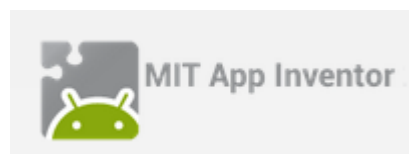


APPINVENTOR: EDICIÓN DE APPS PARA MÓVILES ANDROID

Esta aplicación online nos permite crear apps para móviles Android. Nosotros crearemos apps para poder controlar nuestro Arduino desde el móvil (aprovechando el bluetooth del teléfono) utilizando un módulo bluetooth que se puede acoplar a Arduino.

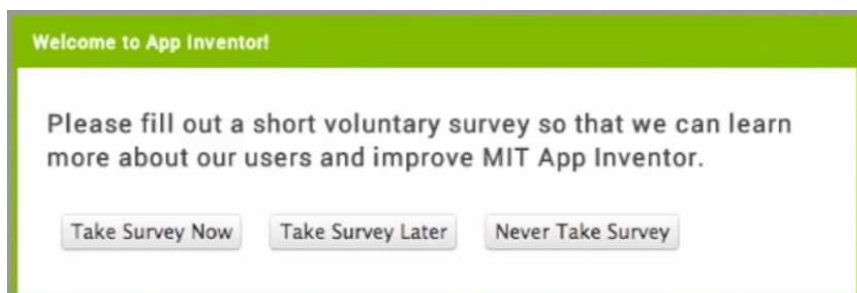


Para utilizar la aplicación necesitaremos tener una cuenta de gmail.

Para acceder a la aplicación entramos en <http://appinventor.mit.edu> y hacemos clic en el icono "Create apps!" de la figura.

Create
apps!

Nos pedirá nuestra cuenta de gmail.



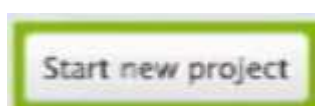
Se la indicaremos haciendo clic sobre Take Survey Now.

Cuando ya le hayamos indicado la cuenta de gmail, nos pedirá autorización para acceder a la cuenta y haremos clic en el botón Permitir.



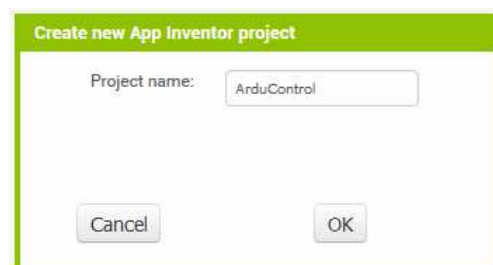
Si aún no tenemos creada una cuenta de gmail, le diremos que más tarde en el botón "Take Survey Later", nos dará la bienvenida (damos continuar) y nos dirá que no tenemos ningún proyecto creado.

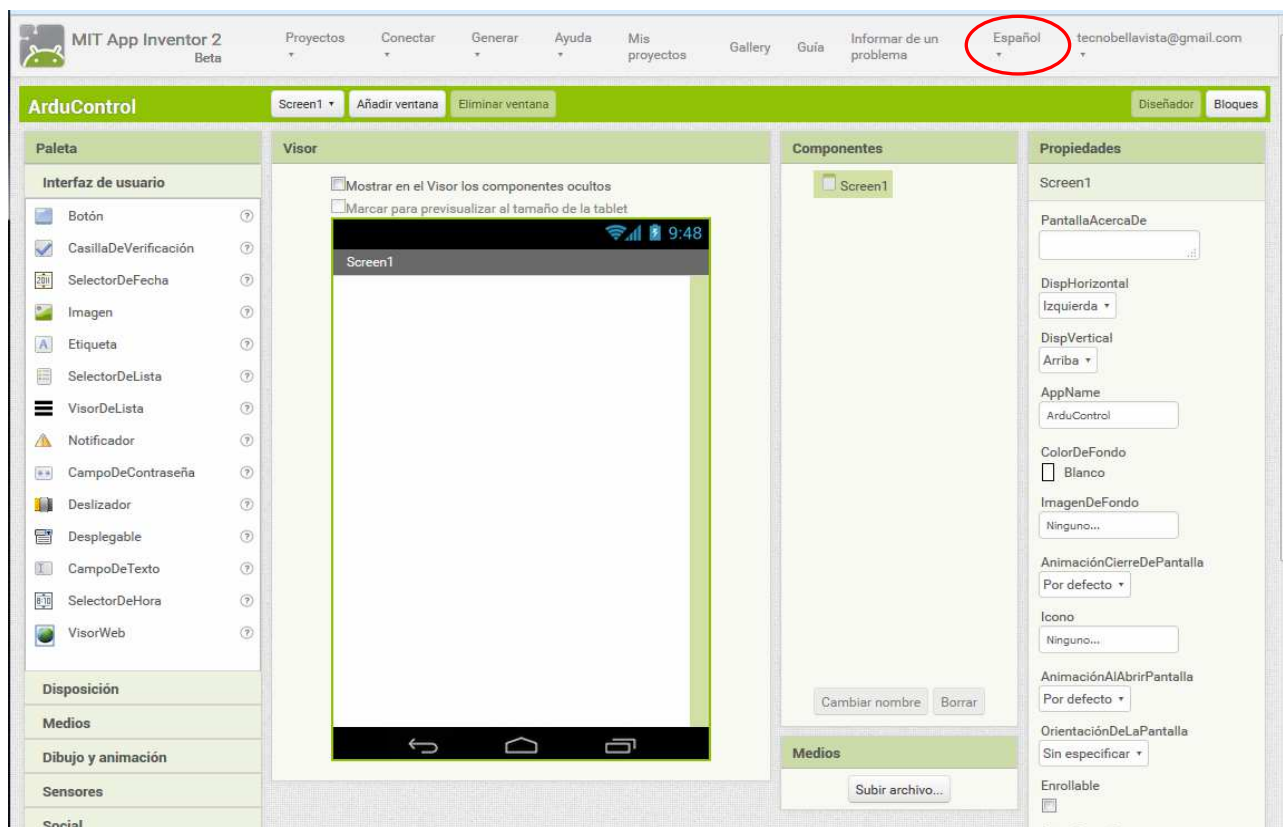
Haremos clic sobre el botón "Start new Project"



Nos pedirá el nombre del proyecto. Le podemos llamar, por ejemplo, ArduControl.

