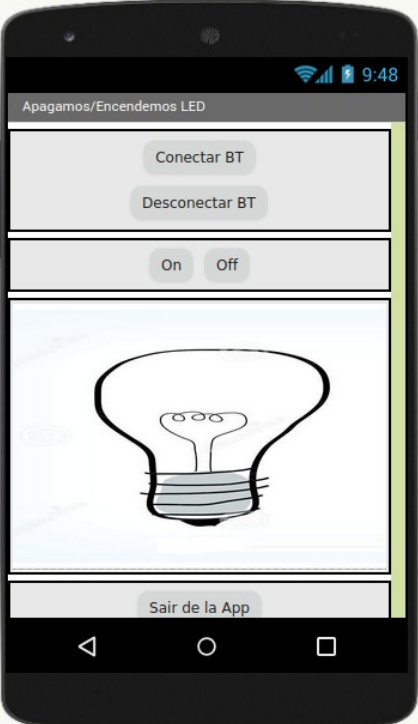




AppInventor – Bluetooth(cliente)

Mostrar en el Visor los componentes ocultos



Componentes no visibles

 BluetoothClient1  Notificador

Screen1

- VerticalArrangement1
 - BTConectar
 - BtDesconectar
- HorizontalArrangement1
 - BON
 - BOFF
- HorizontalArrangement2
 - ImagenBombilla
- HorizontalArrangement3
 - BApagado
 - BluetoothClient1
 - Notificador

Cambiar nombre Borrar

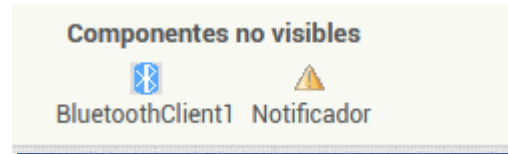
Medios

- bombillaapagada.jpg
- bombilla...ndida.jpg

Subir archivo...

AppInventor – Bluetooth(cliente)

Elementos no visibles con los que trabajaremos



AppInventor – Bluetooth(cliente)

El botón que nos permita conectar a Bluetooth debe ser un

SELECTOR DE LISTA

AppInventor – Bluetooth(cliente)

Procedimiento para conectar Bluetooth

cuando **BTConectar** .AntesDeSelección

ejecutar poner **BTConectar** . Elementos como **BluetoothClient1** . DireccionesYNombres

cuando **BTConectar** .DespuésDeSelección

ejecutar si llamar **BluetoothClient1** .Conectar
dirección **BTConectar** . Selección

entonces llamar **Notificador** .MostrarAlerta
aviso " Conectado "

AppInventor – Bluetooth(cliente)

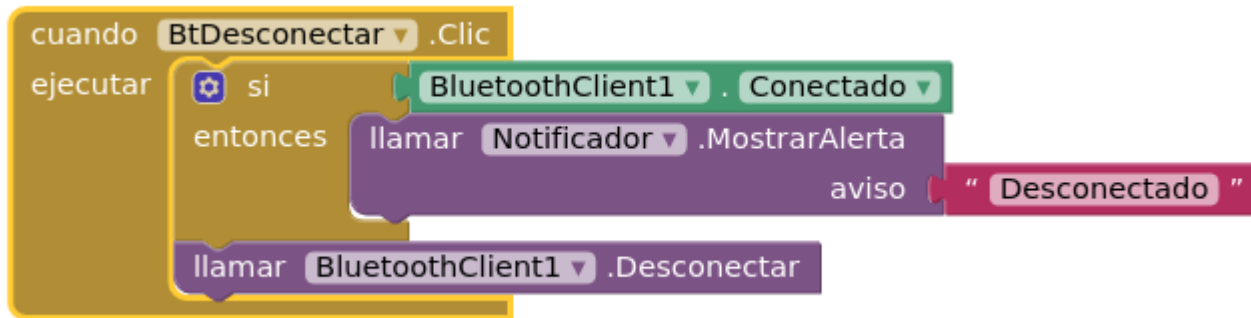
Procedimiento para enviar desde Bluetooth

```
cuando BOFF .Clic
ejecutar
  llamar BluetoothClient1 .EnviarTexto
    texto " Apagado\n "
  poner ImagenBombilla . Foto como " bombillaapagada.jpg "
```

```
cuando BON .Clic
ejecutar
  llamar BluetoothClient1 .EnviarTexto
    texto " Encendido "
  poner ImagenBombilla . Foto como " bombillaencendida.jpg "
```

