

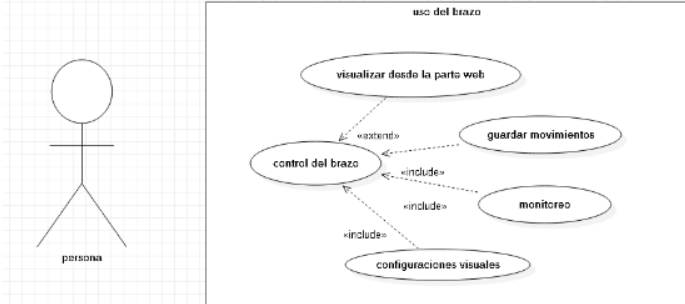
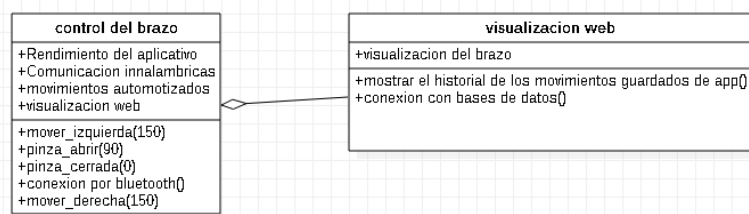
Manual técnico

Brazo robótico


**Instructor:
Hernando Rozo**

**Yerli Cardenas Martinez, Gabriela Bermudez
Veloza**



análisis y desarrollo de sistemas de información
instituto sena regional guaviare
2022

No	Descripción	Detalle												
1	Alcance	el diseño, construcción y control de un sistema de brazo robótico dedicado a la manipulación de objetos, mediante el uso de software de simulación y elementos electromecánicos.												
2	Casos de uso	<div><div></div><table><thead><tr><th>Actor</th><th>Nombre del sistema</th><th>Pre-condición</th><th>Pos-condición</th><th>Flujo básico</th><th>Flujo alternativo</th></tr></thead><tbody><tr><td>persona</td><td>control del brazo</td><td><ul style="list-style-type: none">- Estar conectada a bluetooth al app- Tener el brazo en posición 0</td><td>controlar el brazo a gusto del usuario</td><td><ul style="list-style-type: none">- Tener el dispositivo móvil con app- Conectar la aplicación por medio de bluetooth al brazo.- Ejecutar los movimientos queridos y guardarlos.</td><td><ul style="list-style-type: none">- Descargar la app</td></tr></tbody></table></div>	Actor	Nombre del sistema	Pre-condición	Pos-condición	Flujo básico	Flujo alternativo	persona	control del brazo	<ul style="list-style-type: none">- Estar conectada a bluetooth al app- Tener el brazo en posición 0	controlar el brazo a gusto del usuario	<ul style="list-style-type: none">- Tener el dispositivo móvil con app- Conectar la aplicación por medio de bluetooth al brazo.- Ejecutar los movimientos queridos y guardarlos.	<ul style="list-style-type: none">- Descargar la app
Actor	Nombre del sistema	Pre-condición	Pos-condición	Flujo básico	Flujo alternativo									
persona	control del brazo	<ul style="list-style-type: none">- Estar conectada a bluetooth al app- Tener el brazo en posición 0	controlar el brazo a gusto del usuario	<ul style="list-style-type: none">- Tener el dispositivo móvil con app- Conectar la aplicación por medio de bluetooth al brazo.- Ejecutar los movimientos queridos y guardarlos.	<ul style="list-style-type: none">- Descargar la app									
3	Diagramas uml	<div><div></div></div>												