Nos aparece la pantalla de diseño de la aplicación. Aparece en inglés, pero podemos elegir el idioma desplegando la pestaña English y eligiendo Español.





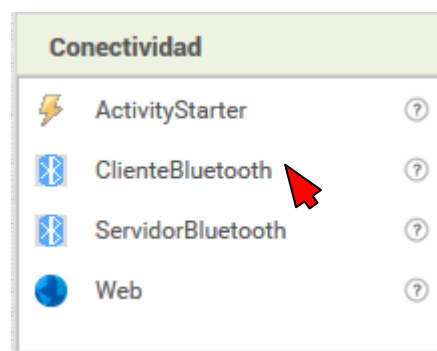
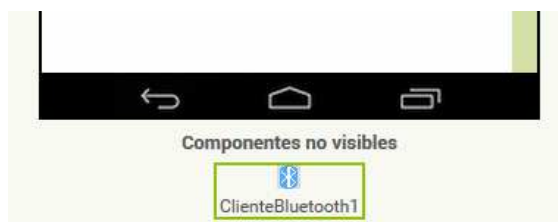
En la parte central está la sección **Visor**, que representa la pantalla de nuestro teléfono móvil.

En el lado izquierdo aparece la sección **Paleta**, donde se relacionan todos los elementos que podemos añadir a nuestra aplicación organizados por tipos en menús desplegables. Sólo hay que seleccionarlos y arrastrarlos hacia la pantalla..

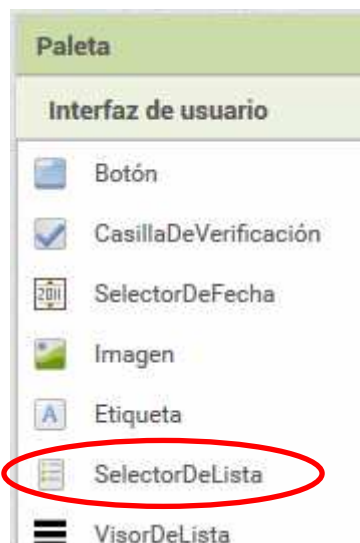
A la derecha del visor aparece la sección **Componentes**, que es donde aparecen los componentes que vamos añadiendo a nuestra aplicación. Aquí podemos cambiar su nombre o borrarlos.

A la derecha del todo aparece la sección **Propiedades**, donde cambiaremos las propiedades de los componentes.

Lo primero que necesitaremos es añadir un módulo bluetooth. Desplegamos la paleta **Conectividad**, y seleccionamos y arrastramos hacia la pantalla el módulo **ClienteBluetooth**. Como es un componente no visible nos aparece debajo de la pantalla.



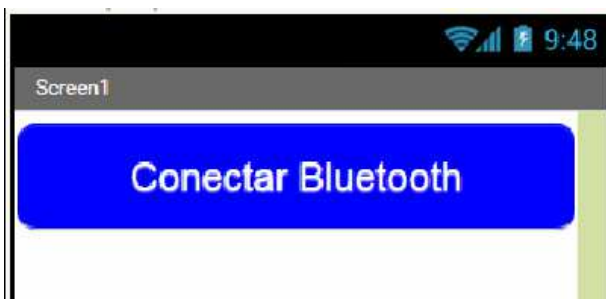
También nos aparecerá en la sección Componentes.



Ahora vamos a seleccionar y arrastrar a la pantalla un “Selector de lista” desde la paleta **Interfaz de usuario**.

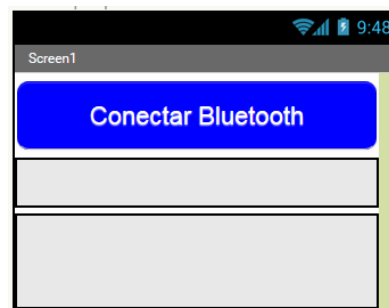


Podemos modificar las propiedades del selector de lista:

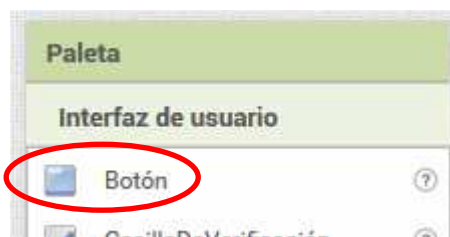


- ColorDeFondo: azul
- Tamaño de letra: 25
- Alto: 60 pixels
- Ancho: Ajustar al contenedor
- Forma: redondeado
- Texto: Conectar Bluetooth
- PosiciónDelTexto: centro
- ColorDeTexto: Blanco

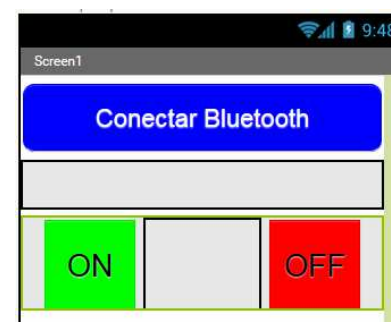
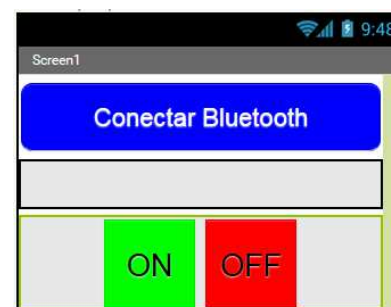
Ahora vamos a dejar un espacio en blanco debajo de este botón para separarlo de los que colocaremos luego. Para ello, desplegamos la paleta **Disposición** y arrastramos a la pantalla un “contenedor horizontal” de 40 píxeles de alto y de ancho ajustado al contenedor. Debajo de este contenedor, dispondremos otro “contenedor horizontal” de 80 píxeles de alto para incluir luego algunos botones. En su propiedad disposición horizontal elegimos Centro.