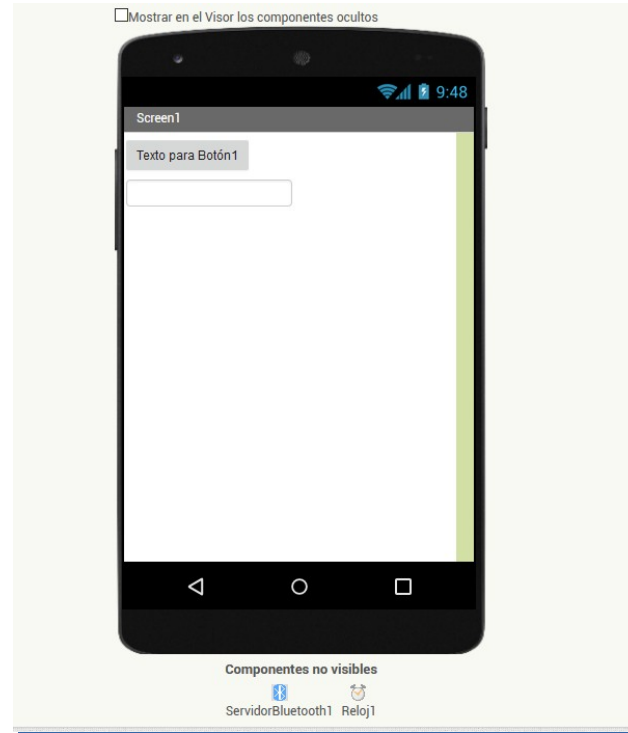
AppInventor – Bluetooth(cliente)

Procedimiento para desconectar Bluetooth



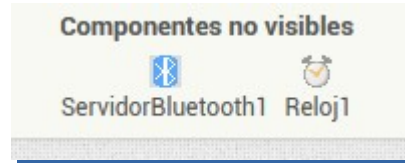
AppInventor – Bluetooth(servidor)

Aplicación



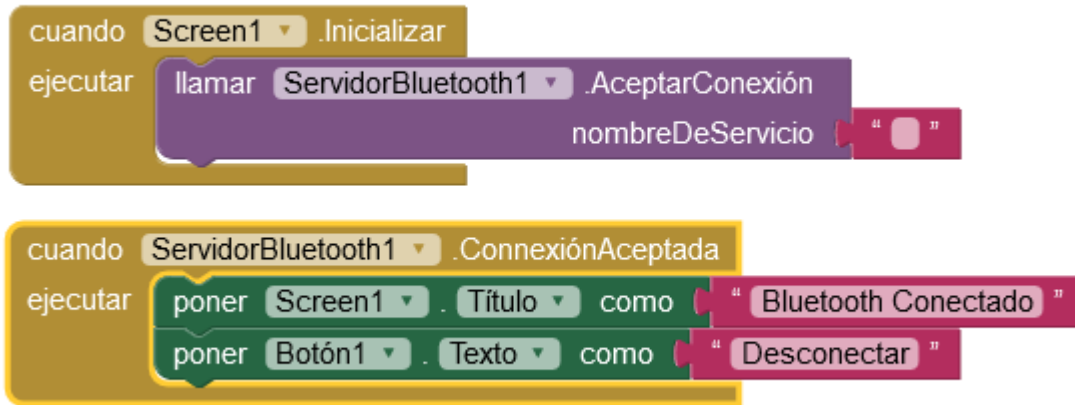
AppInventor – Bluetooth(servidor)