		<div> <div> control del brazo </div> <div> +Rendimiento del aplicativo +Comunicacion innalambricas +movimientos automotizados +visualizacion web </div> <div> +mover_izquierda(150) +pinza_abrir(90) +pinza_cerrada(0) +conexion por bluetooth() +mover_derecha(150) </div> </div> <div> <div> visualizacion web </div> <div> +visualizacion del brazo </div> <div> +mostrar el historial de los movimientos guardados de app() +conexion con bases de datos() </div> </div>
4	Modelo racional de la bases de datos	<div> <div> bd_brazo_robotico tb_registro </div> <div> docper : int(250) antebrazo : int(250) base : int(250) muneca_vertical : int(250) muneca_griro : int(250) pinzas : int(250) brazo : int(250) </div> </div> <div> <div> bd_brazo_robotico tb_historial </div> <div> id : int(250) usu : int(250) descripcion : varchar(500) fecha : timestamp </div> </div> <div> <div> bd_brazo_robotico tb_login </div> <div> id : int(250) id_usu : int(250) pass : varchar(500) docper : int(250) estado : int(11) </div> </div> <div> <div> bd_brazo_robotico tb_persona </div> <div> docper : int(250) nombres : varchar(500) apellidos : varchar(500) celular : varchar(500) email : varchar(500) estado : int(11) </div> </div> <div> <div> bd_brazo_robotico tb_registro </div> <div> # docper : int(250) # antebrazo : int(250) # base : int(250) # muneca_vertical : int(250) # muneca_griro : int(250) # pinzas : int(250) # brazo : int(250) </div> </div> <div> <div> bd_brazo_robotico tb_login </div> <div> # id : int(250) # id_usu : int(250) # pass : varchar(500) # docper : int(250) # estado : int(11) </div> </div>

		
5	descripción del proyecto	<p>Brazo robotico con app y página web.</p> <p>Cada pieza del brazo robótico está diseñada en la app 3D builder y estas mismas fueron impresas en una impresora 3D.</p> <p>La primera fase fue calibrar los motores con las piezas de brazo, para que abran y cierran o se muevan con libertad, para lo cual se hizo un programa sencillo en arduino, y se sube al arduino lo para que el motor vaya del grado 0 al 180 y de vuelta, y se van ensamblando parte a parte comprobando la calibración de los motores y movimientos</p> <p>La segunda fase es armar el código parte por parte para mover los motores como se quería que se movieran. Luego armar incluyendo el módulo bluetooth haciendo que cada parte responda de forma individual al pulso del bluetooth sin que los motores se fuercen y esto arroja la velocidad a la que se debe mover cada motor. Finalmente incluir cada bloque de código y su rapidez en los movimientos, También poner las configuraciones según el pulso y la onda del bluetooth y con eso se terminaría por el momento la parte de la programación.</p> <p>La tercera fase es el circuito, se hace una alimentación detallada forma que se ejecuten los motores y para optimizar los recursos se usan motores como resistencias, así la electricidad que ya pasa con menos fuerza se usa para alimentar los módulos y el arduino ya que no usan tantos amperios(fuerza en la electricidad)</p> <p>Y así se define la corriente y la intensidad a usar.</p> <p>Cuarta fase se hace un prototipo de app desde la que se va usar el brazo y se define la posición inicial para grabarla en el arduino.</p> <p>La fase final es definir los detalles de la app y ponerla la más estética posible, y también acomodar los cables para que no queden todos resueltos ya estaría listo.</p> <p>Se dispone de una programación en arduino, un ciclo for donde la idea es que el motor va de 0 a 180, evalúe dónde está y lo envíe de regreso, mientras el arduino está haciendo el motor tiene la pieza ensamblada y de acuerdo al movimiento si se estanca y queda vibrando como si quisiera moverse más y no puede, hay que soltar la pieza y acomodarla de acuerdo con el movimiento del motor.</p>

