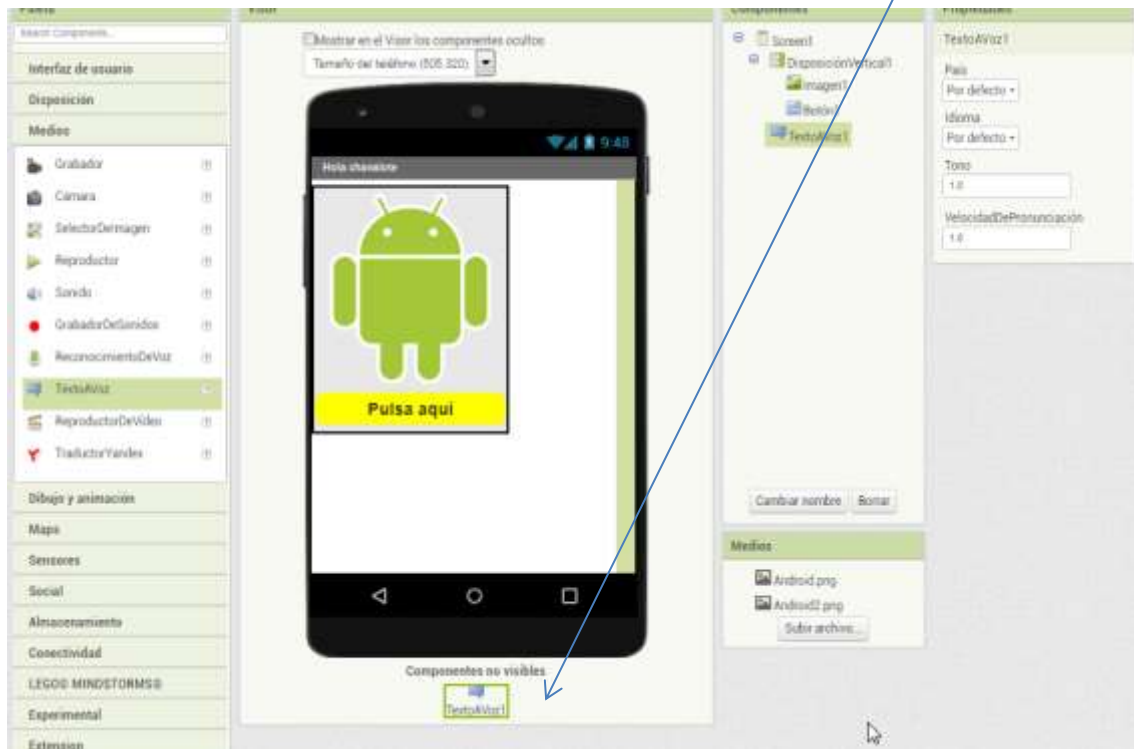


Con esto ya habríamos terminado la parte visual y entraríamos en la parte de programación por bloques.

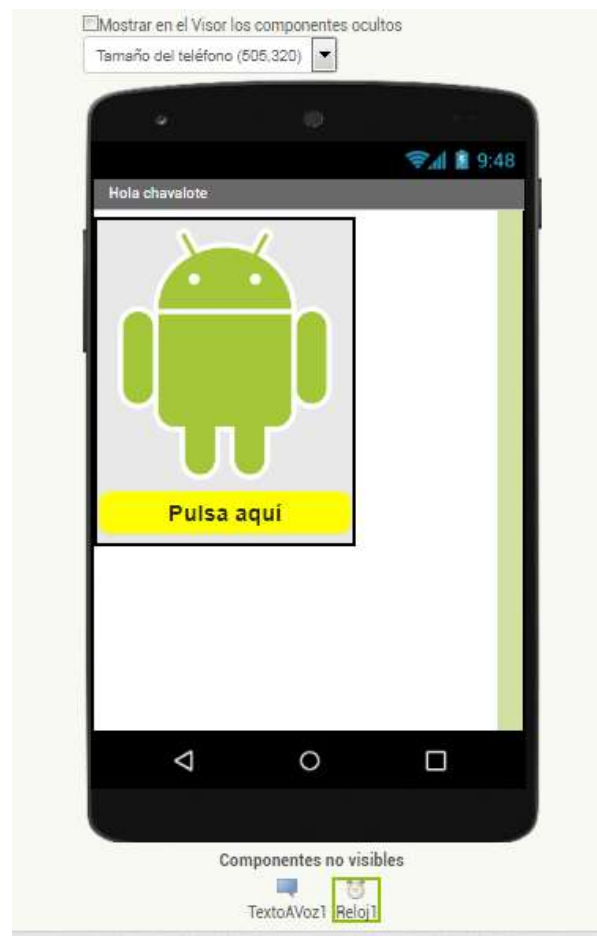
Lo primero cogemos un texto a voz y lo pondremos



