

# *Inteligencia Artificial*

## *Reporte Final*

*Elaborado por:*

*Osiel Chávez Flores*

*Rubén Gómez Barrientos*

*Docente:*

*M.I.S. José Antonio Flores Lara*

*Lugar y Fecha de Entrega:*

*Sombrerete Zacatecas a 15 de Diciembre Del 2015*



*ITSZO*

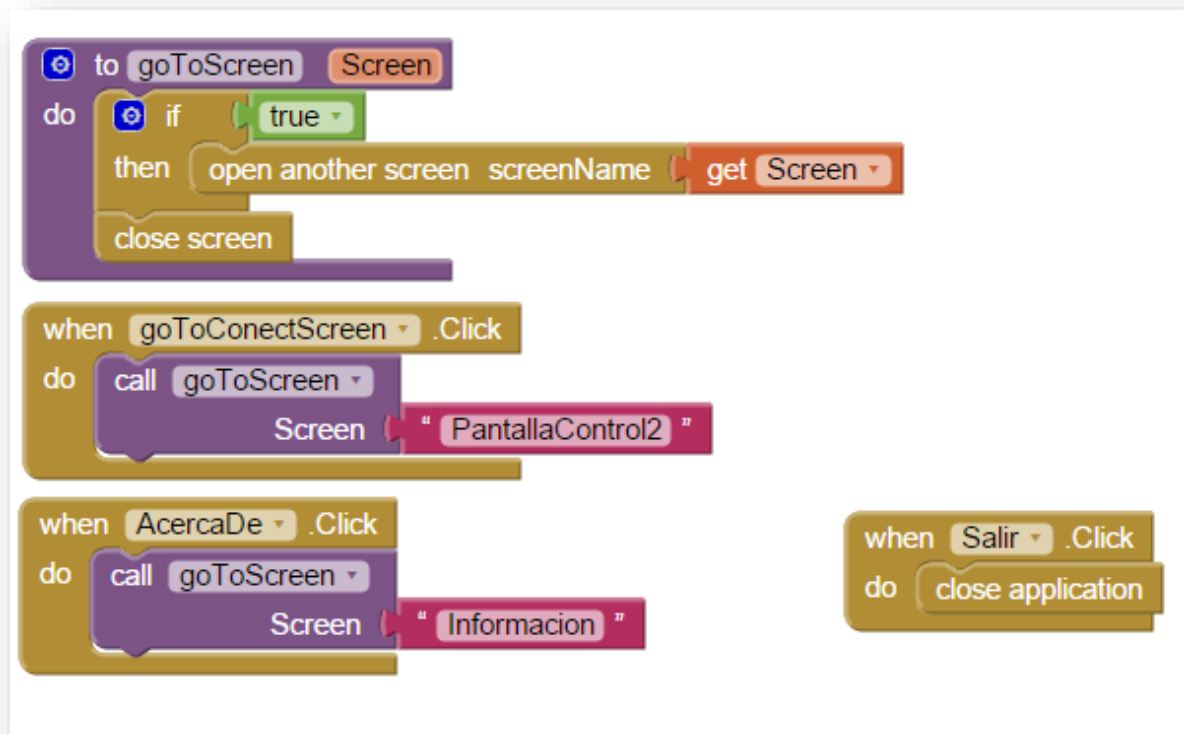
*Ingeniería En Sistemas Computacionales*

## Creación de la App

La aplicación fue creada en la plataforma de Google llamada App Inventor 2, dicha aplicación se creó en esta plataforma por los siguientes motivos, App inventor permite crear aplicaciones de manera visual, a lo que me refiero con esto es que esta plataforma brinda un entorno grafico para el diseño de la interfaz, además de que también brinda un entorno grafico para la lógica de la aplicación, ya que la lógica de la App se realiza mediante ensamblar bloques de instrucciones, a continuación se verá una imagen con la interfaz gráfica de una pantalla de la App y los bloques necesarios para el funcionamiento de dicha pantalla.



Como se puede ver la pantalla consta de 3 botones, el botón que dice pantalla de control que es el que nos va a mandar a la interfaz de control del brazo, el segundo botón que dice acerca de, nos manda a la pantalla donde está toda la información relacionada con el proyecto, por ejemplo institución educativa, carrera, materias, quien lo desarrollo, docente, entre otros datos más, por último el botón que tiene una imagen de apagado es el que nos permite cerrar la aplicación cuando ya terminemos de usarla, a continuación se mostraran los bloques necesarios para esta pantalla.