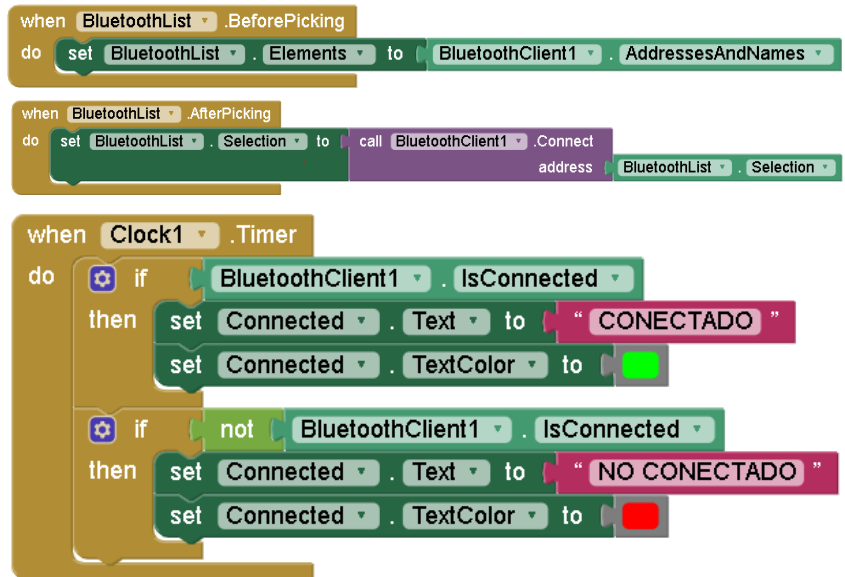
		<p>También en la programación hay puntos máximo y mínimos, donde el motor rota, y uno lo limita para que llegue al cierto punto sin que el motor tenga tanto trabajo.</p> <p>Así se puede optimizar el movimiento del motor sin exigir más potencia en la corriente</p> <p>Código Arduino:</p> <p>Primero, incluimos la librería SoftwareSerial y la biblioteca servo para la comunicación en serie con el módulo Bluetooth. Luego definimos los seis servos, el módulo bluetooth HC-06 y algunas variables para almacenar la posición actual y anterior de los servos y una matriz para almacenar la posición o el número de pasos en modo automático.</p> <p>En la parte de configuración, inicializamos los servos y el módulo bluetooth para mover el brazo a la posición inicial, usamos la función write() para mover los servos a cualquier posición de 0 a 180 grados y en la parte de bucle, usamos la función bluetooth.disponible(), luego verificamos si hay datos entrantes desde la app del celular, si es así, usando la función readString(), leimos los datos como una cadena basada en los datos recibidos y los almacenamos los datos en una variable y le decimos al brazo que hacer.</p> <p>La interfaz gráfica web está diseñada en HTML, CSS3, PHP.</p> <p>Los archivos donde esta el codigo están organizados por carpetas como se muestra en la imagen:</p>
--	--	--

		<div><div>✓ BRAZO_ROBOTICO</div><div><div>> desktop</div><div>> img</div><div>> links</div><div>> login</div><div>> registro</div><div>> registro_movimientos</div><div>> sesiones</div><div>> style</div><div> conexionBase.php</div><div> funciones.php</div></div></div> <div>cada carpeta contiene su archivo correspondiente, como se muestra en la imagen:</div>
--	--	--

		<p>delate es el archivo que elimina los datos de la tabla.</p> <p>El segundo archivo es registro donde está la conexión de la base de datos, toma los datos de toma los datos del apk de los movimientos que son en grados los guarda en una variable, después de eso trae los datos de la base datos y realiza una consulta para insertar esos datos en la tabla de registro.</p> <p>El tercer archivo contiene contiene la parte visual de la tabla y el header.</p> <ol style="list-style-type: none"> 3. La carpeta de IMG tiene las imágenes de la página web. 4. La carpeta de links contiene solo dos scripts de los iconos. 5. La carpeta de 'login' tiene checklogin donde esta la parte de lógica el logeo, ella comprueba si los datos que ingresa el usuario son iguales a los de los de la base de datos y así poder dar ingreso y redireccionar, si las credenciales no coinciden mandara por la url un error 401. El archivo de login es donde esta el formulario donde estan los inputs de ingresar credenciales y el header que tiene el logo, la imagen del logo y el link de registro y login. 6. la carpeta de 'registro', contiene el checkregistro, es la parte lógica, recibe los parámetros de entrada de registro trae el archivo de funciones para encriptar la contraseña, creamos un variable que contenga los campos de la base de datos, otra variable donde tiene los datos de ingreso y realiza una consulta de inter para subir los datos a las tablas de registro y login. El archivo de registro contiene el formulario de entrada, es decir, es la parte visible donde el usuario podrá ingresar los datos para poder registrarse y ser usuario de web. 7. La carpeta de 'style' tiene tres archivos donde están la parte de css que es el embellecimiento de las páginas login, registro, desktop. 8. Por último está la carpeta de 'vistas' donde están los inputs de la toma de los datos de la apk. 9. están dos archivos fuera de todas las carpetas, que son: conexión que es la conexión de la base de datos. y funciones que es el archivo de encriptar y desencriptar las contraseñas.
--	--	---