Dentro de este último contenedor vamos a colocar dos “botones” (uno de encendido y otro de apagado) de una altura de 80 píxeles y una anchura de 80. Para ello, los arrastramos desde la paleta **Interfaz de usuario** y les cambiamos el nombre a Botón_ON y Botón_OFF. También podemos asociar imágenes a los botones haciendo clic en la propiedad Imagen y subiendo un archivo.



Para separar un poco los botones, por estética, vamos a introducir entre ellos un contenedor horizontal de unos 100 píxeles de ancho y 80 de alto.



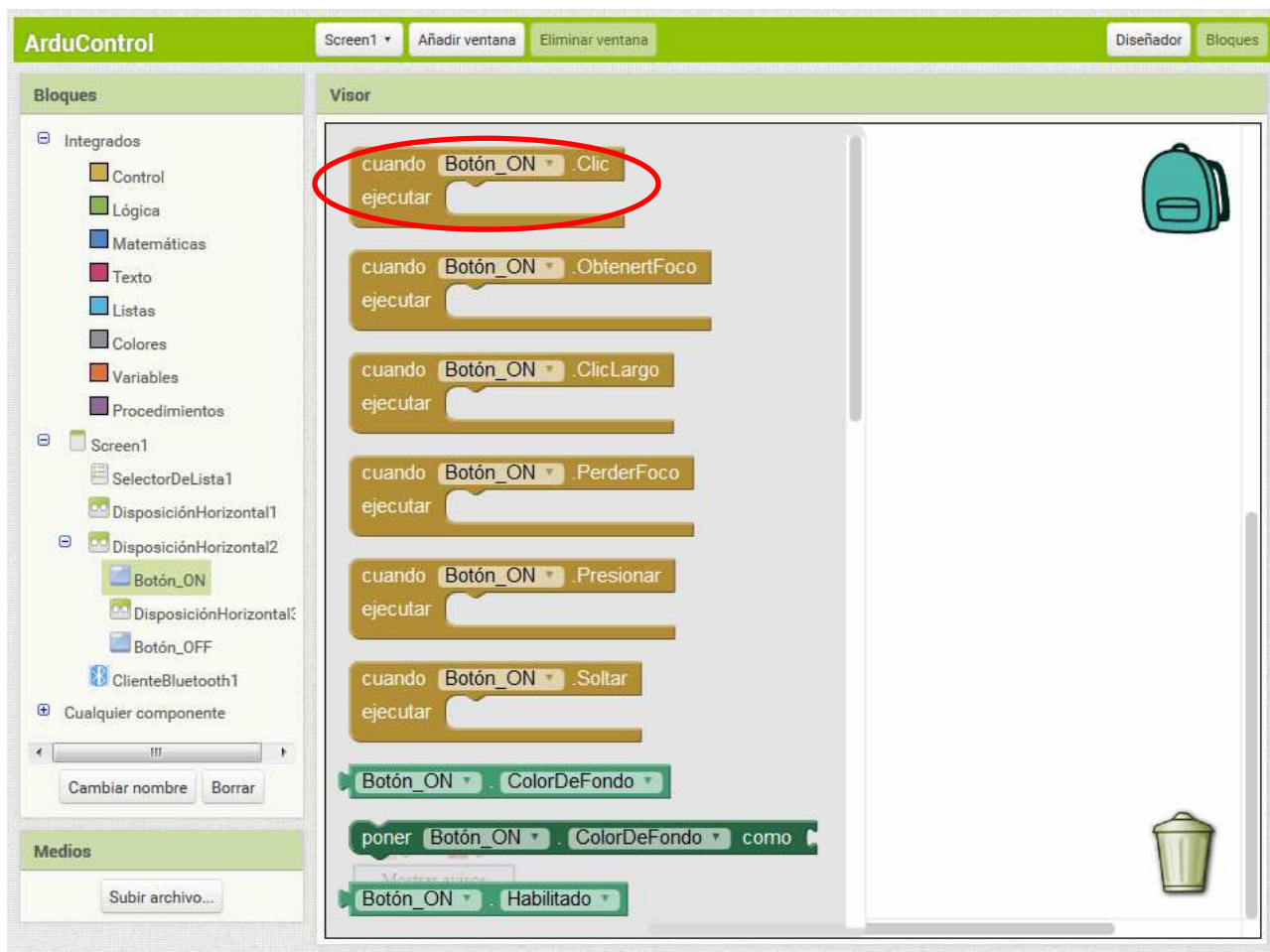
Ya hemos acabado el diseño de esta primera sencilla interfaz.

Ahora pasamos al diseño de los bloques de programación de todos estos botones. Para ello, hacemos clic en la pestaña Bloques, encima de la sección Propiedades.