Elementos no visibles con los que trabajaremos



AppInventor – Bluetooth(servidor)

Para “Aceptar” conexiones desde cliente



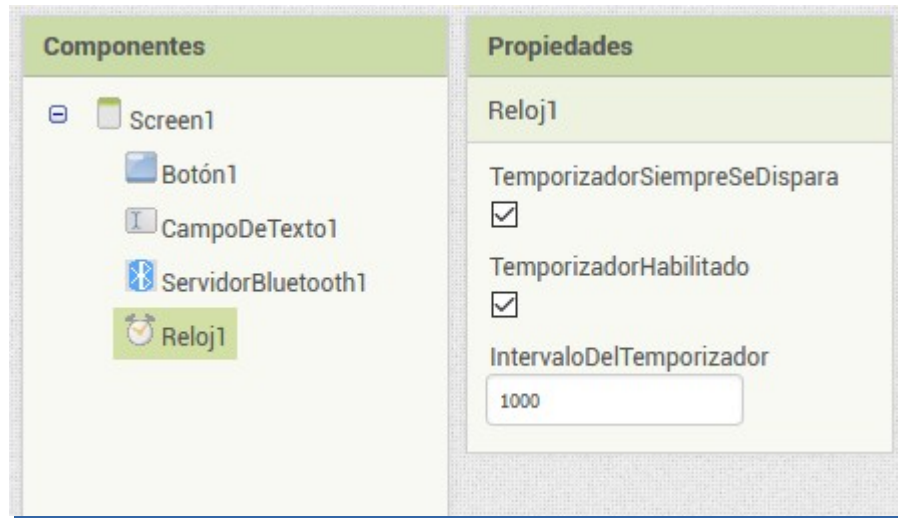
AppInventor – Bluetooth(servidor)

Para “Recibir” datos desde cliente



AppInventor – Bluetooth(servidor)

Para “Recibir” datos desde cliente



AppInventor – Bluetooth(servidor)

Para desconectar Bluetooth



Bluetooth Raspberry Pi

- Tenemos que tener en cuenta que ClientBluetooth de Appinventor trabaja con SDP lo que hace todo pelín más complicado.
- La parte de SDP no es necesaria si estamos vinculando dos clientes bluetooth que solo utilicen RFCOMM

Bluetooth Raspberry Pi



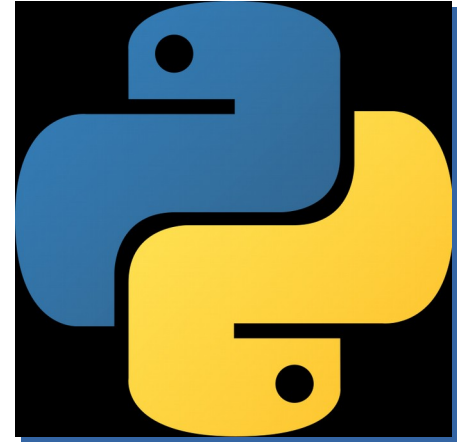
Bluetooth Raspberry Pi

- Para obtener la MAC de nuestro Bluetooth:

`hciconfig`

Bluetooth Raspberry Pi

- Para la instalación de librerías Python:
`sudo apt-get update`
`sudo apt-get install bluetooth`
`sudo apt-get install python-bluez`



Bluetooth Raspberry Pi

- Arrancamos los servicios de bluetooth:
service bluetooth start
service bluetooth status (Obviamente esto solo es necesario como comprobación)