6	documentación del código fuente	<p>Código de arduino:</p> <pre> #include <SoftwareSerial.h> // biblioteca para la comunicacion serial del bluetooth #include <Servo.h> // biblioteca servos servo servo01; servo servo02; servo servo03; //definicion de los servos servo servo04; servo servo05; servo servo06; SoftwareSerial Bluetooth (3, 4); //Arduino (RX, TX) - HC-06 Bluetooth (TX, RX) int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos; //posicion actual int servo1Poss, servo2Poss, servo3Poss, servo4Poss, servo5Poss, servo6Poss; //posicion anterior int servo01sp [50], servo02sp [50], servo03sp [50], servo04sp [50], servo05sp [50], servo06sp [50]; //almacenar posiciones int speedDelay = 20; int indice = 0; String dataIn = "" ; void setup () { servo01.write (5); servo02.write (6); servo03.write (7); servo04.write (8); servo05.write (9); servo06.write (10); Bluetooth.begin (38400); //velocidad en baudios predeterminada del modulo Bluetooth.setTimeout (1); delay(20); //posicion inicial del brazo servo1Poss = 90; servo01.write (servo1Poss); servo2Poss = 150; servo02.write (servo2Poss); servo3Poss = 35 ; servo03.write (servo3Poss); servo4Poss = 140; servo04.write (servo4Poss); servo5Poss = 85; servo05.write (servo5Poss); servo6Poss = 80; servo06.write (servo6Poss); } //verificar si hay datos entrantes if (Bluetooth.available () > 0) { dataIn = Bluetooth.readString () ; // lee los datos como una cadena } </pre>
---	---------------------------------	---

Código por bloques de la apk: conexión por bluetooth



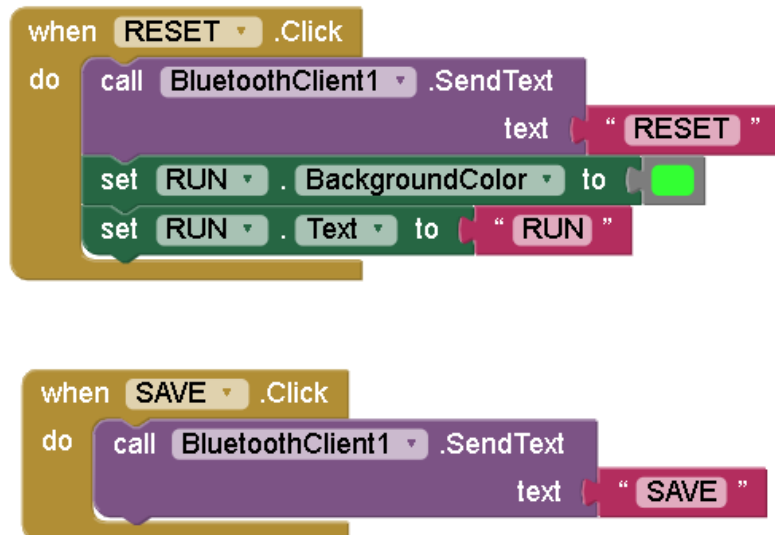
```
when BluetoothList .BeforePicking
do
  set BluetoothList . Elements to BluetoothClient1 . AddressesAndNames

when BluetoothList .AfterPicking
do
  set BluetoothList . Selection to call BluetoothClient1 .Connect
  address BluetoothList . Selection

when Clock1 .Timer
do
  if BluetoothClient1 . IsConnected
  then
    set Connected . Text to " CONECTADO "
    set Connected . TextColor to green
  if not BluetoothClient1 . IsConnected
  then
    set Connected . Text to " NO CONECTADO "
    set Connected . TextColor to red
```

The code consists of three event-driven blocks. The first block, 'when BluetoothList .BeforePicking', sets the 'Elements' property of 'BluetoothList' to 'BluetoothClient1 . AddressesAndNames'. The second block, 'when BluetoothList .AfterPicking', sets the 'Selection' property of 'BluetoothList' to the result of calling 'BluetoothClient1 .Connect' with the 'Selection' property of 'BluetoothList' as the address. The third block, 'when Clock1 .Timer', contains two conditional 'if' statements. The first 'if' statement checks if 'BluetoothClient1 . IsConnected' is true; if so, it sets the 'Text' property of 'Connected' to ' CONECTADO ' and its 'TextColor' to green. The second 'if' statement checks if 'BluetoothClient1 . IsConnected' is false; if so, it sets the 'Text' property of 'Connected' to ' NO CONECTADO ' and its 'TextColor' to red.