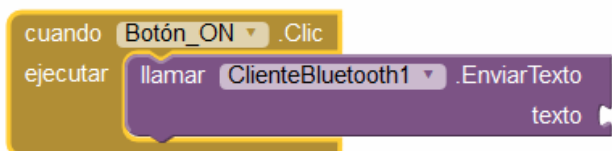
El programa consistirá en que cuando hagamos clic sobre el botón ON, mandaremos a Arduino el texto "1" y cuando hagamos clic sobre el botón OFF, enviaremos el texto "2".

Empezamos por el Botón_ON. Al seleccionarlo en la sección Bloques, aparece en la sección Visor todas las opciones asociadas al botón. Seleccionamos y arrastramos a la zona de programas la primera opción "cuando Botón_ON Clic... ejecutar".

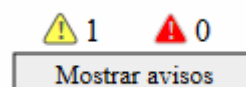


Nota: El icono de la papelera, situado en la esquina inferior derecha de la zona de programas, nos sirve para borrar bloques, arrastrándolos sobre ella. El icono de la mochila, situado en la esquina superior derecha nos sirve para copiar porciones de bloques en otros proyectos o pantallas.

Ahora tenemos que indicar qué queremos que haga el programa al hacer clic en el Botón_ON. Lo que queremos es que llame al cliente Bluetooth y envíe el texto "1". Para ello, seleccionamos el cliente Bluetooth en la sección Bloques y elegimos, de entre las opciones que aparecen, la opción "llamar ClienteBluetooth1. Enviar Texto":



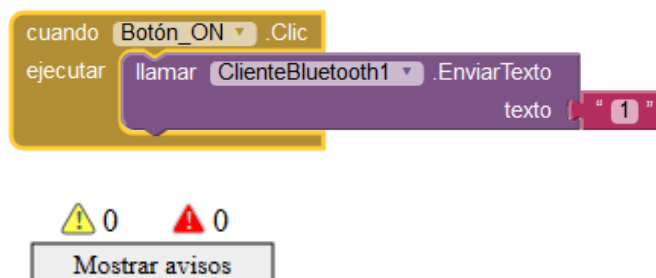
Observamos que nos aparece un aviso en la parte inferior izquierda de la zona de programas.



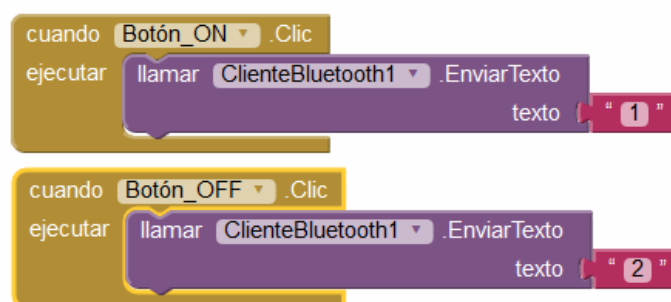
Si hacemos clic sobre el botón “Mostrar avisos”, aparece en el bloque al que se refiere el aviso el icono de aviso. Si hacemos clic sobre él nos informa de lo que ocurre. En este caso nos dice “Debe llenar todos los zócalos con bloques”, debido a que nos falta incluir el bloque de texto que queremos enviar.



Para incluir dicho texto, vamos a la opción Texto de la sección Bloques y elegimos el primero. Escribimos un “1”. Observamos que desaparecen los avisos.



Ahora podemos duplicar el programa del Botón_ON para el Botón_OFF, seleccionando éste último y cambiando el “1” por un “2”. Para duplicar, se hace clic con el botón secundario del ratón sobre el primer bloque de la porción y se selecciona Duplicar.



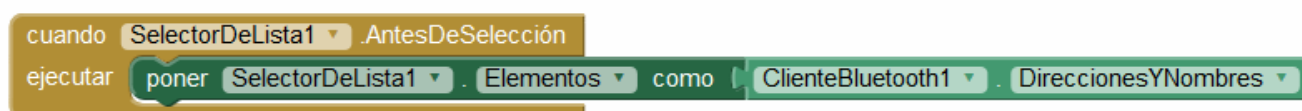
Vamos a programar ahora la conexión del módulo Bluetooth. Para ello, vamos a programar el Selector de lista que colocamos al principio. Lo seleccionamos en la sección Bloques y arrastramos a la zona de programa dos opciones: “Cuando SelectorDeLista.AntesDeSelección... ejecutar” y “Cuando SelectorDeLista.DespuésDeSelección... ejecutar”



Es decir, vamos a indicar lo que tiene que ocurrir antes de que se seleccione el botón y después de seleccionarse. Empezamos con Antes. De las opciones del Selector de Lista elegimos la opción “poner SelectorDeLista.Elementos como”.



A continuación, seleccionamos el cliente Bluetooth de la sección de bloques y elegimos la opción: “ClienteBluetooth.DireccionesYNombres”.

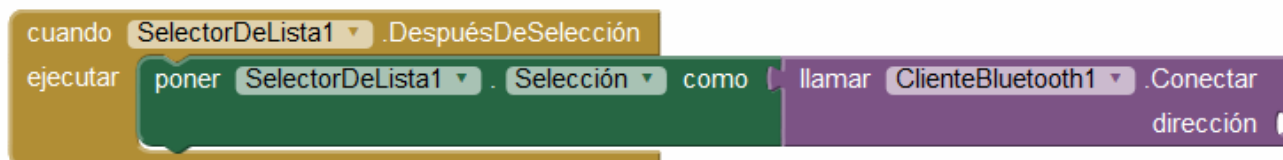


De esta forma, cuando hagamos clic sobre el Selector de Lista, antes de que seleccionemos un elemento de la lista, colocará en la lista los nombres de los módulos bluetooth que estén sincronizados con nuestro teléfono móvil Android.