En el primer bloque se está creando un procedimiento para abrir una pantalla y cerrar la pantalla actual, ya que de lo contrario tendríamos muchas pantallas abiertas a y el funcionamiento del dispositivo no sería el óptimo.

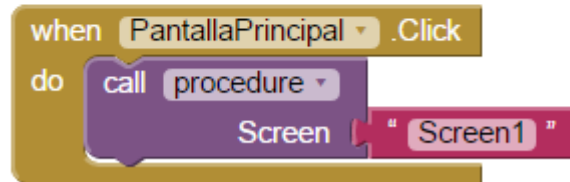
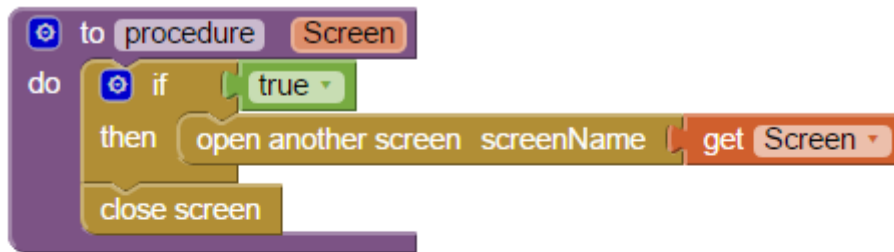
En el segundo bloque se está haciendo que cuando el usuario presione el botón llamado goToConectScreen, este mande llamar al procedimiento antes mencionado y abra la pantalla de control, el segundo bloque tiene el mismo funcionamiento, solo

que abre la pantalla llamada información y el último bloque es el encargado de cerrar la aplicación al momento de presionar el botón llamado salir.

La siguiente imagen presenta a la pantalla donde se encuentra la información relacionada con el proyecto.

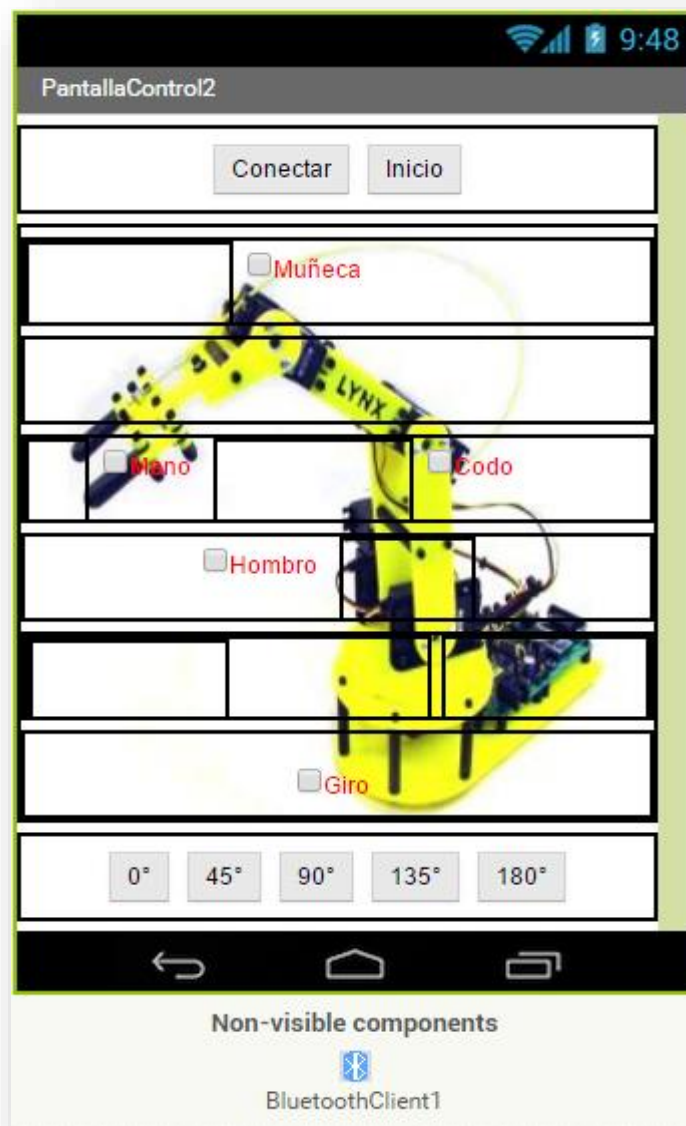


Como se puede ver, esta pantalla consta de varios labels para la información y un botón que nos manda a la pantalla principal. Ahora veremos los bloques que se necesitan en esta pantalla.