Bluetooth Raspberry Pi

- Para vincular nuestro móvil al Bluetooth de la Raspberry Pi

```
sudo bluetoothctl
```

```
[bluetooth]# power on --> activa bluetooth
```

```
[bluetooth]# agent on --> hace que sea el principal
```

```
[bluetooth]# discoverable on --> lo muestra al resto de dispositivos bluetooth para ser vinculado
```

```
[bluetooth]# pairable on --> lo hace vinculable
```

```
[bluetooth]# scan on --> escanea todos los dispositivos disponibles
```

```
[bluetooth]# pair <MAC del móvil a vincular> --> esto lo vincula.
```

```
[bluetooth]# trust <MAC del móvil vinculado> --> para hacerlo vinculable para siempre
```

- Si queremos ver los dispositivos vinculados

```
[bluetooth]# paired-devices
```

Bluetooth Raspberry Pi

- Para vincular nuestro móvil al Bluetooth de la Raspberry Pi

```
sudo bluetoothctl
```

```
[bluetooth]# power on --> activa bluetooth
```

```
[bluetooth]# agent on --> hace que sea el principal
```

```
[bluetooth]# discoverable on --> lo muestra al resto de dispositivos bluetooth para ser vinculado
```

```
[bluetooth]# pairable on --> lo hace vinculable
```

```
[bluetooth]# scan on --> escanea todos los dispositivos disponibles
```

```
[bluetooth]# pair <MAC del móvil a vincular> --> esto lo vincula.
```

```
[bluetooth]# trust <MAC del móvil vinculado> --> para hacerlo vinculable para siempre
```

- Si queremos ver los dispositivos vinculados

```
[bluetooth]# paired-devices
```

Bluetooth Raspberry Pi

- Ahora trabajamos con **SDP** para poder trabajar con APPINVENTOR

Entramos en el fichero (o en el enlace lógico que aparece en `/etc/systemd/system/bluetooth.target.wants/bluetooth.service`) `/lib/systemd/system/bluetooth.service` y comprobamos que la línea `ExecStart` está como sigue:

`ExecStart=/usr/lib/bluetooth/bluetoothd --compat --noplugin=sap`

`ExecStartPost=/usr/bin/sdptool add --channel=22 SP` -->activamos el canal 22 para SP

Bluetooth Raspberry Pi

- Reiniciamos el servicio bluetooth:

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart bluetooth
```

Código Python para el servidor

```
import os
import socket
hostMacAddress = 'B8:27:EB:A7:D3:8A' ← -La mac de vuestra raspberry pi, obtenida con hciconfig
port = 22 #Este puerto debe ser el canal que hemos activado en el servicio Bluetooth
conexiones = 1
size = 1024
s = socket.socket(socket.AF_BLUETOOTH, socket.SOCK_STREAM, socket.BTPROTO_RFCOMM)
s.bind((hostMacAddress,port))
s.listen(conexiones)
try:
    client, address = s.accept()
    while True:
        data = client.recv(size)
        if data:
            print(data)
            client.send(bytes(f"Te he escuchado y has enviado {data}\n", 'UTF-8'))
except:
    print('Cerrando socket')
    client.close()
s.close()Reiniciamos el servicio bluetooth:
```

Firestore

CharlasAppInventor ▾

Realtime Database

Datos Reglas Copias de seguridad Uso

✦ Crea prototipos y pruébalos por completo con la Suite del Emulador Local, que ahora también incluye Firebase Authentication. [Empezar](#)

[🔗](#) [https://charlasappinventor.firebaseio.com/](#)



charlasappinventor-default-rtdb

📄 ventanas

Firestore

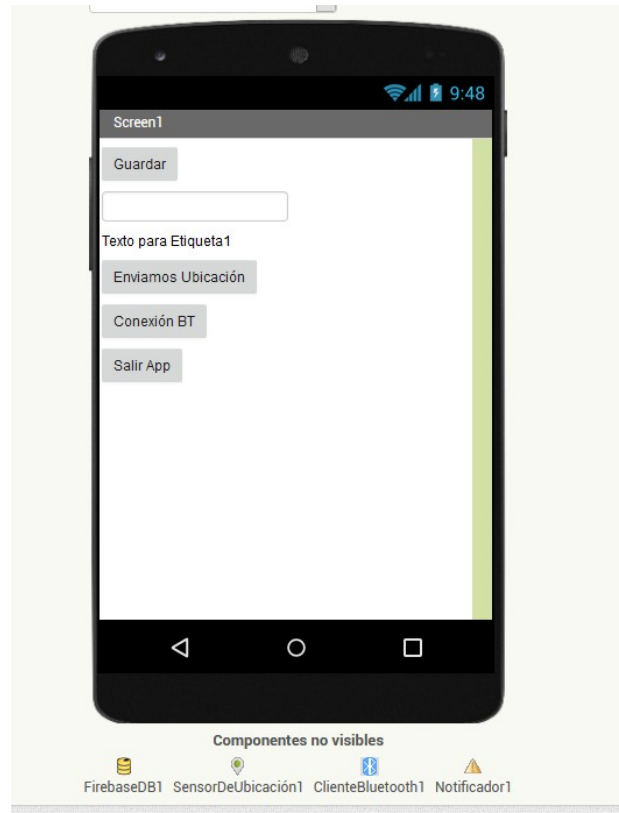
Editar reglas Supervisar reglas

Zona de prueba de reglas

