

3.2.6. Control por pantalla: Mover un robot

En esta actividad aprenderemos a utilizar la pantalla de nuestro dispositivo Android como si fuera un mando de radio control.

3.2.6.1. Programa para Android (código de Bloques en AppInventor)

- **Diseño**

- Utilizamos como base el programa de encender/apagar el led utilizado en la sección 3.2.4.1.
- Quitaremos el botón de encender el LED y en su lugar añadiremos 4 botones que llamaremos Boton_Avanza, Boton_Retrocede, Boton_Izquierda y Boton_derecha, quedando un diseño como el de la Figura 3.2.6-1

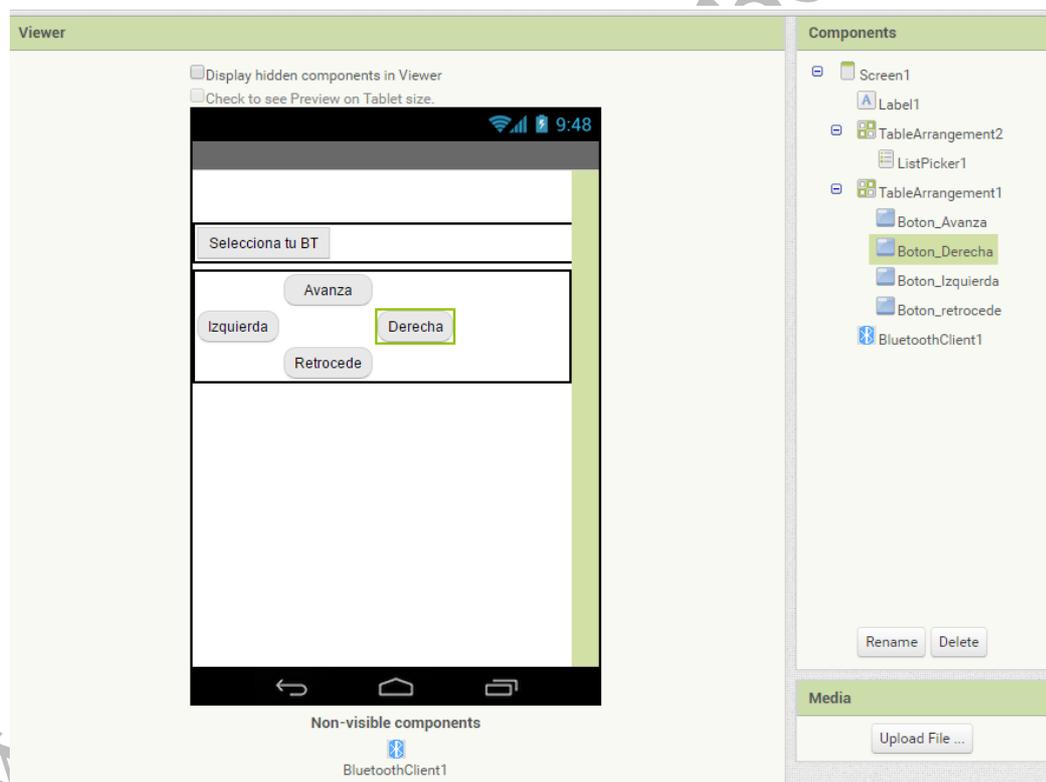


Figura 3.2.6-1 Diseño final de la aplicación de la actividad en AppInventor

- **Programación**

Ahora, en lugar de mandar un “1” ó un “0” por el Bluetooth, como hacíamos en la actividad 3.2.4, mandaremos un “1” si queremos que el robot avance, un “2” si queremos que retroceda, un “3” para que gire a la izquierda, un “4” para que gire a la derecha y un “0” para que pare. Como vemos en la Figura 3.2.6-2, utilizamos los bloques TouchDown de los botones. Este bloque se activa mientras toquemos un botón. Además, también utilizamos el bloque TouchUp, que se activa cuando

dejamos de pulsar el botón, para que cuando esto ocurra se le mande al robot un “o” y se pare.