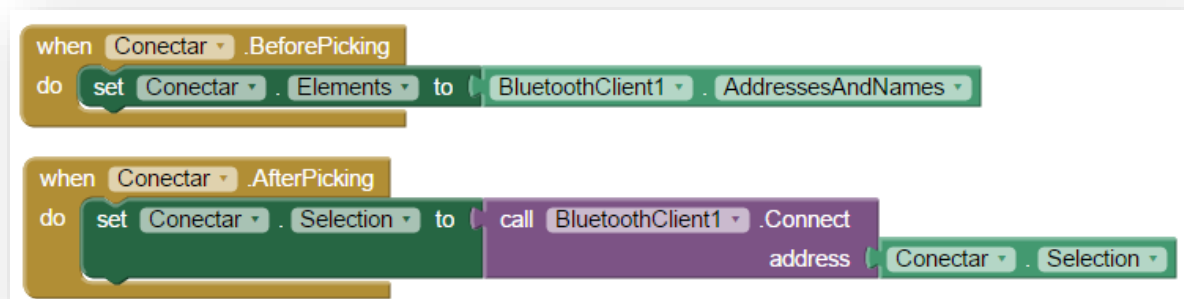


Como se puede ver, consta de pocos bloques ya que solo tiene un botón, el primer bloque al igual que la pantalla principal es un procedimiento que permite abrir una pantalla y cerrar la actual, y el segundo es el que al momento de presionar el botón llamado PantallaPrincioal manda llamar al procedimiento y lo ejecuta.

Por último se describirá la pantalla más importante, la pantalla encargada del control del brazo.



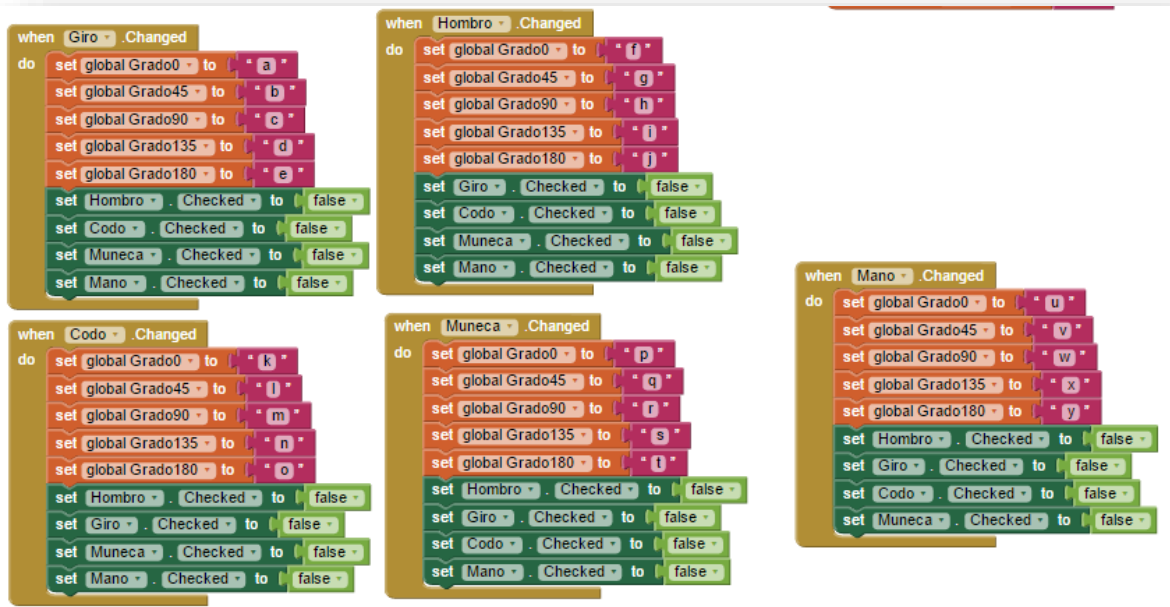
Como vemos, esta pantalla tiene más elementos que las demás, para empezar cuenta con cinco botones que son los encargados de enviar datos al arduino via bluetooth, también cinco check buttons que nos permiten seleccionar la parte del brazo que queramos manipular, un list picker que es el encargado de conectar el celular con el bluetooth arduino y por ultimo otro botón que dice Inicio que nos manda a la pantalla principal, a continuación se mostraran los bloques que se utilizan en esta pantalla.



En estos primeros dos bloques se está haciendo la conexión con el bluetooth, esta es la acción principal de estos bloques.