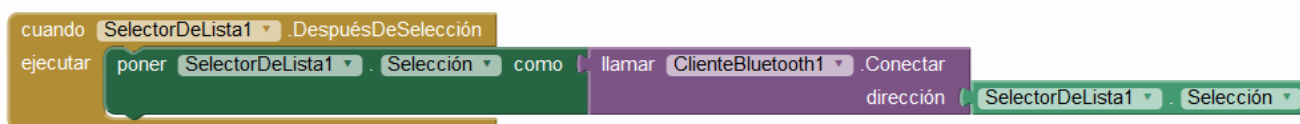
Pasamos ahora a la opción Después. De las opciones del selector de Lista elegimos la opción “poner SelectorDeLista.Selección como”.



A continuación, seleccionamos el cliente Bluetooth de la sección de bloques y elegimos la opción: “llamar ClienteBluetooth.Conectar...dirección”.

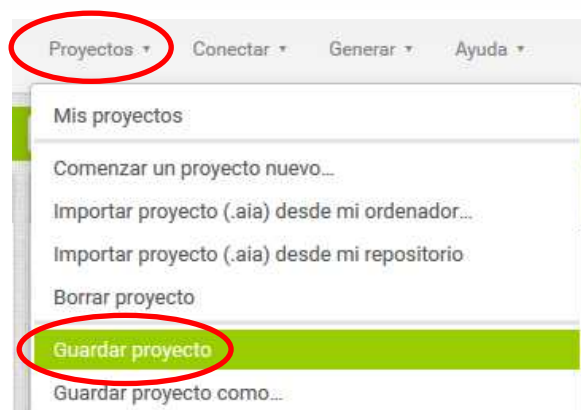
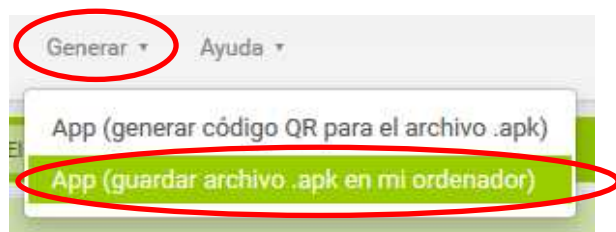


Ahora le indicamos el módulo bluetooth al que tiene que conectarse, que es el que hayamos seleccionado de la lista que nos ha ofrecido. Para ello, seleccionamos el Selector de Lista en la sección Bloques y elegimos la opción “SelectorDeLista.Selección”

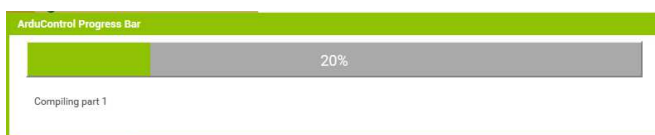


Ya tenemos terminado el proyecto. Lo guardamos haciendo clic en Proyectos > Guardar Proyecto.

Ahora vamos a **generar el instalador** (archivo .apk). Para ello, hacemos clic sobre Generar > App (guardar archivo .apk en mi ordenador).



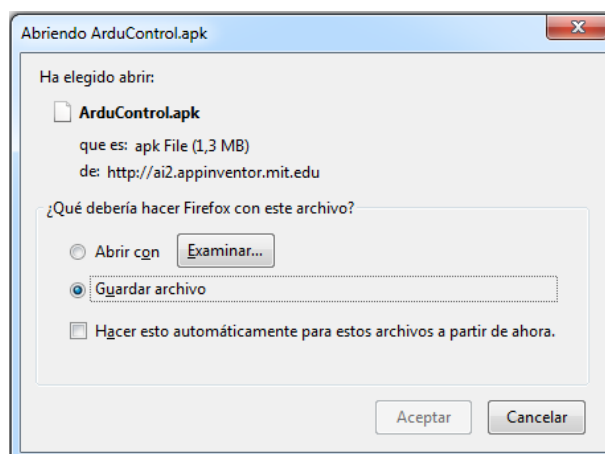
Empezará a construir el archivo instalador



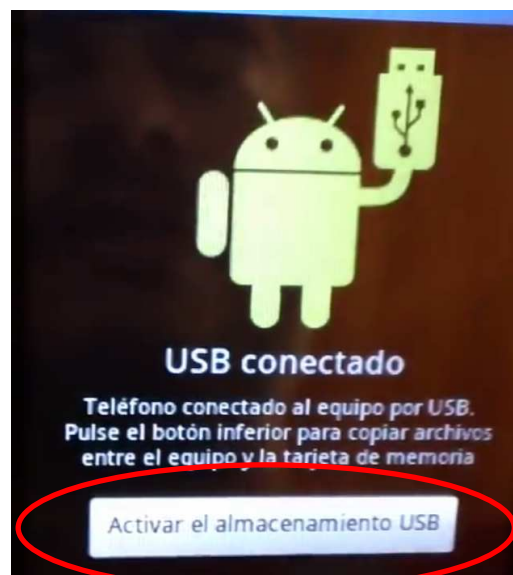
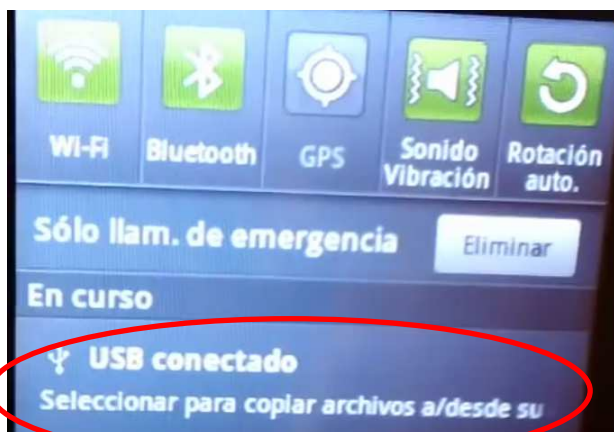
Y después nos pedirá dónde guardarlo:

A continuación tenemos que **montar este archivo en el teléfono móvil** con sistema operativo Android.