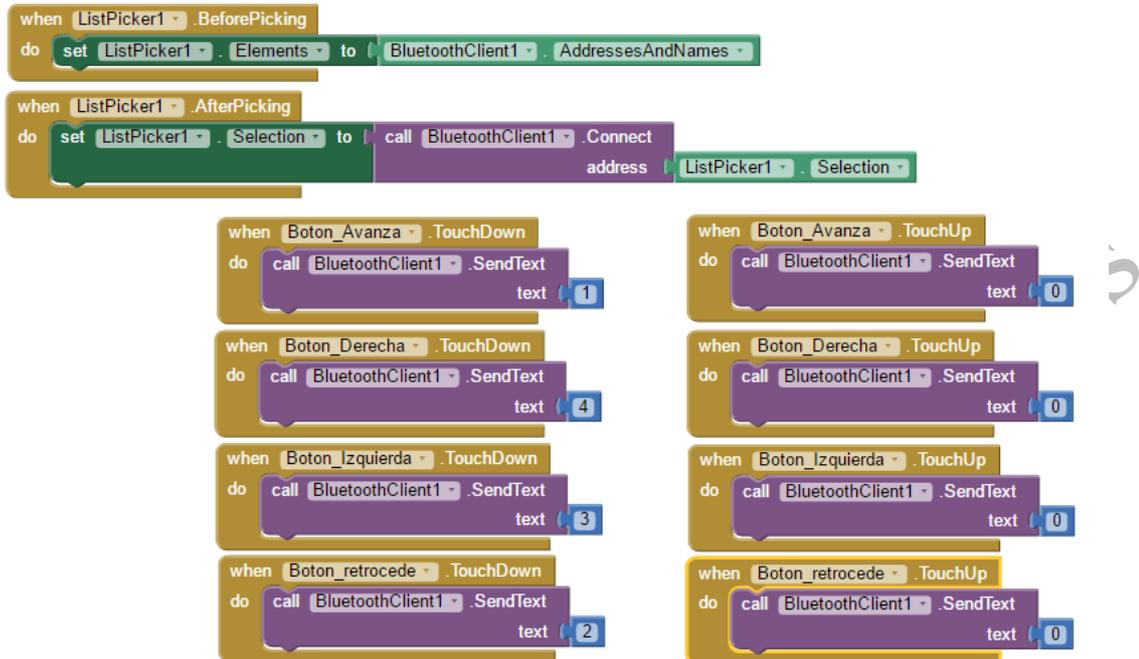


Figura 3.2.6-2 Programa en bloques en AppInventor de la actividad

3.2.6.2. Programa en Arduino (con el IDE de Arduino)

El código en Arduino es muy sencillo, cuando tengamos dato recibido por el puerto serie lo compararemos con los número del 0 al 4 y en función del resultado de la comparación, moveremos las ruedas para que el robot avance, retroceda, gire o se pare.

```
#include <Servo.h>

int input='0'; //variable para guardar la entrada por el serie (bluetooth)
Servo servo_continuo_izq;
Servo servo_continuo_der;

void setup() {

  servo_continuo_izq.attach(10);
  servo_continuo_der.attach(12);
  Serial.begin(19200); //recuerda cambiar esto en funcion del modulo
  //bluetooth que tengas
}

void loop() {
  ///comprobamos si se ha recibido algo del puerto serie
  if(Serial.available()>0)
  {
    input = Serial.read(); //si se ha recibido lo leemos
  }
  // avanza
  if(input == '1')
  {
    servo_continuo_der.write(180);
```

```
servo_continuo_izq.write(0);
}
//retrocede
else if(input == '2')
{
servo_continuo_der.write(0);
servo_continuo_izq.write(180);
}
//derecha
else if(input == '3')
{
servo_continuo_der.write(0);
servo_continuo_izq.write(0);
}
//izquierda
else if(input == '4')
{
servo_continuo_der.write(180);
servo_continuo_izq.write(180);
}
//parar
else if(input == '0')
{
servo_continuo_der.write(90);
servo_continuo_izq.write(90);
}
}
```

Ya sólo te queda subir los programas (a tu Android y a tu Arduino) y comprobar que eres capaz de controlar tu robot remotamente.