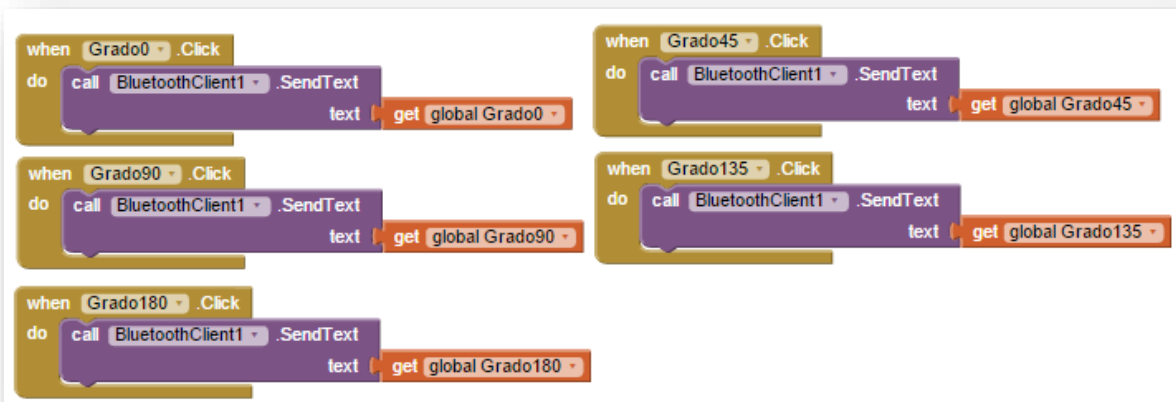


En estos siguientes bloques se están creando 5 variables globales inicializadas con un carácter cualquiera.

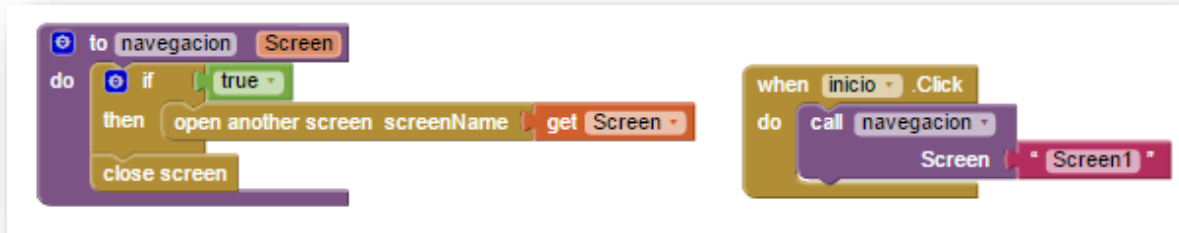


Los 5 bloques anteriores son los necesarios para manipular los check buttons, en estos bloques se está diciendo que cuando uno está presionado, los otros se desmarquen y además que le asigne un valor determinado a cada variable global anteriormente mencionada, por ejemplo cuando el check button llamado Hombro este marcado, la variable Grado0 tendrá un valor igual a "f", la variable Grado45 tendrá un valor igual a "g", la variable Grado90 tendrá un valor igual a "h", la variable global Grado135 tendrá un valor igual a "i" y la variable global Grado180 tendrá un valor de "j", además de que si los otros check buttons están marcados, se desmarcarán automáticamente.

En los siguientes 5 bloques se está diciendo que cuando se presione un botón, este mandará el valor de una determinada variable al arduino para que posteriormente ejecute una acción, a continuación se muestran esos bloques.



Los últimos bloques de esta pantalla son al igual que las demás pantallas, un procedimiento para abrir y cerrar pantallas, a continuación se muestran dichos bloques.



Pues en si así es como se creó la aplicación en la plataforma app inventor 2.

A continuación se dará una explicación del código de arduino.

La primera línea de código es para usar una librería para poder controlar el servo de una manera fácil, la primera línea se ve en la siguiente imagen:

```
#include <Servo.h>
```

Lo siguiente en el código son la declaración de variables, a continuación se ve dicha parte del código.

```
int estado=0;//se inicia la variable estado en 0, esta variable contendrá lo enviado por Bluetooth

//Variables de tipo Servo, que permitirán el control de un determinado servo
Servo myservo6;
Servo myservo3;
Servo myservo5;
Servo myservo9;
Servo myservo10;
```

Como se puede observar, la primera variable que se crea es la llamada estado, dicha variable es de tipo entero y esta inicializada en 0, esta variable juega un papel muy importante en el funcionamiento del brazo, ya que en dicha variable se estarán guardando los datos que el celular envíe vía bluetooth para de esta manera lograr mover el brazo de una manera fácil y cómoda.