```
1 {
2   "rules": {
3     ".read": true,
4     ".write": true
5   }
6 }
```

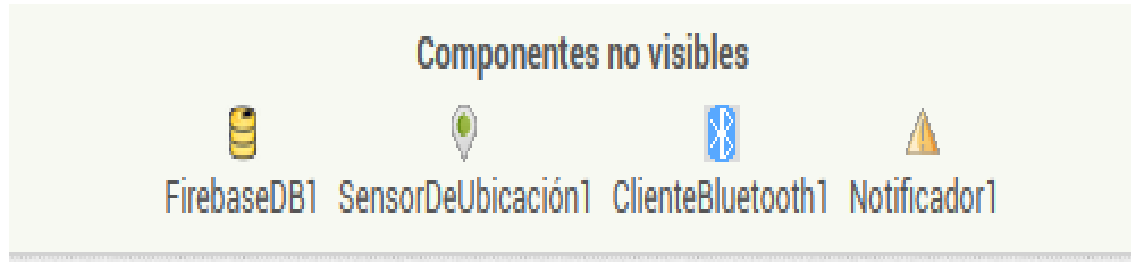

Firestore en AppInventor

- Nuestra Aplicación



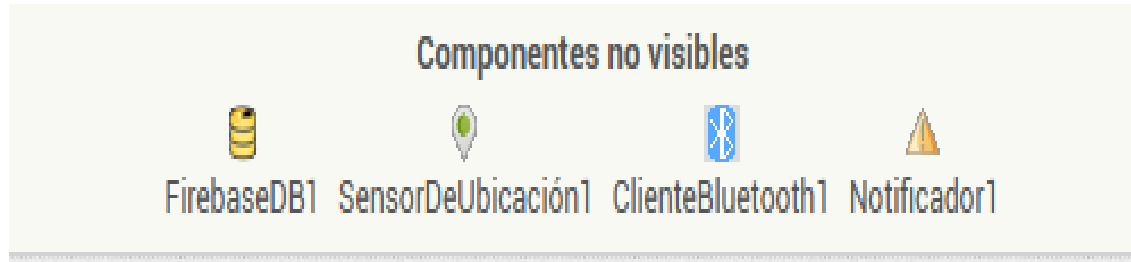
Firestore en AppInventor

- Elementos No Visibles



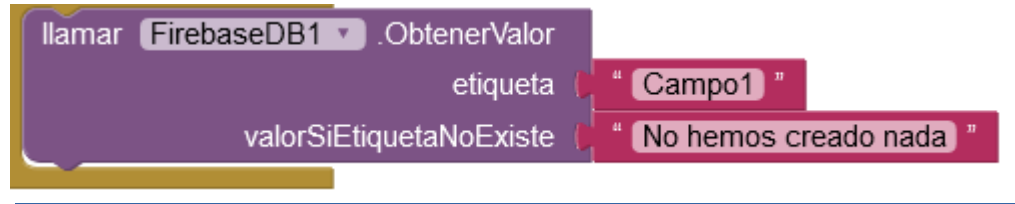
Firestore en AppInventor

- Elementos No Visibles



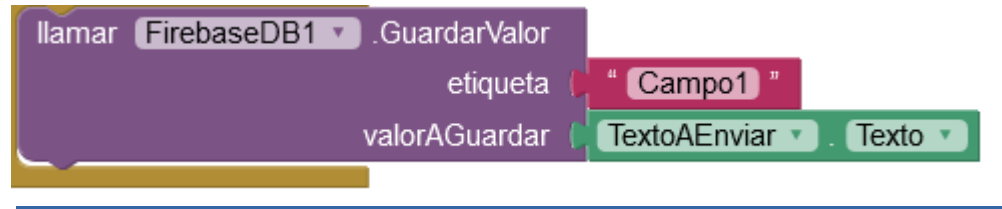