Para ello, lo conectamos a través del cable USB.

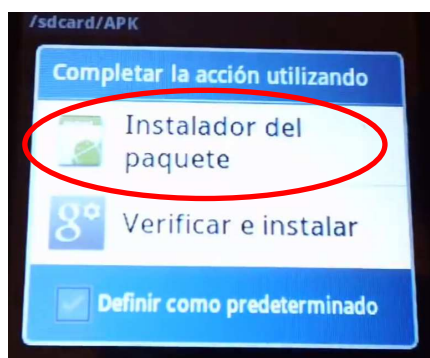
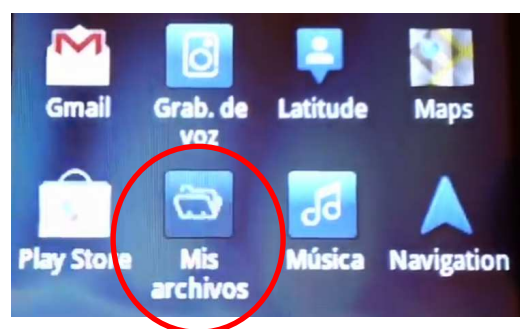


El ordenador debe reconocer al teléfono como dispositivo de almacenamiento (memoria USB). Si no lo reconoce directamente como memoria USB lo activamos.



Una vez podemos acceder a la memoria del móvil desde el ordenador, podemos crear una carpeta para los archivos .apk y en ella copiaremos nuestro archivo ArduControl.apk. A continuación podemos expulsar el teléfono del ordenador.

Ahora, en el móvil, utilizamos una aplicación que nos permita examinar los archivos internos del teléfono. Una típica es la que se llama “Mis Archivos”. A través de ella, entramos en la carpeta donde hemos copiado el archivo .apk y lo seleccionamos. Se nos abre el instalador y elegimos “Instalador de paquete” y Aceptar.



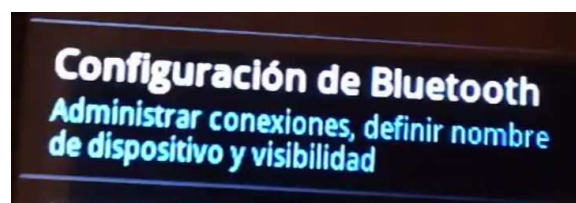
Una vez instalada, podemos abrir la aplicación. Si presionamos los botones ON u OFF nos saldrá un mensaje de error indicando que no está conectado a ningún módulo Bluetooth. Si presionamos sobre el Selector de Lista “Conectar Bluetooth” nos aparecerá una lista de los códigos de los módulos Bluetooth que tengamos sincronizados con nuestro móvil. Seleccionaremos el código correspondiente al módulo conectado a Arduino.

Lo primero que tenemos que hacer es **sincronizar el módulo Bluetooth conectado a Arduino con nuestro móvil.** Para ello:

Primero nos aseguramos de que tenemos activado el Bluetooth del móvil.

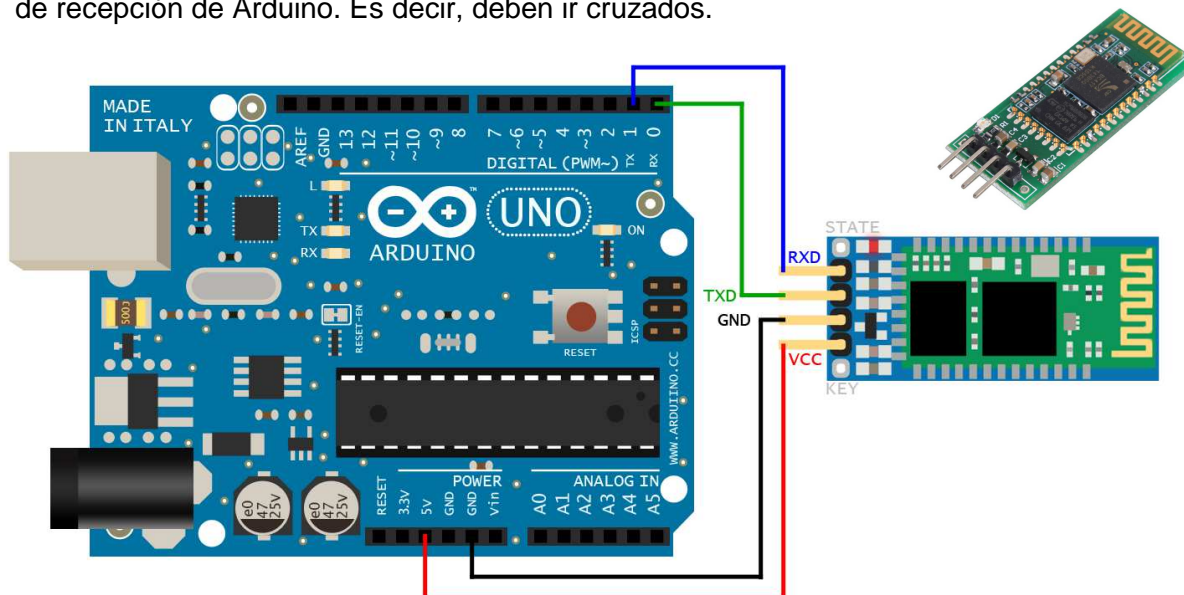
A continuación tenemos que vincular el módulo Bluetooth conectado a Arduino con el del móvil.