Veréis que cuando lo soltéis en el escenario no aparecerá ahí sino que aparecerá en la parte de abajo:



Análogamente pondremos un reloj de la paleta de sensores. Si lo hemos hecho bien el aspecto que tendrá nuestro proyecto será el siguiente:



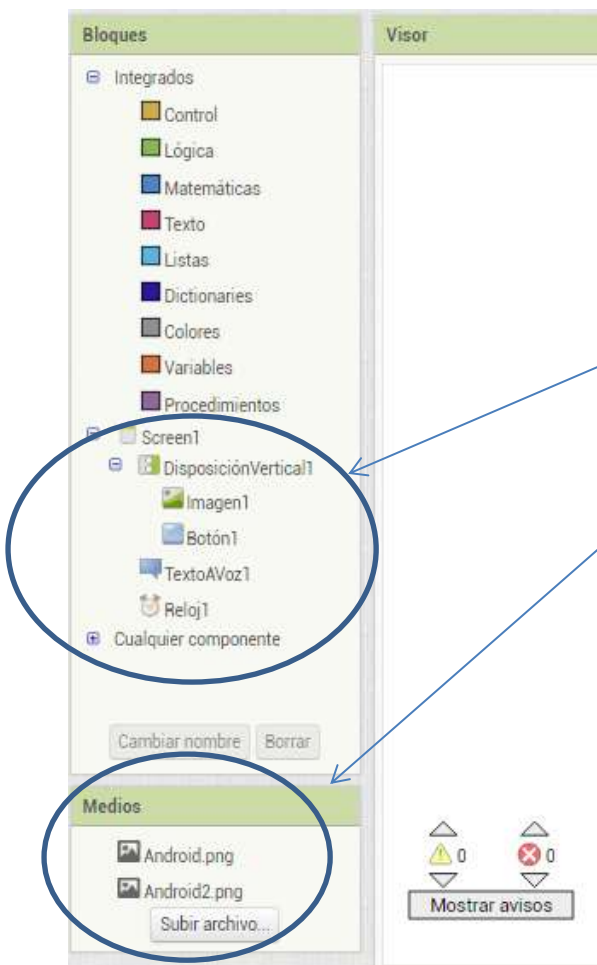
Habr  que configurar en el panel de *propiedades* seleccionando previamente en el de *Componentes* el *texto a voz* como *Pa s* Espa a y como *Idioma* es para seleccionar el espa ol para que hable con nuestro acento. Comprobamos que el reloj tiene como Intervalo de Temporizador 1000 ms, es decir, un segundo.

Ahora s  que es cuando empezamos el dise o con bloques. Para esto pinchamos en bloques y en principio nos tiene que aparecer vac a.



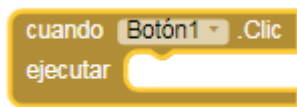
Ver is que en los paneles de la izquierda aparecer  lo siguiente:

Observar que nos aparece lo que hemos a adido.

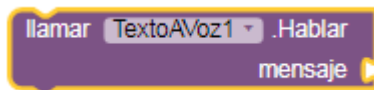


Lo primero que vamos a configurar es el comportamiento del botón.

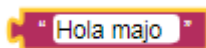
Pincho en *Botón* y selecciono



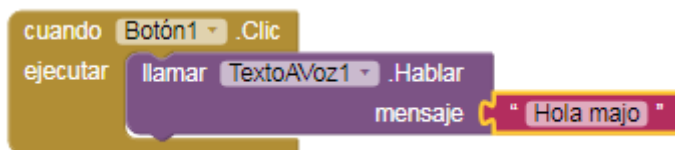
Después pincho en *Texto a Voz* y selecciono



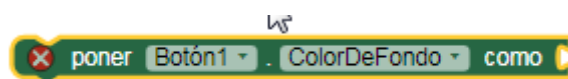
A continuación pincho en *Texto* y selecciono un texto en el que pongo Hola majo.