Firestore en AppInventor

- Pedimos a FireBase un valor



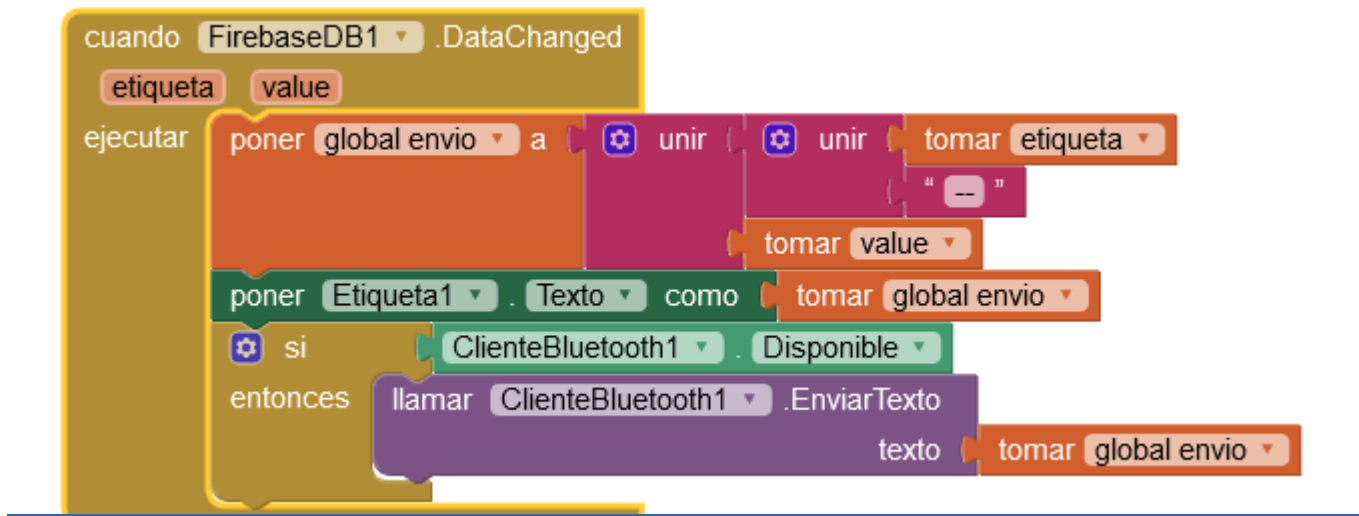
Firestore en AppInventor

- Guardamos en FireBase un valor



Firestore en AppInventor

- Y si cambia algún valor que nos interese...WebSocket



Firestore en AppInventor

- El resto de cosas que hay en la aplicación es unir un poco todo lo que hemos visto

Bluetooth

Notificaciones