Después se ve que se crean 5 variable de tipo Servo, en donde cada una se llama myservo pero con la diferencia de un número al final de cada una, esto es con el



propósito de identificar cual variable va a estar asignada a una salida física de la placa arduino.

El siguiente bloque de instrucciones, es donde se establecen las características de las variables, veámoslo.

```
void setup(){  
  
    Serial.begin(9600); //Se inicializa la comunicación serial con un valor de 9600  
  
    //Se establecen las salidas de las variables tipo Servo  
    myservo3.attach(3); //Salida 3  
    myservo5.attach(5); //Salida 5  
    myservo6.attach(6); //Salida 6  
    myservo9.attach(9); //Salida 9  
    myservo10.attach(10); //Salida 10  
}
```

Como vemos la primera línea dentro del bloque void setup es Serial.begin(9600);, esta línea es para que la comunicación serial comience en 9600, después de esta línea se ven las variables de tipo servo, como se mencionó anteriormente, estas variables se nombraron con un formato determinado para poder identificar cual variable va a cual salida del arduino, entonces en esta parte se están estableciendo dichas salidas, por ejemplo myservo3 se establece a la salida 3, myservo5 a la salida 5 y así sucesivamente se pueden identificar con los nombres de las variables.

Lo siguiente en el código, es el bloque void loop, en donde se estarán ejecutando las ordenes enviadas por el celular, lo primero dentro de este bloque es una condicional en donde se está evaluando si la variable estado contiene un valor diferente de 0, dicha condicional se ve en la siguiente imagen:

```
if(Serial.available()>0){  
    estado = Serial.read();  
}
```

La última parte del bloque void loop, es la encargada de asignar un valor para un determinado servo, dependiendo del valor recibido, esta parte se compone por

varias condicionales if, en donde si el celular envía un valor determinado, dicho valor tiene que caer en una de estas condicionales para que un servo establecido dentro de la condición se mueva a un determinado ángulo, el código de estas condiciones se muestra a continuación:

```
if (estado == 'a'){  
    myservo3.write(0);  
}  
if (estado == 'b'){  
    myservo3.write(45);  
}  
if (estado == 'c'){  
    myservo3.write(90);  
}  
if (estado == 'd'){  
    myservo3.write(135);  
}  
if (estado == 'e'){  
    myservo3.write(180);  
}  
if (estado == 'f'){  
    myservo5.write(0);  
}  
if (estado == 'g'){  
    myservo5.write(45);
```

```
    }  
  
    if (estado == 'h'){  
        myservo5.write(90);  
    }  
  
    if (estado == 'i'){  
        myservo5.write(135);  
    }  
  
    if (estado == 'j'){  
        myservo5.write(180);  
    }  
  
    if (estado == 'k'){  
        myservo6.write(0);  
    }  
  
    if (estado == 'l'){  
        myservo6.write(45);  
    }  
  
    if (estado == 'm'){  
        myservo6.write(90);  
    }  
  
    if (estado == 'n'){  
        myservo6.write(135);  
    }  
  
    if (estado == 'o'){
```

```
        myservo6.write(180);  
    }  
    if (estado == 'p'){  
        myservo9.write(0);  
    }  
    if (estado == 'q'){  
        myservo9.write(45);  
    }  
    if (estado == 'r'){  
        myservo9.write(90);  
    }  
    if (estado == 's'){  
        myservo9.write(135);  
    }  
    if (estado == 't'){  
        myservo9.write(180);  
    }  
    if (estado == 'u'){  
        myservo10.write(0);  
    }  
  
    if (estado == 'v'){  
        myservo10.write(45);  
    }  
}
```

```
}  
  
if (estado == 'w'){  
    myservo10.write(90);  
}  
  
if (estado == 'x'){  
    myservo10.write(135);  
}  
  
if (estado == 'y'){  
    myservo10.write(180);  
}
```

Pues en si así es como funciona la aplicación y el código arduino, para una mayor apreciación del código puede visitar el siguiente link de Dropbox, en donde encontrará el código arduino completo.

<https://db.tt/pMFDuJNs>

Además si desea probar la aplicación también la puede descargar del siguiente link de Dropnox:

<https://db.tt/tyxrcqYV>

Para descargar el proyecto de App Inventor 2 y poder modificarlo visitar el siguiente link:

<http://db.tt/JvQweexq>