Botones



```
when RESET .Click
do
  call BluetoothClient1 .SendText
  text " RESET "
  set RUN . BackgroundColor to green
  set RUN . Text to " RUN "

when SAVE .Click
do
  call BluetoothClient1 .SendText
  text " SAVE "
```

The code consists of two event-driven blocks. The first block, 'when RESET .Click', calls 'BluetoothClient1 .SendText' with the text ' RESET ', sets the 'BackgroundColor' property of 'RUN' to green, and sets the 'Text' property of 'RUN' to ' RUN '. The second block, 'when SAVE .Click', calls 'BluetoothClient1 .SendText' with the text ' SAVE '.

```

initialize global url_form to " https://zoratha.000webhostapp.com/brazo_robotico..."
when registrar .Click
do
  set Web1 . Url to join
    get global url_form
    "?servero_1="
    Servo_01 . ThumbPosition
    "&servero_2="
    Servo_02 . ThumbPosition
    "&servero_3="
    Servo_03 . ThumbPosition
    "&servero_4="
    Servo_04 . ThumbPosition
    "&servero_5="
    Servo_05 . ThumbPosition
    "&servero_6="
    Servo_06 . ThumbPosition
  call Web1 . Get

```

```

when RUN .Click
do
  if RUN . Text = " RUN "
  then
    set RUN . BackgroundColor to 
    call BluetoothClient1 . SendText
      text " RUN "
    set RUN . Text to " PAUSE "
  else if RUN . Text = " PAUSE "
  then
    set RUN . BackgroundColor to 
    call BluetoothClient1 . SendText
      text " PAUSE "
    set RUN . Text to " RUN "

```

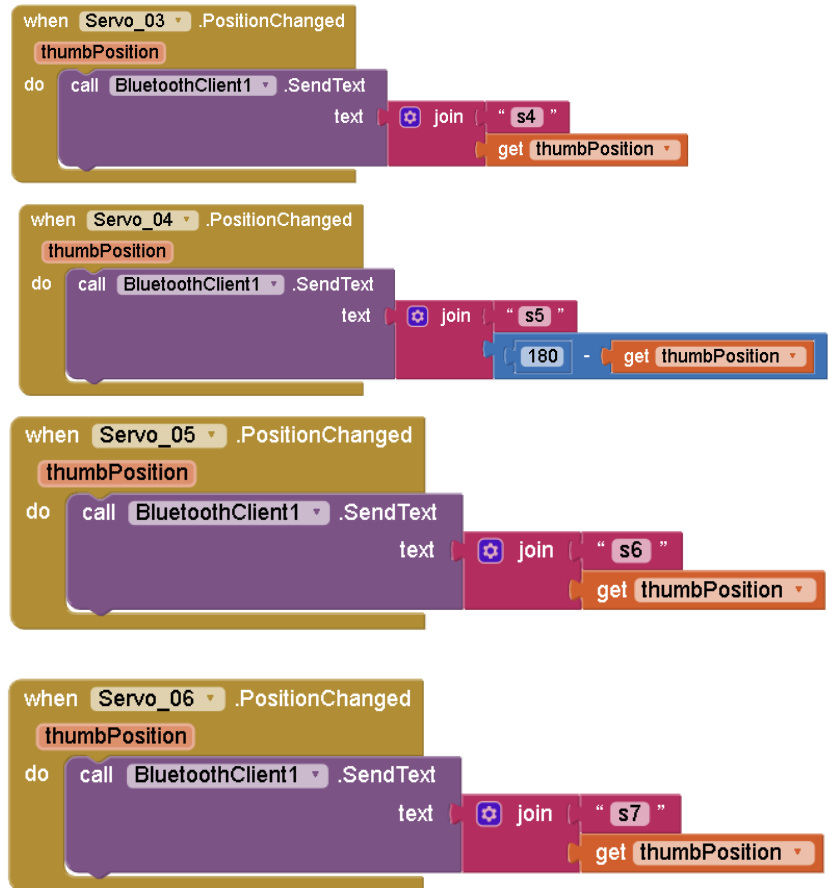
Botones de los movimientos:

```

when Servo_01 . PositionChanged
  thumbPosition
do
  call BluetoothClient1 . SendText
    text join " s1 "
      get thumbPosition

when Servo_02 . PositionChanged
  thumbPosition
do
  call BluetoothClient1 . SendText
    text join " s2 "
      get thumbPosition

```



link del hosting:

https://zoratha.000webhostapp.com/brazo_robotico/login/login.php