En el móvil entramos en Ajustes (o configuración) y seleccionamos Conexión Bluetooth.



A continuación, hacemos clic en buscar, para que busque los que se encuentren a su alcance.

Tenemos que tener encendido el módulo Bluetooth conectado a Arduino. De los cuatro pines del módulo Bluetooth, dos son de alimentación (Vcc y GND) que van conectados a los pines Vcc y GND de Arduino. Los otros dos son uno de recepción (RXD), que debe ir conectado al pin 1 (TX) de transmisión de Arduino, y el otro es el de transmisión (TXD) que debe ir conectado al pin 0 (RX) de recepción de Arduino. Es decir, deben ir cruzados.



Una vez detectado el módulo Bluetooth, lo seleccionamos y hacemos clic en Vincular. Nos pide un PIN que suele ser siempre 1234.

Una vez los tenemos vinculados, nos vamos a la aplicación y presionamos el Selector de Lista "Conectar Bluetooth" con lo que aparece la lista de elementos vinculados y elegimos el código del módulo Bluetooth conectado a Arduino.

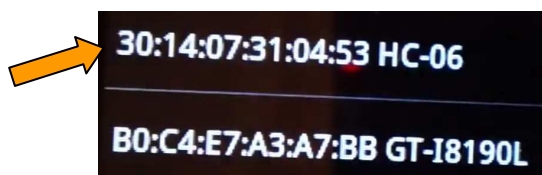
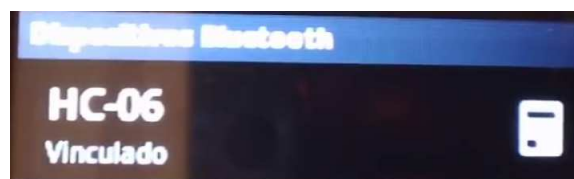
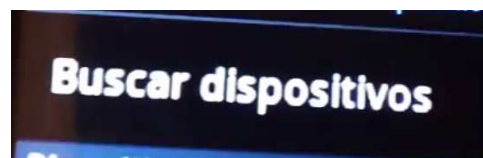
Cuando el LED del módulo Bluetooth conectado a Arduino deje de parpadear habrá establecido la conexión con el móvil, y ya podemos usar la aplicación.

Lógicamente, tenemos que cargar en Arduino un programa que al recibir un "1" por el puerto serie conecte algún elemento (por ejemplo un LED) y al recibir un "2" lo desconecte.

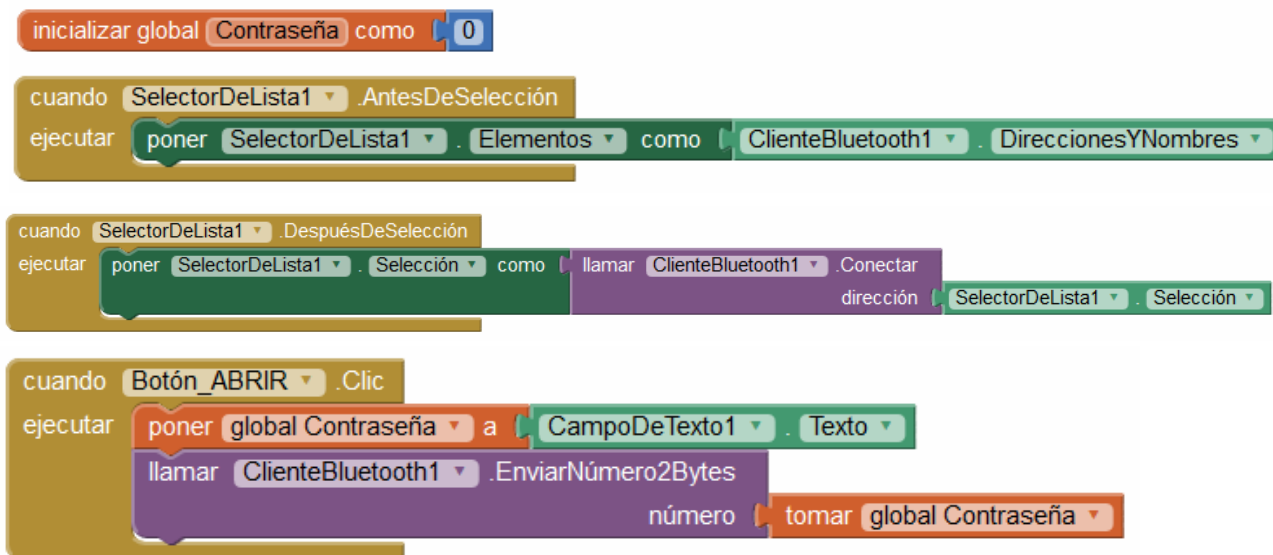
Por ejemplo, el programa adjunto enciende y apaga un LED conectado en el pin 13.

Nota: el módulo bluetooth debe estar desconectado de Arduino en el momento de cargar el programa.

```
int LED=13;
int estado=0;
void setup() {
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
}
void loop() {
  if(Serial.available()>0) {
    estado=Serial.read();
  }
  if(estado=='1') digitalWrite(LED, HIGH);
  if(estado=='2') digitalWrite(LED, LOW);
}
```