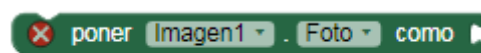
Si todo ha ido bien en nuestra ventana de bloques debería aparecer:



Podemos hacer más cosas como cambiarle el color de fondo. Veréis que cuando cogéis el color de fondo os aparecerá de esta manera



La  es porque da un error porque no le tenemos asignado un color. Cuando se lo asignéis veréis que desaparece. Además debemos cambiar la imagen para que dé la sensación de que el androide saluda. Para ello pinchamos en *Imagen* y seleccionamos este bloque



Hay que poner la extensión del archivo en nuestro caso escribiremos *android2.png*

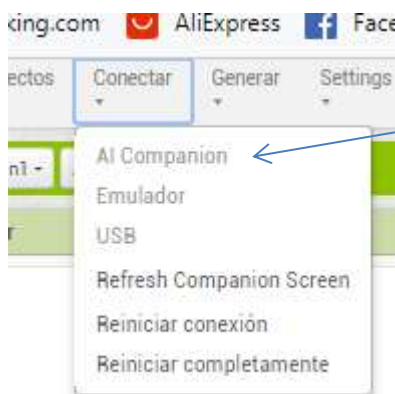
Si todo ha ido como debe deberemos tener en nuestra ventana de bloques lo siguiente:


```

cuando Botón1 .Clic
ejecutar
  llamar TextoAVoz1 .Hablar
    mensaje "Hola majo"
  poner Botón1 .ColorDeFondo como [Cyan]
  poner Imagen1 .Foto como "Android2.png"

```

Si queréis podéis probar lo hecho hasta aquí pasando vuestra App al móvil. Como vuestro ordenador estará en la misma red Wifi que el móvil podéis generar un código QR y pasarlo al móvil escaneándolo con el MIT App Inventor2. Si todo está correcto cuando pulséis el botón en robot levantará el brazo y dirá hola majo. Recuerdo que para generar el código QR hay que pulsar en AI Companion



Faltaría programar el retorno de la aplicación a su estado original. Para eso utilizamos el reloj. Pinchamos el reloj y seleccionamos el bloque:

```

cuando Reloj1 .Temporizador
ejecutar

```

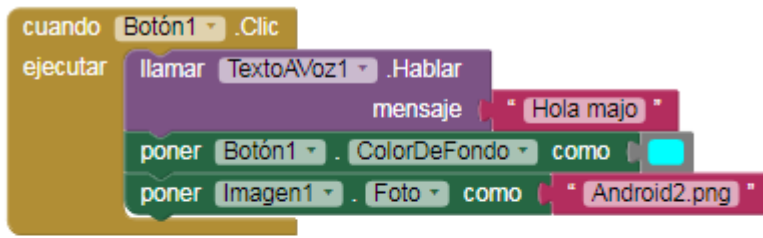
Este bloque se ejecuta cada vez que el temporizador llega a cero. En nuestro caso esto ocurre cada segundo. Y lo que haremos es que en este caso de fin de cuenta ponemos todo en las condiciones iniciales. Por lo que tendremos que agregar en la ventana de bloques lo siguiente:

```

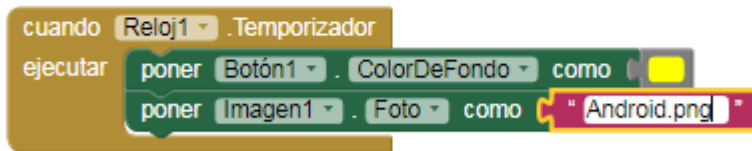
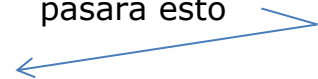
cuando Reloj1 .Temporizador
ejecutar
  poner Botón1 .ColorDeFondo como [Yellow]
  poner Imagen1 .Foto como "Android.png"

```

En definitiva nos tiene que aparecer todo esto:



Cuando pulso el botón pasará esto



Y después de un segundo pasará esto.

Ahora ya podéis volver a probarlo en el móvil. Si por casualidad observáis un parpadeo puede ser porque la pantalla se refresca cada segundo. Puede ser que ocurra o no. Depende del móvil pero si pasa deberéis ir a diseñador y cambiar la configuración de reloj desmarcando la pestaña TemporizadorHabilitado como se muestra:



Y habilitando y deshabilitando después el reloj mediante bloques de programación cuando a nosotros nos interese.

```
cuando Botón1 .Clic
ejecutar
  llamar TextoAVoz1 .Hablar
    mensaje "Hola majo"
  poner Botón1 . ColorDeFondo como [cyan]
  poner Imagen1 . Foto como "Android2.png"
  poner Reloj1 . TemporizadorHabilitado como cierto
```

```
cuando Reloj1 .Temporizador
ejecutar
  poner Botón1 . ColorDeFondo como [yellow]
  poner Imagen1 . Foto como "Android.png"
  poner Reloj1 . TemporizadorHabilitado como falso
```