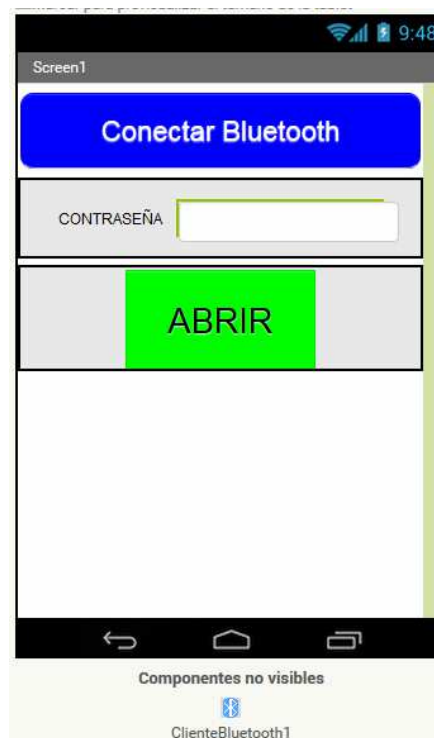
Ejemplo: aplicación que permite enviar un número de 2 bytes por Bluetooth a Arduino. Esto podría aplicarse, por ejemplo, al envío de una clave numérica para que Arduino, en caso de que la clave coincida con una que tiene guardada, ponga en marcha algo, por ejemplo abra una puerta o, en este caso, encienda un LED durante 3 segundos.



```
int LED=13;
int clave=0;

void setup() {
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
}

void loop() {
  if(Serial.available()>0){
    //lee el byte menos significativo y lo guarda
    clave=Serial.read();
    delay(5);
    //le suma a clave el byte más significativo
    //por eso lo multiplica por 256.
    clave=clave + 256 * Serial.read();
    if(clave==1234){ //si la clave introducida
      // es la correcta 1234
      digitalWrite(LED,HIGH);
      delay(3000);
      digitalWrite(LED,LOW);
    }
    clave=0;
  }
}
```



Aclaraciones

- 1.- Si el número fuera de 1 byte, sólo se podría mardar de 0 a 255. Para que la clave pueda ser de cuatro cifras se requiere al menos 2 bytes, que permite enviar de 0 a 65535.
- 2.- Al enviar un número de 2 bytes en serie, primero se envía el menos significativo y luego el más significativo.
- 3.- Serial.read() lee un byte cada vez.

Ejemplo: aplicación que combina las anteriores. Al presionar un botón, envía un texto (un carácter) para indicar el botón presionado. En caso de ser el '1', a continuación envía un número de 2 bytes con la clave escrita previamente en el campo "CONTRASEÑA", si la contraseña es correcta (en este caso 1234) acciona un servomotor; en caso del ser el '2' cambia el estado de las luces y en caso del ser el '3' cambia el estado del aire acondicionado.

```

inicializar global Contraseña como 0

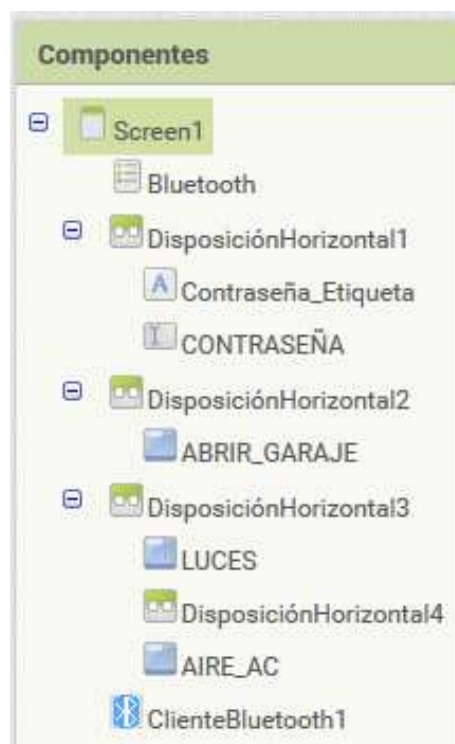
cuando Bluetooth .AntesDeSelección
ejecutar poner Bluetooth .Elementos como ClienteBluetooth1 .DireccionesYNombres

cuando Bluetooth .DespuésDeSelección
ejecutar poner Bluetooth .Selección como llamar ClienteBluetooth1 .Conectar
dirección Bluetooth .Selección

cuando ABRIR_GARAJE .Clic
ejecutar poner global Contraseña a CONTRASEÑA .Texto
llamar ClienteBluetooth1 .EnviarTexto
texto "1"
llamar ClienteBluetooth1 .EnviarNúmero2Bytes
número tomar global Contraseña

cuando LUCES .Clic
ejecutar llamar ClienteBluetooth1 .EnviarTexto
texto "2"

cuando AIRE_AC .Clic
ejecutar llamar ClienteBluetooth1 .EnviarTexto
texto "3"
    
```



```
#include <Servo.h>
Servo miservo;
int Luces=12;
int AireAco=11;
int boton=0;
int estado_luces=0;
int estado_aire=0;
int clave=0;

void setup() {
  Serial.begin(9600);
  pinMode(Luces, OUTPUT);
  pinMode(AireAco, OUTPUT);
  digitalWrite(Luces,LOW);
  digitalWrite(AireAco,LOW);
  miservo.attach(9);
  miservo.write(0);
}
void loop() {
  if(Serial.available()>0){
    boton=Serial.read();
  }
  if(boton=='1'){
    delay(5);
    clave=Serial.read();
    delay(5);
    clave=clave + 256 * Serial.read();
    if(clave==1234){
      miservo.write(90);
      delay(5000);
      miservo.write(0);
    }
    clave=0;
  }
  if(boton=='2'){
    if(estado_luces==0){
      digitalWrite(Luces,HIGH);
      estado_luces=1;
    }
    else{
      digitalWrite(Luces,LOW);
      estado_luces=0;
    }
  }
  if(boton=='3'){
    if(estado_aire==0){
      digitalWrite(AireAco,HIGH);
      estado_aire=1;
    }
    else{
      digitalWrite(AireAco,LOW);
      estado_aire=0;
    }
  }
  boton=0;
}
```